

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Институт информационных технологий и анализа данных**

наименование института

Допускаю к защите

Руководитель ООП

  
подпись

**Р.В. Кононенко**

И.О. фамилия

**Разработка методики применения ЭЦП  
для оптимизации работы со студентами в ИРНТУ**

наименование темы

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
к выпускной квалификационной работе бакалавра  
Программа бакалавриата  
Автоматизированные системы обработки информации и управления

наименование программы

по направлению подготовки  
09.03.01 «Информатика и вычислительная техника»

Код и наименование направления подготовки

**0. 020.00.00 ПЗ**

обозначение документа

Разработал студент группы **АСУб-21-1**

Руководитель

Нормоконтроль

  
подпись

**В.А. Колупаев**

И.О. Фамилия

  
подпись

**П.А. Петров**

И.О. Фамилия

  
подпись

**П.А. Петров**

И.О. Фамилия

Иркутск 2025 г.



**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Институт информационных технологий и анализа данных**  
наименование института

УТВЕРЖДАЮ

Директор института ИТ и АД

А.С. Говорков

« 08 » 04. 2025 г.

**ЗАДАНИЕ**

на выпускную квалификационную работу студенту Колупаеву Вячеславу  
Андреевичу

группы АСУ6-21-1

1 Тема работы: Разработка методики применения ЭЦП для оптимизации  
работы со студентами в ИРНТУ

Утверждена приказом по университету от 15.04.2025 № 873

2 Срок представления студентом законченной работы в ГЭК 02.06.2025 г.

3 Исходные данные

3.1 СТО 005-2020 «Система менеджмента качества. Учебно-  
методическая деятельность. Оформление курсовых проектов (работ) и  
выпускных квалификационных работ технических направлений  
подготовки и специальностей»

3.2 Материалы преддипломной практики

4 Содержание расчетно-пояснительной записки (перечень подлежащих  
разработке вопросов):

4.1 Анализ предметной области

4.2 Теоретическое проектирование интеграции ЭЦП

4.3 Методика применения ЭЦП

4.4 Разработка и тестирование прототипа

4.5 Рекомендации по внедрению и дальнейшему развитию

5 Перечень графического материала (с указанием обязательных чертежей)

6 Дополнительные задания и указания

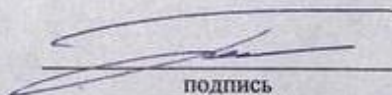


# Календарный план

Разделы	Месяцы и недели											
	апрель				май				июнь			
Введение				+	+							
1. Анализ предметной области				+	+	+	+					
2. Теоретическое проектирование интеграции ЭЦП						+	+					
3. Методика применения ЭЦП						+	+	+				
4. Разработка и тестирование прототипа							+	+	+			
5. Рекомендации по внедрению и дальнейшему развитию								+	+			
Заключение								+	+			
Оформление пояснительной записки							+	+	+			
Подготовка к защите ВКР										+	+	+

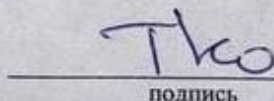
Дата выдачи задания « 21 » апреля 2025 г.

Руководитель работы

  
подпись

П.А. Петров  
И.О. Фамилия

Руководитель ООП

  
подпись

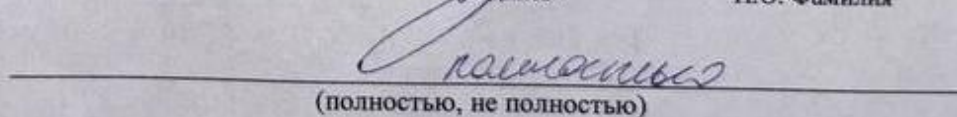
Р.В. Кононенко  
И.О. Фамилия

Задание принял к исполнению студент

  
подпись

В.А. Колупаев  
И.О. Фамилия

План выполнен

  
(полностью, не полностью)

Руководитель работы « 29 » мая 2025 г.

