Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ (РГГМУ)

Институт Информационных систем и геотехнологий

**1 КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ**

БАКАЛАВРСКАЯ РАБОТА

На тему

“Автоматизированное формирование сопроводительной документации образовательной программы”

**Исполнитель** Цветков Владислав Олегович

**Руководитель** к. т. н., доцент – Яготинцева Наталья Владимировна

**«К защите допускаю»**

**Заведующий кафедрой** _____

(подпись)

_____

(ученая степень, ученое звание)

_____

(фамилия, имя, отчество)

«___»_____2023 г.

Санкт–Петербург

2023г.

**Оглавление**

## Ведение

В условиях современного делопроизводства – требуется обрабатывать все больший объём документов. Это необходимо по разным и малосвязанным причинам. начиная от выдачи чека за покупки, заканчивая подписанием государственных законов в госдуме.

Документы являются неотъемлемой частью современного мира, и их обработка — это важный, кропотливый и время затратный процесс. Который, зачастую, требует особой квалификации от исполнителя.

Автоматизация документооборота позволяет эффективнее решать задачи в условиях современного делопроизводства. При должном уровне автоматизации допущение ошибки становится не просто маловероятным, а невозможным.

Что касается формирования сопроводительной документации образовательной программы, то ситуация аналогичная. Многие учебные заведения пренебрегают возможностью оптимизировать процесс формирования документов. В результате чего, преподавателям, как и некоторым другим сотрудникам, приходится брать на себя лишнюю ответственность по обработке соответственных документов. Подобное не только может негативно сказаться на качестве получаемых документов, но и отобразится на основной профессиональной деятельности преподавателей.

Таким образом внедрение в бизнес-процессы учебного заведения средств автоматизации документооборота снимет с преподавателей лишнюю ответственность снизив нагрузку и увеличив время на профессиональную деятельность, что повысит общую эффективность системы. На этом и основывается актуальность данной работы.

**Объектом исследования** является процесс формирования сопроводительной документации образовательной программы

**Предметом исследования** является проблема автоматизации процесса формирования сопроводительной документации дисциплины

**Цель работы** разработать прототип приложения для автоматизированной обработки сопроводительной документации образовательной программы.

**Задачи:**

1. Произвести анализ предметной области
2. Проанализировать существующие решения
3. Создать модель информационной системы
4. Разработать интерфейс
5. Реализовать прототип

**Используемые методы проектирования:**

- Концептуальное проектирование при помощи диаграммы прецедентов и диаграмм последовательностей
- Объектно-ориентированное проектирование при помощи диаграммы классов
- Проектирование физического представления при помощи диаграммы развертывания
- Проектирования базы данных, схемой базы данных

**Используемые средства проектирования:**

- Draw.io – для отрисовки всех видов диаграмм

**Используемые средства в реализации:**

- Язык программирования Python
- Фреймворк Flask с необходимыми модулями
- База данных MariaDB

- Программное обеспечение Nginx
- Средства виртуализации Docker

Используемые средства разработки:

- Visual Studio Code как IDE и текстовый редактор
- Mozilla Firefox как IDE
-

# 2 Предпроектный анализ

Перед началом проектирования системы необходимо выполнить соответственные приготовления, а именно:

1. Изучить аналоги, произвести анализ применяемых решений, выявить преимущества и недостатки – определить целесообразность разработки целевого проекта
2. Распланировать и зафиксировать сроки реализации
3. Рассчитать затраты на разработку, то есть, рассчитать стоимость приобретения и обслуживания средств разработки
4. Определить и зафиксировать требования к целевому прототипу
5. Определить преимущества и недостатки гипотетической системы, выявить угрозы и дополнительные возможности

Не все вышеперечисленные пункты являются необходимыми, но в рамках данной работы данные действия были сочтены оптимальными и минимально необходимыми

## 2.1 Анализ предметной области

Автоматизация документооборота широкая и развитая сфера бизнеса. Существует множество средств оптимизации работы с документами различных масштабов и специфик. В рамках данной работы будут рассмотрены только средства автоматизации создания сопроводительной документации дисциплин образовательной программы. Таких на сегодняшний день, среди относительно распространенных всего 3:

1. «Юрайт» – сервис с шаблонами учебных документов, позволяет сформировать документ после заполнения простой формы. Имеет ограниченный функционал, но можно использовать на бесплатной основе.
2. «Рабочие программы» - средства автоматизирования от «ММИС» для формирования учебной документации. Продвинутое программное обеспечение для ВУЗов, имеет существенно больший функционал, чем есть в «Юрайт», например существует база для хранения общих входных данных. Распространяется только платно.
3. «Планы»- у сервиса от «ТУСУР» схожий с «Рабочими программами» функционал, имеет существенное преимущество в виде предоставлении вычислительных ресурсов и доступа к ним посредством web-интерфейса. Также распространяется строго платно.

Для выявления преимуществ и недостатков следует рассмотреть сервисы более детально на примере формирования РПД.

### 2.1.1 Анализ сервиса «Юрайт»

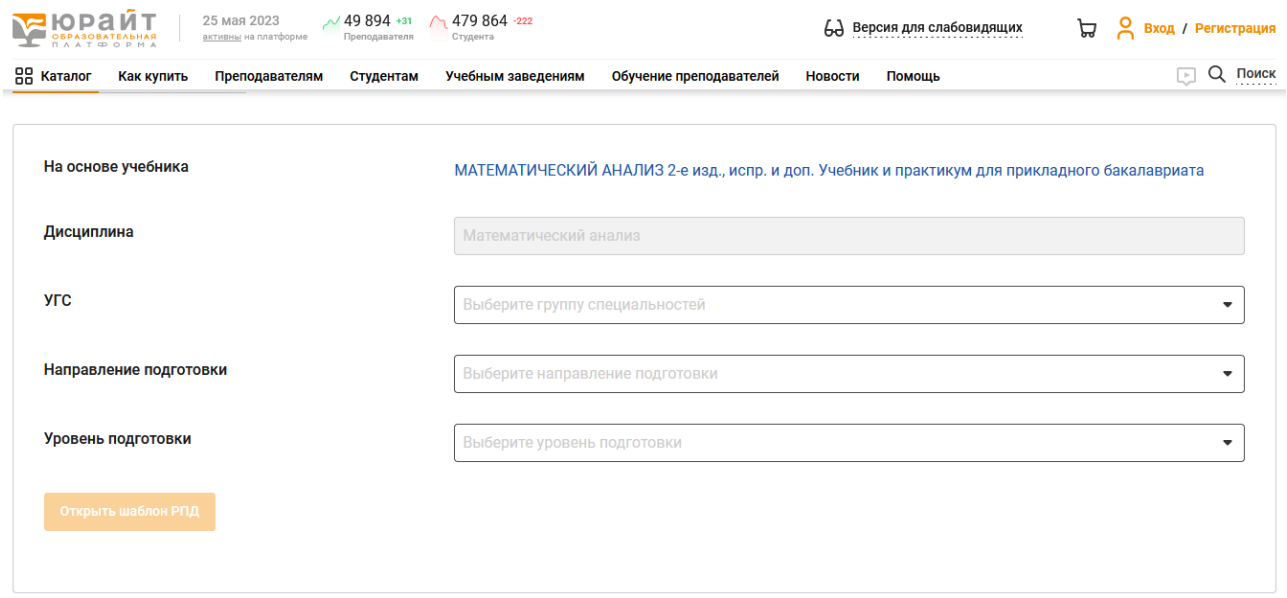



Рисунок 1. Форма создания РПД

Сервис «Юрайт» имеет обширную базу данных учебников, которые используются для введения основной информации в целевой документ. После выбора учебника, требуется заполнить оставшиеся 4 поля и нажать кнопку загрузки РПД.

**Преимущества:**

- Очень простой в использовании
- Большая база учебников
- Web-интерфейс
- Бесплатная основа

**Недостатки:**

- Не гибкий
- Не учитывается специфика ВУЗа
- Невозможно внедрить на уровне ВУЗа

### 2.1.2 Анализ ПО «Рабочие программы»

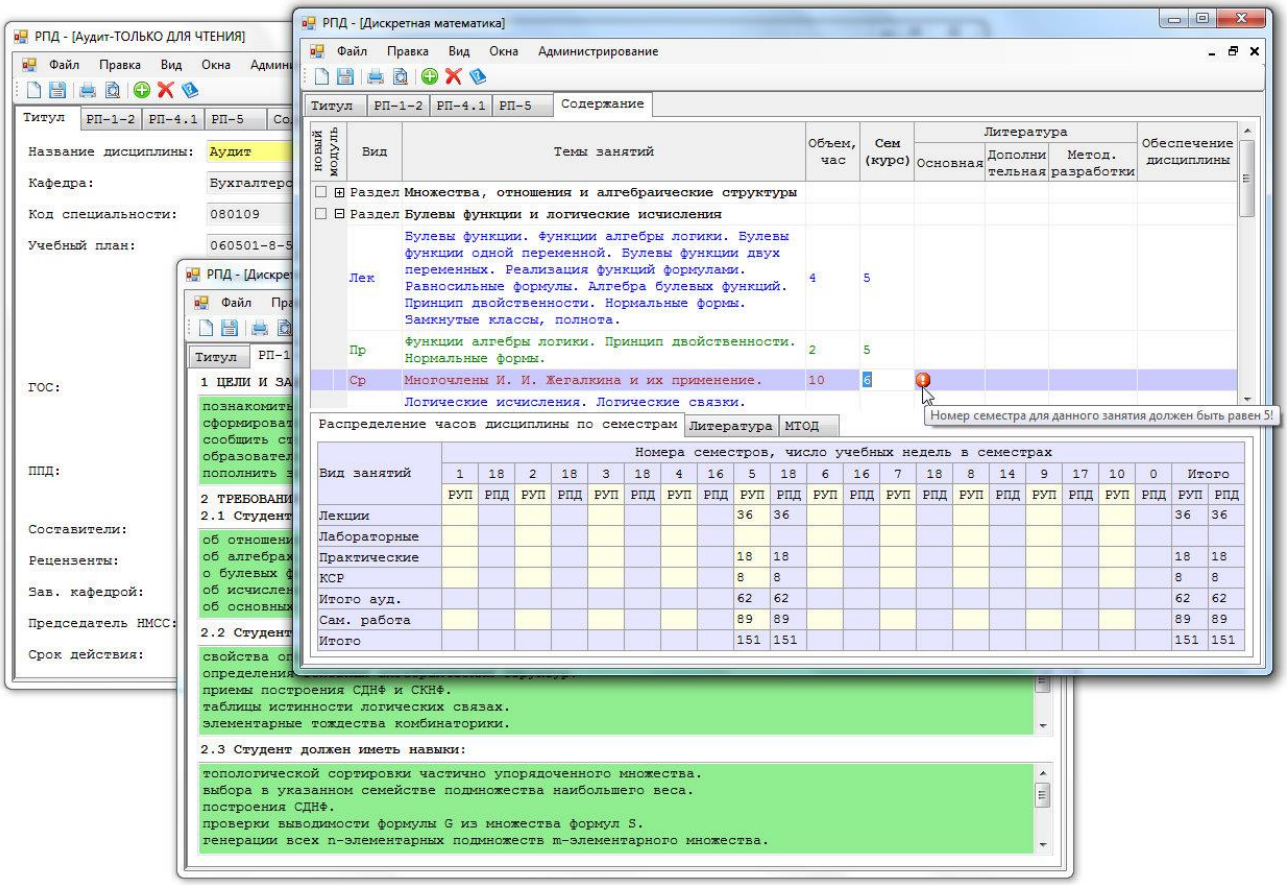



Рисунок 2. Интерфейс «Рабочих программ»

Очень мощный инструмент для обработки сопроводительной документации. Требует предварительной настройки и некоторого времени на освоение. После приобретения можно развернуть на собственном сервере и встроить в информационную систему Уза. Можно развернуть с web-интерфейсом.

Преимущества:

- Общая база предварительных значений
- Широкий функционал
- Web-версия
- Можно встроить в ИС УЗа

Недостатки:

- Непрост в освоении
- Требует ручное заполнения базы значений
- Не предоставляются аппаратные средства
- Платная основа

### 2.1.3 Анализ сервиса «Планы»

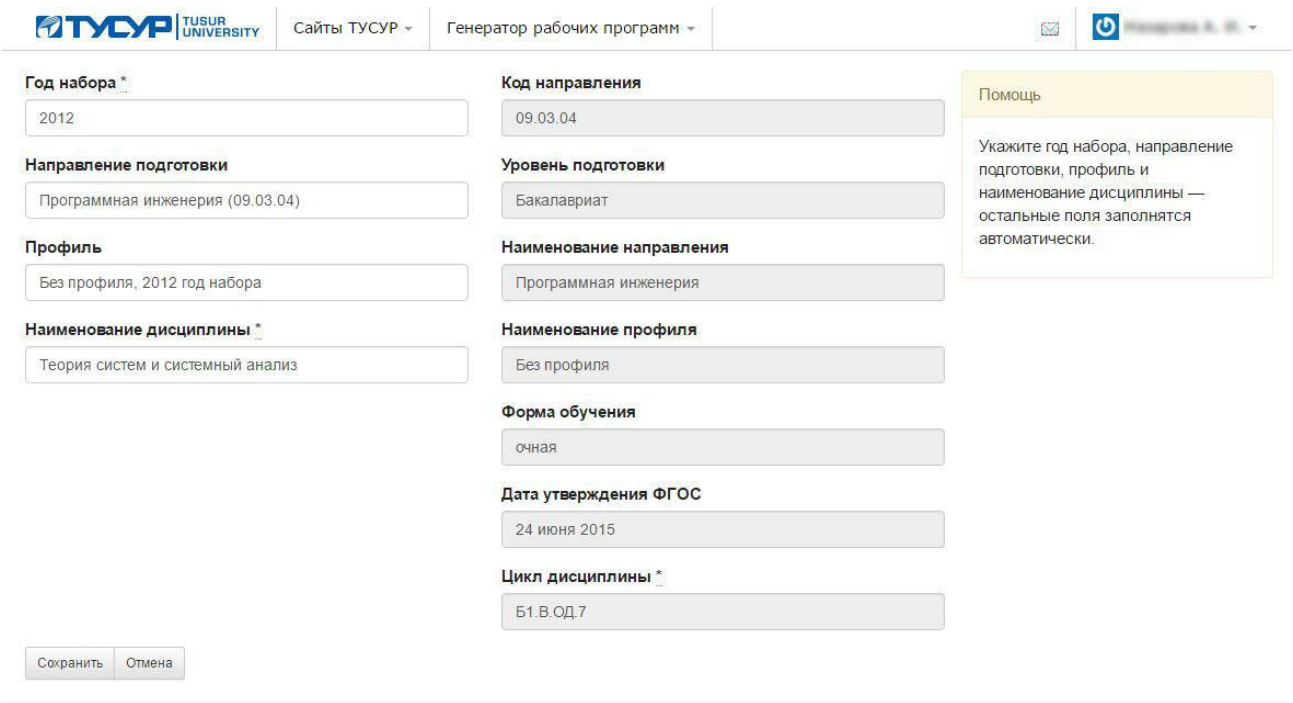



Рисунок 3. Интерфейс сервиса «Планы»

Сервис «Планы» хотя и имеет схожий с «Рабочими программами функционал», но все же менее функционален. Поставляется по подписке, не требует дополнительных аппаратных ресурсов.

Преимущества:

- Общая база предварительных значений
- Относительно простой в освоении
- Web-интерфейс
- Не требует дополнительных аппаратных средств

Недостатки:

- Требует ручное заполнения базы значений
- Платная основа

### 2.1.4 Сравнительный анализ программных средств

Таблица 1. Сравнительный анализ аналогов

| Наименование ПО | "Планы" | "Генератор Рабочих Программ" | Юрайт | Прототип |
|---|---|---|---|---|
| Шаблоны документов | + | + | + | + |
| Общая база входных данных | + | + | - | + |
| Web-интерфейс | + | + | + | + |
| Использование содержимого документа в качестве входных данных | - | - | - | + |
| Интеграция с ИС УЗа | - | + | - | + |
| Бесплатная основа | - | - | + | ? |

На таблице представлены наиболее важные аспекты сравнения. Как можно заметить, ни один из рассмотренных аналогов не поддерживает использования существующих документов для извлечения входных данных, чем уступают целевому прототипу. Также не все рассмотренные аналоги поддерживают интеграцию с информационной системой учебного заведения, что хоть и является сомнительным преимуществом, но очень полезной возможностью в перспективе, а именно извлечением необходимых документов из актуальной документной базы, для автоматического заполнения собственной базы входных значений.

Исходя из описанного в данной главе можно сделать вывод, что итоговый прототип не только имеет право на существование, но и потенциально является конкурентноспособным продуктом, с видимыми преимуществами.

