

**«Санкт-Петербургский государственный электротехнический университет
«ЛЭТИ» им. В.И.Ульянова (Ленина)»
(СПбГЭТУ «ЛЭТИ»)**

Направление	09.03.04 — Программная инженерия
Профиль	Программная инженерия
Факультет	КТИ
Кафедра	МОЭВМ

К защите допустить

Зав. кафедрой

А.А. Лисс

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
БАКАЛАВРА**

**Тема: РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ WEB-ПРИЛОЖЕНИЯ
ГОСУДАРСТВЕННОЙ ИТОГОВОЙ АТТЕСТАЦИИ ВУЗА**

Студент		<hr/>	А.Е. Козиков
		<i>подпись</i>	
Руководитель	к.т.н., доцент	<hr/>	С.А. Беляев
		<i>подпись</i>	
Консультанты	к.т.н., доцент	<hr/>	М.М. Заславский
		<i>подпись</i>	
	к.э.н., доцент	<hr/>	Г.В. Голигузова
		<i>подпись</i>	
		<hr/>	

Санкт-Петербург

2024

ЗАДАНИЕ

НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

УТВЕРЖДАЮ

Зав. кафедрой МО ЭВМ

А.А. Лисс

« » 2024 г.

Студент Козиков А.Е.

Группа 0304

Тема работы: Разработка серверной части web-приложения государственной
итоговой аттестации вуза

Место выполнения ВКР: Санкт-Петербургский государственный электро-
технический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

Исходные данные (технические требования):

ОС Ubuntu 22.04, технологии программирования выбираются исполнителем

Содержание ВКР:

В работе рассматриваются следующие разделы: обзор и анализ существующих решений, архитектура и реализация приложения.

Перечень отчетных материалов: пояснительная записка, иллюстративный материал.

Дополнительные разделы: экономическое обоснование

Дата выдачи задания

Дата представления ВКР к защите

« » 20 г.

« 20 Г.

Студент

А.Е. КОЗИКОВ

Руководитель

С.А. Беляев

К.Т.Н., ДОЦЕНТ

КАЛЕНДАРНЫЙ ПЛАН ВЫПОЛНЕНИЯ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Утверждаю

Зав. кафедрой МО ЭВМ

А.А. Лисс

« 20 Г.

Студент Козиков А.Е.

Группа 0304

Тема работы: Разработка серверной части web-приложения государственной
итоговой аттестации вуза

№ п/п	Наименование работ	Срок вы- полнения
1	Обзор литературы по теме работы	01.02 — 10.02
2	Анализ существующих решений	10.02 — 15.02
3	Описание архитектуры	15.02 — 01.03
4	Реализация приложения	01.03 — 20.04
5	Тестирование приложения	20.04 — 01.05
5	Оформление пояснительной записки	01.05 — 05.05
6	Оформление иллюстративного материала	05.05 — 20.05

Студент

А.Е. КОЗИКОВ

Руководитель
к.т.н., доцент

С.А. Беляев

РЕФЕРАТ

Пояснительная записка 65 стр., 36 рис., 7 табл., 9 ист.

УНИВЕРСИТЕТ, ЦИФРОВИЗАЦИЯ, ДОКУМЕНТООБОРОТ, ГОСУДАРСТВЕННАЯ ИТОГОВАЯ АТТЕСТАЦИЯ,

Объектом исследования является автоматизация процесса проведения государственной итоговой аттестации в Санкт-Петербургском государственном электротехническом университете «ЛЭТИ» им. В.И. Ульянова (Ленина).

Предметом исследования является серверная часть web-приложения, совместимая с информационными системами вуза и предоставляющая возможность для автоматизации всех шагов процесса государственной итоговой аттестации в Санкт-Петербургском государственном электротехническом университете «ЛЭТИ» им. В.И. Ульянова (Ленина).

Цель работы: Разработать серверную часть web-приложения государственной итоговой аттестации вуза.

В работе описывается разработка серверной части программного комплекса для проведения государственной итоговой аттестации вуза. Проводится исследование существующих решений для поставленной цели, выделяются критерии для сравнения решений. Делается вывод об актуальности разработки приведённого решения. Описываются бизнес-процессы, архитектура серверной части, технологии, этапы разработки и тестирования, результаты апробации приложения в вузе.

ABSTRACT

Explanatory note: 65 pages, 36 figures, 7 tables, 9 references.

UNIVERSITY, DIGITALIZATION, DOCUMENT FLOW, STATE FINAL CERTIFICATION.

The object of the research is the automation of the process of conducting the state final certification at the Saint Petersburg State Electrotechnical University "LETI" named after V.I. Ulyanov (Lenin).

The subject of the research is the server-side of a web application, compatible with the university's information systems, providing automation for all steps of the state final certification process at the Saint Petersburg State Electrotechnical University "LETI" named after V.I. Ulyanov (Lenin).

The aim of the work is to develop the server-side of the web application for the university's state final certification.

The document describes the development of the server-side of the software complex for conducting the university's state final certification. It conducts research on existing solutions for the stated purpose, identifies criteria for comparing solutions, and concludes on the relevance of developing the proposed solution. It describes business processes, the architecture of the server-side, technologies, stages of development and testing, and the results of the application's testing at the university.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	11
1 АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ.....	12
1.1 Введение.....	12
1.2 Цель	12
1.3 Проблема.....	12
1.4 Задачи	12
1.5 Поиск и краткое описание существующих решений	13
1.6 Сравнение существующих решений	14
1.7 Рассмотрение аналогов.....	15
1.7.1 DIRECTUM и MS Office.....	15
1.7.2 «1С: Документооборот» и «1С: Университет»	16
1.7.3 A-Delo.....	17
1.7.4 Odoo	18
1.7.5 «Мой СФУ»	18
1.8 Вывод.....	19
2 АРХИТЕКТУРА.....	21
2.1 Введение.....	21
2.1 Архитектура серверной части.....	21
2.1.1 Уровень бизнес-логики.....	23
2.1.2 Уровень доступа к данным	24
2.1.3 Уровень хранения данных.....	25
2.2 Процессы системы	25
2.2.1 Подготовка заявки на список председателей на текущий год	26
2.2.1.1 Формирование перечня ГЭК.....	26
2.2.1.2 Заполнение персональных данных председателей.....	27
2.2.1.3 Формирование обоснования на председателей ГЭК.....	29
2.2.1.4 Приказ об утверждении председателей ГЭК	30
2.2.2 Подготовка на кафедрах и в деканатах.....	31
2.2.2.1 Формирование перечня доп. разделов ВКР	31

2.2.2.2 Утверждение тем, доп. разделов, руководителей и консультан- тов ВКР.....	32
2.2.2.3 Приказ о составе ГЭК.....	35
2.2.2.4 График защит ВКР	38
2.2.3 Проведение защит, документы секретаря ГЭК.....	40
2.2.3.1 Протокол ГЭК на студента	40
2.2.3.2 Сводная ведомость на заседание ГЭК	41
2.2.4 Отчет председателя.....	42
2.3 Архитектура базы данных.....	42
2.3.1 Диаграмма тем ВКР	42
2.3.2 Диаграмма ГЭК	45
2.3.3 Диаграмма персональных данных члена комиссии	46
2.3.4 Диаграмма экзаменов	48
2.4 Диаграммы классов бизнес процессов.....	49
2.4.1 Шаблона фабрики	50
2.4.2 Шаблон адаптера.....	50
2.4.3 Шаблон декоратор и синглтон.....	51
2.5 Выводы.....	51
3 РЕАЛИЗАЦИЯ	53
3.1 Введение.....	53
3.2 Реализация бизнес процессов	53
3.2.1 Служебные модули	53
3.2.1.1 Загрузка файлов.....	53
3.2.1.2 Справочники	53
3.2.1.3 Логирование	53
3.2.2 Системные модули.....	54
3.2.2.1 Перечень ГЭК.....	54
3.2.2.2 Темы ВКР.....	54
3.2.2.3 Интеграция с ИС	55
3.3 Тестирование	56

3.3.1 Модульное и интеграционное тестирование	56
3.3.2 Тестирование SQL инъекций	56
3.3.3 Нагрузочное и стресс тестирование	57
4 ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ	59
4.1 Введение.....	59
4.2 Расходы на оплату труда	59
4.3 Расходы на содержание и эксплуатацию оборудования.....	61
4.4 Амортизационные отчисления	62
4.5 Накладные расходы	62
4.6 Вывод.....	63
ЗАКЛЮЧЕНИЕ	64
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	65

ВВЕДЕНИЕ

В современном мире информационных технологий, ускоренное развитие цифровой инфраструктуры становится необходимостью во многих сферах деятельности. Особое внимание заслуживает образовательная сфера, где применение новых технологий позволяет значительно упростить и ускорить процесс управления образовательными и административными задачами. Одним из ключевых процессов в данной сфере является организация и проведение государственной итоговой аттестации (ГИА) в высших учебных заведениях.

Информационная система, представляющая собой web-приложение, предоставит инструменты для автоматизированно нормированного ведения и редактирования необходимых документов, просмотра актуальных данных и управления протекающими процессами в зависимости от уровня доступа. По сравнению с заполнением бумажных экземпляров, благодаря цифровизации процессов будет достигнуто упрощение и ускорение внутреннего документооборота, нормирование процессов внесения и редактирования данных, что уменьшит количество ошибок и ускорит исполнение работы. Web-приложение позволит независимо от лицензий проприетарного программного обеспечения, выполнять работу на мобильных и настольных устройствах. С помощью REST API будет достигнута интеграция с уже существующими информационными системами вуза.

Целью работы является разработка серверной части web-приложения государственной итоговой аттестации вуза.

Предлагаемое решение должно решать следующие **задачи**:

- Подготовка заявки на список председателей на год
- Подготовка на кафедрах и в деканатах
- Проведение защит, документы секретаря ГЭК
- Отчет председателя
- Документы на оплату

А также соответствовать следующим критериям:

- Интеграция с информационными системами ВУЗа.
- Доступность системы для клиентов внутри и за пределами ВУЗа через сеть интернет с мобильных и настольных устройств.
- Контроль доступа к данным с помощью выделения ролей в системе.
- Приложение должно стремиться минимизировать использование проприетарных и малознакомых для команды разработки технологий. Это влияет на возможность поддержания и развития приложения.

Объектом исследования является автоматизация процесса проведения государственной итоговой аттестации в Санкт-Петербургском государственном электротехническом университете «ЛЭТИ» им. В.И. Ульянова (Ленина).

Предметом исследования является серверная часть web-приложения, совместимая с информационными системами вуза и предоставляющая возможность для автоматизации всех шагов процесса государственной итоговой аттестации в Санкт-Петербургском государственном электротехническом университете «ЛЭТИ» им. В.И. Ульянова (Ленина).

В данной работе будут исследованы основные аспекты и проблемы, связанные с созданием серверной части такого приложения, включая: рассмотрение существующих аналогов (глава 1), проектирование архитектуры (глава 2), описание реализации и тестирование (глава 3), экономическое обоснование решения (глава 4).

1 АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ

1.1 Введение

В главе 1 описана цель, проблема и постановка задач, рассмотрены существующие решения с их анализом и сравнительной таблицей. На основании чего делается вывод о необходимости разработки нового решения для поставленных задач.

1.2 Проблема

Информационная система (ИС) «Государственная итоговая аттестация» направлена на цифровизацию процессов проведения государственной итоговой аттестации. Информационная система, представляющая собой web-приложение, предоставит инструменты для:

- Автоматизированно нормированного ведения и редактирования необходимых документов.
- Просмотра актуальных данных и управления протекающими процессами в зависимости от уровня доступа.

По сравнению с заполнением бумажных экземпляров, благодаря цифровизации процессов будет достигнуто

- Упрощение и ускорение внутреннего документооборота.
- Нормирование процессов внесения и редактирования данных, что уменьшит количество ошибок и ускорит исполнение работы.

1.3 Поиск и краткое описание существующих решений

В качестве аналогов рассматривалось программное обеспечение (ПО), обеспечивающее полную или частичную цифровизацию процесса проведения итоговой аттестации или документооборота вуза. Поиск осуществлялся в агрегаторах научных статей Google Scholar и eLibrary по запросам: «Автоматизация итоговой государственной аттестации», «Автоматизация документооборота вуза».

Краткий обзор найденных аналогов:

- DIRECTUM и MS Office — использование системы электронного документооборота DIRECTUM и офисного пакета приложений Microsoft Office.
- «1С: Документооборот» и «1С: Университет» — интеграция двух продуктов компании «1С» для документооборота и редактирования готовых шаблонов документов.
- A-Delo — проприетарная информационная система, разработанная для автоматизации процессов и управления документооборотом в учебных заведениях.
- Odoo — интегрированная система управления организациями, которая распространяется по лицензии LGPL.
- «Мой СФУ» — корпоративный социальный сетевой сервис Сибирского федерального университета, который представляет инструменты для ведения и контроля данных сотрудников и студентов вуза.

1.4 Сравнение существующих решений

Для сравнения аналогов был выбран ряд критериев, которыми можно охарактеризовать каждую технологию.

1. Нормирование документации. Критерий оценивает возможность реализации нормирования и шаблонизации документации: частичная (\pm), когда шаблон необходимо редактировать в отдельных редакторах, или полная (+) шаблонизация, с экспортом уже готового документа.
2. Интеграция по REST API с ИС. Критерий оценивает возможности интеграции с существующими ИС вуза: полная (+) интеграция включает двусторонний обмен данными между ИС, частичная (\pm) позволяет только считывать данные по API.
3. Модель распространения технологий. Критерий оценивает доступность технологии с рассмотрения доступности технологии: пропри-

старная технология с платным доступом (\pm), закрытая технология (-)
или технология с открытым исходным кодом (+).

Результаты сравнения, представленные в таблице 1, дадут общую оценку технологию и понимание актуальности предлагаемого решения для решения поставленной задачи.

Таблица 1 — Сравнение аналогов

Аналог	Нормирование документации	Интеграция по REST API с ИС	Модель распространения технологий
DIRECTUM и MS Office	±	±	±
«1С»	±	±	±
A-Delo	-	±	±
Odoo	-	+	+
«Мой СФУ»	+	+	-

1.5 Рассмотрение аналогов

1.5.1 DIRECTUM и MS Office

Статья «Автоматизация процессов управления в учебно-методической деятельности структурного подразделения университета» [1] рассматривает возможности автоматизации процесса управления учебной деятельностью в структурных подразделениях университета.

Организация документооборота в исследуемом контексте предусматривает использование системы электронного документооборота DIRECTUM и офисного пакета приложений Microsoft Office.

DIRECTUM — проприетарное решение для:

- Автоматизация документооборота
- Хранения, безопасность и контроля доступа
- Управления задачами и процессами

DIRECTUM предлагает локальное и облачное хранение данных, в зависимости от этого меняется модель распространения ПО.

Microsoft Office — это пакет офисных приложений, разработанный и распространяемый корпорацией Microsoft по подписочному плану.

Из пакета приложений для создания и редактирования рабочих документов предлагается использовать Microsoft Word и Microsoft Excel.

Данные приложения:

- Обеспечивают нестрогую шаблонизацию документов (невозможно разграничить права на заполнение и редактирование шаблона)
- Нет автоматической связи с базами данных. Такая возможность снизила бы количество опечаток, за счёт автоматической сверки данных
- Имеют веб, настольные и мобильные версии. Изменения документа можно сохранять на облачный сервер Microsoft

Таким образом, предлагается интеграция программного обеспечения, автоматизирующий процессы работы с документацией и её распространения.

С одной стороны, использование популярных проприетарных решений выгодно за счёт стандартизации инструментов: новых сотрудников не нужно будет учить работе приложением. Однако опыт 2022 года говорит о ненадёжности использования проприетарного ПО и необходимости рассматривать выбор в пользу аналогов с открытым исходным кодом [2].

1.5.2 «1С: Документооборот» и «1С: Университет»

Статья «Применение технологий 1С для развития компетенций цифровой экономики» [3] описывает интеграцию систем «1С: Университет» и «1С: Документооборот» для автоматизации бизнес-процессов в университетах. Оба приложения являются проприетарным продуктом компании «1С», распространяемым по лицензионным копиям.

«1С: Университет» отвечает за работу с документами и имеет готовые шаблонные решения для типовых задач вуза, в том числе для проведения государственной итоговой аттестации. Несмотря на это, зачастую для создания шаблонов документов необходимо адаптировать имеющиеся решения под специфические условия вуза [4]. Для этого необходимо использовать в связке с «1С: Университет» поддерживаемые текстовые редакторы (MS Word, LibreOffice Writer и другие).

«1С: Документооборот» предназначен для автоматизации процессов электронного документооборота с поддержкой контроля доступа. Имеет как локальную, так и облачную версию хранения данных.