дата

  
подпись

П.А. Петров  
И.О. Фамилия

## Аннотация

**Тема:** Разработка методики применения ЭЦП для оптимизации работы со студентами в ИРНИТУ.

**Ключевые слова:** электронная цифровая подпись, информационная безопасность, документооборот, цифровизация, образовательные учреждения, программный прототип, ИРНИТУ.

**Выполнил:** Колупаев Вячеслав Андреевич, 09.03.01 «Информатика и вычислительная техника», АСУБ–21–1.

**Руководитель:** Петров Павел Александрович, доцент, кандидат экономических наук.

Дипломная работа посвящена разработке и внедрению методики применения простой электронной цифровой подписи (ЭЦП) для оптимизации административных процессов и повышения эффективности взаимодействия со студентами в ИРНИТУ. В рамках исследования выполнен анализ нормативной базы и особенностей электронного документооборота в образовательных организациях, а также изучены применимые виды ЭЦП. Разработан программный прототип, реализующий генерацию ключей, подписание документов, аутентификацию пользователей и контроль заявок в цифровом формате.

Система прошла успешное тестирование в реальных условиях, подтвердив соответствие поставленным требованиям, удобство использования и высокий уровень защищённости информации. На основе анализа существующих информационных систем вуза предложены рекомендации по дальнейшему внедрению, а также перспективы масштабирования и интеграции решения с государственными платформами. Практическая значимость работы заключается в возможности переноса методики на другие вузы и вкладе в цифровую трансформацию образовательного процесса.

**Объем пояснительной записки** – 96 страниц.

**Объем приложений** – 13 страниц.

**Количество рисунков** – 34.

**Количество таблиц** – 5.

**Количество использованных источников** – 31.

## Содержание

Введение.....	7
1 Анализ предметной области .....	9
1.1 Теоретические основы ЭЦП .....	9
1.1.1 Основные работы принципы криптографии .....	9
1.1.2 Криптографические алгоритмы и математическая база .....	10
1.1.3 Процедуры формирования и проверки цифровой подписи.....	11
1.1.4 Виды электронной цифровой подписи .....	13
1.1.5 Процедура получения электронной цифровой подписи .....	17
1.1.6 Проблемы и направления развития теоретических основ ЭЦП.....	18
1.2 Нормативная база .....	19
1.2.1 Обзор российского законодательства .....	19
1.2.2 Международные стандарты .....	20
1.2.3 Особенности правового статуса ЭЦП в образовательной сфере .....	21
1.3 Применение ЭЦП в системах управления.....	22
1.3.1 Анализ существующих решений и практических кейсов.....	22
1.3.2 Выявление преимуществ и ограничений применения ЭЦП.....	23
1.4 Особенности документооборота в образовательных учреждениях.....	24
1.4.1 Современные тенденции и проблемы .....	24
1.4.2 Роль цифровых технологий в оптимизации административных процессов .....	25
2 Теоретическое проектирование интеграции ЭЦП.....	27
2.1 Описание существующих процессов работы со студентами .....	27
2.2 Анализ существующих информационных систем ИРНИТУ .....	34
2.2.1 Moodle .....	34
2.2.2 1–С Университет .....	35
2.2.3 Тандем .....	36
2.2.4 Апекс .....	37
2.3 Перспективы внедрения ЭЦП в информационные системы ИРНИТУ .....	38
2.4 Выбор вида ЭЦП и обоснование .....	41
2.5 Подписываемые типы документов в ИРНИТУ .....	45
3 Методика применения ЭЦП.....	48
3.1 Формирование требований к системе .....	48
3.2 Техническая архитектура решения .....	50
3.2.1 Описание выбора стека технологий .....	50
3.2.2 Описание модулей Backend .....	50
3.2.3 Логика клиентского взаимодействия и поток данных .....	51
3.3 Модель безопасности и управления ключами .....	52
3.4 Методика сопровождения и контроля корректности работы компонентов ЭЦП.....	55
4 Разработка и тестирование прототипа .....	57
4.1 Техническое задание на разработку прототипа .....	57
4.2 Пользовательские интерфейсы .....	57