## 2.2 Сроки реализации прототипа

Для повышения вероятности завершения проекта следует выделить основные задачи, и примерные сроки их завершения, так как согласно статистике, проекты без даты завершения чаще всего никогда не завершаются. В случае данной работы следует выделять 5 основных задач:

1. Предпроектный анализ
2. Проектирование
3. Реализация
4. Тестирование и отладка
5. Оформление документации

При расчете на год получается следующая диаграмма (Рисунок 4).

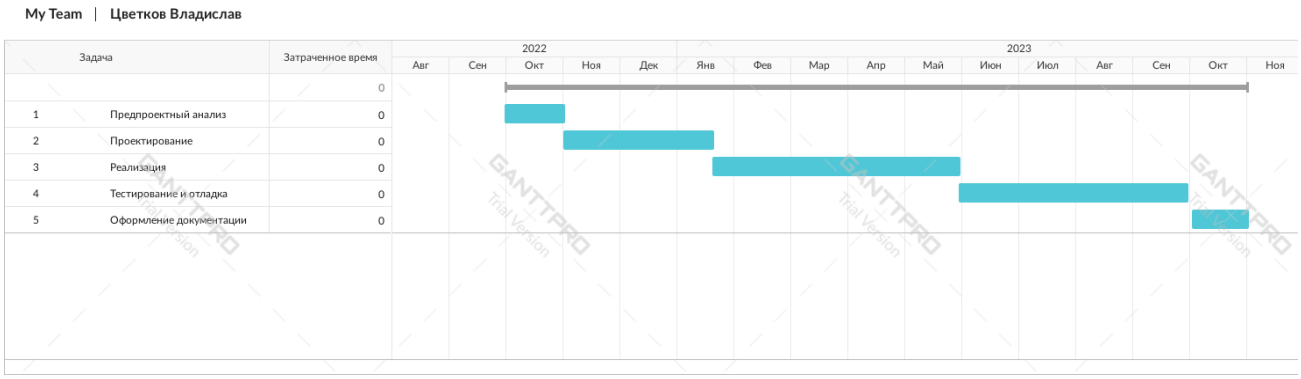



Рисунок 4. Диаграмма Ганта

Как можно увидеть на разработку и отладку выделено одинаковое большее количество времени, а именно по 4 месяца, так как согласно статистике именно эти 2 этапа занимают большую часть времени. Обычно отладка занимает меньше времени, чем разработка, но так как система является своего рода уникальным прототипом, требуется больше внимания уделить именно практическому опыту использования программы.

Что касается других этапов, ввиду узкой специфики проекта, на аналитическую часть не требуется выделять больше времени, чем 1 месяц. Этап документации же, хоть и является трудоемким, как правило не сопровождается

неожиданными трудностями, как этап разработки или отладки. Этап проектирования получил 2 месяца, так как от него по большей части зависит сложность и трудоемкость этапа реализации, на него не стоит выделять мало времени.

## 2.3 Технико-экономическое обоснование

В разработке участвует всего один человек, у проекта нет инвестированного бюджета, поэтому данная работа попадает в категорию инди-проектов. Такие проекты, как правило, тяжело поддаются оценке трудозатрат и обобщаются всего одним правилом: «При бюджете N желательно потратить меньше». Найм дополнительного персонала не требуется, таким образом, следует рассчитать стоимость инструментов для разработки и соответственных дополнительных сервисов. Исходя из данных суждений получается следующий прогноз затрат:

Таблица 2. Фактические затраты

| Наименование ресурса | Количество | Стоимость |
|---|---|---|
| Средства построения моделей Draw.io | 1 | бесплатно |
| IDE Visual Studio Code | 1 | бесплатно |
| Средства разработки и отладки Python | 1 | бесплатно |
| Средства разработки и отладки Firefox | 1 | бесплатно |
| VDS + открытый IP адрес | 1 год | 15 000 |
| | Итого: | 15 000 |

Всего затрат получилось на 15 000, то есть придется заплатить из бюджета разработки только за сервер, для отладки в условиях открытой сети. Если считать все расходы, и ресурсы, которые уже есть в распоряжении, включая человеческий труд, а для данного проекта хватит одного разработчика, владеющего полным стеком технологий, то следует добавить ещё такие пункты:

Таблица 3. Расчет стоимости существующих ресурсов

| Наименование ресурса | Количество | Стоимость |
|---|---|---|
| Разработчик<br>(Full-stack python) | 1 на 1 год | 720 000<br>(60 000/мес.) |
| Минимально функциональный ноутбук для разработки | 1 | 15 000 |
| | Итого: | 735 000 |

Как можно заметить из данной таблицы, стоимость проекта существенно возросла с добавлением человеческих ресурсов; итого, получается 750 000 рублей.

Рассчитывать окупаемость проекта пока нет смысла, так как пока предполагается использование только в одной конкретной организации. Сам проект не предполагает принесение дохода, но он предполагает снижение трудозатрат и ответственности соответственных кадров, а именно преподавателей.

## 2.4 Требования к системе

Требования один из обязательных пунктов для определения, ведь они отвечают за конечный вид системы. Без описания требований представление конечного вида системы попросту невозможно. Требования выделяются функциональные и нефункциональные.

Функциональные требования:

1. Возможность добавления документов для ввода, администратором
2. Система должна преобразовывать документы в данные для базы входных значений
3. Возможность редактировать базу входных значений для администратора
4. Программа должна генерировать документы на основе данных из базы данных и введенных пользователем
5. Осуществление пользовательских данных осуществляется посредством заполнения форм
6. Выгрузка документов должна быть доступна для выгрузки как пользователем, так и администратором
7. Администратор должен иметь возможность заполнять формы пользователей
8. Администратор может иметь все возможности пользователя, то есть должен иметь возможность заполнять форму на свое имя

Нефункциональные требования:

1. Система должна быть аппаратно-независимой
2. Дизайн интерфейса должен быть выполнен в соответствии с особыми дизайнерскими элементами РГГМУ
3. Интерфейс должен минимально загруженным функционалом

## 2.5 SWOT-анализ

SWOT - Strengths (сильные стороны), Weaknesses (слабые стороны), Opportunities (возможности) Threats (угрозы). Данный метод ещё называют методом стратегического планирования, он предназначен для определения основных свойств системы, чтобы выявить опасные моменты и определить оптимальные пути реализации. Представление по модели SWOT не является обязательным, но является полезным дополнением для определения путей реализации.

| | **Сильные стороны** | **Слабые стороны** |
|---|---|---|
| **Внешняя среда** | **Возможности:**<br>• Потенциально высокая степень автономности<br>• Функциональное преимущество перед аналогами<br>• Потенциал для улучшения | **Угрозы:**<br>• Низкая конкурентоспособность на старте<br>• Необходимость в дальнейшей доработке<br>• Зависимость от качества входных данных |
| **Внутренняя среда** | **Преимущества:**<br>• Легкость освоения<br>• Относительная простота развертывания<br>• Аппаратная независимость<br>• Низкие системные требования<br>• Строгое ограничение ответственности | **Недостатки:**<br>• Привязка к бизнес-процессам конкретного предприятия<br>• Необходимость развертывания<br>• Неполная автономность |

# 3 Моделирование системы

Перед тем как приступить к реализации, следует сначала спроектировать систему. Это необходимо чтобы составить наиболее полный образ конечного результата, тем самым сделав разработку более очевидной и менее рискованной. Кроме того, это поможет определить и зафиксировать сроки. Если задача однозначна определена, то её гораздо проще выполнить.

Для данной работы будут задействованы следующие подходы:

1. Моделирование прецедентов
2. Объектно-ориентированное моделирование
3. Составление модели развертывания
4. Составление модели базы данных

Конечно, существуют и другие полезные методологии, но вышеперечисленные были выбраны как оптимальные.

## 3.1 Диаграмма прецедентов

Первым шагом составления концепции прототипа будет составление диаграммы прецедентов. Данная диаграмма отображает отношения между участниками бизнес-процессов (актёрами) и методами взаимодействия с системой (прецедентами). Ниже представлена соответственная диаграмма прецедентов.

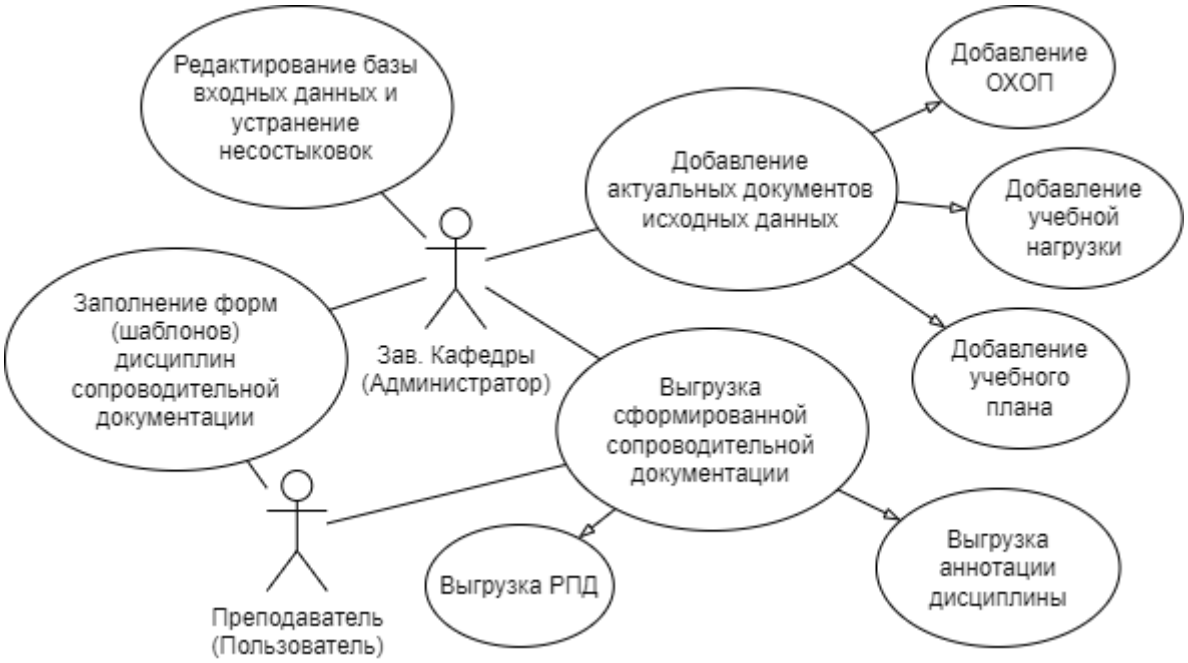



Рисунок 5. Диаграмма прецедентов

На данной диаграмме представлены пользователь (предполагается, что это преподаватель), и администратор (заведующий кафедрой, к примеру). Как можно увидеть на диаграмме, всего система предполагает 4 метода взаимодействия:

1. Добавление документа

   *- ОХОПП, учебный план и учебная нагрузка. Система должна будет извлечь из этих документов соответственную информацию необходимую для заполнения соответственных документов*
2. Редактирование БД

*- Данная опция будет полезна для отладки, но основное ее назначение – исправление расхождений в данных. Опечатки в фамилии преподавателя, например.*

3. Заполнение форм

   *- Формы нужны чтобы внести недостающую для генерации документов информации. Предполагается, что это только та информация, которую определяет преподаватель.*

4. Выгрузка итоговых документов

   *- После внесения в базу всех необходимых данных это опция открывает доступ к полученным документам.*

Далее представлены таблицы и схемы для более удобного представления порядка действий основный прецедентов. Стоит отметить, что для данного прототипа пока не представлены исключения, так как исключения серьезно усложнят разработку, на начальном этапе. Но нет никакой проблемы в том, чтобы добавить исключения при модификации конечной системы в будущем.

На таблице ниже представлен ход событий для прецедента «Добавление актуальных документов исходных данных». Так как данный прецедент реализует «Добавление ОХОП», «Добавление учебной нагрузки» и «Добавление учебного плана» - данные прецеденты не требуют аналогичного описания, так как имеют идентичный ход событий.

Таблица 4. Типичный ход сценария
«Добавление актуальных документов исходных данных»

| Главный раздел | |
|---|---|
| Вариант использования | Добавление актуальных документов исходных данных |
| Актёры | Зав. Кафедры (Администратор) |
| Краткое описание | Загрузка документа на сервер |
| Цель | Предоставить исходные данные для генератора документов |
| Ход событий | |
| Действия актеров | Отклик системы |
| 1. Инициация добавления документа | |
| | 2. Запрос документа |
| 3. Выбор и отправка документа для выгрузки | |
| | 3. Обработка документа |
| | 4. Выгрузка данных в БД |
| | 5. Возврат статуса обработки |

В соответствии с данными таблицы была составлена следующая диаграмма последовательности.

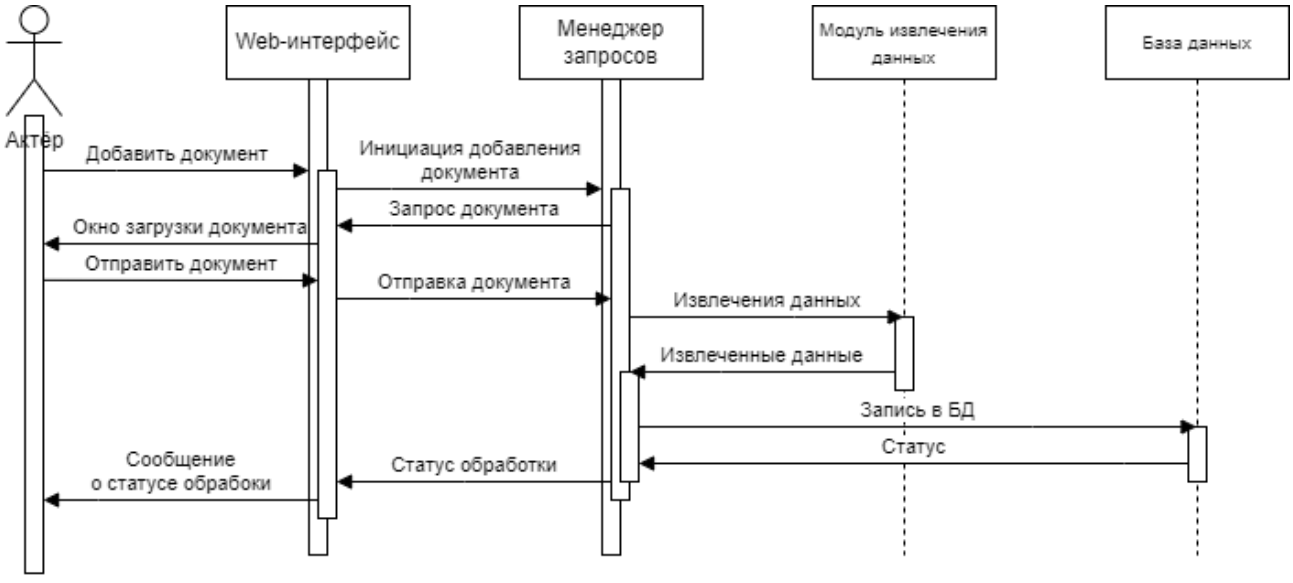



Рисунок 6. Диаграмма последовательности для прецедента
«Добавление актуальных документов исходных данных»

Диаграмма последовательности представляет события в рамках жизненного цикла одного события прецедента

Далее аналогичным прецеденту «Добавление актуальных документов исходных данных» образом представлены сценарии и диаграммы последовательности для прецедентов: «Редактирование базы входных данных и устранение несостыковок», «Заполнение форм (шаблонов) дисциплин сопроводительной документации» и «Выгрузка сформированной сопроводительной документации». Так как «Выгрузка сформированной сопроводительной документации» реализуется идентичным образом «Выгрузке РПД» и «Выгрузке аннотации» - данные прецеденты развернутого описания не требуют.