За счёт готовой интеграции приложений «1С», не нужно тратить дополнительные усилия для настройки совместной работы приложений. И в то же время, оба приложения самостоятельны и могут быть заменены в будущем. Но такая интеграция подойдёт только как настольное приложение для использования только сотрудниками вуза: для доступа остальным пользователям нужна будет либо лицензия, либо отдельное приложение.

1.5.3 A-Delo

Статья «Подходы к автоматизации документооборота в вузе» [5] описывает разработку и внедрение информационной системы A-Delo.

A-Delo — это проприетарная информационная система, разработанная для автоматизации процессов и управления документооборотом в учебных заведениях. Она предназначена для упрощения и оптимизации работы с документами, включая их регистрацию, обработку, хранение и обмен.

Однако система не подразумевает создание, редактирование или шаблонизацию документов. Таким образом эта система покрывает задачи, связанные с документооборотом. Но вопрос выбора инструментов для работы с самими документами предлагается решать самим пользователям.

Авторы статьи для интеграции с ИС A-Delo реализуют интеграцию с пакетами MS Office и OpenOffice. Для хранения данных используется MS SQL Server.

Несмотря на то, что A-Delo имеет веб версию, представленная в статье реализация требует установки и подключения системы к компьютеру пользователя.

1.5.4 Odoo

В статье «Автоматизация информационных процессов проведения государственной итоговой аттестации выпускников вуза» говорится об использовании программы Odoo [6] для автоматизации бизнес-процессов.

Odoo — это интегрированная система управления организациями, которая распространяется по лицензии LGPL, то есть свободно, с предоставлением исходного кода. Такой подход позволяет пользователям разрабатывать собственные модули для приложения и настраивать интеграцию с другими системами. Приложение Odoo предоставляет широкий функционал для хранения и работы с данными: систему управления контентом (CMS) и взаимоотношениями с пользователями (CRM), в которую в том числе входит редактор сайтов. Таким образом у разработчиков появляется возможность построить клиент-серверное приложение с разграниченными правами доступа пользователей.

Но данное решение потребует доработки и интеграции с текстовыми редакторами, так как инструменты Odoo позволяют лишь управлять, но не редактировать документы.

1.5.5 «Мой СФУ»

В статье «Автоматизация итоговой государственной аттестации выпускников» [7] рассматривается построение архитектуры веб приложения, предназначенного для автоматизации бизнес-процессов.

«Мой СФУ» — корпоративный социальный сетевой сервис Сибирского федерального университета, который представляет инструменты для ведения и контроля данных сотрудников и студентов вуза. Сервис состоит из автоматизированных информационных систем (АИС), реализованных как веб приложения.

АИС «Диплом-мастер» предоставляет функционал для хранения данных о выпускниках, руководителях и темах ВКР. В системе реализована ролевая система, с разграничением доступа и редактирования данных.

Веб приложение для решения подобной задачи позволяет:

- Создать шаблоны, заполняемые на основе введенных данных. С возможностью последующего экспорта в подходящий формат текстового редактора (в рассматриваемой статье был выбран пакет MS Office).
- Синхронизировать данные из баз данных других АИС вуза.
- Получать доступ к приложению для компьютеров и мобильных устройств.
- Избежать необходимости подключения к внутренним сетям вуза и установки специализированных программ для документооборота (как, например, рассмотренные выше: Odoo, A-Delo, «1С: Документооборот»).

Такая АИС является собственностью вуза и может иметь свой патент. Это означает независимость от решений третьих лиц, владеющих правами на ПО, по вопросу его распространения. Но поддержка и обеспечение подобной системы является задачей учебного заведения.

1.6 Вывод

Разбор существующих решений говорит об актуальности проблемы автоматизации процессов государственной итоговой аттестации. Однако представленные аналоги не решают всех поставленных задач: предлагают слабое взаимодействие данных системы с внешними ИС, нестрогую шаблонизацию, являются труднодоступными или дорогостоящими.

2 АРХИТЕКТУРА

2.1 Введение

Во второй главе описывается архитектура представленного решения: серверная часть, взаимодействующая с клиентской частью и базой данных.

2.1 Архитектура серверной части

Построение архитектуры серверного приложения и выбор технологий основывался на:

- Возможности выполнить поставленные задачи.
- Покрыть все недостатки представленных аналогов.
- Использовать поддерживаемые технологии, знакомые внутри команды разработчиков вуза. Для длительной поддержки разработанного приложения.

На рис. 1 представлена архитектура серверной части веб приложения.

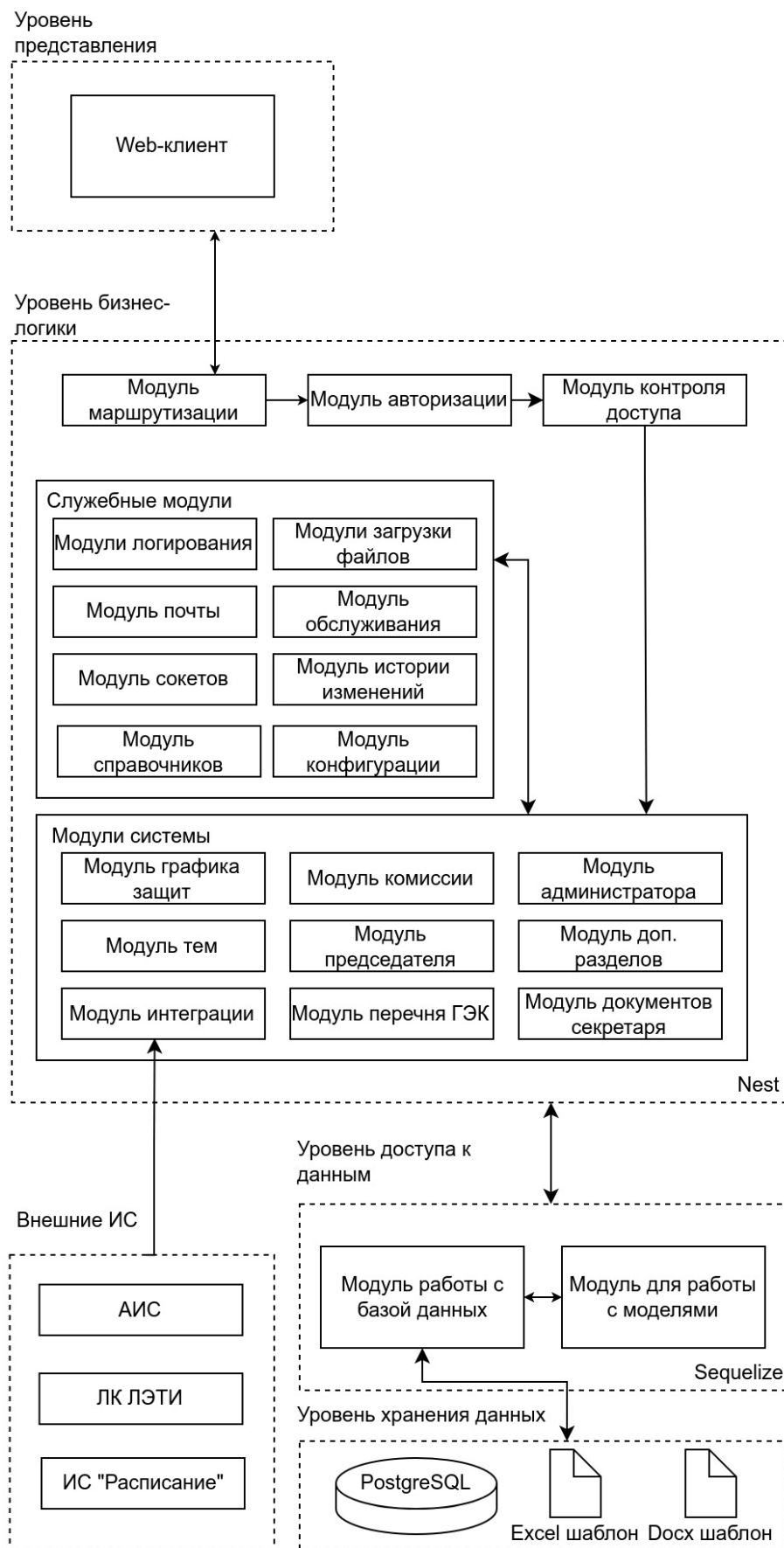


Рисунок 1 — Архитектура серверной части

2.1.1 Уровень бизнес-логики

Уровень бизнес логики реализован с помощью технологии NestJS. Выбор NestJS обоснован его модульной структурой, основанной на принципах SOLID. Он предоставляет готовое архитектурное решение для разработки масштабируемых веб-приложений, благодаря использованию TypeScript и системы модулей, что существенно ускоряет разработку и улучшает поддержку приложения. NestJS является бесплатным и распространяется по лицензии MIT.

В качестве языка программирования выбран TypeScript, так как строгая типизация позволяет избежать большинства ошибок на стадии написания кода, а также улучшает читаемость кода, помогая быстрее новым разработчикам разобраться в написанной программе.

Серверная часть состоит из модулей — самостоятельных единиц, отвечающих за весь жизненный цикл запроса и создание ответа. К модулям подключаются: пред- и пост- обработчики данных, контроллеры и сервисы.

Контроллеры, подключенные к модулям, содержат информацию о точке запроса, необходимые обработчики, вызывают метод для реализации запроса и возвращают ответ.

Промежуточные обработчики отвечают за: преобразование (парсинг, декодирование) данных, контроль доступа.

Благодаря шаблону внедрения зависимостей (Dependency Injection), связанные с контроллерами сервисы реализуют основную бизнес-логику процесса: взаимодействие с базой данных, обработку полученных данных.

На уровне бизнес-логики находится корневой модуль, который NestJS использует для построения графа приложения, маршрутизации, подключающий все основные модули и обработчики:

- ConfigModule — модуль конфигурации для получения данных из файла окружения

- **LoggerModule** — модуль логирования, для сохранения логов работы серверной части в файловое хранилище
- **SequelizeModule** — модуль для использования объектно-реляционного отображения (ORM) базы данных в приложении
- **SentryModule** — модуль для использования sentry.io для ведения журнала ошибок
- **GeneralModule** — служебный модуль, объединяющий в себе модули, используемые в системных модулях: модули истории, логирования, сокетов
- **AuthModule** — модуль авторизации пользователей: через ЛК, через логин и пароль, через Digital Authorizer. Выполняет роль в проверке прав в RightsGuard
- **DictsModule** — модуль справочников
- **SystemModule** — системный модуль, содержащий модули:
 - **AdminModule** — модуль администратора
 - **TopicsModule** — модуль тем ВКР
 - **IntegrationModule** — модуль интеграции с другими ИС
 - **ExamScheduleModule** — модуль расписания защит
 - **AdditionalSectionsModule** — модуль доп. разделов ВКР
 - **CommissionModule** — модуль комиссии
 - **ExamCommissionModule** — модуль перечня ГЭК
 - **MainPageModule** — модуль главной страницы.
- **FileLoaderModule** — модуль загрузки файлов на основе промежуточного обработчика Multer

2.1.2 Уровень доступа к данным

Sequelize — это ORM, которая предоставляет интерфейс для взаимодействия с реляционными базами данных, в том числе PostgreSQL. Sequelize позволяет использовать объектно-ориентированный подход в работе с данными, вместо написания SQL запросов, что делает разработку быстрее и безопаснее. Поддержка миграций позволяет непре-

рывно вносить изменения в базу данных, а асинхронные транзакции обеспечивают атомарность запросов.

Уровень доступа к данным представлен модулем `SequelizeModule`, который реализован в `NestJS`. В качестве сервиса используется `SequelizeConfigService`, наследующий класс `SequelizeOptionsFactory`. Сервис конфигурирует подключение к базе данных и добавление моделей таблиц.

2.1.3 Уровень хранения данных

На уровне хранения данных используется БД `PostgreSQL` и шаблоны `docx` и `excel` для документов.

`PostgreSQL` — это система управления реляционными базами данных (СУБД), распространяющаяся на лицензии открытого исходного кода (Open Source license). Данная база данных выделяется среди прочих поддержкой большого числа типов данных (например позволяет хранить массивы) и объектно-реляционной моделью, которая хорошо сочетается с `Sequelize`.

Для экспорта файлов со стороны сервера используются шаблоны `docx` и `excel`, хранящиеся локально на сервере. Работа с форматом `excel` реализована благодаря пакету `exceljs`, а создание документов `docx` происходит с помощью `docx-templates`. Такая реализация подготовки документов гарантирует строгую шаблонизацию, при которой данные берутся из базы данных, что убирает вероятность опечаток в процессе заполнения.

2.2 Процессы системы

В процессе проведения ГИА можно выделить следующие ключевые этапы:

1. Подготовка заявки на список председателей на текущий год
2. Проведение подготовки на кафедрах и в деканатах
3. Проведение защит, оформление документов секретаря ГЭК

4. Оформление отчета председателя

5. Оформление документов на оплату

Рассмотрим подробно каждый из бизнес процессов, используя BPMN-диаграммы.

2.2.1 Подготовка заявки на список председателей на текущий год

2.2.1.1 Формирование перечня ГЭК

Данный этап представлен на рис. 2, начинается с формирования кафедр перечня ГЭК по своим образовательным программам. Формируются проекты ГЭК, заполняется контингент защищающихся групп.

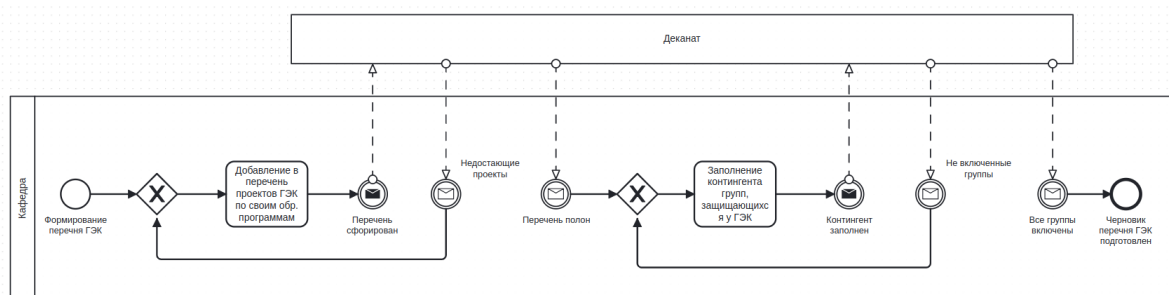


Рисунок 2 — Формирование перечня ГЭК: кафедра

На рис. 3 деканат контролирует процесс заполнения проектов ГЭК для своих кафедр. После окончательной корректировки составленного перечня, деканат обязан уведомить УОО о готовности проектов ГЭК для утверждения.

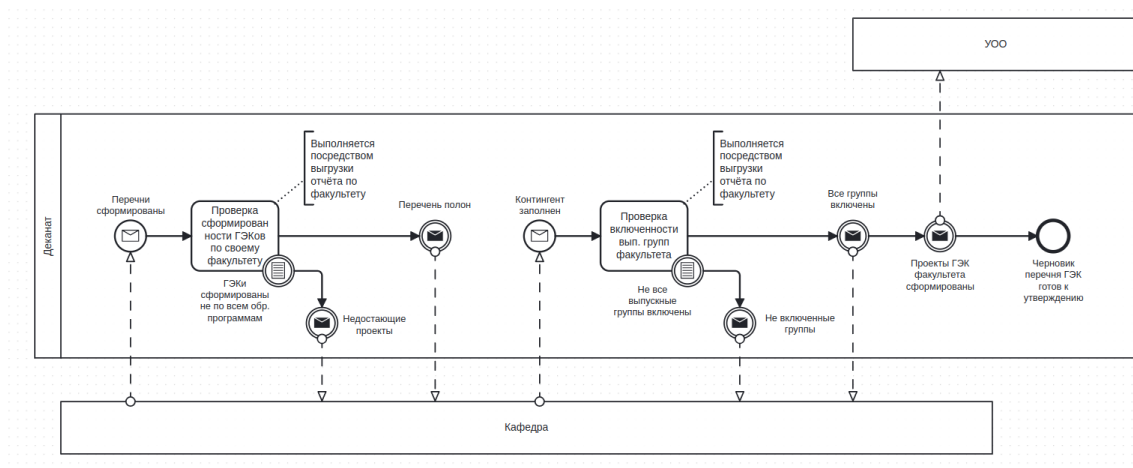


Рисунок 3 — Формирование перечня ГЭК: деканат

Финальное формирование перечня представлено на рис. 4. Финальные корректировки осуществляются УОО, который также назначает номера ГЭК. После присвоения номера проект ГЭК считается утвержденным, что позволяет его использовать в последующих этапах системы.