4.2.1 Экран «Вход в систему» .....	57
4.2.2 Экран «Панель управления» .....	61
4.2.3 Экран «Документы» .....	63
4.2.4 Экран «Подписи» .....	66
4.2.5 Экран «Создание ЭЦП» .....	68
4.3 Описание БД .....	70
4.4 Проведение тестирования .....	72
4.5 Анализ результатов тестирования .....	84
5 Рекомендации по дальнейшему внедрению ЭЦП .....	85
5.1 Предложения по интеграции приложения в процессы университета .....	85
5.2 Перспективы дальнейшего развития и масштабирования системы .....	87
5.3 Организационные и административные барьеры при внедрении ЭЦП .....	89
5.4 Рекомендации по нормативному обеспечению и внутренней политике ..	90
Заключение .....	93
Список использованных источников .....	94
Приложения .....	97
Приложение А .....	97
Приложение Б .....	98
Приложение В .....	100
Приложение Г .....	104
Приложение Д .....	108
Приложение Е .....	109

## Введение

В современном мире информационные технологии играют ключевую роль в развитии экономики, общества и образовательной сферы [1]. Они обеспечивают оперативное взаимодействие между государственными структурами, бизнесом, образовательными учреждениями и здравоохранением. При этом традиционные методы документооборота зачастую не отвечают современным требованиям по скорости, безопасности и удобству обмена информацией. Рост информационных угроз, необходимость защиты критической инфраструктуры [2] и стремление к технологической независимости требуют внедрения новых, более надёжных и эффективных цифровых решений.

Особое значение эти проблемы приобретают в образовательной сфере, где оперативное взаимодействие со студентами и сотрудниками является залогом качественного учебного процесса. В условиях высокой нагрузки на административные службы, длительных сроков оформления документов и риска ошибок ручного ввода традиционные методы работы с документацией оказываются неэффективными. Электронная цифровая подпись (ЭЦП) позволяет автоматизировать обмен информацией, резко сократить временные затраты и повысить прозрачность процессов, одновременно придавая документам необходимую юридическую значимость [3].

С одной стороны, внедрение электронной цифровой подписи служит ключевым фактором комплексной автоматизации делопроизводства: каждый этап жизненного цикла документа – от инициирования студентом заявления до издания приказа – переводится в сквозной электронный формат с регламентированными маршрутами согласования и автоматическим наложением подписи, которая фиксируется в системе. Это устраняет дублирование операций, минимизирует ошибки ручного ввода, сокращает трудоёмкость административного персонала и позволяет интегрировать процессы подписания с существующими информационными системами вуза, формируя единое пространство данных для аналитического учёта и управленческих решений. Одновременно ЭЦП повышает устойчивость инфраструктуры документооборота к несанкционированным изменениям, обеспечивая подтверждённое авторство и целостность документов в соответствии с действующими нормативно-правовыми требованиями.

Цель данной работы – разработка и апробация методики применения ЭЦП для повышения эффективности работы с документами и улучшения взаимодействия со студентами в ИРНИТУ. Для достижения этой цели необходимо решить следующие задачи:

- изучить теоретические основы ЭЦП, а также нормативно-правовую базу, регулирующую её применение в образовательной сфере [4] [5];
- провести анализ существующих процессов документооборота и взаимодействия со студентами в ИРНИТУ, выявить проблемные зоны и узкие места;

- определить ключевые этапы административных процессов, где внедрение ЭЦП способно обеспечить наибольшую оптимизацию;
- разработать модель интеграции ЭЦП в систему документооборота с учётом существующей инфраструктуры образовательного учреждения;
- создать прототип решения и провести экспериментальное тестирование для оценки эффективности внедрения разработанной методики;
- оценить достигнутые результаты с применением статистических методов и сформулировать рекомендации по дальнейшему внедрению.

Практическая значимость работы заключается в возможности использования разработанной методики для повышения оперативности и безопасности документооборота в ИРНИТУ. Реализация предложенного решения позволит снизить временные и ресурсные затраты, минимизировать риск ошибок при оформлении документов и создать основу для дальнейшей цифровизации образовательных процессов. Полученные результаты могут быть полезны не только для ИРНИТУ, но и для других образовательных учреждений, стремящихся к повышению эффективности управления информационными ресурсами и обеспечению безопасности обмена данными.