Таблица 5. Типичный ход сценария
«Редактирование базы входных данных и устранение несостыковок»

| Главный раздел | |
|---|---|
| Вариант использования | Редактирование базы входных данных и устранение несостыковок |
| Актёры | Зав. Кафедры (Администратор) |
| Краткое описание | Загрузка документа на сервер |
| Цель | Предоставить исходные данные для генератора документов |
| Ход событий | |
| Действия актеров | Отклик системы |
| 1. Инициация просмотра данных | |
| | 2. Сбор данных из БД |
| | 3. Формирование таблицы значений |
| | 4. Возврат формы |
| 5. Манипуляции нат данными | |
| 6. Инициация сохранения | |
| | 7. Обработка таблицы и сохранение данных |
| | 8. Возврат статуса обработки |

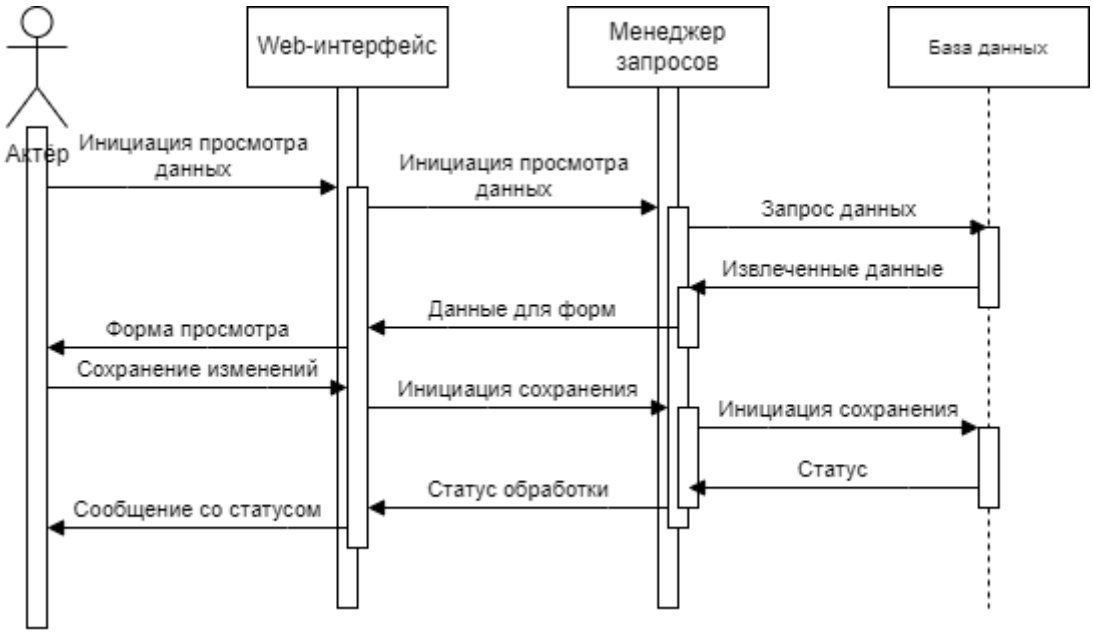



Рисунок 7. Диаграмма последовательности для прецедента «Редактирование базы входных данных и устранение несостыковок»

Таблица 6. Типичный ход сценария
«Заполнение форм (шаблонов) дисциплин сопроводительной документации»

| Главный раздел | |
|---|---|
| Вариант использования | Заполнение форм (шаблонов) дисциплин сопроводительной документации |
| Актёры | Зав. Кафедры (Администратор) или Преподаватель (Пользователь) |
| Краткое описание | Формирование и передача документа актёру |
| Цель | Получить итоговый документ |
| Ход событий | |
| Действия актеров | Отклик системы |
| 1. Инициация заполнения формы | |
| | 2. Сбор данных из БД |
| | 3. Формирование формы |
| | 4. Возврат формы |
| 5. Манипуляции с формой | |
| 6. Инициация сохранения | |
| | 7. Обработка формы |
| | 8. Сохранения значений в БД |

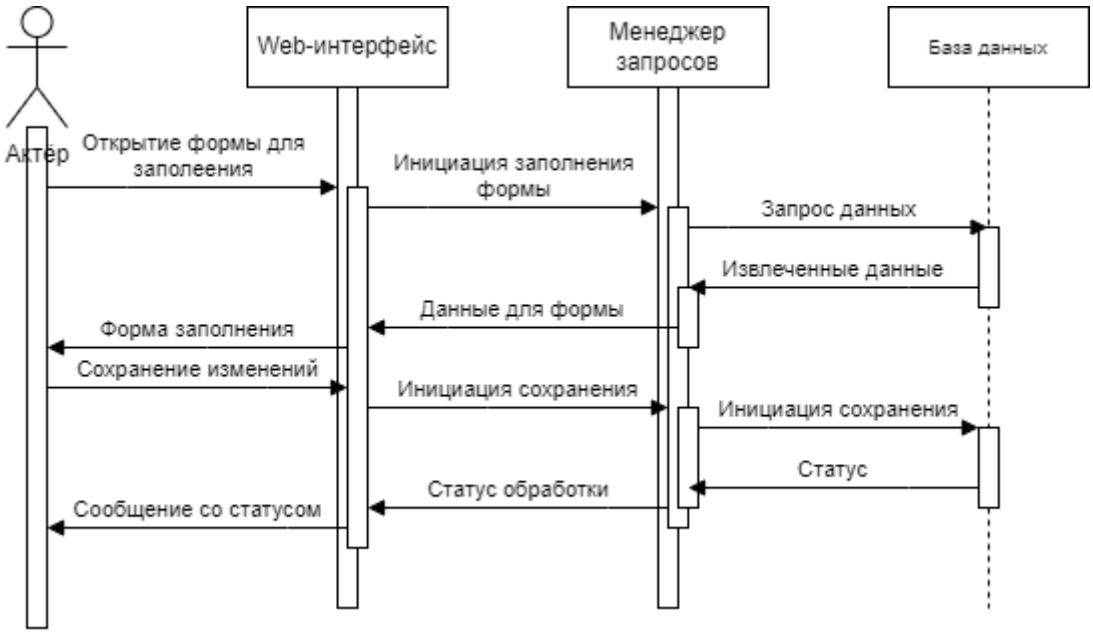



Рисунок 8. Диаграмма последовательности для прецедента
«Заполнение форм (шаблонов) дисциплин сопроводительной документации»

Таблица 7. Типичный ход сценария
«Выгрузка сформированной сопроводительной документации»

| Главный раздел | |
|---|---|
| Вариант использования | Выгрузка сформированной сопроводительной документации |
| Актёры | Зав. Кафедры (Администратор) или Преподаватель (Пользователь) |
| Краткое описание | Формирование и передача документа актёру |
| Цель | Получить итоговый документ |
| Ход событий | |
| Действия актеров | Отклик системы |
| 1. Инициация получения документа | |
| | 2. Сбор данных из БД |
| | 3. Формирование документа |
| | 4. Возврат документа |

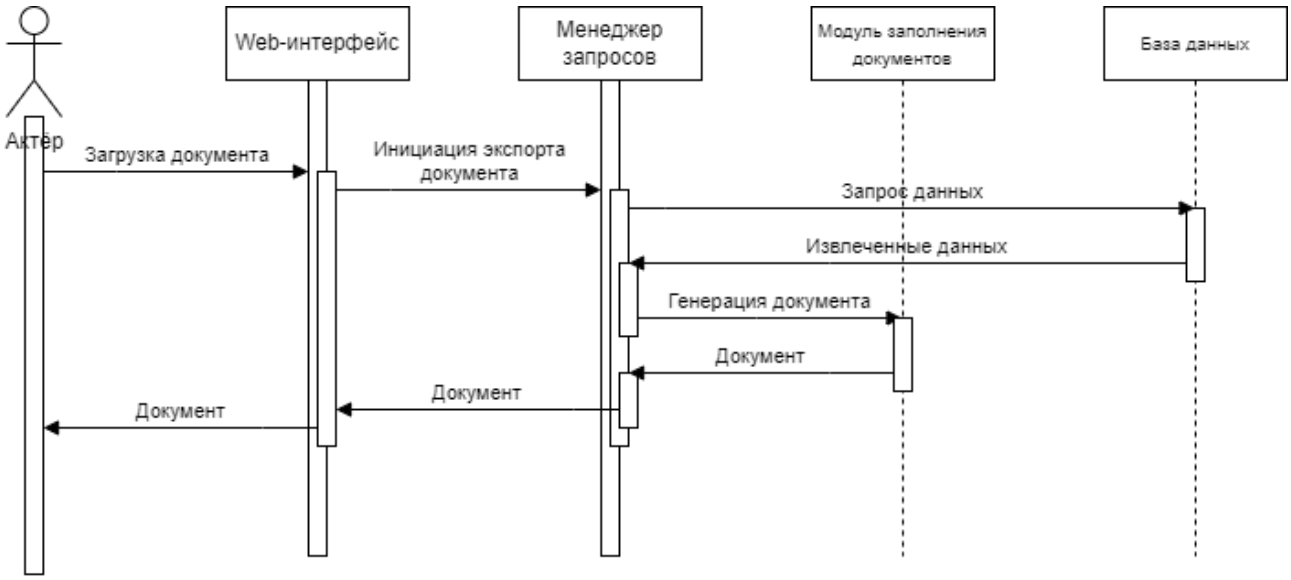



Рисунок 9. Диаграмма последовательности для прецедента
«Выгрузка сформированной сопроводительной документации»

## 3.2 Диаграмма классов

Диаграмма классов представляет систему в виде объектов (классов). Подобный подход помогает более детально представить логическую структуру системы, а также показывает зависимости и виды взаимодействий между компонентами. Ниже представлена диаграмма классов для целевого прототипа.

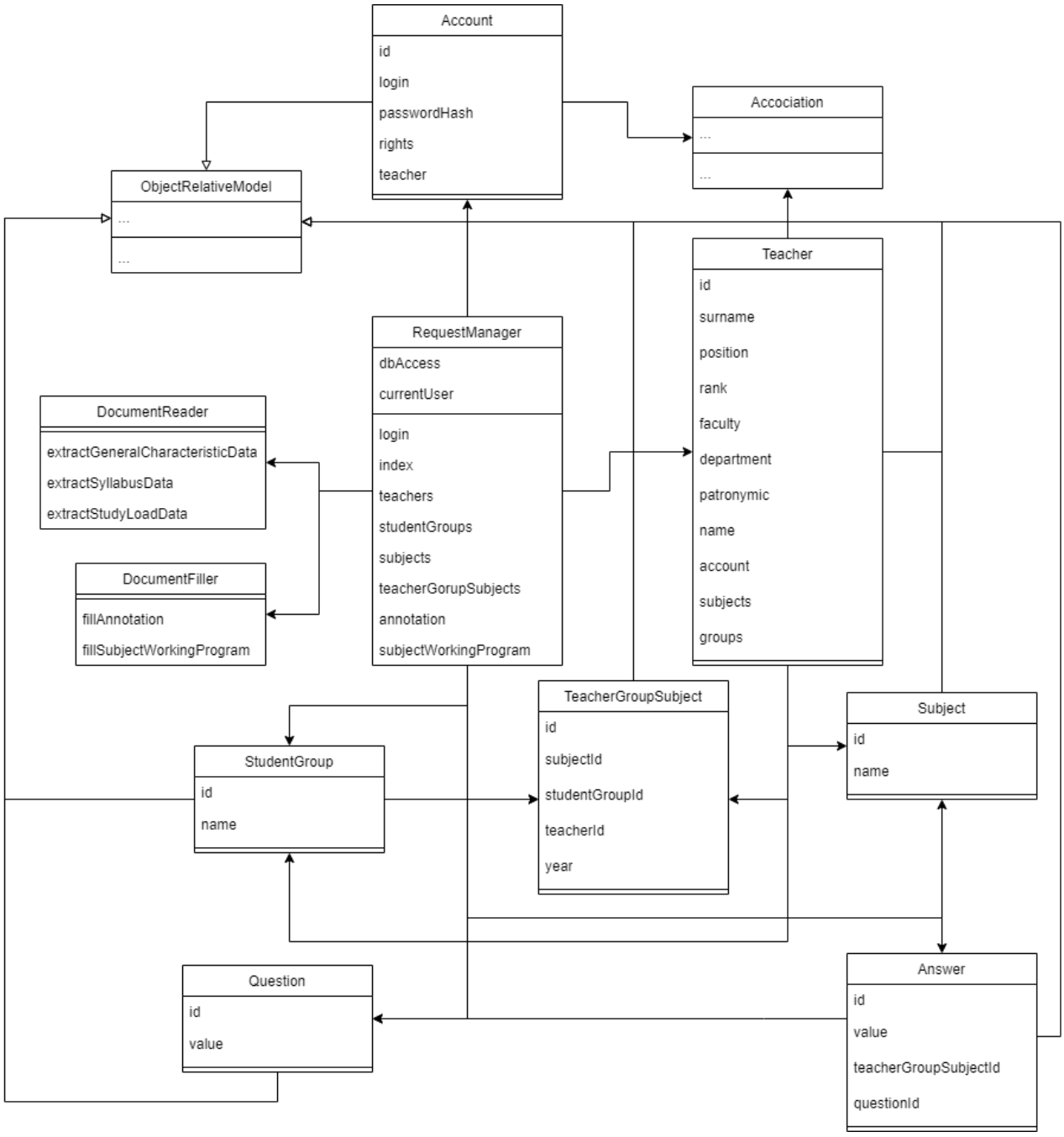



Рисунок 10. Диаграмма классов

Как видно из диаграммы, система состоит из четырех основных компонентов (групп компонентов):

1. Менеджер запросов (RequestManager)
   - *Данный компонент представляет собой ядро web-сервера. Он определяет вид html-страницы и является посредникам между web-интерфейсом и другими компонентами сервера*
2. Модуль извлечения данных (DocumentReader)
   - *Отвечает за прочтение содержимого документа и отбор значений в соответствии с запрошенным методом.*
3. Модуль генерации документов (DocumentFiller)
   - *Занимается заполнением документов предоставленными для ввода данными, в соответствии с запрошенным методом.*
4. Модели реляционных данных (ORM-objects)
   - *Являются посредниками между БД и программным кодом. ORM-компоненты позволяют манипулировать данными БД как классическими программными объектами. Это очень удобно, так как из кода исключаются SQL-запросы.*

Самое полезное в диаграмме классов, то, что в ней однозначно определяется реализация системы. Представленные в диаграмме классов компоненты являются идентичными структурам объектно-ориентированных языков.

### 3.3 Диаграмма развертывания

Диаграмма развертывания представляет физическое представление, то есть, определяет из каких аппаратных средств будет состоять целевая система и какие технологии взаимодействия будет использовать.

В рамках данной работы представлено три диаграммы. Так как система предполагается легко масштабируемой за счет микросервисной архитектуры, следует предусмотреть несколько сценариев развертывания: для одного сервера с одним экземпляром приложения, для одного сервера с несколькими экземплярами приложения и для нескольких серверов соответственно.

Ниже представлены соответственные диаграммы.

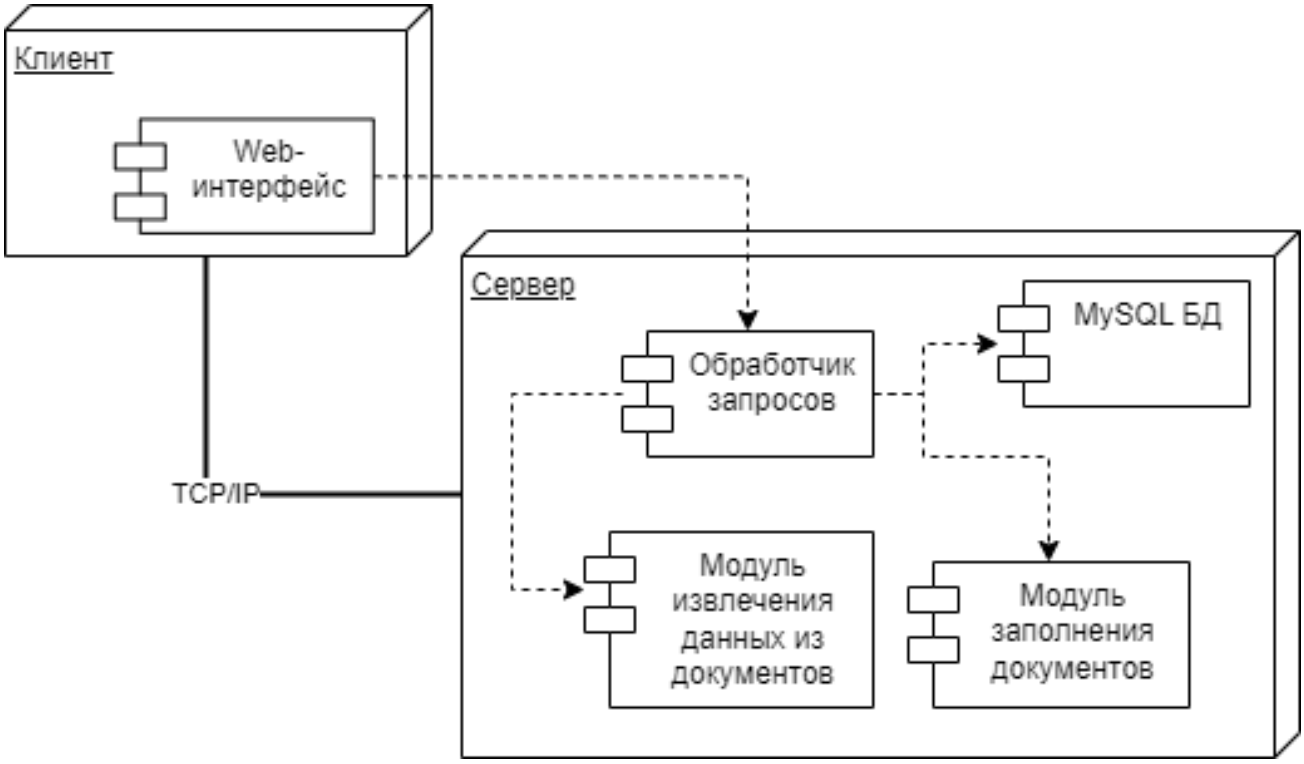



Рисунок 11. Диаграмма развертывания для одного сервера с одним экземпляром приложения