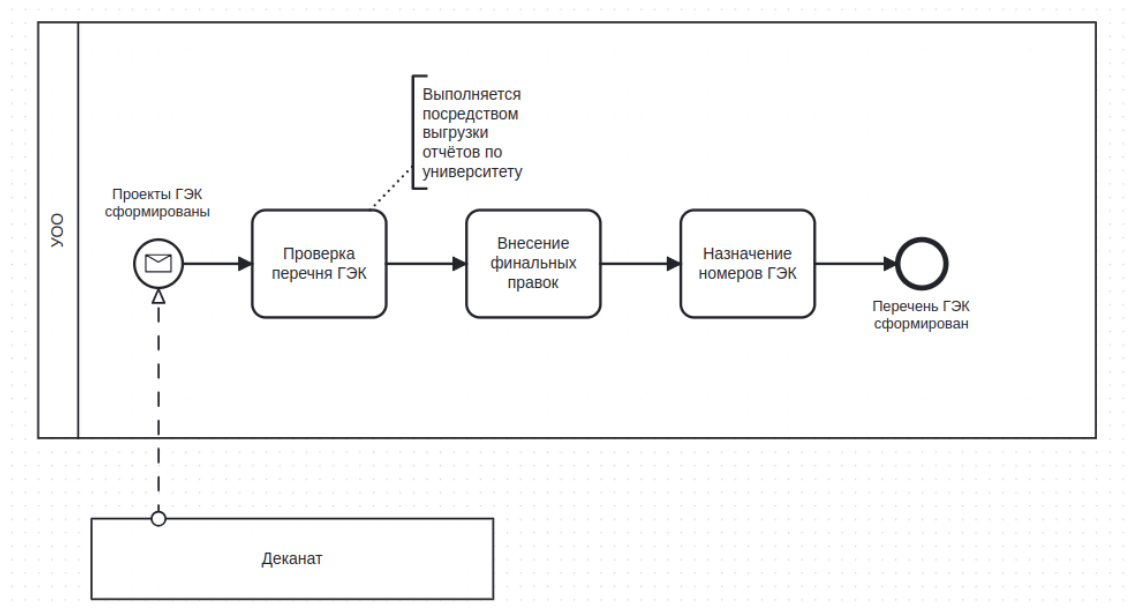
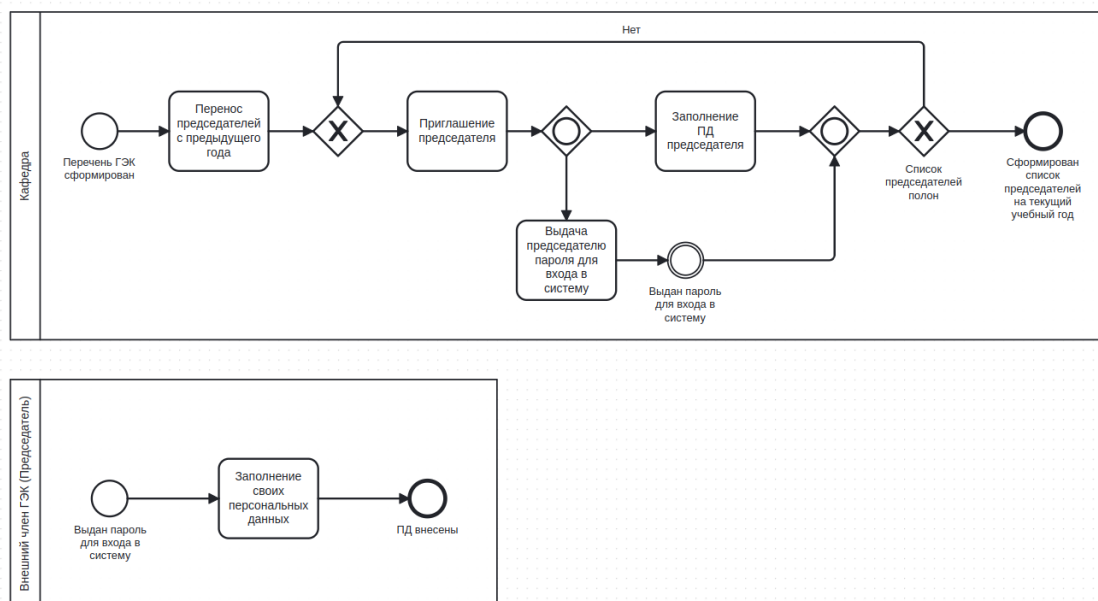


Рисунок 4 — Формирование перечня ГЭК: УОО

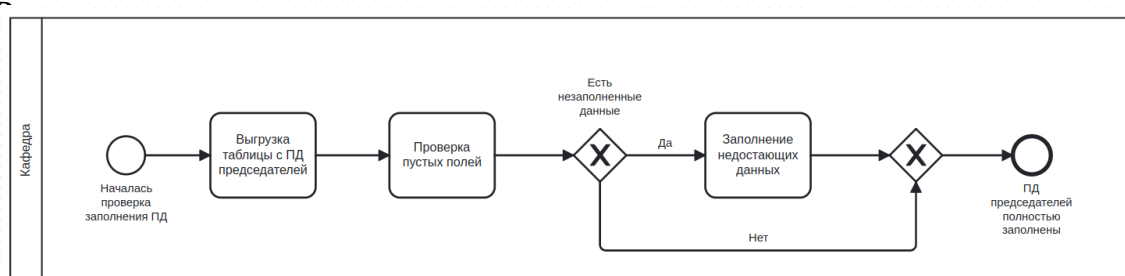
2.2.1.2 Заполнение персональных данных председателей

Затем составляется список председателей на текущий учебный год и вводятся их персональные данные (ПД). Процесс показан на рис. 5. Полномочия на выполнение указанных бизнес-процессов принадлежат не только кафедре, но и деканату, а также УОО.



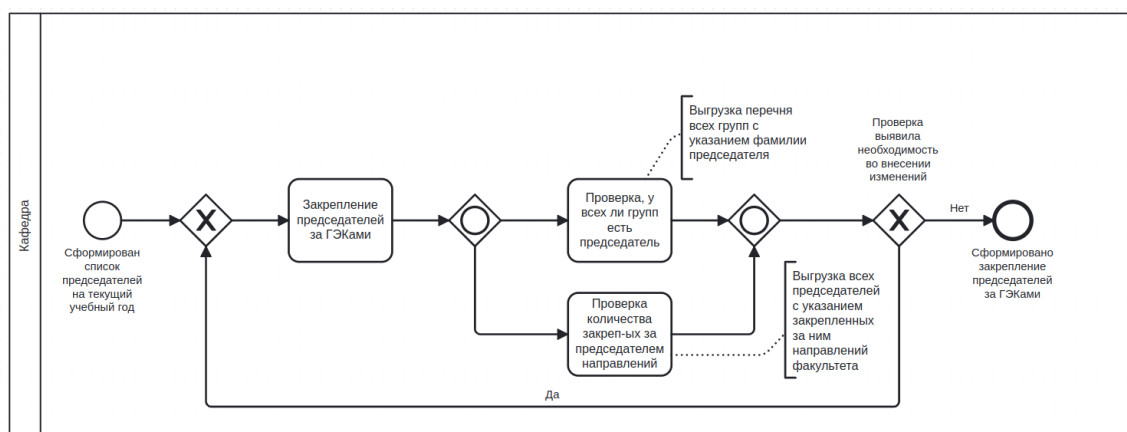
— Формирование списка председателей

Новому председателю может быть предоставлен пароль для доступа к системе, что позволит ему самостоятельно заполнить свой профиль. При первом переносе председателя на следующий учебный год будет создана новая запись, независимая от предыдущего года, с копированием текущей информации о ПД, но без автоматической актуализации при последующих изменениях. При повторном переносе председателя на следующий год его ПД будут автоматически актуализированы. После составления списка председателей производится проверка заполнения их персональных данных, описанная на рис. 6.



ок 6 — Проверка ПД

В то же время, помимо проверки заполнения персональных данных председателей, может проводиться работа по назначению председателей для конкретных комиссий. Закрепление председателей показано на рис. 7.

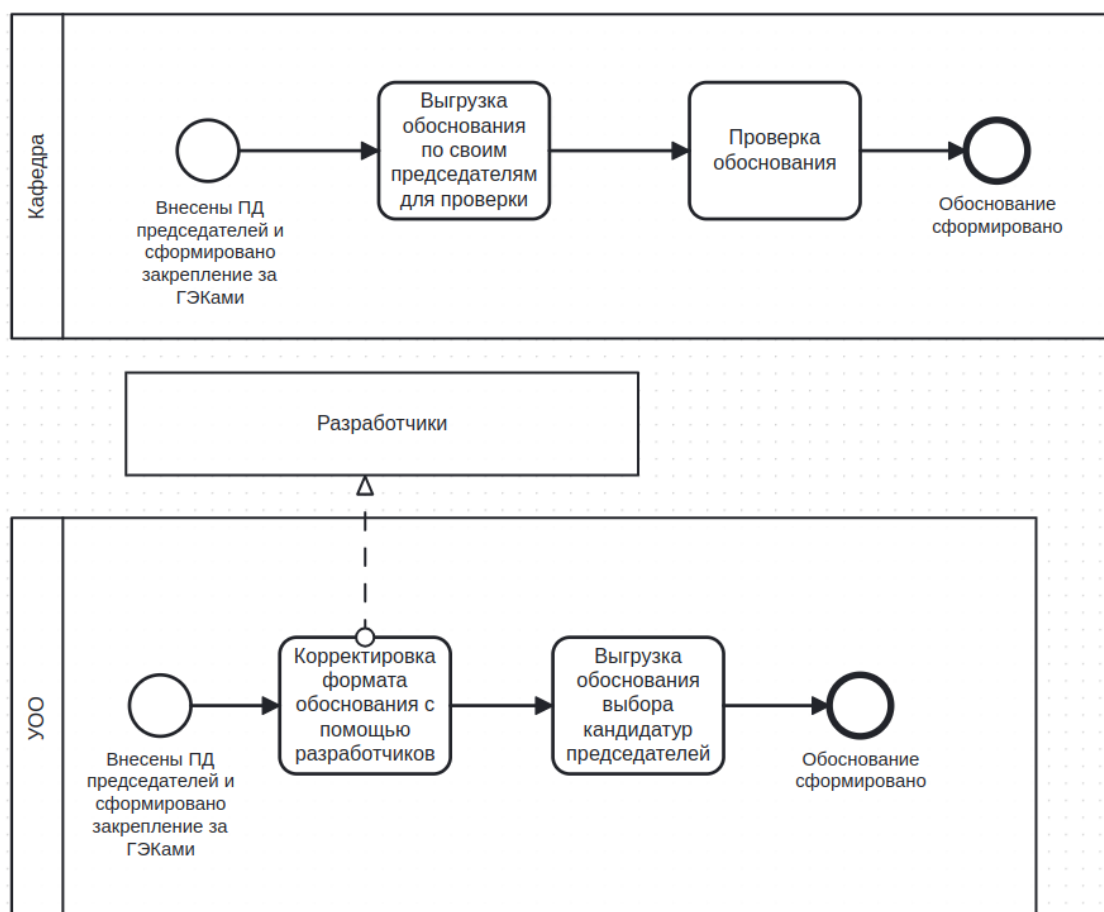


7 — Закрепление председателей

По завершении этого этапа должно быть завершено закрепление председателей за всеми проектами ГЭК, а также заполнены все персональные данные председателей.

2.2.1.3 Формирование обоснования на председателей ГЭК

Формирование обоснования показано на рис. 8. Подтверждение выбора кандидатур на должность председателей формируется путем выгрузки данных. Информация, включенная в выгрузку, является наиболее актуальной на момент ее составления. Формат обоснования может быть изменен разработчиками по запросу УОО.



боснование на председателя ГЭК

2.2.1.4 Приказ об утверждении председателей ГЭК

Завершающим этапом процесса является составление приказа б утверждении председателей ГЭК, описанное на рис. 9.

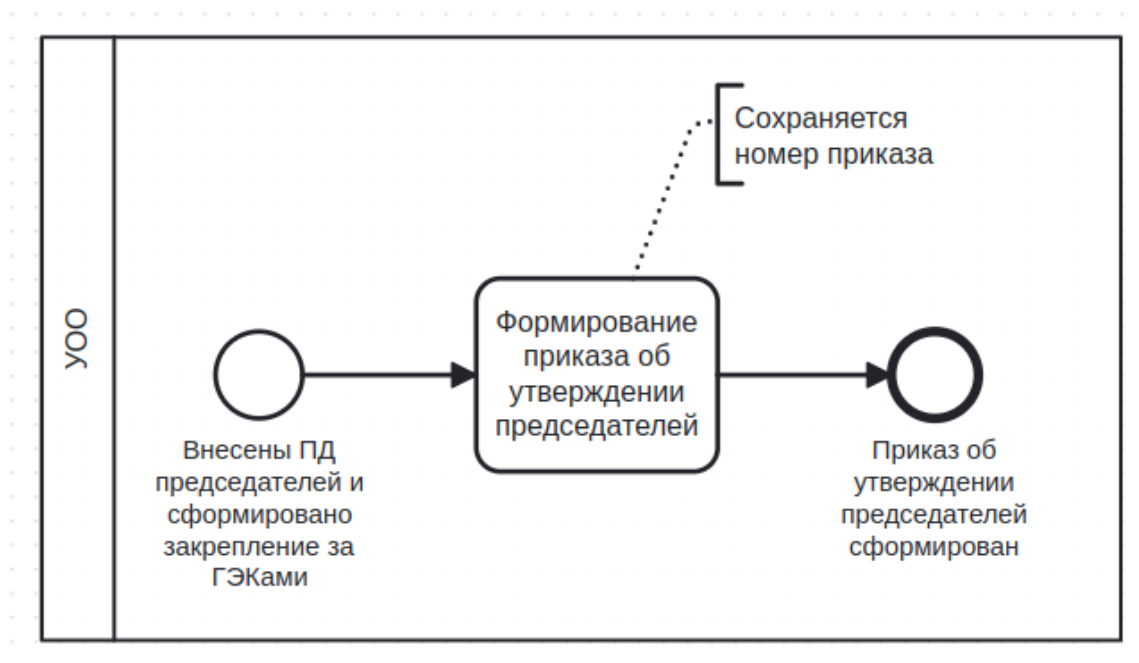


Рисунок 9 — Приказ об утверждении председателей

2.2.2 Подготовка на кафедрах и в деканатах

2.2.2.1 Формирование перечня доп. разделов ВКР

Формирования перечня доп. разделов, показанное на рис. 10, начинается с того, что УОО формирует основной перечень разделов, который может быть дополнен или отредактирован кафедрами, ответственными за реализацию дополнительных разделов. Процесс завершается выгрузкой распоряжения о текущем перечне дополнительных разделов на учебный год.

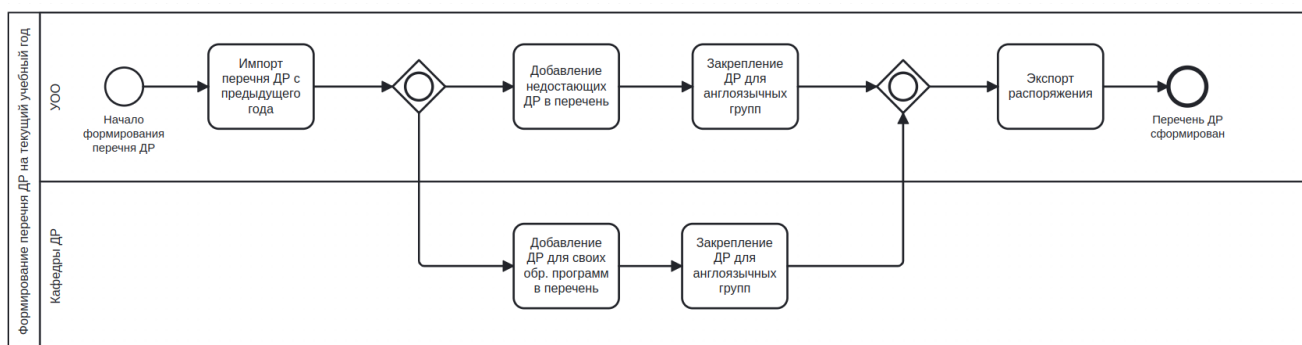
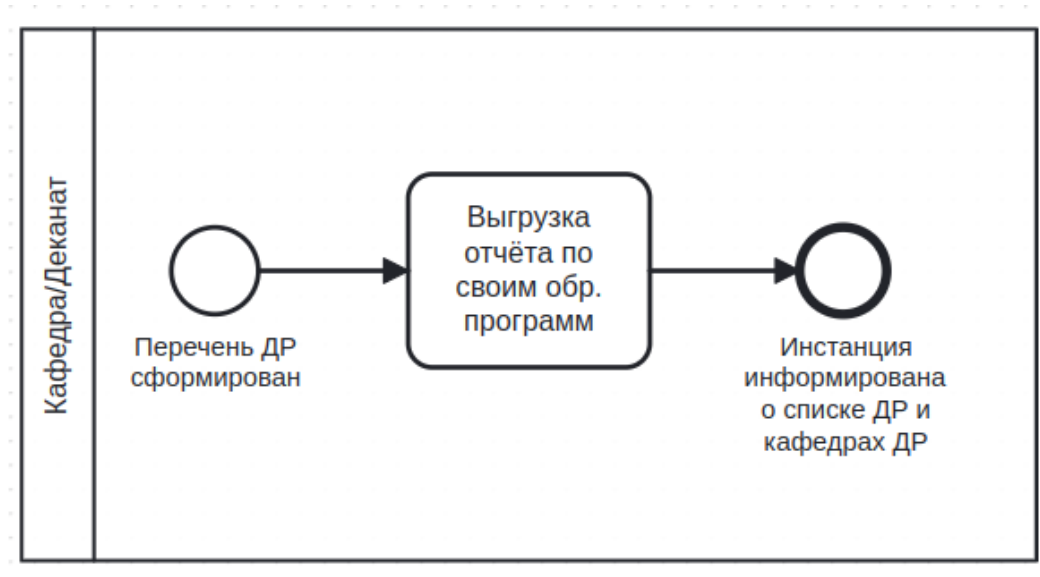


Рисунок 10 — Формирования перечня доп. разделов

Кафедры и деканаты имеют доступ ко всему списку дополнительных разделов, а также могут получать отчеты о списках разделов и кафедр по своим образовательным программам. Выгрузка информации описана на рис. 11.