На рисунках представленных ниже присутствует артефакт «Контроллер запросов» - он служит для распределения и переренаправления запросов свободным приложениям или серверам (в качестве контроллера может выступать Nginx или Apache2, например) и узел «Контейнер» - он изолирует программы от

других процессов в системе (чаще всего для создания контейнеров используется Docker)

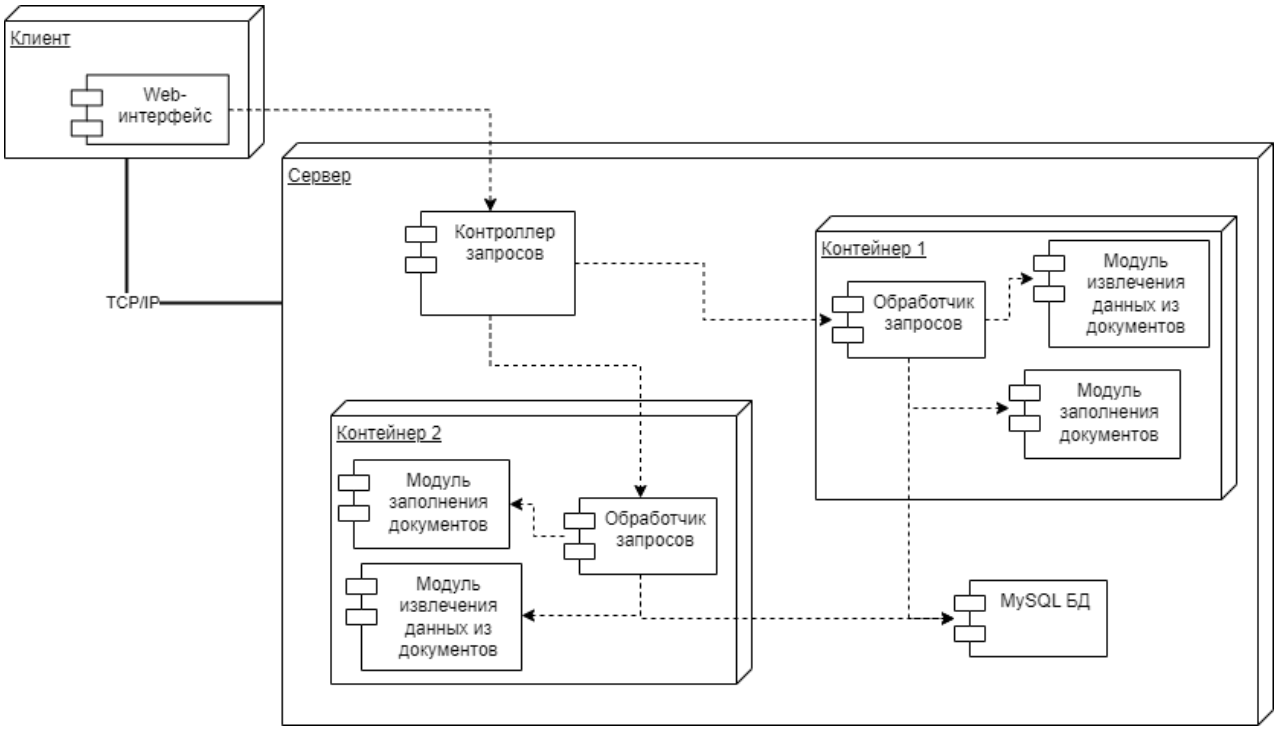



Рисунок 12. Диаграмма развертывания для одного сервера с несколькими экземплярами приложения

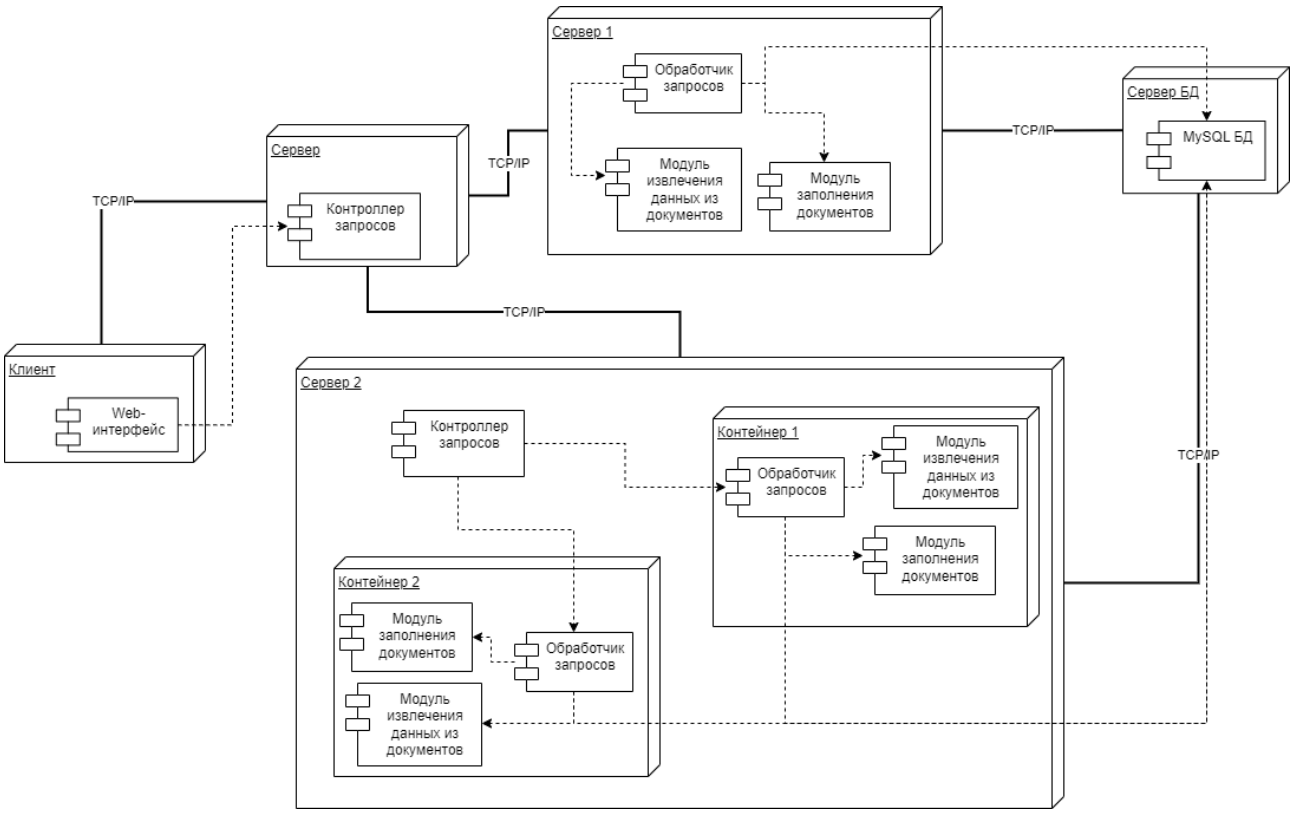



Рисунок 13. Диаграмма развертывания для нескольких серверов

## 3.4 Проектирование БД

И последним шагом для проектирования целевого прототипа будет составление схемы БД, так как информационная система по определению предполагает использование базы данных. Представленная ниже схема БД составлена независимо от языка запросов и совместима с ORM из диаграммы классов.

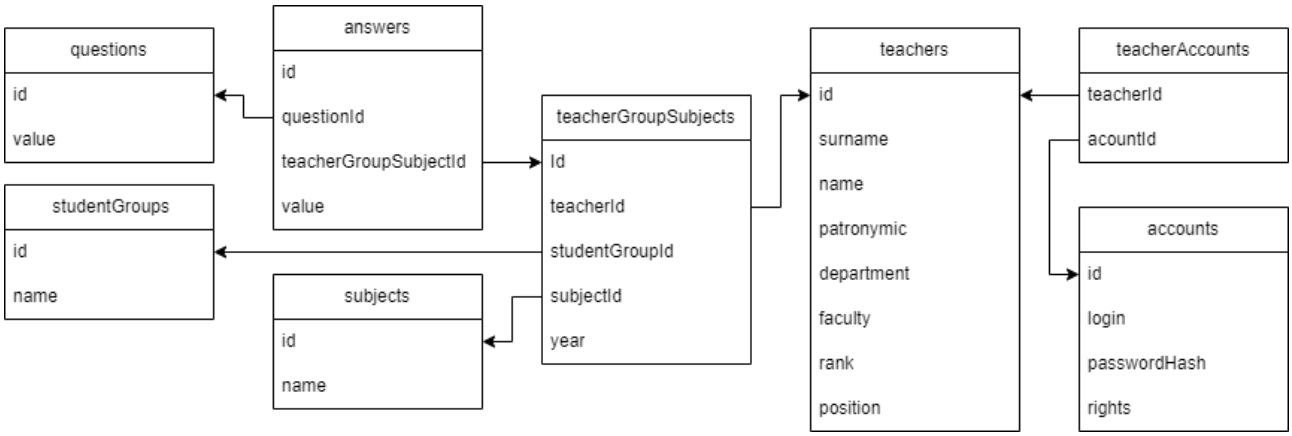



Рисунок 14. Схема БД

# 4 Реализация

## 4.1 Выбор языка программирования

Для реализации проекта требуется определит инструменты, при помощи которых будет производиться разработка. Некоторые разработчики рекомендуют начать с выбора фреймворка – это хороший путь, но не единственный верный. Фреймворки разрабатываются под конкретные задачи, с учетом специфики языка. В рамках данной работы в первую очередь будет учитываться специфика языка программирования. Хотя в идеале, нужно выбирать фреймворк совместно с языком программирования с учетом специфики поставленных задач.

Выбор языка программирования в первую очередь определяется задачами, которые, собственно, этим языком необходимо будет решить. Согласно требованиям, разрабатывается web-приложение, а значит, для разработки подойдут такие языки, как: Java, JavaScript, C#, Ruby и Python.

Помимо web-части будет разработаны еще и прикладные модули, в таком случае из перечня исключается JavaScript и Ruby.

Далее стоит учесть, что от программы требуется аппаратная независимость, но Java, C# и Python – это языки кроссплатформенной разработки, а значит являются подходящими.

В рамках данной работы выбор пал на Python, так как это простой язык с множеством полезных библиотек, которые легко подключаются к любому проекту, в любой операционной системе.

## 4.2 Выбор фреймворка

Для языка web-разработки на языке Python существует несколько десятков подходящих фреймворков: Django, CherryPy, Pyramid, Flask и много других. Все они решают примерно одинаковые задачи и схожи в использовании.

Для реализации проекта требуется встроенная система авторизации и аутентификации, такая есть и в Djangoб, и в CherryPy, и в Pyramid, и в Flask.

Далее требуется ORM-система. Вот ORM, из перечисленных выше фреймворков есть только в Django и Flask.

Самым популярным фреймворком для web-разработки в Python на сегодняшний день является Django. Это хороший фреймворк с открытым исходным кодом и широким функционалом. Но, в рамках данной работы будет использоваться фреймворк Flask, так как это самый легкий фреймворк среди многофункциональных, как в плане освоения, так и в плане потребления аппаратных ресурсов. Что достигается благодаря упрощенной структуре и концепции расширяемости, то есть фреймворк изначально поставляется с ограниченным функционалом, а расширяется благодаря загрузке и подключению дополнительных компонентов по мере необходимости.

## 4.3 Выбор БД

Учитывая, что для Flask существует модуль SQLAlchemy, который позволяет работать с любой базой данных через реляционные объекты, то выбор базы данных становится не так важен. Но, всё же есть некоторые аспекты, которые стоит учесть.

Целевое приложение, прототип, рассчитано, по крайней мере, пока что на одну организацию, и не будет сильно нагружаться запросами, а значит тяжелые базы данных вроде Postgres использовать не совсем целесообразно, так как такие базы данных скорее предназначены для масштабных высоконагруженных систем, с соответственными аппаратными требованиями.

С другой стороны, есть вероятность, что потребуется перенести базу данных на отдельный сервер, таким образом отпадают такие средства как SQLite.

Исходя из полученных требований к базе данных, оптимальным решением будет использовать MySQL. В рамках данного проекта будет использоваться одна из версий SQLite, под названием MariaDB. Данное средство подходит как для малых, так и для масштабных информационных структур и имеет изначально низкие системные требования.

## 4.4 Проектирование Web-интерфейсов

В Flask используется удобный механизм создания шаблонов и генератор html-страниц Jinja2. Это очень легкий в использовании модуль, который идет в комплекте с основными компонентами Flask. Хотя и используется генератор html-кода, писать на html всё же придется.

Помимо прочего в рамках данной работы будет использоваться bootstrap5 для создания элементов с предварительно прописанными стилями. В flask есть bootstrap3, но это довольно старая версия, в ней не реализованы некоторые требуемые элементы. Поэтому придется отдельно догрузить модуль flask-bootstrap в котором есть подходящая версия bootstrap.

После подключения всех необходимых модулей удалось создать следующий шаблон:


Рисунок 15. Базовый шаблон web-интерфейса

Данный шаблон может быть использован в других шаблонах как исходный, тогда будут наследоваться его элементы, что позволит существенно сократить количество кода в проекте.

Теперь можно приступить к созданию остальных шаблонов. Предварительно требуется создать следующие шаблоны:

1. Форма администрирования списка групп
2. Форма администрирования списка предметов
3. Форма администрирования списка преподавателей
4. Форма выбора шаблонов и загрузки
5. Форма заполнения шаблона Аннотации
6. Форма заполнения шаблона РПД
7. Форма авторизации

### 4.4.1 Форма администрирования списка групп

Данная форма предназначена для загрузки учебных планов, а также для редактирования списка групп в случае, если наименования групп по тем или иным причинам были считаны некорректно.

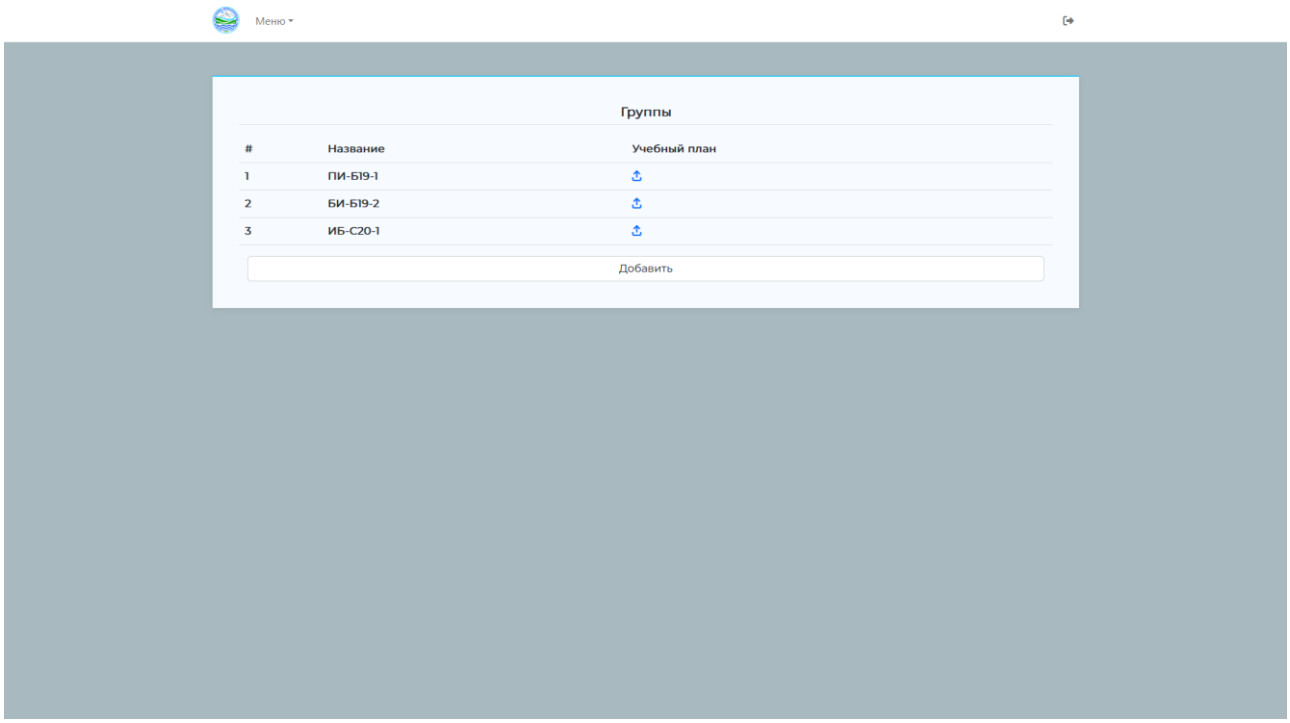



Рисунок 16. Форма администрирования списка групп

### 4.4.2 Форма администрирования списка предметов

Как и форма с группами, форма администрирования предназначена для выгрузки документа и редактирования значений при необходимости.

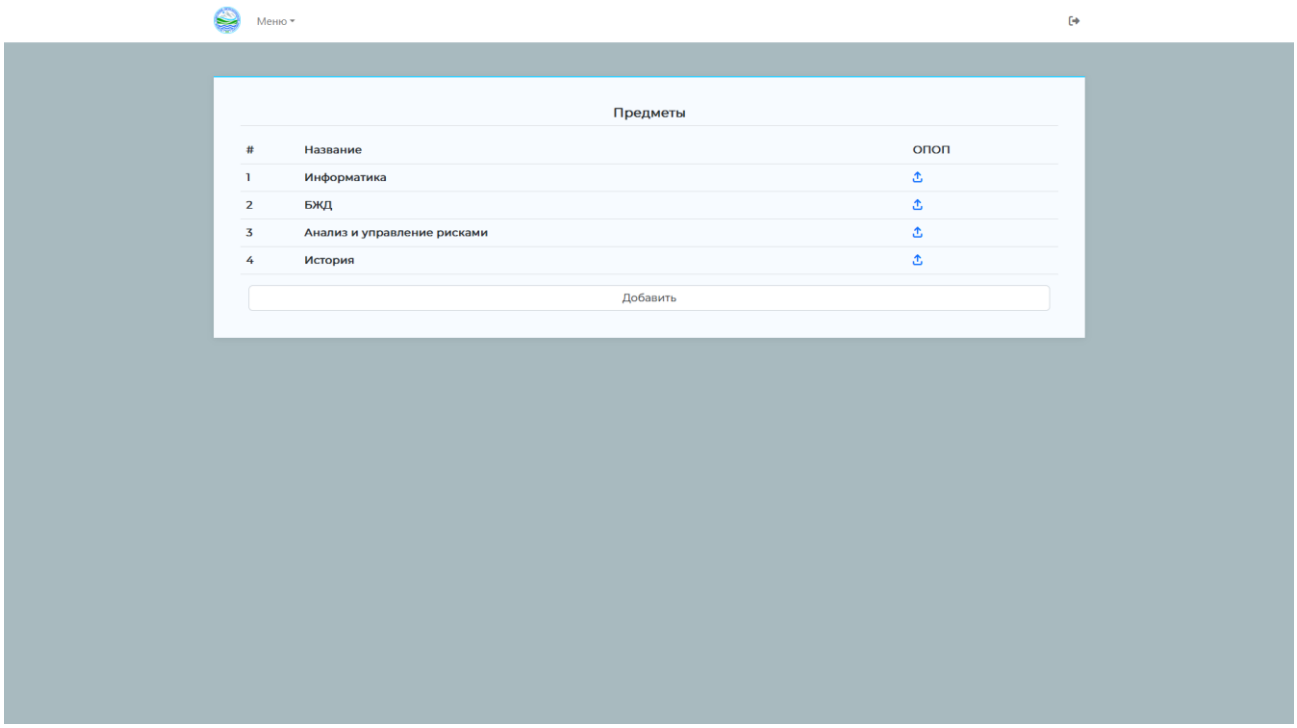



Рисунок 17. Форма администрирования списка предметов.

### 4.4.3 Форма администрирования списка преподавателей

Форма администрирования списка преподавателей содержит основную информацию о преподавателях и позволяет обновлять и удалять существующие учетные записи, также добавлять новых преподавателей, кроме того, в данной форме отображается статус заполнения всех шаблонов документов. При переходе по ссылке статуса пользователь перенаправляется на страницу с выбором шаблонов для загрузки. Данная форма доступна только администратору. Стоит обратить внимание на то, что администратор имеет возможность редактировать ответы в формах любого преподавателя и соответственно имеет права на загрузку итоговых документов для любой группы любой дисциплины.

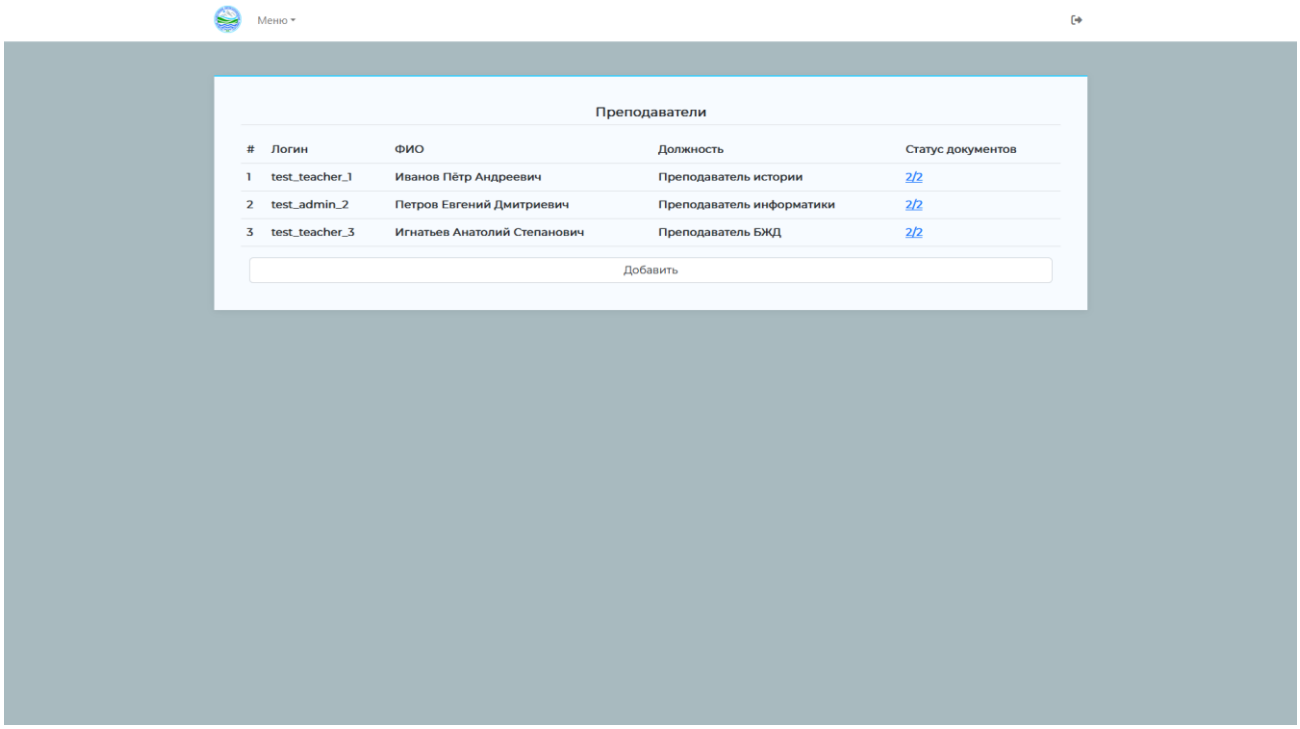



Рисунок 18. Форма администрирования списка преподавателей

### 4.4.4 Форма выбора шаблонов и загрузки

Форма выбора шаблонов и загрузки позволяет переходить к заполнению форм соответственных документов. Кроме того, для готовых документов предоставляется возможность загрузки документа с сервера.

Стоит отметить, что преподаватель не может редактировать данный список. Права на внесение изменений имеются только у администратора, так как данный список заполняется автоматически при выгрузке насервер учебного плана.

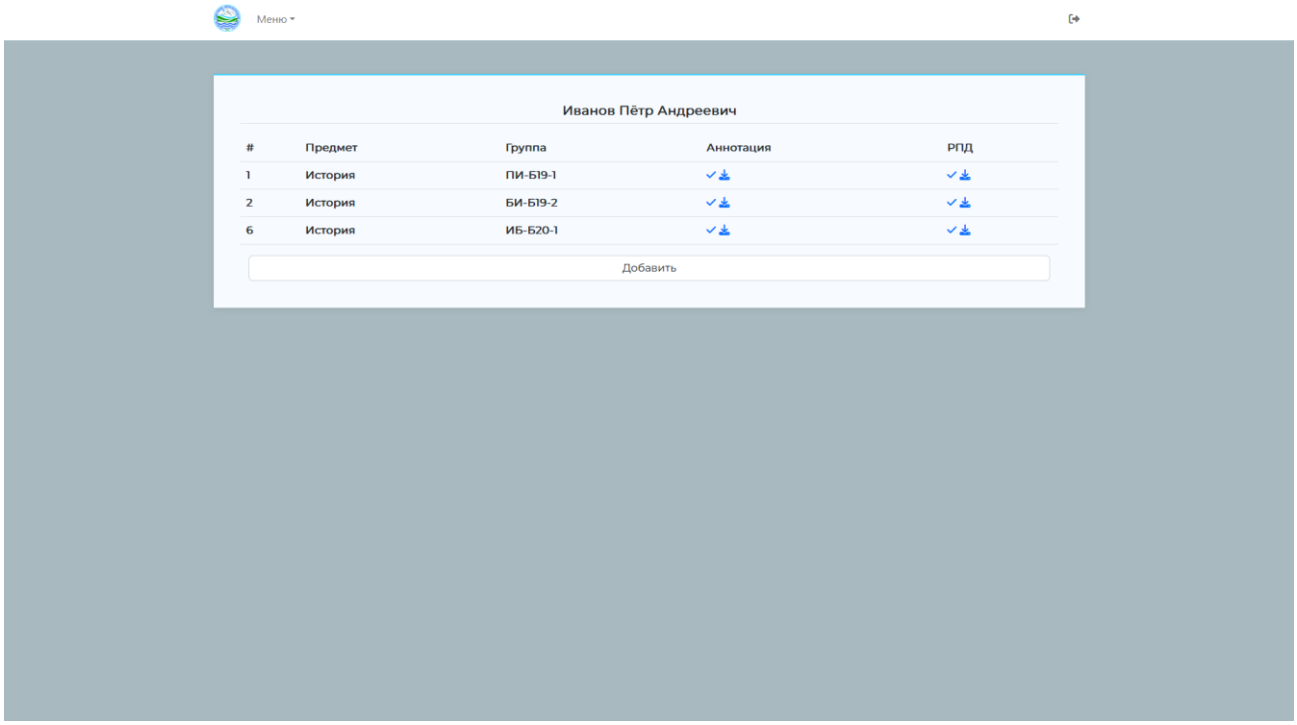



Рисунок 19. Форма выбора шаблонов и загрузки

### 4.4.5 Форма заполнения шаблона аннотации

Форма заполнения шаблона аннотации запрашивает соответственную недостающую информацию для заполнения аннотации. Также результаты, внесенные в данную форму, будут использоваться для формирования рабочей программы дисциплины. Без заполнения этой формы переход на форму РПД невозможен.

При полном заполнении формы, и нажатии на кнопку сохранить пользователю будет предложено загрузить документ.

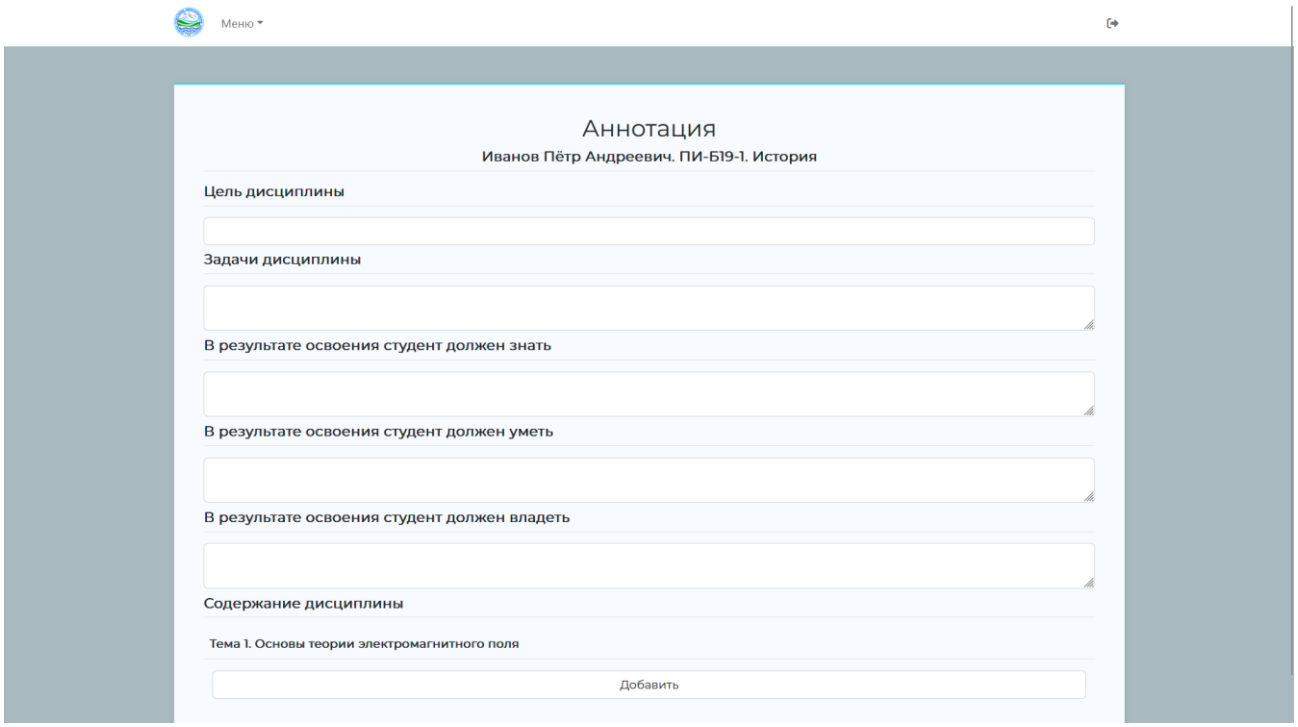



Рисунок 20. Форма заполнения шаблона аннотации.

### 4.4.6 Форма заполнения шаблона РПД

Как было описано в форме аннотации, для перехода на форму рабочей программы дисциплины требуется заполненная форма аннотации, в противном случае переход на данную форму невозможен.

Как и для формы с аннотацией по окончанию заполнения, пользователю будет предложено загрузить документ.

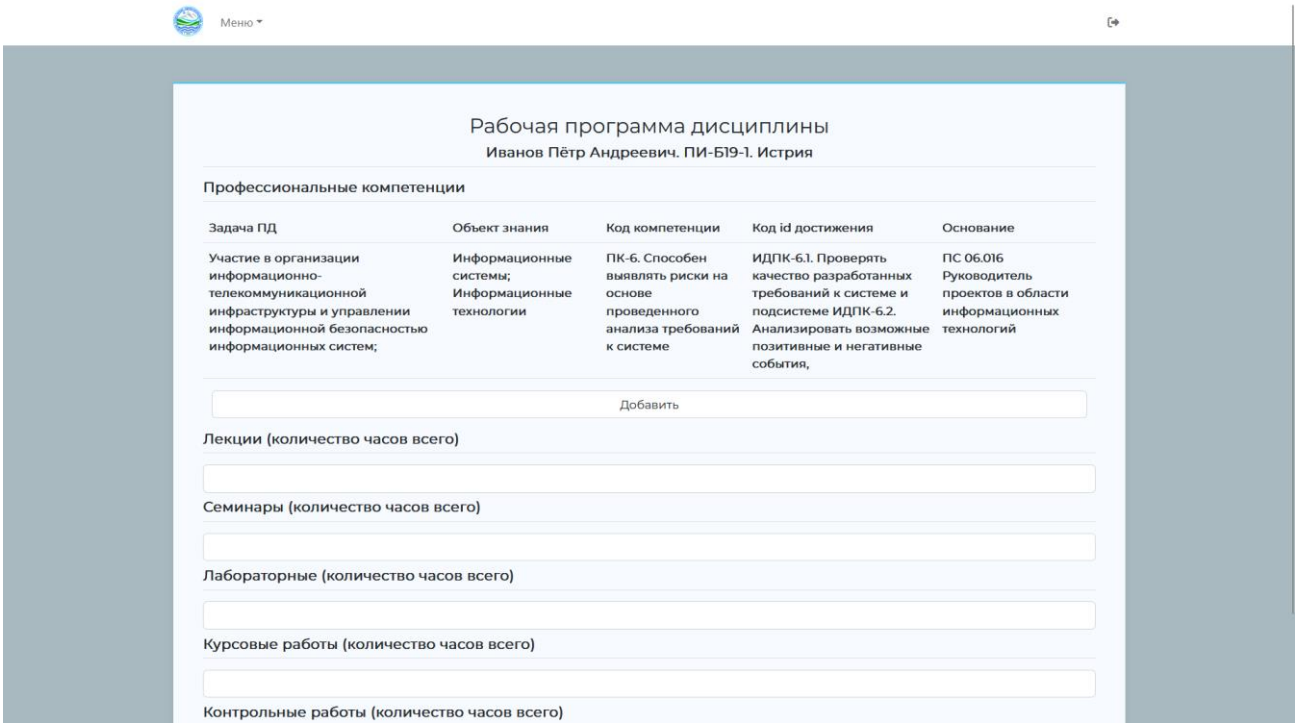



Рисунок 21. Форма заполнения РПД

### 4.4.7 Форма авторизации

Для определения прав пользователя, а также личной информации, если имеется, требуется авторизация. Flask имеет все необходимые компоненты для выполнения авторизации. Остается только определить форму. Так как в системе всегда будет один пользователь с правами разработчика (Не удаляемый администратор), то регистрация производится только из интерфейса администратора, путем создания учетной записи для преподавателя. Таким образом форма для регистрации не требуется.