1

— Информирование о доп. разделах

2.2.2.2 Утверждение тем, доп. разделов, руководителей и консультантов ВКР

Процесс, показанный на рис. 12, начинается с создания списка предлагаемых тем для выпускных квалификационных работ на соответствующих кафедрах.

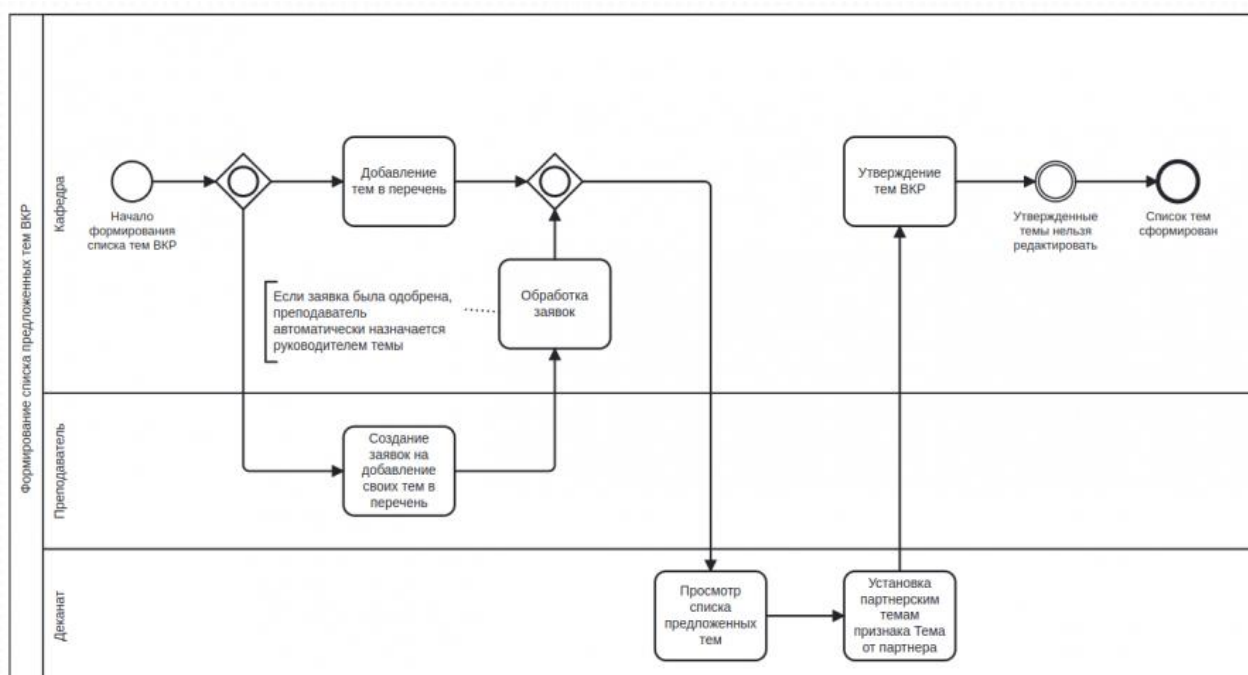


Рисунок 12 — Формирование списка тем

Далее, как демонстрирует рис. 13, студенты бронируют темы ВКР.

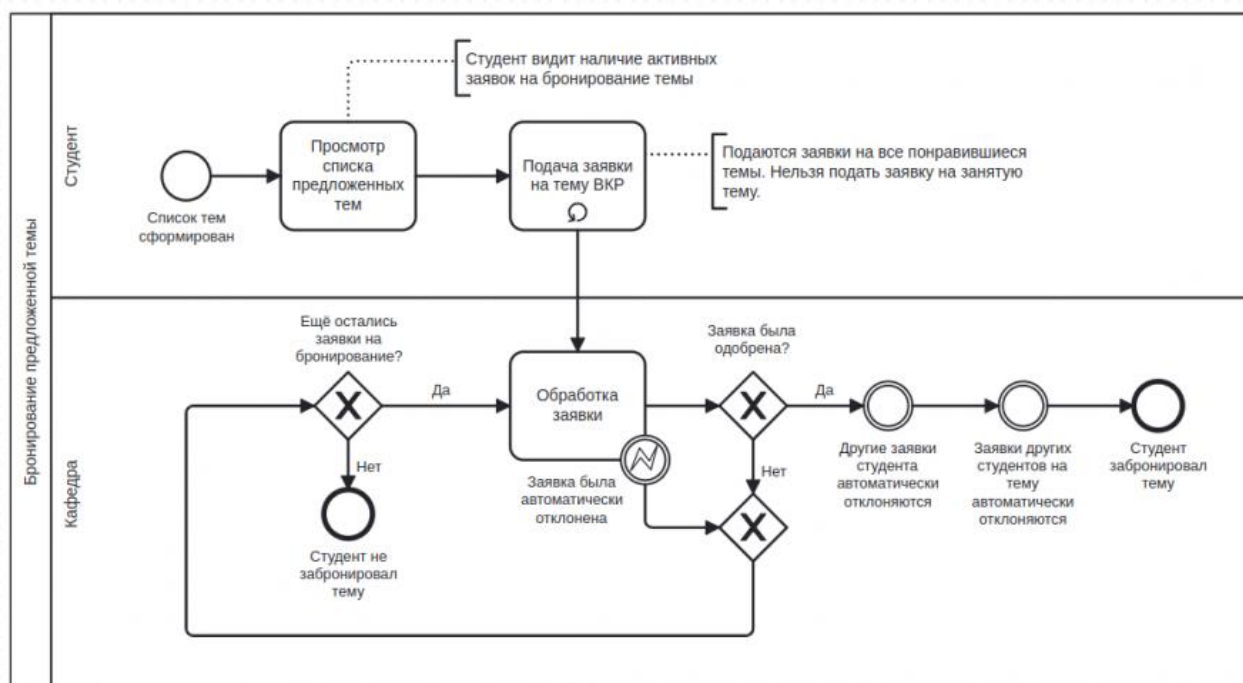


Рисунок 13 — Бронирование тем

Студентам предоставляется возможность подать заявку на свою тему для выпускной квалификационной работы (рис. 14).

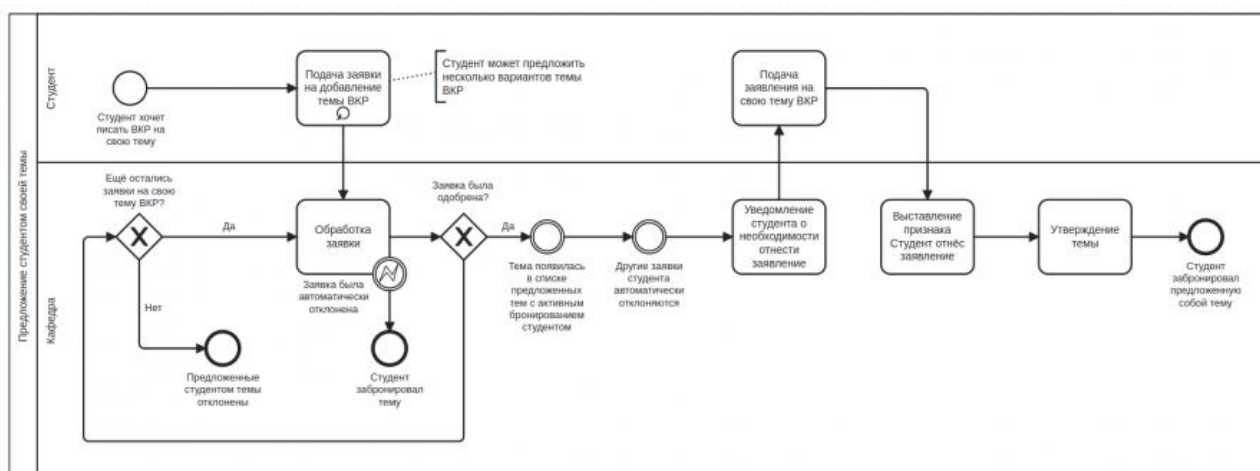


Рисунок 14 — Подача темы студентом

Когда бронирование темы подтверждается, другие заявки студента автоматически отклоняются, а также отклоняются заявки других студентов на забронированную тему. В случае, если студент остается без темы, ему необходимо повторно подать заявку на свободную тему, или кафедра может самостоятельно назначить тему при закреплении.

Затем происходит закрепление и уточнение тем ВКР. Процесс начинается на кафедрах, где заполняется доступная информация, добавляются записи о внешних консультантах, либо их персональные данные заполняются самостоятельно, либо им выдается пароль для входа в систему. Темы проходят проверку в деканатах, после чего УОО отправляет их в приказ (рис. 15).

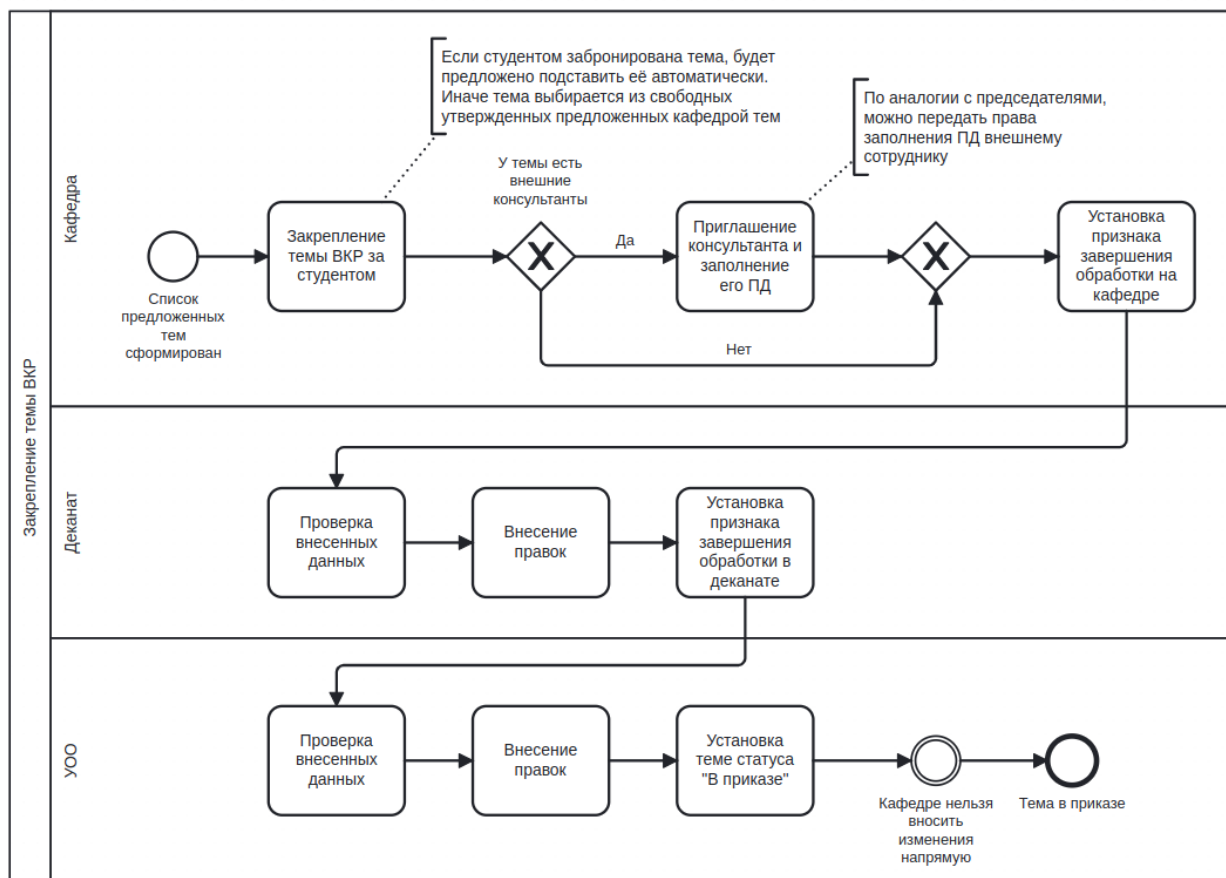
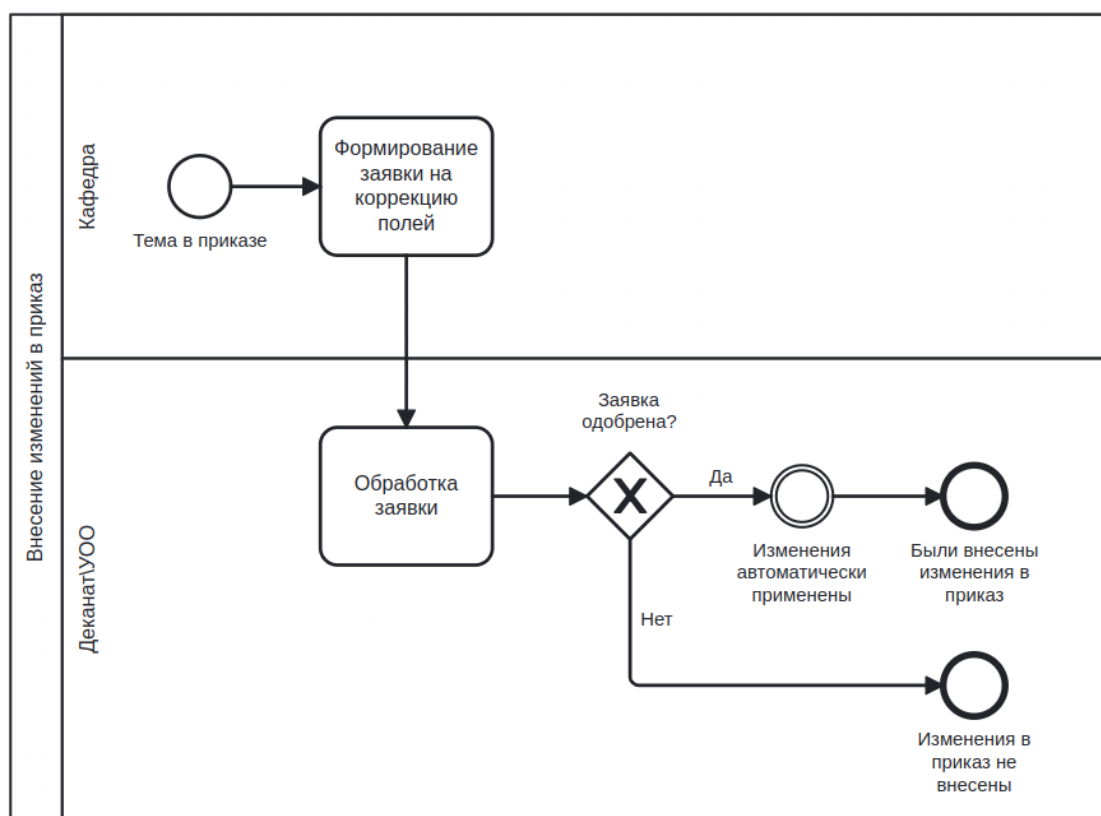


Рисунок 15 — Закрепление тем

После отправки тем в приказ возможность внесения изменений (рис. 16) в закрепление на кафедре блокируется. Кафедра может подать заявку на коррекцию полей, которая обрабатывается в деканате или УОО. Деканат и УОО имеют право свободно вносить изменения в темы в приказе без участия кафедры.