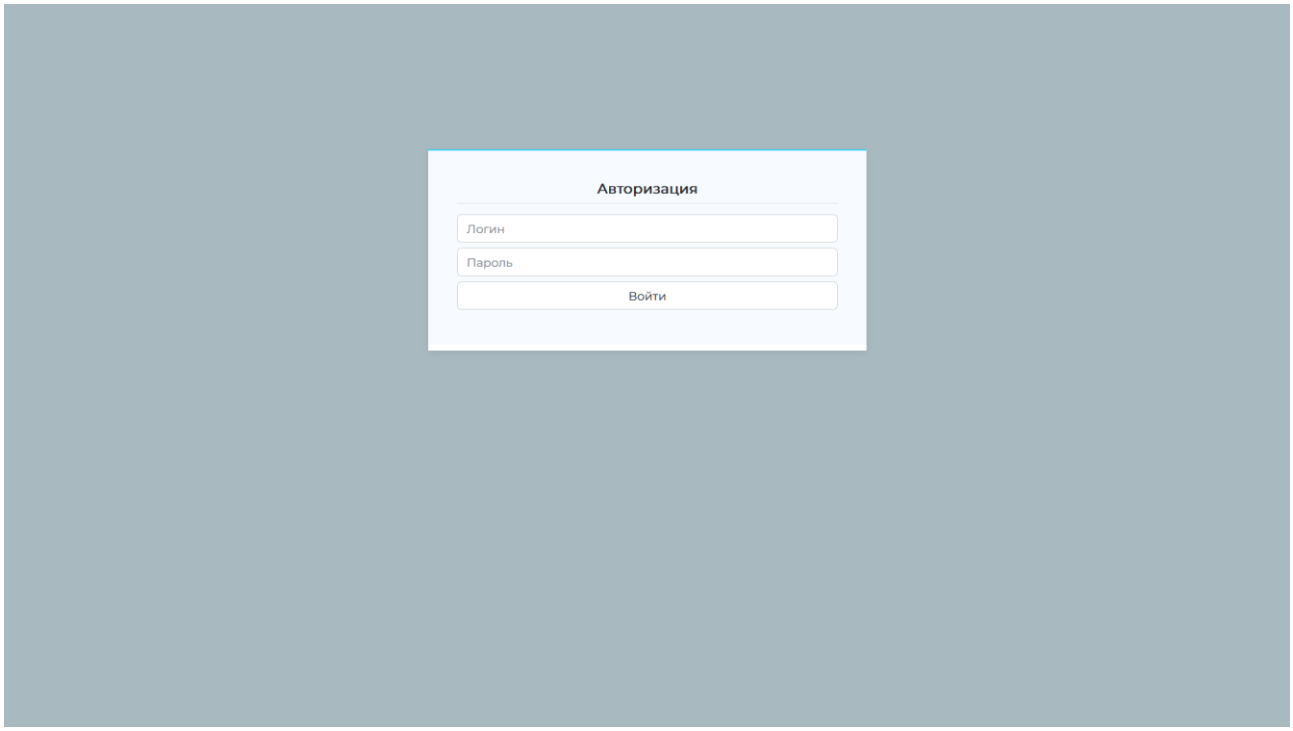



Рисунок 22. Форма авторизации

## 4.5 Реализация ORM-объектов

ORM – объекты определяются в коде, затем при запуске программы Flask пытается соотнести все объекты, которые обнаруживает с соответственными таблицами в базе данных. Если соотнести не удалось, то Flask попытается создать недостающие таблицы.

В соответствии со схемой БД были созданы следующие модели:

```
# Файл models.py

from typing import List
from sqlalchemy import ForeignKey
from sqlalchemy.orm import DeclarativeBase, Mapped, mapped_column,  relationship

# Базовый класс ORM
class Base(DeclarativeBase):
    pass

# Буффер названий таблиц
class TableName:
    Account = 'Accounts'
    Teacher = 'Teachers'
    TeacherAccount = 'TeacherAccounts'
    StudentGroup = 'StudentGroups'
    Subject = 'Subjects'
    TeacherGroupSubject = 'TeacherGroupSubjects'
    Question = 'Questions'
    Answer = 'Answers'

class Account(Base):
    __tablename__ = TableName.Account

    id: Mapped[int] = mapped_column(primary_key = True)
    login: Mapped[str] = mapped_column(nullable = False)
    passwordHash: Mapped[str] = mapped_column(nullable = False)
    rights: Mapped[str] = mapped_column(nullable = False)

    teacher: Mapped['Teacher'] = relationship(back_populates = 'account',
            secondary = TableName.TeacherAccount)
```

```python
class Teacher(Base):
    __tablename__ = TableName.Teacher

    id: Mapped[int] = mapped_column(primary_key = True)
    surname: Mapped[str] = mapped_column(nullable = False)
    name: Mapped[str] = mapped_column(nullable = False)
    patronymic: Mapped[str] = mapped_column(nullable = False)
    department: Mapped[str] = mapped_column(nullable = False)
    faculty: Mapped[str] = mapped_column(nullable = False)
    rank: Mapped[str] = mapped_column(nullable = False)
    position: Mapped[str] = mapped_column(nullable = False)

    account: Mapped['Teacher'] = relationship(back_populates = 'teacher',
            secondary = TableName.TeacherAccount)
    studentGroups: Mapped['StudentGroup'] = relationship(
            back_populates = 'teachers',
            secondary = TableName.TeacherGroupSubject)

class TeacherAccount(Base):
    __tablename__ = TableName.TeacherAccount

    acountId: Mapped[int] = mapped_column(ForeignKey(TableName.Account + '.id'),
            primary_key = True)
    teacherId: Mapped[int] = mapped_column(ForeignKey(TableName.Teacher + '.id'),
            primary_key = True)

class StudentGroup(Base):
    __tablename__ = TableName.StudentGroup

    id: Mapped[int] = mapped_column(primary_key = True)
    name: Mapped[str] = mapped_column(nullable = False)

class Subject(Base):
    __tablename__ = TableName.Subject

    id: Mapped[int] = mapped_column(primary_key = True)
    name: Mapped[str] = mapped_column(nullable = False)
```

```
class TeacherGroupSubject(Base):
    __tablename__ = TableName.TeacherGroupSubject

    id: Mapped[int] = mapped_column(primary_key = True)
    teacherId: Mapped[int] = mapped_column(
            ForeignKey(TableName.Teacher + '.id'), nullable = False)
    studentGroupId: Mapped[int] = mapped_column(
            ForeignKey(TableName.StudentGroup + '.id'),
            nullable = False)
    subjectId: Mapped[int] = mapped_column(
            ForeignKey(TableName.Subject + '.id'), nullable = False)
    year: Mapped[int] = mapped_column(nullable = False)

    answers: Mapped[List['Answer']] = relationship()

class Question(Base):
    __tablename__ = TableName.Question

    id: Mapped[int] = mapped_column(primary_key = True)
    name: Mapped[str] = mapped_column(nullable = False)

class Answer(Base):
    __tablename__ = TableName.Answer

    id: Mapped[int] = mapped_column(primary_key = True)
    questionId: Mapped[int] = mapped_column(
            ForeignKey(TableName.Question + '.id'), nullable = False)
    teacherGroupSubjectId: Mapped[int] = mapped_column(
            ForeignKey(TableName.TeacherGroupSubject + '.id'),
            nullable = False)
    value: Mapped[str] = mapped_column(nullable = False)

    question: Mapped['Question'] = relationship()
```

## 4.6 Определение извлекаемых данных

Для работы с документами Microsoft для питона были созданы следующие библиотеки: openpyxl и python-docx для Excel и Word соответственно. Они позволяют как создавать, так и редактировать документы Microsoft.

Первым делом следует определить элемента подлежащие считыванию. Потребуются следующие документы:

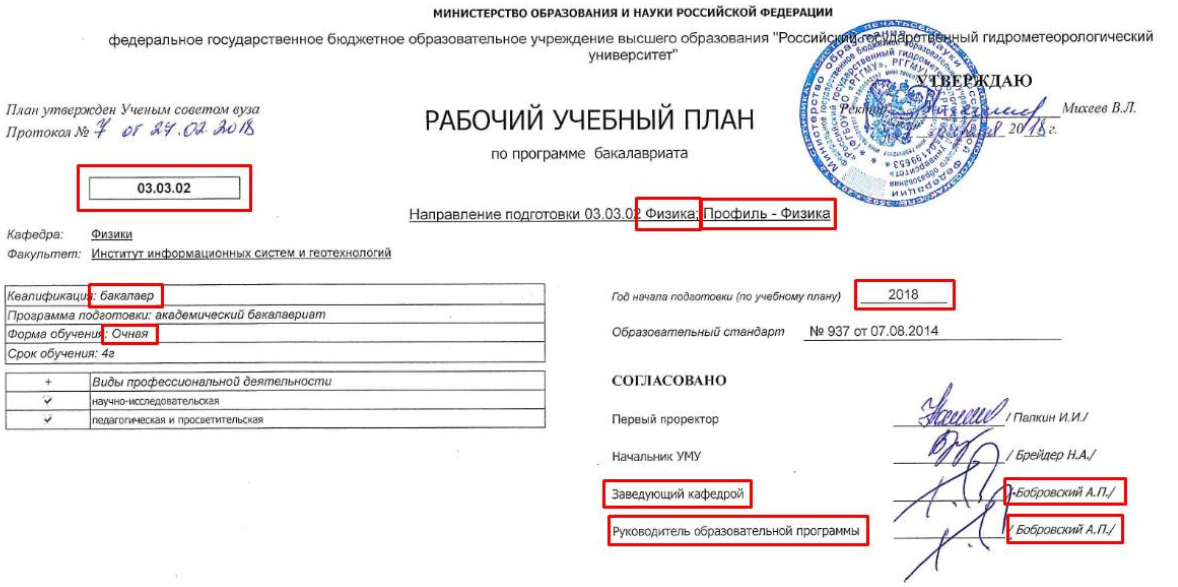
- Учебный план
- Учебная нагрузка
- ОХОП

В будущем данные из этих документов следует сохранять в базу данных, но, для прототипа будет достаточно просто сохранить значения в текстовом файле, .

### 4.6.1 Учебный план

Учебный план — это xlsx-файл, значит извлекать данные должно быть довольно просто. Для этого потребуется определить положение нужных ячеек, затем просто извлечь данные. Из учебного плана требуются 3 листа:

1. Титул
2. План
3. Компетенции



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования "Российский государственный гидрометеорологический университет"

УТВЕРЖДАЮ

Ректор Михеев В.Л.

План утвержден Ученым советом вуза
Протокол № 7 от 27.02.2018

РАБОЧИЙ УЧЕБНЫЙ ПЛАН

по программе бакалавриата

03.03.02

Направление подготовки 03.03.02 Физика; Профиль - Физика

Кафедра: Физики

Факультет: Институт информационных систем и геотехнологий

| |
|---|
| Квалификация: бакалавр |
| Программа подготовки: академический бакалавриат |
| Форма обучения: Очная |
| Срок обучения: 4г |

| | |
|---|---|
| + | Виды профессиональной деятельности |
| ✓ | научно-исследовательская |
| ✓ | педагогическая и просветительская |

Год начала подготовки (по учебному плану) 2018

Образовательный стандарт № 937 от 07.08.2014

СОГЛАСОВАНО

Первый проректор / Палкин И.И./

Начальник УМУ / Брейдер Н.А./

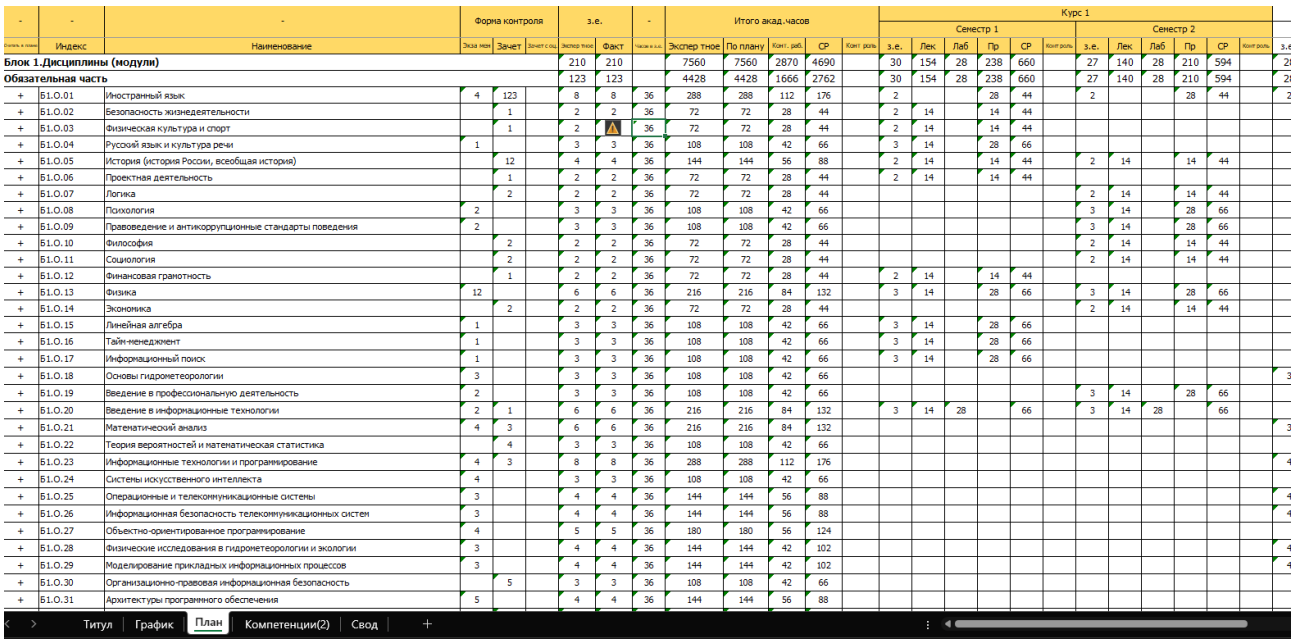
Заведующий кафедрой / Бобровский А.П./

Руководитель образовательной программы / Бобровский А.П./

Рисунок 23. Учебный план титульный лист

Из титульного листа необходимо взять

- Код дисциплины
- Направление подготовки
- Профиль
- Год начала обучения
- Фамилию и инициалы заведующего кафедрой
- Фамилию и инициалы руководителя образовательной программы