Б

несение изменений в приказ

По результатам формирования закрепления тем ВКР участникам системы доступна выгрузка информации о закреплении в формате Excel для бакалавриата или магистратуры. Кафедры работают только с группами своей кафедры, деканаты — только с группами факультета, а УОО — с группами всех факультетов.

Для кафедр доп. разделов доступна выгрузка пофамильных списков выбравших тот или иной доп. раздел.

2.2.2.3 Приказ о составе ГЭК

После утверждения приказа о составе ГЭК начинается процесс заполнения составов комиссий (рис. 17). Формируется список внешних членов ГЭК на текущий учебный год, где заполняются их персональные данные. Аналогично председателям, внешним членам ГЭК может быть выдан пароль для доступа к системе, что позволит им самостоятельно внести свои персональные данные и прикрепить копии документов.

Также составляется список секретарей. После подготовки списков участников комиссий, кафедры заполняют черновики составов ГЭК.

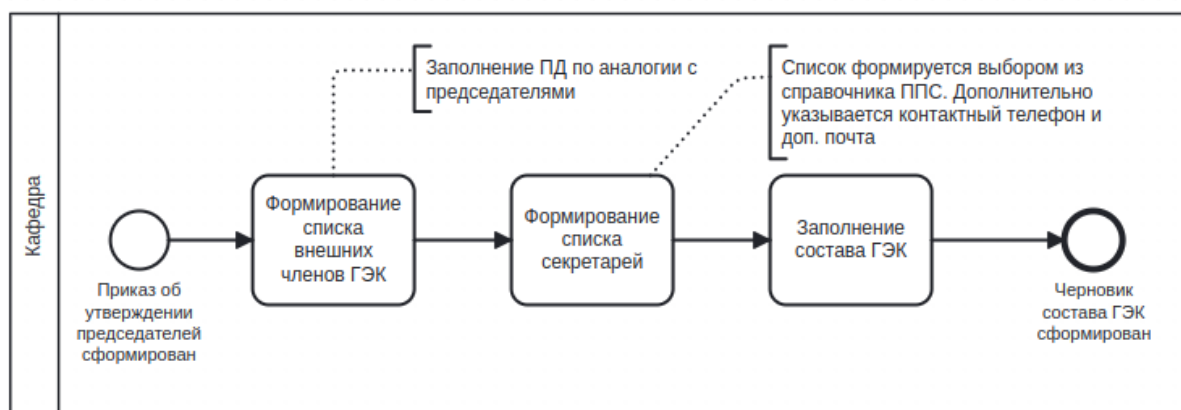


Рисунок 17 — Формирование черновика состава ГЭК

После того, как черновики составов ГЭК были сформированы для всех кафедр, производится проверка процента участия сторонних членов комиссии (рис. 18). После этого кафедра устанавливает признак «Кафедра закончила редактирование» для всех своих ГЭК, и черновые пакеты составов ГЭК отправляются на проверку в деканат.

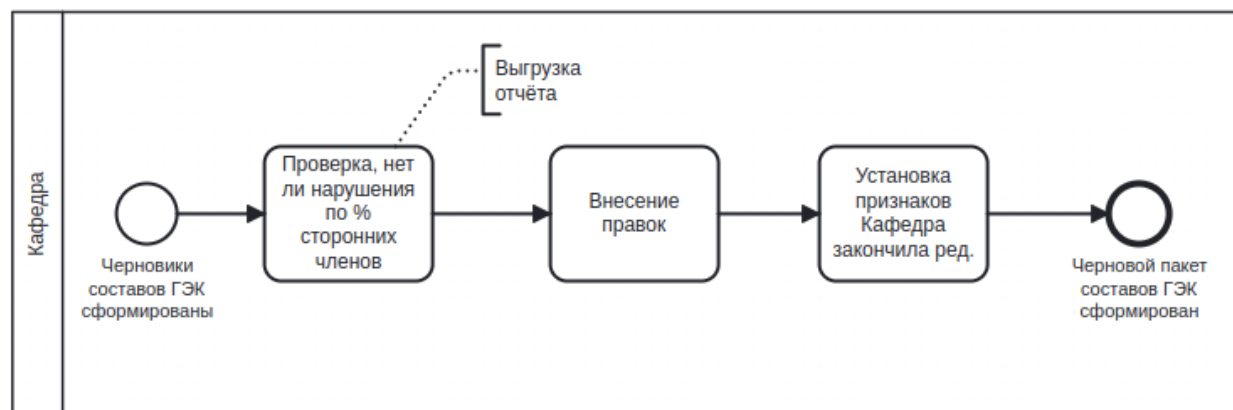


Рисунок 18 — Проверка процента сторонних членов ГЭК

После проверки в деканате черновика состава ГЭК, деканат вносит необходимые правки и устанавливает признак «Деканат закончил редактирование» (рис. 19). Таким образом, деканат осуществляет проверку всех составов ГЭК, подготовленных кафедрами.

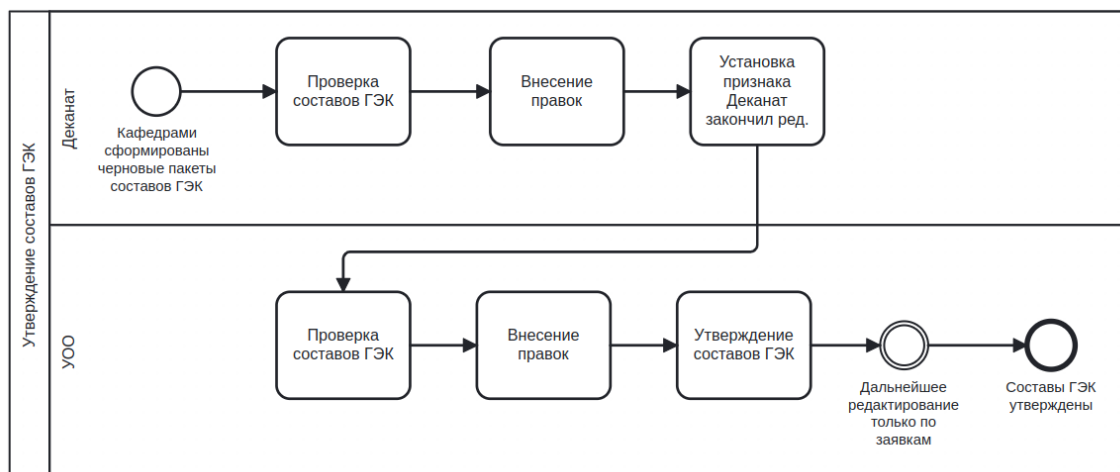


Рисунок 19 — Утверждение составов ГЭК

После окончательной проверки и внесения финальных правок, УОО утверждает состав ГЭК. После утверждения состава ГЭК блокируется возможность вносить изменения в состав напрямую. Дальнейшая работа осуществляется только по заявкам (рис. 20).

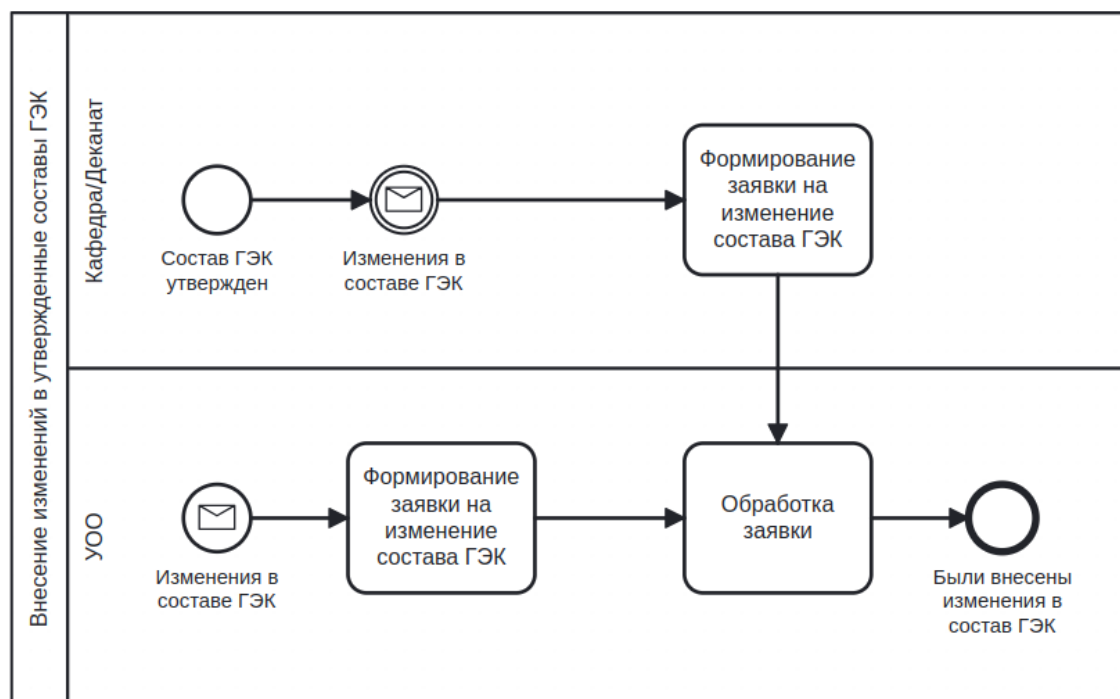


Рисунок 20 — Внесение изменений в утвержденный состав ГЭК

После утверждения всех составов ГЭК или внесения изменений в уже утвержденные составы, формируется приказ об утверждении составов ГЭК (рис. 21). Приказу присваивается номер, который будет виден в выгрузке отчетов для проверки.

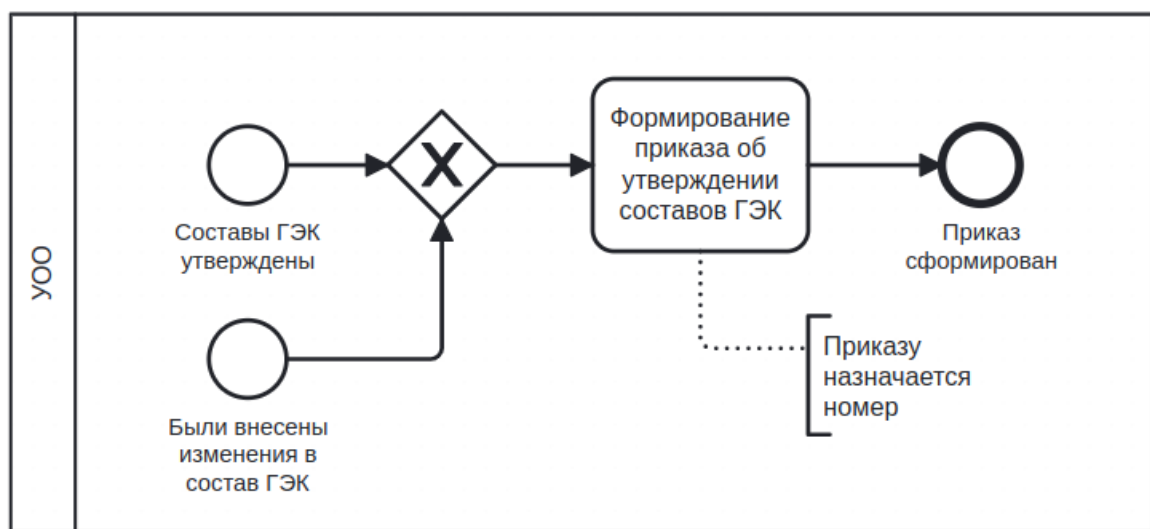


Рисунок 21 — Формирование приказа

2.2.2.4 График защит ВКР

Для каждого ГЭКа формируется график защит выпускных квалификационных работ, в котором указывается контингент студентов, которые будут защищаться (рис. 22).

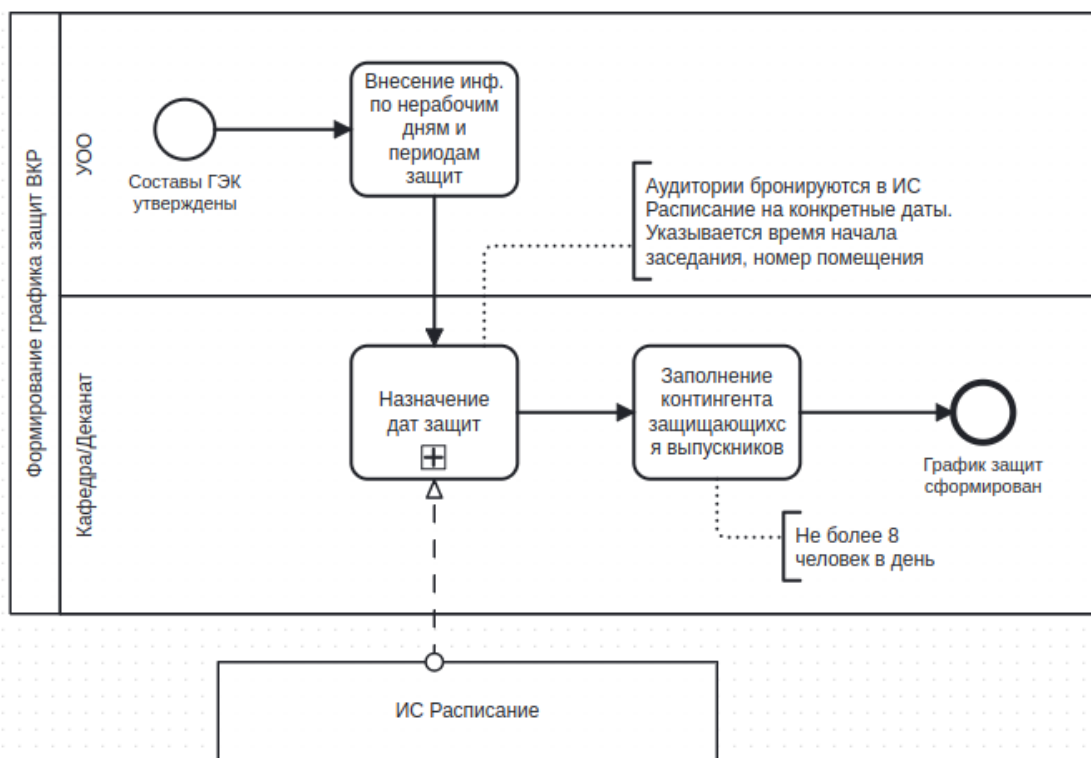
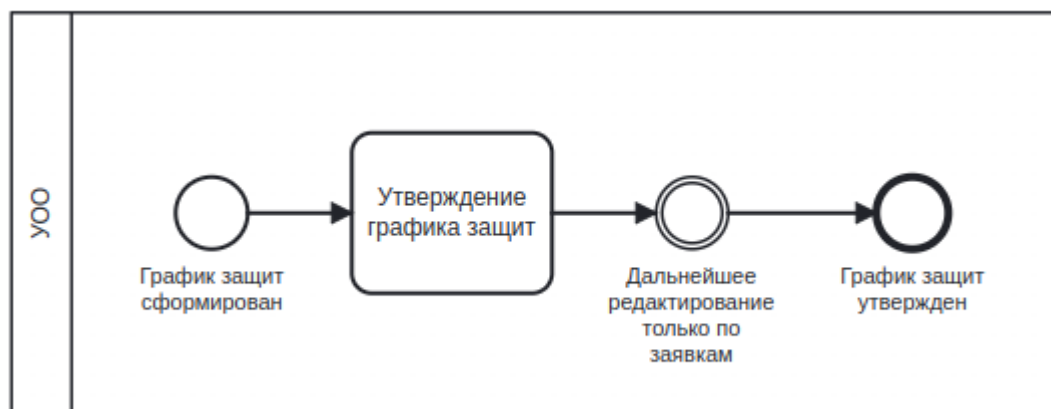


Рисунок 22 — Формирование графика защит ВКР

Далее график защит утверждается УОО отдельно для каждого ГЭК (рис. 23).



23 — Утверждение графика защит ВКР

После утверждения графика защит кафедра и деканат могут вносить изменения только по заявкам, УОО может вносить изменения напрямую (рис. 24).