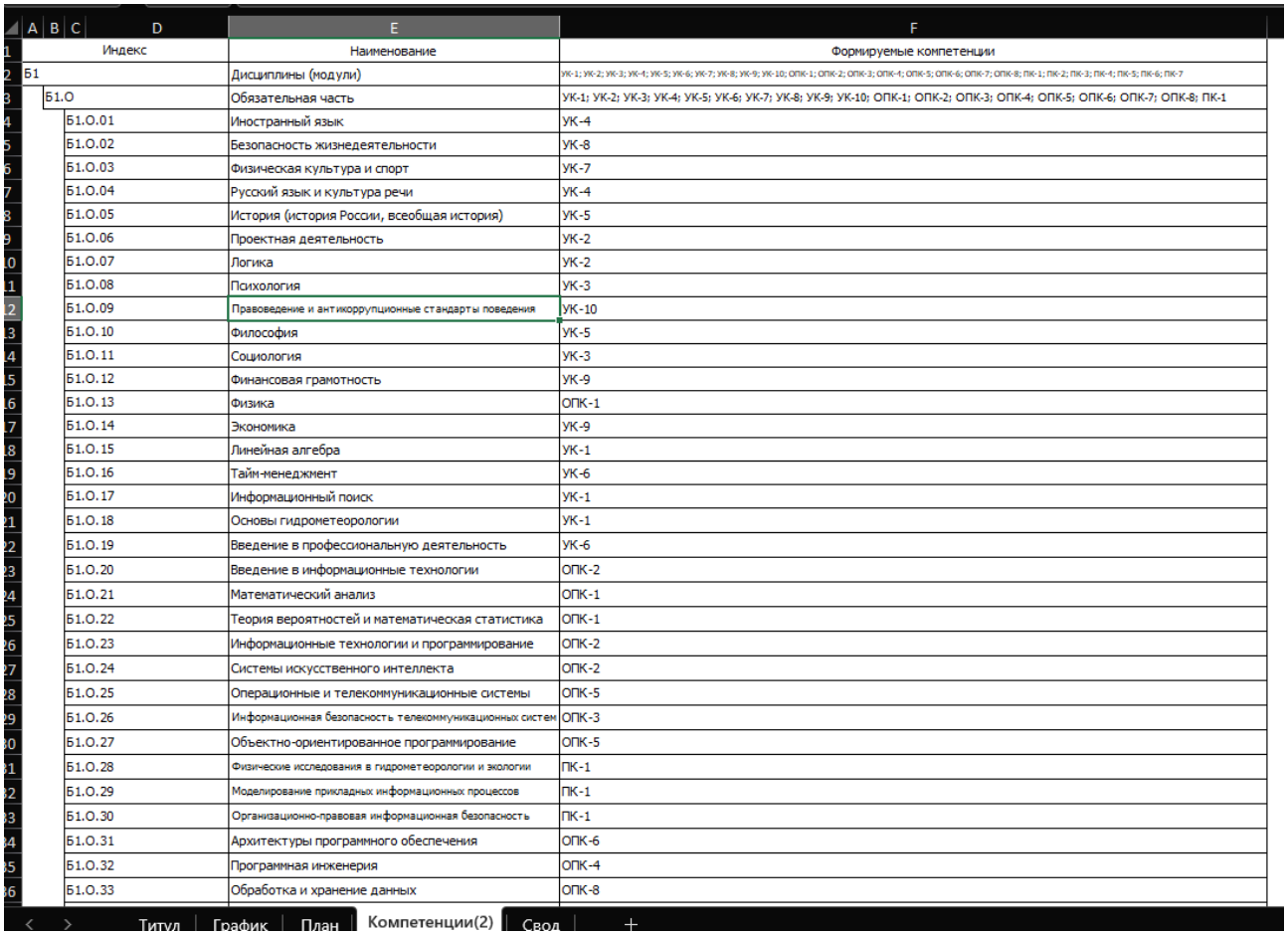


| - | - | - | Форма контроля | | | з.е. | | - | Итого акад.часов | | | | | Курс 1 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | Семестр 1 | | | | | | Семестр 2 | | | | | |
| Скачать в плане | Индекс | Наименование | Экза мен | Зачет | Зачет с оц | Экспер тное | Факт | Часов в з.е. | Экспер тное | По плану | Конт. раб. | СР | Конт роль | з.е. | Лек | Лаб | Пр | СР | Конт роль | з.е. | Лек | Лаб | Пр | СР | Конт роль |
| **Блок 1.Дисциплины (модули)** | | | | | | 210 | 210 | | 7560 | 7560 | 2870 | 4690 | | 30 | 154 | 28 | 238 | 660 | | 27 | 140 | 28 | 210 | 594 | |
| **Обязательная часть** | | | | | | 123 | 123 | | 4428 | 4428 | 1666 | 2762 | | 30 | 154 | 28 | 238 | 660 | | 27 | 140 | 28 | 210 | 594 | |
| + | Б1.О.01 | Иностранный язык | 4 | 123 | | 8 | 8 | 36 | 288 | 288 | 112 | 176 | | 2 | | | 28 | 44 | | 2 | | | 28 | 44 | |
| + | Б1.О.02 | Безопасность жизнедеятельности | | 1 | | 2 | 2 | 36 | 72 | 72 | 28 | 44 | | 2 | 14 | | 14 | 44 | | | | | | | |
| + | Б1.О.03 | Физическая культура и спорт | | 1 | | 2 | ⚠ | 36 | 72 | 72 | 28 | 44 | | 2 | 14 | | 14 | 44 | | | | | | | |
| + | Б1.О.04 | Русский язык и культура речи | 1 | | | 3 | 3 | 36 | 108 | 108 | 42 | 66 | | 3 | 14 | | 28 | 66 | | | | | | | |
| + | Б1.О.05 | История (история России, всеобщая история) | | 12 | | 4 | 4 | 36 | 144 | 144 | 56 | 88 | | 2 | 14 | | 14 | 44 | | 2 | 14 | | 14 | 44 | |
| + | Б1.О.06 | Проектная деятельность | | 1 | | 2 | 2 | 36 | 72 | 72 | 28 | 44 | | 2 | 14 | | 14 | 44 | | | | | | | |
| + | Б1.О.07 | Логика | | 2 | | 2 | 2 | 36 | 72 | 72 | 28 | 44 | | | | | | | | 2 | 14 | | 14 | 44 | |
| + | Б1.О.08 | Психология | 2 | | | 3 | 3 | 36 | 108 | 108 | 42 | 66 | | | | | | | | 3 | 14 | | 28 | 66 | |
| + | Б1.О.09 | Правоведение и антикоррупционные стандарты поведения | 2 | | | 3 | 3 | 36 | 108 | 108 | 42 | 66 | | | | | | | | 3 | 14 | | 28 | 66 | |
| + | Б1.О.10 | Философия | | 2 | | 2 | 2 | 36 | 72 | 72 | 28 | 44 | | | | | | | | 2 | 14 | | 14 | 44 | |
| + | Б1.О.11 | Социология | | 2 | | 2 | 2 | 36 | 72 | 72 | 28 | 44 | | | | | | | | 2 | 14 | | 14 | 44 | |
| + | Б1.О.12 | Финансовая грамотность | | 1 | | 2 | 2 | 36 | 72 | 72 | 28 | 44 | | 2 | 14 | | 14 | 44 | | | | | | | |
| + | Б1.О.13 | Физика | 12 | | | 6 | 6 | 36 | 216 | 216 | 84 | 132 | | 3 | 14 | | 28 | 66 | | 3 | 14 | | 28 | 66 | |
| + | Б1.О.14 | Экономика | | 2 | | 2 | 2 | 36 | 72 | 72 | 28 | 44 | | | | | | | | 2 | 14 | | 14 | 44 | |
| + | Б1.О.15 | Линейная алгебра | 1 | | | 3 | 3 | 36 | 108 | 108 | 42 | 66 | | 3 | 14 | | 28 | 66 | | | | | | | |
| + | Б1.О.16 | Тайм-менеджмент | 1 | | | 3 | 3 | 36 | 108 | 108 | 42 | 66 | | 3 | 14 | | 28 | 66 | | | | | | | |
| + | Б1.О.17 | Информационный поиск | 1 | | | 3 | 3 | 36 | 108 | 108 | 42 | 66 | | 3 | 14 | | 28 | 66 | | | | | | | |
| + | Б1.О.18 | Основы гидрометеорологии | 3 | | | 3 | 3 | 36 | 108 | 108 | 42 | 66 | | | | | | | | | | | | | 3 |
| + | Б1.О.19 | Введение в профессиональную деятельность | 2 | | | 3 | 3 | 36 | 108 | 108 | 42 | 66 | | | | | | | | 3 | 14 | | 28 | 66 | |
| + | Б1.О.20 | Введение в информационные технологии | 2 | 1 | | 6 | 6 | 36 | 216 | 216 | 84 | 132 | | 3 | 14 | 28 | | 66 | | 3 | 14 | 28 | | 66 | |
| + | Б1.О.21 | Математический анализ | 4 | 3 | | 6 | 6 | 36 | 216 | 216 | 84 | 132 | | | | | | | | | | | | | 3 |
| + | Б1.О.22 | Теория вероятностей и математическая статистика | | 4 | | 3 | 3 | 36 | 108 | 108 | 42 | 66 | | | | | | | | | | | | | |
| + | Б1.О.23 | Информационные технологии и программирование | 4 | 3 | | 8 | 8 | 36 | 288 | 288 | 112 | 176 | | | | | | | | | | | | | 4 |
| + | Б1.О.24 | Системы искусственного интеллекта | 4 | | | 3 | 3 | 36 | 108 | 108 | 42 | 66 | | | | | | | | | | | | | |
| + | Б1.О.25 | Операционные и телекоммуникационные системы | 3 | | | 4 | 4 | 36 | 144 | 144 | 56 | 88 | | | | | | | | | | | | | 4 |
| + | Б1.О.26 | Информационная безопасность телекоммуникационных систем | 3 | | | 4 | 4 | 36 | 144 | 144 | 56 | 88 | | | | | | | | | | | | | 4 |
| + | Б1.О.27 | Объектно-ориентированное программирование | 4 | | | 5 | 5 | 36 | 180 | 180 | 56 | 124 | | | | | | | | | | | | | |
| + | Б1.О.28 | Физические исследования в гидрометеорологии и экологии | 3 | | | 4 | 4 | 36 | 144 | 144 | 42 | 102 | | | | | | | | | | | | | 4 |
| + | Б1.О.29 | Моделирование прикладных информационных процессов | 3 | | | 4 | 4 | 36 | 144 | 144 | 42 | 102 | | | | | | | | | | | | | 4 |
| + | Б1.О.30 | Организационно-правовая информационная безопасность | | 5 | | 3 | 3 | 36 | 108 | 108 | 42 | 66 | | | | | | | | | | | | | |
| + | Б1.О.31 | Архитектуры программного обеспечения | 5 | | | 4 | 4 | 36 | 144 | 144 | 56 | 88 | | | | | | | | | | | | | |

Титул | График | План | Компетенции(2) | Свод

Рисунок 24. Учебный план

Значения из листа с планом следует сохранить целиком, можно в csv-файле



| | A B C D | E | F |
|---|---|---|---|
| 1 | Индекс | Наименование | Формируемые компетенции |
| 2 | Б1 | Дисциплины (модули) | УК-1; УК-2; УК-3; УК-4; УК-5; УК-6; УК-7; УК-8; УК-9; УК-10; ОПК-1; ОПК-2; ОПК-3; ОПК-4; ОПК-5; ОПК-6; ОПК-7; ОПК-8; ПК-1; ПК-2; ПК-3; ПК-4; ПК-5; ПК-6; ПК-7 |
| 3 | Б1.О | Обязательная часть | УК-1; УК-2; УК-3; УК-4; УК-5; УК-6; УК-7; УК-8; УК-9; УК-10; ОПК-1; ОПК-2; ОПК-3; ОПК-4; ОПК-5; ОПК-6; ОПК-7; ОПК-8; ПК-1 |
| 4 | Б1.О.01 | Иностранный язык | УК-4 |
| 5 | Б1.О.02 | Безопасность жизнедеятельности | УК-8 |
| 6 | Б1.О.03 | Физическая культура и спорт | УК-7 |
| 7 | Б1.О.04 | Русский язык и культура речи | УК-4 |
| 8 | Б1.О.05 | История (история России, всеобщая история) | УК-5 |
| 9 | Б1.О.06 | Проектная деятельность | УК-2 |
| 10 | Б1.О.07 | Логика | УК-2 |
| 11 | Б1.О.08 | Психология | УК-3 |
| 12 | Б1.О.09 | Правоведение и антикоррупционные стандарты поведения | УК-10 |
| 13 | Б1.О.10 | Философия | УК-5 |
| 14 | Б1.О.11 | Социология | УК-3 |
| 15 | Б1.О.12 | Финансовая грамотность | УК-9 |
| 16 | Б1.О.13 | Физика | ОПК-1 |
| 17 | Б1.О.14 | Экономика | УК-9 |
| 18 | Б1.О.15 | Линейная алгебра | УК-1 |
| 19 | Б1.О.16 | Тайм-менеджмент | УК-6 |
| 20 | Б1.О.17 | Информационный поиск | УК-1 |
| 21 | Б1.О.18 | Основы гидрометеорологии | УК-1 |
| 22 | Б1.О.19 | Введение в профессиональную деятельность | УК-6 |
| 23 | Б1.О.20 | Введение в информационные технологии | ОПК-2 |
| 24 | Б1.О.21 | Математический анализ | ОПК-1 |
| 25 | Б1.О.22 | Теория вероятностей и математическая статистика | ОПК-1 |
| 26 | Б1.О.23 | Информационные технологии и программирование | ОПК-2 |
| 27 | Б1.О.24 | Системы искусственного интеллекта | ОПК-2 |
| 28 | Б1.О.25 | Операционные и телекоммуникационные системы | ОПК-5 |
| 29 | Б1.О.26 | Информационная безопасность телекоммуникационных систем | ОПК-3 |
| 30 | Б1.О.27 | Объектно-ориентированное программирование | ОПК-5 |
| 31 | Б1.О.28 | Физические исследования в гидрометеорологии и экологии | ПК-1 |
| 32 | Б1.О.29 | Моделирование прикладных информационных процессов | ПК-1 |
| 33 | Б1.О.30 | Организационно-правовая информационная безопасность | ПК-1 |
| 34 | Б1.О.31 | Архитектуры программного обеспечения | ОПК-6 |
| 35 | Б1.О.32 | Программная инженерия | ОПК-4 |
| 36 | Б1.О.33 | Обработка и хранение данных | ОПК-8 |

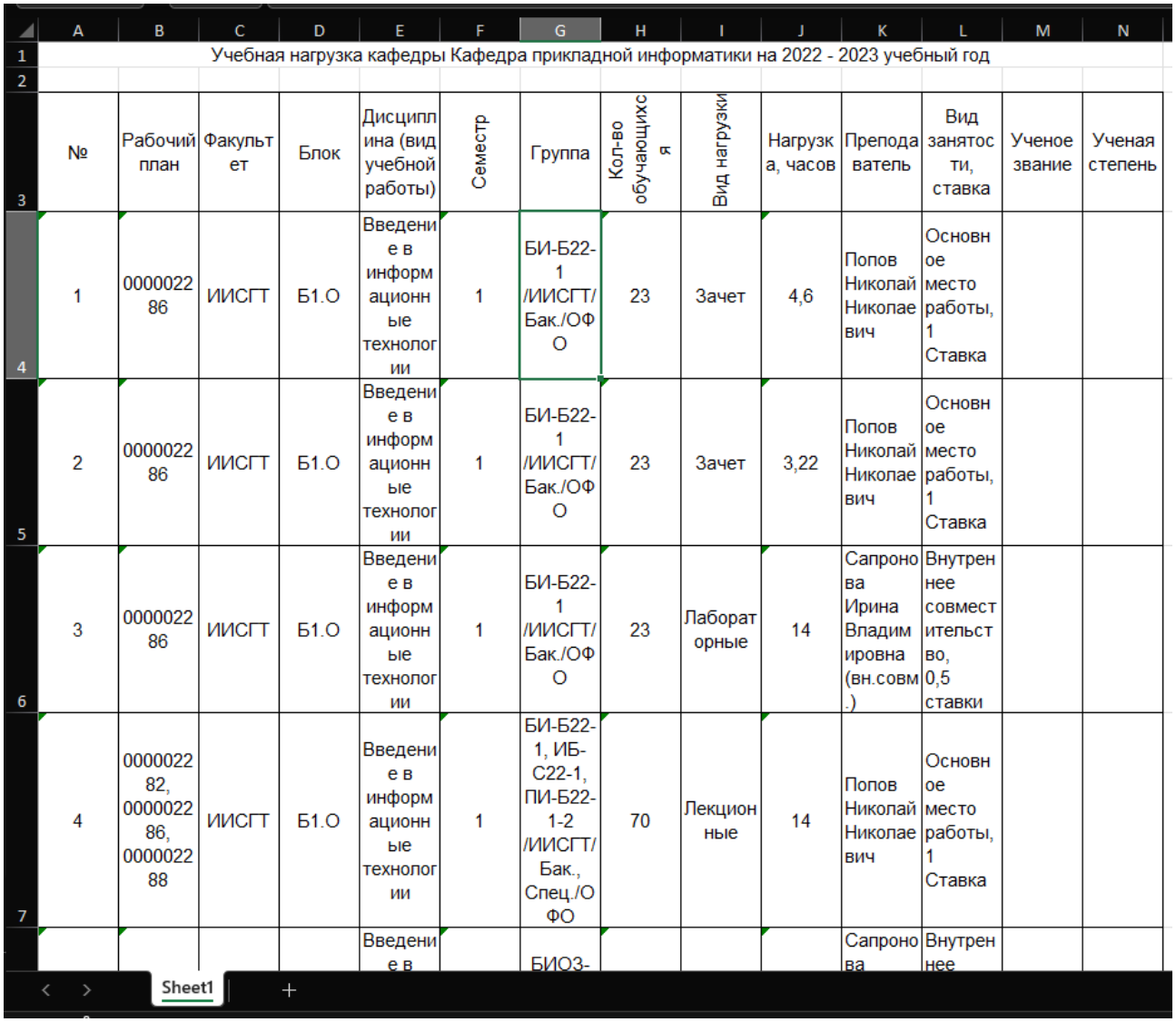
Титул | График | План | Компетенции(2) | Свод

Рисунок 25. Компетенции

Аналогично следует сохранить все значение из компетенции

### 4.6.2 Учебная нагрузка

Как и учебный план, учебная нагрузка – это xlsx-файл. Но имеет всего один лист. Из учебной нагрузки понадобится вся информация, поэтому таблицу придется извлечь целиком.



Учебная нагрузка кафедры Кафедра прикладной информатики на 2022 - 2023 учебный год

| № | Рабочий план | Факультет | Блок | Дисциплина (вид учебной работы) | Семестр | Группа | Кол-во обучающихся | Вид нагрузки | Нагрузка, часов | Преподаватель | Вид занятости, ставка | Ученое звание | Ученая степень |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 000002286 | ИИСГТ | Б1.О | Введение в информационные технологии | 1 | БИ-Б22-1/ИИСГТ/Бак./ОФО | 23 | Зачет | 4,6 | Попов Николай Николаевич | Основное место работы, 1 Ставка | | |
| 2 | 000002286 | ИИСГТ | Б1.О | Введение в информационные технологии | 1 | БИ-Б22-1/ИИСГТ/Бак./ОФО | 23 | Зачет | 3,22 | Попов Николай Николаевич | Основное место работы, 1 Ставка | | |
| 3 | 000002286 | ИИСГТ | Б1.О | Введение в информационные технологии | 1 | БИ-Б22-1/ИИСГТ/Бак./ОФО | 23 | Лабораторные | 14 | Сапронова Ирина Владимировна (вн.совм.) | Внутреннее совместительство, 0,5 ставки | | |
| 4 | 00000228 2, 000002286, 000002288 | ИИСГТ | Б1.О | Введение в информационные технологии | 1 | БИ-Б22-1, ИБ-С22-1, ПИ-Б22-1-2/ИИСГТ/Бак., Спец./ОФО | 70 | Лекционные | 14 | Попов Николай Николаевич | Основное место работы, 1 Ставка | | |
| | | | | Введение в | | БИОЗ- | | | | Сапронова | Внутреннее | | |

Sheet1

Рисунок 26. Учебная нагрузка

### 4.6.3 Общая характеристика образовательной программы

С ОХОП ситуация несколько усложняется, так как это docx-файл. Здесь не получится определить откуда читать информацию. Но, благодаря особенности хранения данных документа, можно извлечь все абзацы с индексацией по соответственным главам и параграфам.

СОДЕРЖАНИЕ



Рисунок 27. ОХОП оглавление

Так как документ большой, на рисунке представлено только оглавление

## 4.7 Создание шаблонов для заполнения документов

При помощи средств Python можно создавать документы с нуля, и заполнять их данными после создания. Но у этого подхода есть некоторые проблемы, а именно отсутствие настроек шрифта и положения текста, поэтому итоговый документа будет создаваться не с нуля, а из шаблона, где вместо специального маркера будет заполняться значения. В качестве маркера будет использоваться подобная запись #marker#, где маркер – имя маркера. Для работы модуля заполнения документа будет требоваться имя маркера и соответственный ему текст, для заполнения таблиц будет использоваться такой подход:

| | | |
|---|---|---|
| #col_1# | #col_2# | #col_3# |

Модуль должен будет получить маркеры каждой колонки и список сгруппированных по колонкам значений. Маркеры будут заменены, а таблица будет расширяться вниз по мере заполнения значениями.

Стоит отметить, что, если в исходном документе где-то будет использоваться # - все поломается. Чтобы избежать ошибок все # в документе должны быть продублированы (экранирование ##) тогда при проходе модуля заполнения все ## будут заменены на # и потерь данных или порчи документа не будет. В исходных документах в рамках данной работы # нигде не используется, поэтому проблем нет. Похожая система и для списков

- #val#

Для прототипа потребуется пока только 2 документа:

- Аннотация дисциплины
- Рабочий план дисциплины

### 4.7.1 Аннотация

Для аннотации была составлена следующая форма:

**Аннотация к рабочей программе дисциплины**
**#subject#**
Направление подготовки - **#direction#**
Направленность (профиль) - **#profile#**

Квалификация выпускника - **#qualification#**

**Цель дисциплины** - #subjectGoal#

**Задачи дисциплины**:
- #subjectObjectives#

**В результате освоения дисциплин студент должен**
**Знать**:
- #studentKnews#

**Уметь**:
- #studentAbles#

**Владеть**:
- #studentMusters#

**Содержание дисциплины:**
Тема 1. #themes#

Рисунок 28. Шаблон аннотации

### 4.7.2 Рабочий план дисциплины

Ниже представлен шаблон для РПД

**1. Цель и задачи освоения дисциплины**
**Цель** - #subjectGoals#

**Задачи**:
- #subjectObjections#

**2. Место дисциплины в структуре основной профессиональной образовательной программы**

Дисциплина изучается в 6 семестре и относится к дисциплинам формируемой участниками образовательных отношений.

Изучение дисциплины требует входных компетенций, знаний, умений и навыков, предусмотренных следующими курсами:
- Информатика и программирование
- Информационные системы и технологии

Результаты изучения данной дисциплины применимы в написании ВКР, а также при прохождении дисциплин Интерактивные информационные системы, Распределенные информационные системы.

**3. Перечень планируемых результатов обучения**

Процесс изучения дисциплины направлен на формирование компетенции ПК-3.

Таблица 1.

Профессиональные компетенции

| **Код и наименование профессиональной компетенции** | **Код и наименование индикатора достижения профессиональной компетенции** | **Результаты обучения** |
|---|---|---|
| #competition# | # achievements# | *#results#* |

**4. Структура и содержание дисциплины**

**4.1. Объем дисциплины**

Объем дисциплины составляет 3 зачетные единицы, 108 академических часа.

Таблица 2.

Объем дисциплины по видам учебных занятий в академических часах

| **Объём дисциплины** | **Очная форма обучения** |
|---|---|
| **Объем дисциплины** | **#allHours#** |
| **Контактная работа обучающихся с преподавателем (по видам аудиторных учебных занятий) – всего:** | |
| в том числе: | - |
| лекции | #lectureHours# |
| лабораторные занятия | **#labHours#** |
| **Самостоятельная работа (далее – СРС) – всего:** | **#iwHowrs#** |
| **Вид промежуточной аттестации** | #attestationType# |

**4.2. Структура дисциплины**

Таблица 3.

Структура дисциплины для очной формы обучения

| **№** | **Тема** | **С** | **Виды учебной** | **Формы** | **Формируемые** | **Индикаторы** |
|---|---|---|---|---|---|---|

Рисунок 29. Шаблон РПД страница 3

| | дисциплины | семестр | работы, в т.ч. самостоятельная работа студентов, час. | | | текущего контроля успеваемости | компетенции | достижения компетенций |
|---|---|---|---|---|---|---|---|---|
| | | | Лекции | Лабораторные работы | СРС | | | |
| #themeIndex# | #themes# | #semester# | #themeLectureHours# | #themelabHours#4 | #themeIWHours# | #controlMethods# | #competitions# | #achivments# |
| | **ИТОГО** | - | #lectureHours# | #labHours# | #iwHours# | - | - | - |

**4.3. Содержание разделов дисциплины**

**Тема 1. #themesWithContext#**

**4.4. Содержание лабораторных**

Таблица 4.

Содержание лабораторных занятий для очной формы обучения

| № темы дисциплины | Тематика лабораторных занятий | Всего часов |
|---|---|---|
| #subjectIndex# | #labThemeName# | #labThemeHours# |

**5. Перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине**

#eduResources#

**6. Оценочные средства для текущего контроля успеваемости и промежуточной аттестации по итогам освоения дисциплины**

Рисунок 30. Шаблон РПД страница 4

Учет успеваемости обучающегося по дисциплине осуществляется по 100-балльной шкале. Максимальное количество баллов по дисциплине за один семестр – 100:

- максимальное количество баллов за выполнение всех видов текущего контроля - #controlMaxMark#;
- максимальное количество баллов за посещение лекционных занятий – #lectureMaxMark#;
- максимальное количество баллов за прохождение промежуточной аттестации – #examMaxMark#.

**6.1. Текущий контроль**

Типовые задания, методика выполнения и критерии оценивания текущего контроля по разделам дисциплины представлены в Фонде оценочных средств по данной дисциплине.

**6.2. Промежуточная аттестация**

Форма промежуточной аттестации по дисциплине – **#endingType#.**

Форма проведения: устно по билетам

**Перечень вопросов для подготовки к зачету:**

ПК-3

1. #examQuestions#

Критерии оценивания:

**ПК-3**

**«Отлично»** - ставится студенту, ответ которого содержит:

- глубокое знание программного материала, а также основного содержания и новаций лекционного курса, по сравнению с учебной литературой;
- знание концептуально-понятийного аппарата всего курса;

а также свидетельствует о способности:

- самостоятельно критически оценивать основные положения курса;
- увязывать теорию с практикой.

Оценка «отлично» не ставится в случаях систематических пропусков студентом семинарских и лекционных занятий по неуважительным причинам, а также неправильных ответов на дополнительные вопросы преподавателя.

**«Хорошо»** - ставится студенту, ответ которого свидетельствует о полном знании материала по программе, а также содержит в целом правильное, но не всегда точное и аргументированное изложение материала.

Оценка «хорошо» не ставится в случаях пропусков студентом семинарских и лекционных занятий по неуважительным причинам.

**«Удовлетворительно»** - ставится студенту, ответ которого содержит:

- поверхностные знания важнейших разделов программы и содержания лекционного курса;
- затруднения с использованием научно-понятийного аппарата и терминологии курса;
- стремление логически четко построить ответ, а также свидетельствует о возможности последующего обучения.

**«Неудовлетворительно»** - ставится студенту, имеющему существенные пробелы в знании основного материала по программе, а также допустившему принципиальные ошибки при изложении материала.

**6.3. Балльно-рейтинговая система оценивания**

Таблица 5.

Рисунок 31. Шаблон РПД страница 5

Распределение баллов по видам учебной работы

| **Вид учебной работы, за которую ставятся баллы** | **Баллы** |
|---|---|
| Посещение лекционных занятий | 0-10 |
| Опрос | 0-30 |
| Сдача лабораторных работ | 0-30 |
| Промежуточная аттестация | 0-30 |
| **ИТОГО** | **0-100** |

Минимальное количество баллов для допуска до промежуточной аттестации составляет 40 баллов при условии выполнения всех видов текущего контроля.

Таблица 6.

Балльная шкала итоговой оценки на экзамене

| **Оценка** | **Баллы** |
|---|---|
| Отлично | 85-100 |
| Хорошо | 65-84 |
| Удовлетворительно | 40-64 |
| Неудовлетворительно | 0-39 |

**7. Методические указания для обучающихся по освоению дисциплины**

#eduRecomendations#

**8. Учебно-методическое и информационное обеспечение дисциплины обеспечение дисциплины**

**8.1. Перечень учебной литературы**

1. #eduExResources#

**8.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"**

1. #eduWebResources#

**8.3. Перечень программного обеспечения**

1. #eduSofts#

**8.4. Перечень информационных справочных систем**

1. #infos#

**8.5. Перечень профессиональных баз данных**

1. #specialDBs#

**9. Материально-техническое обеспечение дисциплины**

#otherResources#

**10. Особенности освоения дисциплины для инвалидов и лиц с ограниченными возможностями здоровья**

Обучение обучающихся с ограниченными возможностями здоровья при необходимости осуществляется на основе адаптированной рабочей программы с использованием специальных методов обучения и дидактических материалов, составленных с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся (обучающегося).

При определении формы проведения занятий с обучающимся-инвалидом

Рисунок 32. Шаблон РПД страница 6

## Заключение

Внедрение технологий автоматизации как правило оказывает положительное влияние на бизнес-процессы. Нельзя недооценивать важность оптимизации общей бизнес-системы. Автоматизация документов-сопроводительной документации — это шаг к созданию более эффективной образовательной системы.

В ходе выполнения работы, был создан прототип приложения автоматизации сопроводительной документации образовательной программы. Что является фактом достижения поставленной цели. Полученный прототип послужит материалом для создания более продвинутой, комплексной информационной системе формирования документов с еще большой степенью автоматизации.

В первой главе был произведен анализ предметной области и рассмотрены существующие решения, а также выявлены сильные и слабые стороны полученного прототипа. Согласно сделанным в этой главе выводам, разработанное приложение потенциально является конкурентноспособным и может в дальнейшем развиваться для коммерческих целей.

Во второй главе были построены: концептуальная, объектно-ориентированная и физическая модели информационной системы. В результате чего представился образ конечного прототипа достаточно полный для выбора оптимального пути реализации.

В третей главе были разработаны интерфейс и прототип конечной программы. Получен ценный опыт разработки с использованием полного стека технологий Flask.

Таким образом, все поставленные задачи выполнены, а цель достигнута.

## Список использованных источников


5. ГОСТ 34.602-2020 . МГС. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. – М.:Изд-во стандартов, 2021 – 30с.
6. 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.
7. ГОСТ 19.701-90 (ИСО 5807-85). ЕСПД. Схемы алгоритмов, программ, данных и систем. Условные обозначнеия и правила выполнения. – М.:Изд-во стандартов, 1991. – 26 с.
8. ГОСТ 7.1-2003. Библиографическая запись. Библиографическое описание. Общие требования и правила составления.
9. ГОСТ 7.32-2001 «Отчет о научно-исследовательской работе. Структура и правила оформления».
10. Роберт Мартин «Чистый код» [Текст]
11. Банда 4х «Паттерны ООП» [Текст]
12. Гультяев, А.К. Проектирование и дизайн пользовательского интерфейса[Текст] / А.К. Гультяев - СПб.: КОРОНАпринт, 2000. - 349 с.
13. Вендров, А.М. Современные методы и средства проектирования информационных систем [Текст] / А.М. Вендров - М.: Финансы и статистика, 2008. – 65 с.
14. Илюшечкин, В.М. Основы использования и проектирования баз данных[Текст] / В.М. Илюшечкин - М.: Издательство Юрайт 2010. -213с.
15. Карпова, Т.С. Базы данных: модели, разработка, реализация[Текст] / Т.С. Карпова – СПб.: Питер, 2002. – 304с.
16. ВикипедиЯ [Электронный ресурс]. – Режим доступа: https://www.wikipedia.org/

17. PVS-Studio [Электронный ресурс]. – Режим доступа: https://pvs-studio.ru
18. MDN [Электронный ресурс]. – Режим доступа: https://developer.mozilla.org/ru/
19. Юрайт [Электронный ресурс]. – Режим доступа: https://urait.ru/info/rpd-service
20. Amazon [Электронный ресурс]. – Режим доступа: https://aws.amazon.com
21. НОУ Интуит [Электронный ресурс]. – Режим доступа: https://intuit.ru
22. PythonRu [Электронный ресурс]. – Режим доступа: https://pythonru.com
23. РГГУ [Электронный ресурс]. – Режим доступа: https://www.rsuh.ru
24. Python Documentation [Электронный ресурс]. – Режим доступа: https://www.python.org
25. Flask Documentation [Электронный ресурс]. – Режим доступа: https://flask.palletsprojects.com/
26. SQLAlchemy Documentation [Электронный ресурс]. – Режим доступа: https://www.sqlalchemy.org/
27. Jinja2 Documentation [Электронный ресурс]. – Режим доступа: https://jinja.palletsprojects.com
28. Тусур [Электронный ресурс]. – Режим доступа: https://workprogram.tusur.ru
29. Форум для студентов и программистов [Электронный ресурс]. – Режим доступа: http://fkn.ktu10.com
30. КиберСофт [Электронный ресурс]. – Режим доступа: https://kiber-soft.net
31. ППУ [Электронный ресурс]. – Режим доступа: https://pstu.ru/
32. Tproger [Электронный ресурс]. – Режим доступа: https://tproger.ru
33. CyberForum [Электронный ресурс]. – Режим доступа: https://www.cyberforum.ru/
34. StudLance [Электронный ресурс]. – Режим доступа: https://studlance.ru
35. YouTube [Электронный ресурс]. – Режим доступа: https://www.youtube.com
36. SEWiki [Электронный ресурс]. – Режим доступа: http://sewiki.ru
37. HeadHunter [Электронный ресурс]. – Режим доступа: https://spb.hh.ru/?hhtmFrom=article
38. Visual paradigm online. Online Productivity suite [Электронный ресурс]. – Режим доступа: https://online.visual-paradigm.com

39. Хабр [Электронный ресурс]. – Режим доступа: https://habr.com/ru/flows/develop
40. Служба поддержки Microsoft [Электронный ресурс]. – Режим доступа: https://support.microsoft.com
41. GitHub Docs [Электронный ресурс]. – Режим доступа: https://docs.github.com
42. Business studio [Электронный ресурс]. – Режим доступа: https://www.businessstudio.ru
43. Enter Chain [Электронный ресурс]. – Режим доступа: https://www.enterchain.ru