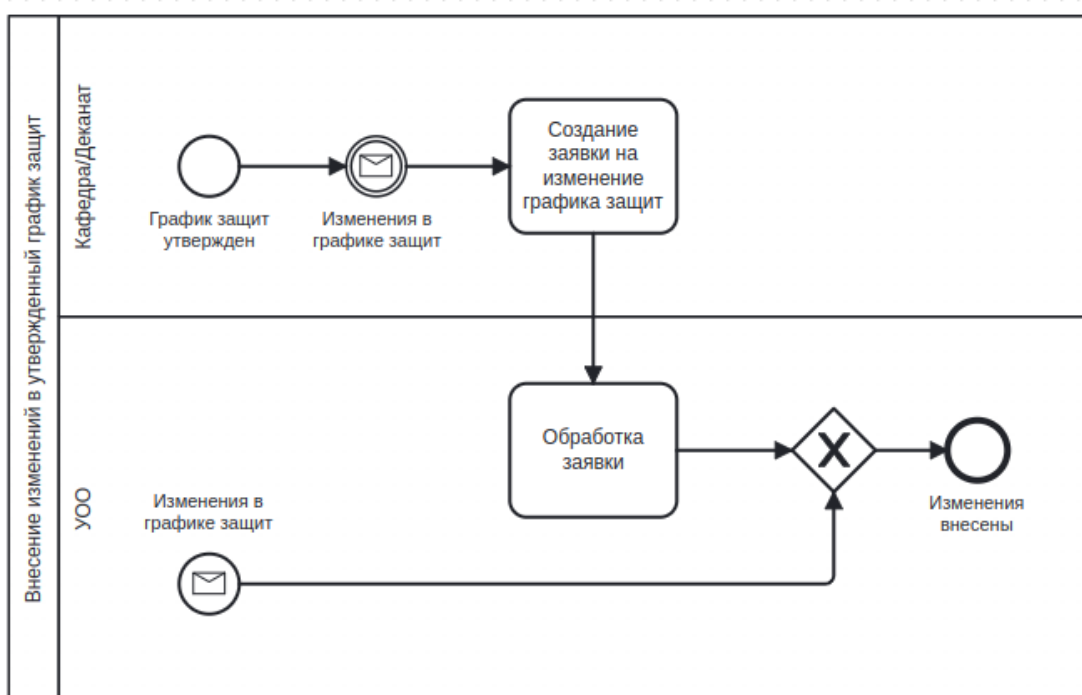
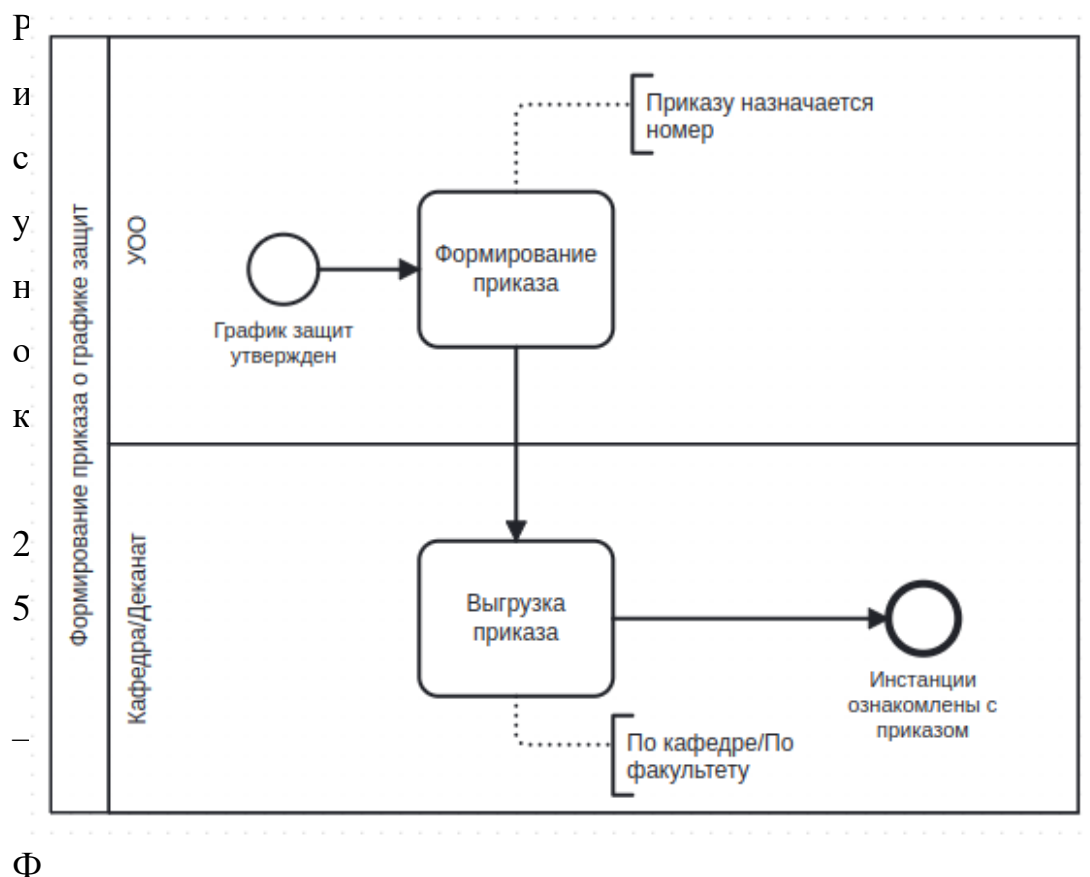


Рисунок 24 — Внесение изменений в утвержденный график защит

После утверждения графика защит формируется приказ о графике защит, в котором присваивается номер (рис. 25). Этот номер становится видимым другим участникам системы при выгрузке приказа.



ормирование приказа о графике защит

2.2.3 Проведение защит, документы секретаря ГЭК

2.2.3.1 Протокол ГЭК на студента

Перед началом заседания ГЭК секретарь заполняет основную информацию о представленных на защиту ВКР. Во время заседания фиксируется присутствие защищающихся и членов комиссии, а также использование ДТО во время проведения защиты. По окончании заседания секретарь дополняет информацию в протоколе (оценка, степень, заданные вопросы), выбирает подписантов среди присутствующих членов ГЭК и печатает протоколы (рис. 26).

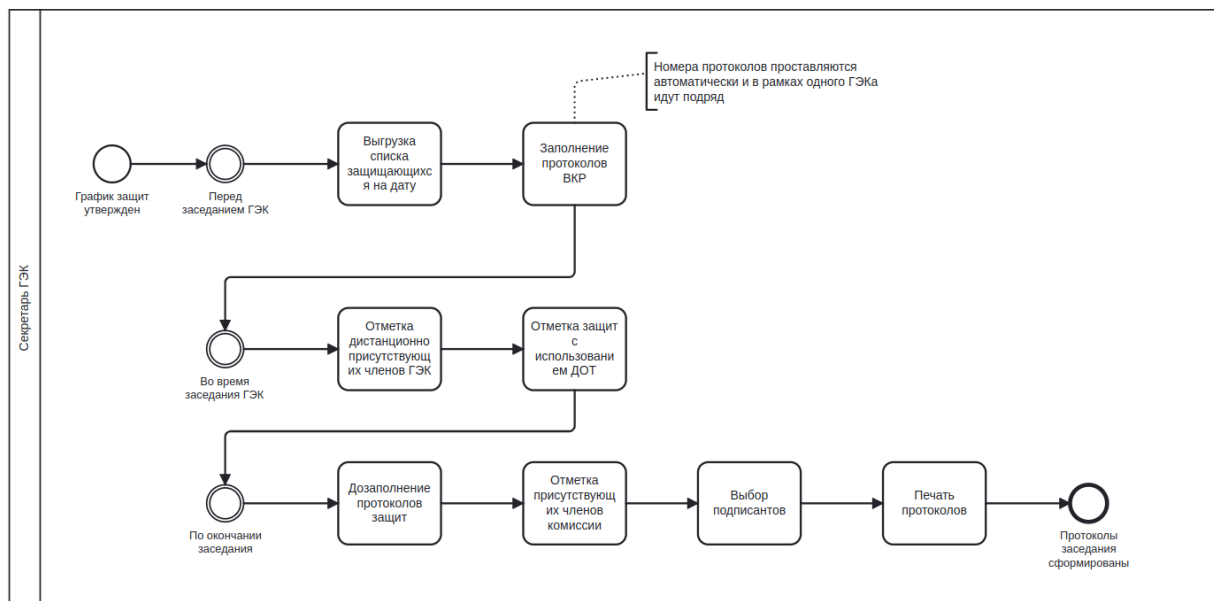
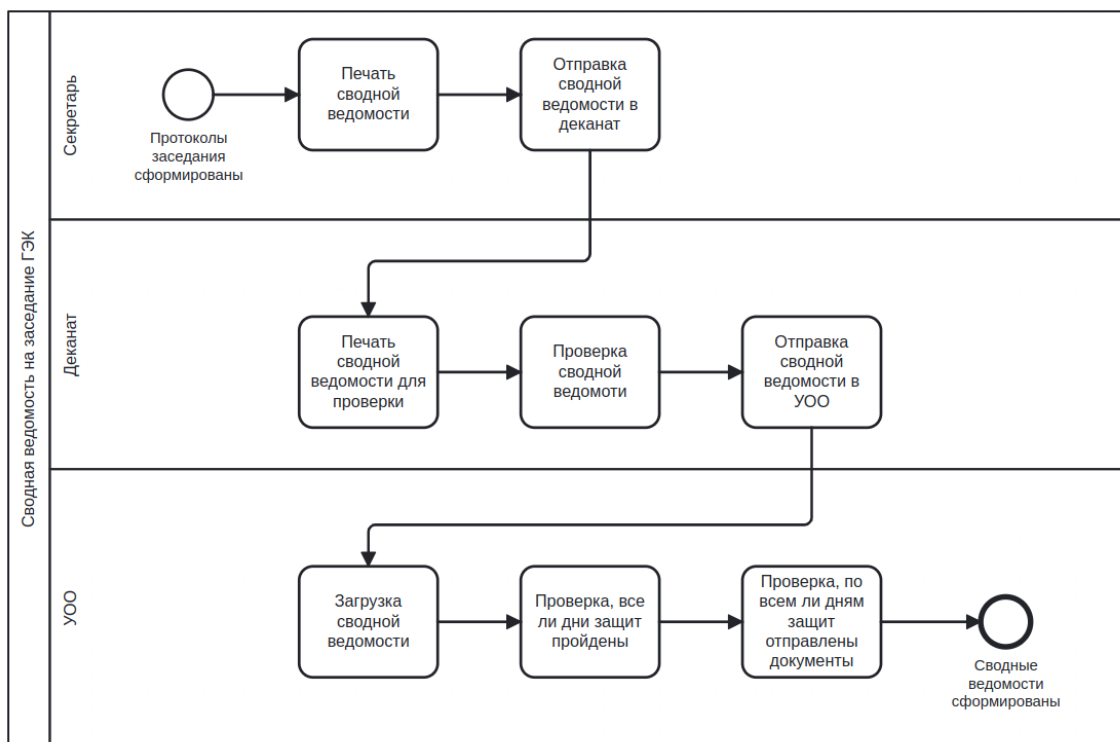


Рисунок 26 — Формирование протоколов заседания

2.2.3.2 Сводная ведомость на заседание ГЭК

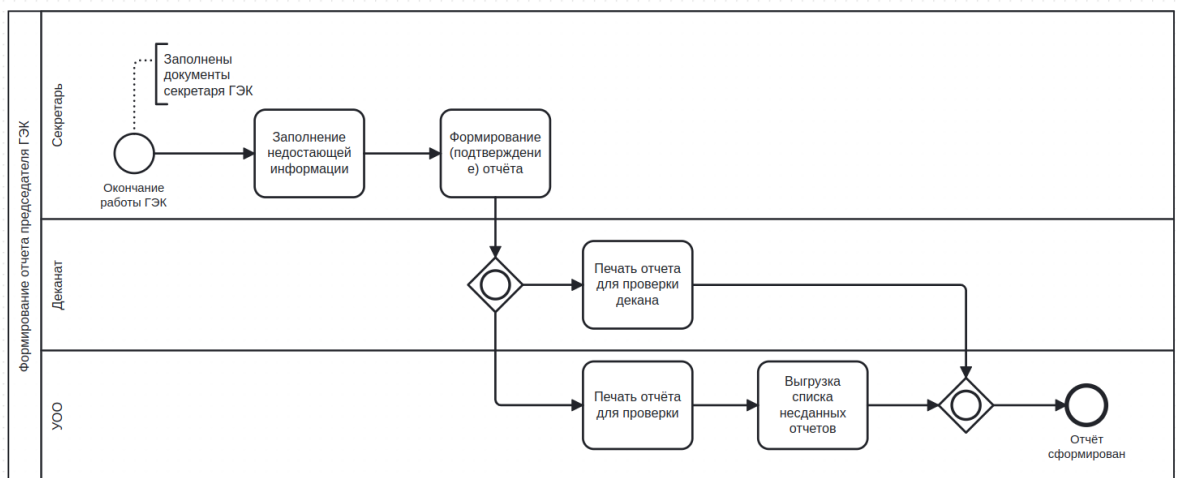
После печати протоколов секретарь должен составить сводную ведомость и отправить ее в деканат для проверки (рис. 27). Деканат уведомляет УОО о возможности загрузки сводной ведомости. УОО контролирует, чтобы все дни защит были завершены, и на всех днях защит были сформированы сводные ведомости.



Формирование сводной ведомости

2.2.4 Отчет председателя

По завершении работы ГЭК секретарь заполняет информацию о сроках работы ГЭК, выделяет замечания, описывает подготовку к допуску и защите, а также информацию о проведении защит и анализе качества выпускных работ и их защиты студентами. Затем секретарь формирует и утверждает отчет (рис. 28). Декану предоставляется выгрузка отчета для проверки. УОО контролирует сдачу всех отчетов, также ему доступна выгрузка несданных отчетов. Участники процесса имеют быстрый доступ к контактам секретаря.



28 — Формирование отчётов ГЭК

2.3 Архитектура базы данных

Для хранения информации на стороне сервера была создана база данных, в соответствии с потребностями бизнес-процессов. Всего база данных включает 67 сущностей. Ниже будут подробнее рассмотрены их ER-диаграмм.

2.3.1 Диаграмма тем ВКР

Для хранения тем ВКР используется сущность Topics (рис. 29). Она содержит:

- Идентификаторы для справочников (studyLevelId — уровень обучения, studentEducationId — обучение студента, leaderDepartmentId — кафедра руководителя)

- Идентификаторы для связанных пользователей (userCreatedId — создатель темы, depEndUserId — сотрудник кафедры, facultyEndUserId — сотрудник деканата)
- Статусы темы (approvalStatus — статус согласования, applicationStatus — статус заявки на тему, status — статус темы)
- Ссылка на доп. раздел (additionalSectionId)
- Название (title — на русском, engTitle — на английском) и обоснование (justificationTopic)
- Флаги (fromPartner — партнёрская, fromStudent — студенческая, isApplication — заявка, isApplicationInDep — прошла ли заявка на кафедре, approvedForSelect — доступна ли тема для назначения в темы ВКР)
- Дополнительные сведения (externalConsultComment — комментарий внешнего консультанта, year — год, type — тип)

Таким образом, сущность Topics содержит всю информацию для того, чтобы тема могла быть предложена студентом или руководителем, прошла все этапы проверки на кафедре и в деканате, описанные в процессе 2.2.2.

- В пункте 2.2.2.1 УОО создаёт сущности доп. Разделов (AdditionalSection). После чего привязанные по departmentId кафедры могут добавлять или редактировать свои доп. разделы. Флаг isEng отвечает за принадлежность к наличие английского названия раздела. Поле year отвечает за год создания и используется для экспорта актуальных распоряжений.
- В пункте 2.2.2.2 кафедра формирует список тем ВКР. Во время формирования преподаватели и студенты могут добавлять темы ВКР. Кафедра подтверждает или отклоняет темы, изменяя флаг applicationStatus.
- При закреплении тем за студентами изменяется статус approvalStatus. После закрепления тем и подготовки всех данных

происходит проверка в деканате и установка флага готовности к приказу (status), который УОО может впоследствии изменить на «В приказе».



Рисунок 29 — ER диаграмма тем ВКР

2.3.2 Диаграмма ГЭЖ

Сущность ExamCommission отвечает за хранение информации о комиссии и содержит:

- Идентификаторы для справочников (specialityId — специальности, departmentId — кафедры, depEndUserId — работник с кафедры, facultyEndUserId — работник с деканата)
- Даты (startDate — дата начала проведения, endDate — окончания проведения ГЭК, year — год проведения)
- Номер ГЭК (number)
- Статус ГЭК (status)

При подготовке заявок на список председателей, описанной в процессе 2.2.1, создаётся запись в таблицу ExamCommission со статусом (status) «Черновик». После чего статус и номер ГЭКа изменяются УОО при утверждении перечня.

Для связи групп, форм обучения и протоколов с комиссией используются таблицы CommissionGroup, CommissionStudyForm и CommissionProtocols соответственно (рис. 30).

Для хранения информации о составе ГЭК используется сущность CommissionMember. Она содержит идентификатор кафедры, ГЭКа и пользователя, информацию о роли (председатель, секретарь, член, заместитель председателя).



Рисунок 30 — ER диаграмма ГЭК

2.3.3 Диаграмма персональных данных члена комиссии

Для хранения персональных данных члена комиссии используется сущность `CommissionMemberData` (рис. 31). Запись `CommissionMemberData` создаётся при добавлении членов ГЭК, заполнении данных председателем (процесс 2.2.1.2) или внешних консультантов (процесс 2.2.2.3).

Сущность связана с пользователем (сущность `Users`) с помощью идентификатора, через которого можно связать `CommissionMember` с его персональными данными. Кафедра выполняет проверку персональных данных председателей, исходя из привязки `CommissionMember` к кафедре.



Рисунок 31 — ER диаграмма члена комиссии

2.3.4 Диаграмма экзаменов

Сущность Exams предназначена для хранения информации экзаменов:

- Идентификатор комиссии (commissionId) и промежутка защит (intervalId)
- Дата (date), время (time) и аудитория (auditoriumNumber)
- Подписанты (signers)

- Флаги (isAccepted — утверждён ли график, isApplication — является подэкзаменом)

Запись в таблицу создаётся при назначении даты защиты (процесс 2.2.2.4): создаются интервалы защит (DictProtectionIntervals), после чего создаётся экзамен, привязанный к интервалу. После создания экзамена добавляются студенты (StudentEducationExams), не более 8 человек и формируется график защит (меняя статус isAccepted). На рис. 32 показана ER диаграмма экзаменов в связи с ключевыми для неё таблицами.

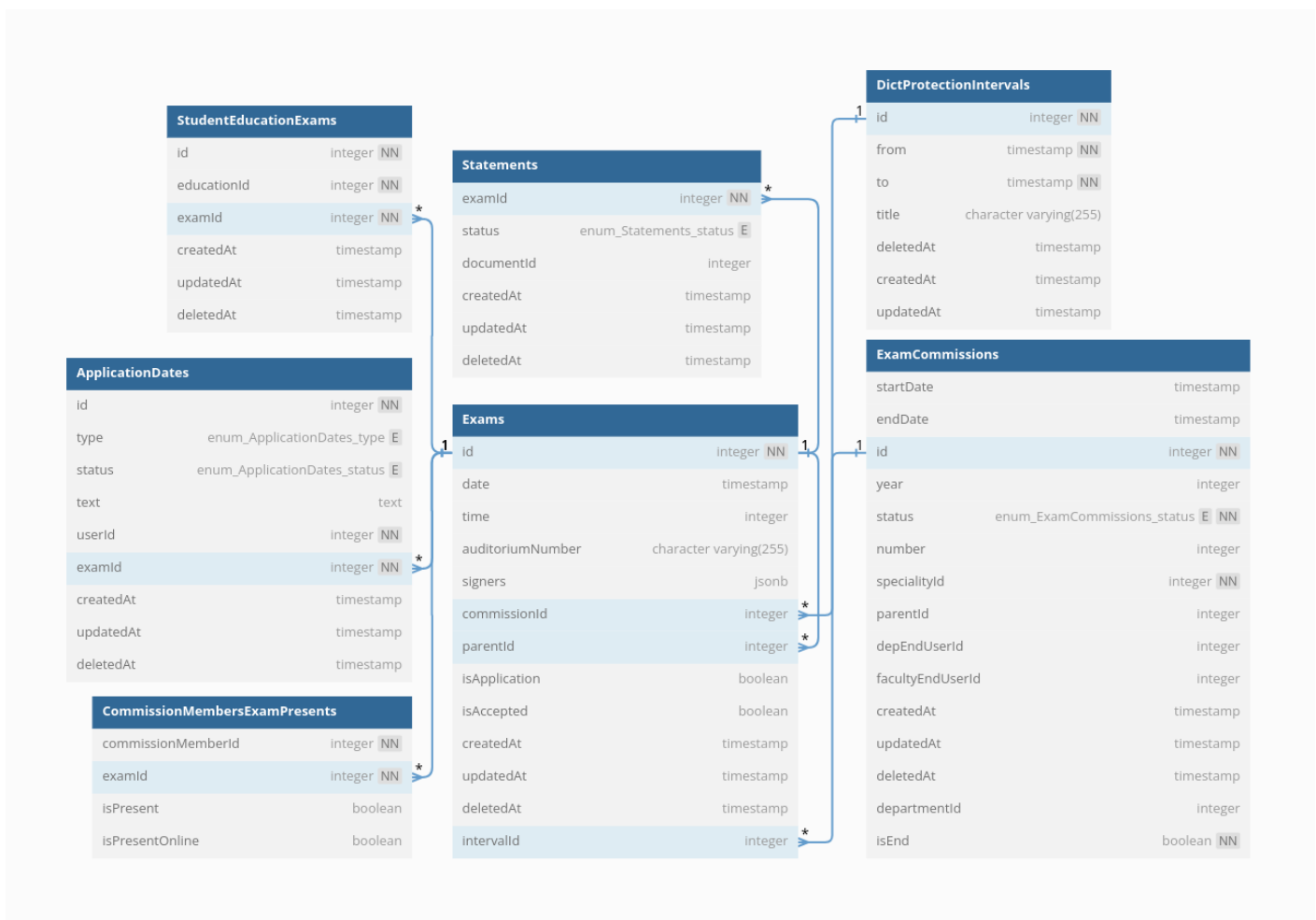


Рисунок 32 — ER диаграмма экзаменов

2.4 Диаграммы классов бизнес процессов

Для реализации серверной части приложения были построены диаграммы классов, реализующие основные шаблоны системы.

2.4.1 Шаблон фабрики

Шаблон фабрики используется для того, чтобы определить фабричный метод, реализация которого будет представлена в наследующемся классе.

Так на рис. 33 показан SequelizeModule, отвечающий за подключение к базе данных. По умолчанию он использует класс SequelizeOptionsFactory с фабричным методом createSequelizeOptions(), который содержит необходимую конфигурацию для подключения к базе данных.

Класс SequelizeConfigService определяет метод createSequelizeOptions, используя данные окружения с помощью ConfigService.

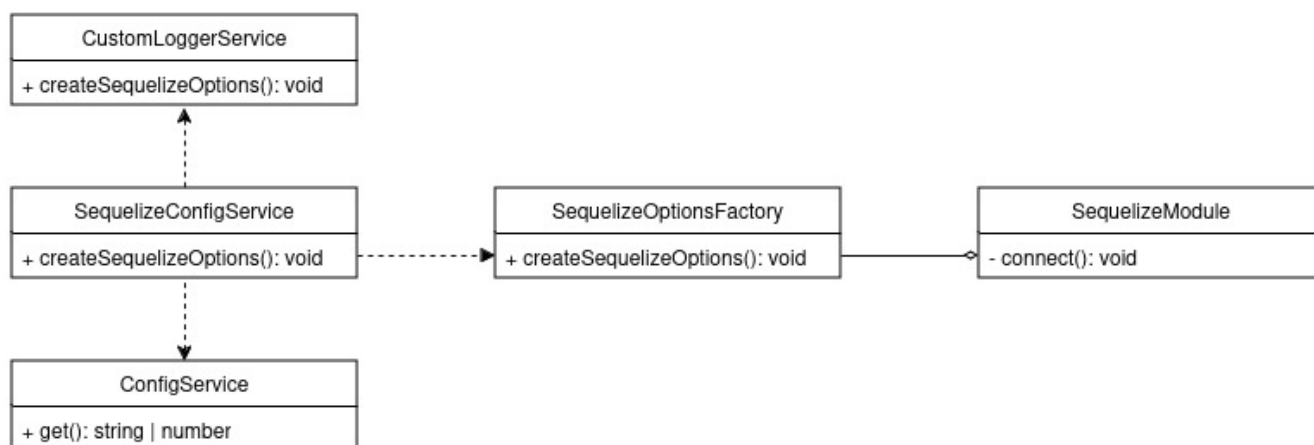


Рисунок 33 — Диаграмма классов для SequelizeConfigService

2.4.2 Шаблон адаптера

На рис. 34 изображена диаграмма класса RightsGuard, который реализует проверку прав доступа клиента к исполнению запроса. RightsGuard реализует класс CanActive, позволяя использовать себя в корневом модуле (AppModule) для предобработки запросов.

RightsGuard использует AuthService для того, чтобы получить данные пользователя и произвести проверку, обращаясь к базе данных. Таким образом RightsGuard является адаптером, между методом проверки (checkRights) и вызовом этой проверки (canActive).

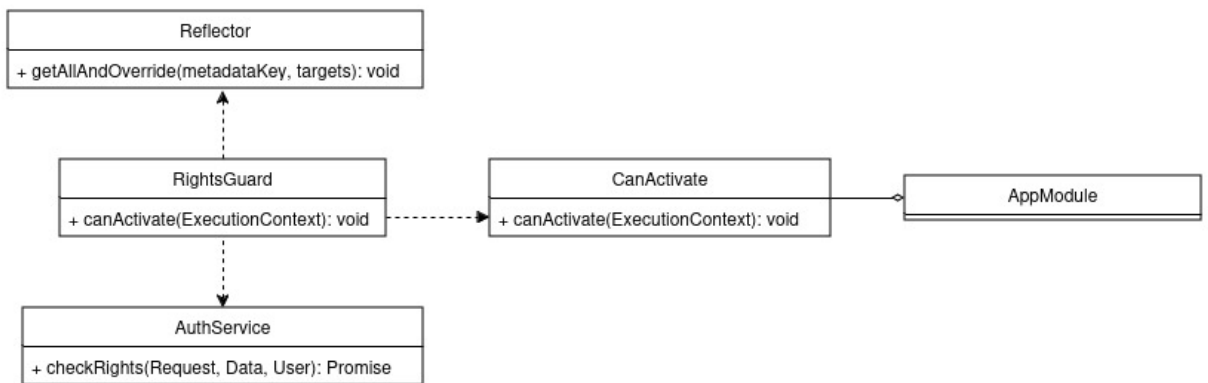


Рисунок 34 — Диаграмма классов для RightsGuard

2.4.3 Шаблон декоратор и синглтон

Внедрение зависимостей — это шаблон на котором построен NestJS. На рис. 35 приведена диаграмма модуля ServiceModule. Диаграмма отражает, что ServiceModule является связующим звеном для ServiceController и ServiceService. Объекты этих классов являются одиночками, то есть существуют в единственном экземпляре, и хранятся во внешнем контейнере приложения. Каждый раз, когда контроллер обращается к сервису, он пытается либо создать, либо получить уже существующий объект.

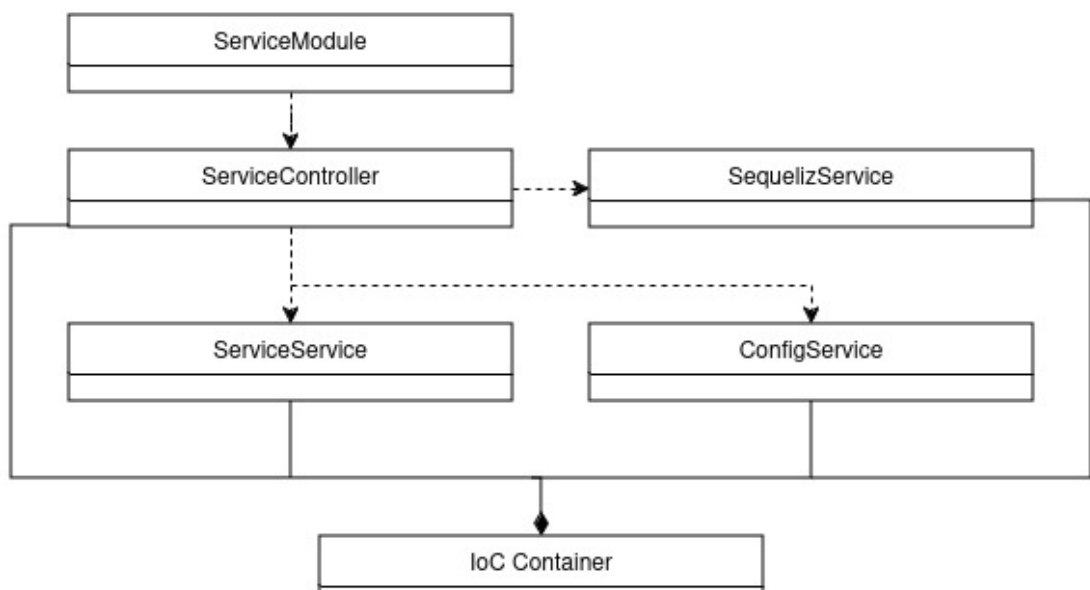


Рисунок 35 — Диаграмма классов для ServiceModule

2.5 Выводы

Исходя из поставленных задач, для реализации и поддержки системы было подготовлено:

- Описание архитектуры и технологий серверной части: бизнес-логика, уровень доступа к данным и уровень хранения данных
- Описание бизнес процессов, которые необходимо реализовать, с использованием BPMN-диаграмм.
- Описание архитектуры базы данных, из 67 сущностей, основные сущности были описаны с использованием ER-диаграмм.
- Описание использованных шаблонов программирования, для них на примерах были построены UML-диаграмм классов.

3 РЕАЛИЗАЦИЯ

3.1 Введение

В данной главе описываются детали реализации серверной части предложенного решения. Отдельно рассматриваются запросы бизнес процессов, интеграция с другими ИС, тестирование приложения и результаты апробации в вузе.

3.2 Реализация бизнес процессов

Для реализации бизнес процессов были разработаны модули, указанные на рис. 1. Ниже подробнее рассматриваются два ключевых модуля: служебный и системный.

3.2.1 Служебные модули

Служебные модули отвечают за обслуживание системы.

3.2.1.1 Загрузка файлов

Модуль `FileLoaderModule` отвечает за загрузку файлов от клиента локально на сервер. Для этого сервис `LoaderMulterService` использует пакет `Multer`, который является обработчиком промежуточного слоя и имеет готовые инструменты для обработки файлов, приходящих в запросах типа `multipart/form-data`.

`LoaderMulterService` реализует пакетный класс `MulterOptionsFactory`, конфигурируя:

- Директорию сохранения файлов
- Ограничения на формат и размер файлов

3.2.1.2 Справочники

`DictsModule` — модуль справочников системы, который отвечает за запросы к общеузовской информации: департаменты, факультеты, формы обучения, уровни обучения, звания и должности, специальности, рабочие дни и интервалы защит. Информация в словарях в основном приходит из внешних ИС и должна поддерживаться актуальной во всей системе.

3.2.1.3 Логирование

LoggerModule — модуль, отвечающий за логирование изменений в системе и сохранение этой информации на сервере. Логирование реализовано благодаря библиотеке winston. Благодаря реализации встроенного класса LoggerService, NestJS позволяет создать своего обработчик запросов, в зависимости от типов сообщений системы.

3.2.2 Системные модули

Системные модули содержат в себе модули, реализующие основную бизнес логику приложения: обрабатывают основные запросы, приходящие от клиента. Рассмотрим три важных контроллера: контроллер ГЭКов, ВКР и интеграции с другими ИС.

3.2.2.1 Перечень ГЭК

ExamCommissionModule — модуль, обрабатывающий запросы, связанные с перечнями ГЭК:

- Получение по id и выгрузка в таблицу
- Добавление, удаление и обновление
- Настройка членов ГЭК
- Экспорт в excel связанных данных: списка и состава ГЭК, утверждения председателей, списка групп с председателями и направлениями

Экспорт в excel реализован с помощью провайдера commissionReportExportExcel, который использует интерфейс пакета exceljs для создания и работы с документом.

Все права на получение, редактирование, добавление и удаление информации описаны в соответствии с процессами. С помощью сервиса авторизации (AuthService) система, используя токен пользователя в запросе, проверяет приходящий запрос на соответствие необходимым правилам, например: наличие нужной роли.

3.2.2.2 Темы ВКР

TopicsModule — модуль тем ВКР. Контроллер модуля обрабатывает запросы:

- Выгрузку тем в таблицу или получение по id
- Добавление/удаление консультантов
- Добавление/обновление/удаление тем
- Экспорт тем в excel и по токenu

Методы для выгрузки в excel идентичны описанным в п. 3.2.2.1. Экспорт по токenu реализован для того, чтобы другие ИС могли получить данные по запросу. Для этого в двух системах создаётся токен, записывается в файл окружения, после чего сервер проверяет запрос на наличие такого токена и на разрешение обрабатывать запрос по токenu.

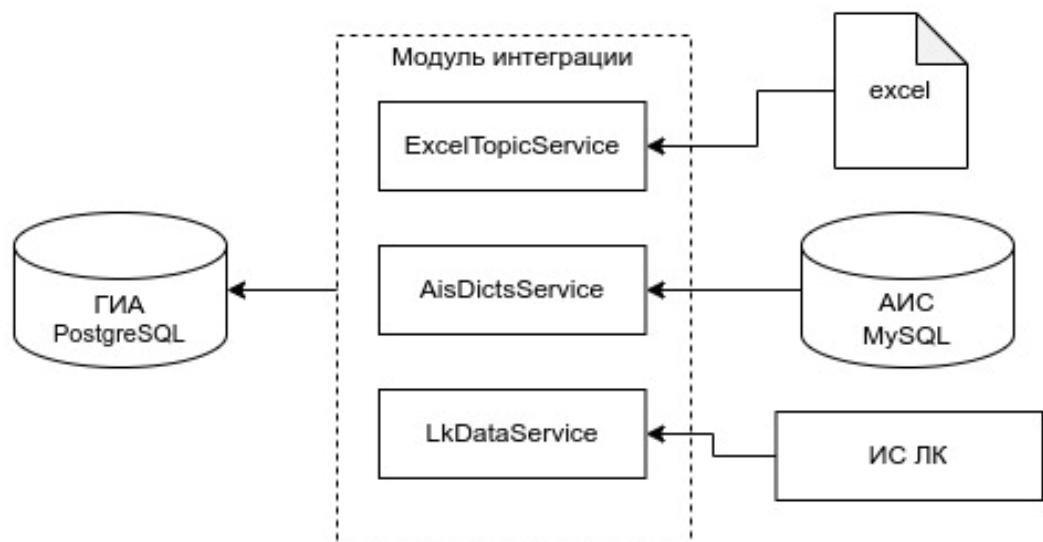
3.2.2.3 Интеграция с ИС

IntegrationModule — модуль для импорта данных из других ИС (IntegrationService) и excel документов (ExcelTopicService), представлен на рис. 36. Права на импорт данных есть как у администраторов, так и у УОО.

Импорт из документов идентичен п. 3.2.2.1: файл загружается на сервер, после чего с помощью библиотеки exceljs происходит чтение и обновление информации в базе данных.

Импорт из ИС «Личный кабинет» проводится по выделенному на стороне АИС интерфейсу, с предоставлением специального токена (BEARER_TOKEN).

Импорт справочников проводится с помощью прямых запросов в базу данных АИС, через настроенный в базе аккаунт с доступом на чтение. С помощью библиотеки mssql создаётся подключение к базе, после чего посылаются запросы, информация из которых обрабатывается и попадает в базу данных сервера приложения.



Модуль интеграции ИС «ГИА»

3.3 Тестирование

Было проведено тестирование реализованной программы с целью выявления ошибок и ограничений системы:

3.3.1 Модульное и интеграционное тестирование

Интеграционное и модульное тестирование серверной части реализовано с помощью библиотеки `jest`.

Система содержит 36 модульных тестов, проверяющих:

- Исполнение транзакций для: тем ВКР и графиков защиты
- Получению информации из справочников
- Загрузке файлов

NestJS создаёт автоматические интеграционные тесты, проверяющие модули на верность сборки композиции, куда можно добавить дополнительные проверки.

Также с помощью программы Postman были написаны интеграционные тесты интерфейса приложения, которые проверяли приложение на безопасность и устойчивость, при попытке введения данных, отличных от предоставляемого интерфейсом клиента.

3.3.2 Тестирование SQL инъекций

Запросы были дополнительно проверены на наличие sql инъекций с помощью программы sqlmap. В ходе тестирования инъекций обнаружено не было.

3.3.3 Нагрузочное и стресс тестирование

Для уточнения ограничений системы с помощью библиотеки k6 были написаны нагрузочные и стресс-тесты, которые, варьируя количество виртуальных пользователей, частоту отправки запросов и количество запросов, позволяют узнать как система ведёт себя под нагрузками.

Для того, чтобы пользователям было комфортно использовать систему, установим пороговое среднее время ожидания ответа от сервера в 1000 мс. Написанные тесты показывают, что:

- Стабильно система способна обрабатывать до 250 запросов в секунду с длительностью ожидания менее 0.8 секунды
- Предельно система обрабатывает до 270 запросов в секунду от 150 пользователей с длительностью ожидания более 1 секунды.

3.4 Апробация в вузе

На момент написания ВКР серверная часть ИС «ГИА» была развёрнута на серверах и проходит стадию внедрения. Разработчикам клиентской части было предоставлено API серверной части, используя библиотеку Swagger. Пользователям была предоставлена документация и доступ к веб приложению для начала работы. В системе уже было создано:

- 28 полностью подтверждённых тем
- 259 внешних консультанта
- 9 утверждённых ГЭК
- 37 экзаменов

Система продолжает дорабатываться и улучшаться, на основании найденных ошибок и пожеланий пользователей.

3.5 Выводы

В главе 3 рассмотрены детали разработки серверной части веб приложения государственной итоговой аттестации.

Описаны детали работы ключевых модулей приложения, которые реализуют обслуживание системы, интеграцию с другими ИС, загрузку и выгрузку файлов, обработку пользовательских запросов.

Описано проведённое тестирование: модульное, благодаря которым в ходе разработки были выявлены ошибки, интеграционное, которое при обновлениях позволяет автоматически проверять работоспособность старых и новых запросов. Нагрузочное и стресс тестирование установило ограничения системы на допустимое количество пользователей и запросов в секунду: до 250 запросов при нагрузочном, и до 270 при стресс тестировании.

Серверная часть внедрена в вуз и предоставлена пользователям. О функционировании системы говорят успешно добавленные записи, однако система продолжает поддерживаться и улучшаться, чтобы пополнять доступный функционал и повышать удобство для пользователей.

4 ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ

4.1 Введение

В данной главе будет проведено экономическое обоснование для разработки серверной части веб приложения государственной итоговой аттестации. Используем методическое пособие [8], на основе которого рассчитаем следующие значения:

- Расходы на оплату труда
- Затраты на услуги сторонних организаций
- Величина амортизационных отчислений
- Затраты на программное обеспечение
- Затраты на электроэнергию

После чего вычислим совокупные затраты.

В заключении необходимо проверить актуальность предложенного решения, сравнив его с указанными в главе 1 аналогами.

4.2 Расходы на оплату труда

В разработке серверной части веб приложения участвовали два сотрудника университета: студент, на позиции разработчика, и преподаватель, на позиции руководителя разработки.

В таблице 2 приведём фактические этапы разработки для разработчика, на момент написания ВКР:

Таблица 2 — Фактические этапы разработки: разработчик

№ п/п	Наименование работы	Продолжительность, дней
1	Разработка технического задания с руководителем	7
2	Построение архитектуры серверной части и выбор технологий	2
3	Построение архитектуры базы данных	5
4	Разработка серверной части	30
5	Тестирование серверной части	7
6	Исправление ошибок, выявленных в ходе тестирования	5
7	Разворачивание веб приложения на сервере	1

В таблице 3 приведём фактические этапы разработки для руководителя:

Таблица 3 — Фактические этапы разработки: руководитель

№ п/п	Наименование работы	Продолжительность, дней
1	Разработка технического задания с разработчиком	2
2	Согласование технического задания с заказчиком	1
3	Руководство ходом разработки	5
4	Демонстрация результата заказчику	1
5	Составление плана разработки на основании обратной связи от заказчика	1

При условии, что средняя заработная плата инженера-программиста составляет 42 тыс. руб. в месяц за полный рабочий день, а руководителя 84 тыс. руб в месяц, то получаем ставку в 2000 руб. и 4000 руб. в день соответственно.

Рассчитаем расходы на основную заработную плату по формуле: $Z = \sum T_i \cdot C_i$,

где Z — расходы на основную заработную плату, k — количество исполнителей, T_i — время i исполнителя, C_i — ставка исполнителя.

Получим: $Z = 7 \cdot 4000 + 57 \cdot 2000 = 142$ тыс. руб.

Расходы на дополнительную заработную плату исполнителей определяются по формуле, с учётом нормативов отчисления страховых взносов на обязательное социальное, пенсионное и медицинское страхование, равным 15%: $Z_{\text{доп}} = 142 \cdot 10^3 \cdot 0.15 = 21.3 \text{ тыс. руб.}$

Отчисления на страховые взносы, на обязательное социальное, пенсионное и медицинское страхование с основной и дополнительной заработной платы исполнителей: $Z_{\text{соц}} = (142 + 21.3) \cdot 0.3 = 48.99 \text{ тыс. руб.}$

Сведём результаты в итоговой таблице 4.

Таблица 4 — Расходы на оплату труда

Основная з.п., тыс. руб.	Дополнительная з.п., тыс. руб.	Социальные отчисления, тыс. руб.	Итого, тыс. руб.
142	21.3	48.88	212.18

4.3 Расходы на содержание и эксплуатацию оборудования

Затраты на содержание и эксплуатацию оборудования определяются из расчета на 1 час работы оборудования с учетом стоимости и производительности.

Для разворачивания веб приложения необходимо оборудование для сервера. Себестоимость одного машино-часа работы оборудования: C_i , следовательно расходы на эксплуатацию оборудования: $Z_{\text{эо}} = C_i \cdot 24 = 878.233$

Данная сумма ежедневно тратится на поддержание серверов, на котором разворачивается несколько веб приложений. Из расчётов, что приложение было развёрнуто лишь во время демонстрационного периода получаем (1 день): 878.233 руб.

4.4 Амортизационные отчисления

Амортизационные отчисления необходимо посчитать для оборудования на котором велась разработка (Lenovo Legion Y530-15ich) и для оборудования, на котором размещаются сервера для приложения.

Согласно постановлению правительства [9], использованные средства принадлежат второй группе со сроком полезного использования от 2 до 3 лет включительно, следовательно: $H_{ai} = 33,3\%$.

Амортизационные отчисления за год (A_i) и за время использования во время написания ВКР ($A_{iВКР}$) для ноутбука и сервера приведены в таблице 5.

Таблица 5 — Амортизационные отчисления

№	Оборудование	Начальная стоимость, тыс. руб.	A_i , тыс. руб.	$A_{iВКР}$, тыс. руб.
1	Lenovo Legion Y530-15ich	30	9.99	1.58175
2	DELL R630 8SFF	85	28.305	0.078625

Суммарно амортизационные отчисления составляют 1.660375 тыс. руб.

4.5 Накладные расходы

Во время разработки и поддержки веб приложения были произведены накладные расходы на электроэнергию для питания ноутбука и сервера.

Ноутбук, мощностью 135ватт, работал в течение 57 дней разработки по 8 часов в сутки. Сервер, мощностью 750ватт, работал 24 часа в течение 1 дня демонстрации приложения.

Тариф на электроэнергию в дневное время в Санкт-Петербурге составляет 6.51 руб/кВтч.

Сведём накладные расходы в таблице 6:

Таблица 6 — Накладные расходы

№	Оборудование	Часовой расход, руб	Суммарное работы, час	Итого
1	Lenovo Legion Y530-15ich	0.87885	456	400.7556
2	DELL R630 8SFF	4.8825	24	117.18

Суммарно получаем накладные расходы на сумму 517.9356 рублей.

4.6 Вывод

В результате были подсчитаны затраты на разработку представленного в ВКР серверного веб приложения, куда вошли пункты оплаты труда, амортизационных отчислений, накладных расходов и расходы на содержание оборудования. Результаты сведены в таблицу 7.

Таблица 7 — Суммарные затраты

№	Статья расходов	Сумма, руб
1	Оплата труда	212180
2	Расходы на содержание и эксплуатацию оборудования	878.233
3	Амортизационные отчисления	1660.375
4	Накладные расходы	517.9356
Итого		215236.5436

Статья «Обоснование эффективности внедрения систем электронного документооборота и поддержки принятия решений в условиях цифровых трансформаций» [10] доказывает эффективность замены бумажного документооборота на цифровой.

Рассматривая аналоги, приведённые в главе 1, можно сделать вывод о том, что технологии либо не являются доступными, либо требуют доработок, а значит время, необходимое на изучение документации новых технологий, их внедрение и обучение, добавится к приведённым в таблице 2 этапам, что повысит стоимость разработки систем.

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы (ВКР) было разработано web-приложение для проведения государственной итоговой аттестации вуза.

Перед разработкой были поставлены задачи и проведён анализ реализованных технологий. Анализ проводился по основным выделенным критериям. В результате было установлено, что проблема создания web-приложения для проведения государственной итоговой аттестации вуза актуальна, но существующие аналоги не удовлетворяют всем критериям сравнения.

Для разработки собственного решения были описаны бизнес-процессы системы, построено 27 BPMN-диаграмм, выделены ключевые роли (6 ролей) и 10 сценариев использования. Выбраны технологии и создана архитектура серверной части, построено 3 диаграммы классов, и базы данных, было сделано 5 ER диаграмм.

В соответствии с выделенными задачами и описанными процессами была разработана серверная часть web-приложения, предоставляющая документацию к своему интерфейсу, используя Swagger.

Проведено тестирование полученной программы: 36 модульных тестов с покрытием в 35%, интеграционные тесты для интерфейса и тестирование на взлом, инструментом sqlmap. Нагрузочные и стресс тесты проводились для всего приложения, чтобы определить ограничения системы.

Проведена апробация приложения в вузе, в результате чего на момент написания ВКР, было создано: 28 полностью подтверждённых тем, 259 внешних консультанта, 9 утверждённых ГЭК, 37 экзаменов. В среднем приложением пользовалось 3000 человек в неделю. Апробация демонстрирует работоспособность разработанной программы, а пожелания пользователей создают направления для развития и улучшения системы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Решение от 20 февраля 2018 года N 40-5 «Развитие информатизации системы образования. Совершенствование законодательства в области электронного обучения и дистанционных образовательных технологий» URL: <https://docs.cntd.ru/document/556985932>
2. Развитие цифровой экономики в России. Доклад Всемирного Банка 20 декабря 2016г. URL: <http://gosbook.ru/node/94904>
3. Лёвина А. И., Гусынина Д. А. Автоматизация процессов управления учебнометодической деятельностью структурного подразделения университета // ФУНДАМЕНТАЛЬНЫЕ И ПРИКЛАДНЫЕ ИССЛЕДОВАНИЯ В ОБЛАСТИ УПРАВЛЕНИЯ, ЭКОНОМИКИ И ТОРГОВЛИ. — 2017.
4. Волканин Л.С., Хачай А.Ю. ПРАКТИЧЕСКИЕ СЦЕНАРИИ БЕСШОВНОЙ ИНТЕГРАЦИИ «1С:УНИВЕРСИТЕТ» И «1С:ДОКУМЕНТООБОРОТ» ПРИ АВТОМАТИЗАЦИИ БИЗНЕС-ПРОЦЕССОВ ВУЗА. Информатика и образование. 2018;(3):39-43. URL: <https://info.infojournal.ru/jour/article/view/274/275>
5. Клишин А. П. и др. Подходы к автоматизации документооборота в вузе // Вестник Новосибирского государственного университета. Серия: Информационные технологии. — 2017. — Т. 15. — №. 1. — С. 36-46. URL: <https://cyberleninka.ru/article/n/podhody-k-avtomatizatsii-dokumentoooborota-v-vuze>
6. Тынченко В. В., Тынченко Я. А. Автоматизация информационных процессов проведения государственной итоговой аттестации выпускников вуза // Решетневские чтения. — 2016. — Т. 2. — №. 20. — С. 169-170.
7. Замыслова, С. С. Автоматизация итоговой государственной аттестации выпускников / С. С. Замыслова, М. А. Едунова // Проспект Свободный — 2023 : Материалы XIX Международной научной конференции студентов, аспирантов и молодых ученых, Красноярск, 24—29 апреля

2023 года. — Красноярск: Сибирский федеральный университет, 2023.
— С. 2294-2297. URL: <https://elibrary.ru/item.asp?id=54740294>

8. Т. Н. Лебедева ВЫПОЛНЕНИЕ ДОПОЛНИТЕЛЬНОГО РАЗДЕЛА ВКР «ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ» СПЕЦИАЛИСТАМИ ФКТИ СПбГЭТУ «ЛЭТИ». - Санкт-Петербург: Издательство СПбГЭТУ «ЛЭТИ», 2022. - 37 с.
9. РФ от 01.01.2002 № 1 (ред. От 18.11.2022) «О классификации основных средств, включаемых в амортизационные группы».
10. М.А. Мирошниченко, К.А. Кузнецова, О.В. Могильда ОБОСНОВАНИЕ ЭФФЕКТИВНОСТИ ВНЕДРЕНИЯ СИСТЕМ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА И ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ В УСЛОВИЯХ ЦИФРОВЫХ ТРАНСФОРМАЦИЙ // Вестник Академии знаний. 2020. №5 (40). URL: <https://cyberleninka.ru/article/n/obosnovanie-effektivnosti-vnedreniya-sistem-elektronnogo-dokumentoooborota-i-podderzhki-prinyatiya-resheniy-v-usloviyah-tsifrovyyh> (дата обращения: 10.05.2024).