Структура работы включает:

- обзор теоретических основ и нормативно-правовой базы;
- анализ текущего состояния документооборота в ИРНИТУ;
- разработку методики применения ЭЦП;
- экспериментальное моделирование и прототипирование;
- оценку результатов эксперимента и обсуждение практических рекомендаций.

Таким образом, проведённое исследование направлено на решение актуальной задачи оптимизации работы с документами и взаимодействия со студентами посредством внедрения современных цифровых технологий, что в перспективе способствует повышению эффективности образовательного процесса и укреплению информационной безопасности.



# 1 Анализ предметной области

## 1.1 Теоретические основы ЭЦП

Электронная цифровая подпись (ЭЦП) – ключевой механизм полной автоматизации и безопасности электронного документооборота, позволяющий в онлайн-режиме подтвердить подлинность, целостность и авторство цифровых документов [6]. В основе ЭЦП лежит криптография с открытым ключом, при которой используются две взаимосвязанные криптографические сущности: закрытый ключ, сохраняемый в секрете владельцем, и открытый ключ, доступный для проверки подписи другими участниками системы. Такая схема обеспечивает не только аутентификацию подписанта, но и обнаружение любых изменений в содержимом документа, а также гарантирует невозможность отказа от совершённого действия [7]. (см. Рисунок 1.1)



Рисунок 1.1 – Схема работы криптографии с открытым ключом

### 1.1.1 Основные работы принципы криптографии

Концепция асимметричной криптографии, впервые предложенная У. Диффи и М. Хеллманом в 1976 г., основывается на использовании пары взаимосвязанных ключей: закрытого, который остаётся строго конфиденциальным у владельца, и открытого, предназначенного для свободного распространения [8]. Закрытым ключом формируется электронная подпись, а открытым – проверяется её подлинность. Благодаря одностороннему характеру этих операций (подпись возможна только с закрытым ключом, но проверка – с открытым) достигается высокий уровень доверия к авторству документа.

Перед подписью данные пропускают через криптографическую хэш-функцию, превращающую любой массив информации в компактный дайджест

фиксированной длины [9]. Подписывается именно этот дайджест, что значительно ускоряет расчёты и, главное, гарантирует жёсткую привязку подписи к конкретному содержимому: изменение хотя бы одного бита порождает иной хэш и делает подпись недействительной.

Чтобы связать открытый ключ с конкретным пользователем, в электронном документообороте применяют инфраструктуру открытых ключей (PKI) [10]. Удостоверяющий центр выпускает цифровой сертификат, в котором подтверждает принадлежность ключа своему владельцу, фиксирует период его действия и задаёт процедуры отзыва. Тем самым криптографические гарантии подписи дополняются юридически значимым механизмом идентификации.

В результате модель асимметричной подписи обеспечивает три фундаментальных свойства:

- аутентичность: подпись однозначно подтверждает, что документ создал владелец соответствующего закрытого ключа.

- целостность: любое изменение в документе приводит к несоответствию между исходной подписью и результатом проверки, что позволяет сразу обнаружить несанкционированные правки.

- неотрекаемость: из-за уникальности и контролируемости закрытого ключа подписавший не может впоследствии убедительно отрицать факт подписи.

Благодаря этим качествам асимметричная криптография стала фундаментом современного электронного документооборота, электронных торгов, банковских систем и нормативных требований (в России – 63–ФЗ «Об электронной подписи») [3]. Именно на перечисленных свойствах, а не на конкретных алгоритмах или математических построениях, держится практическая ценность электронной подписи, что оправдывает её широкое распространение в государственных, корпоративных и публичных информационных системах.

### **1.1.2 Криптографические алгоритмы и математическая база**

Формирование электронной цифровой подписи (ЭЦП) осуществляется с применением криптографических алгоритмов, основанных на решении вычислительно трудных математических задач. На практике наиболее широко применяются следующие алгоритмы шифрования:

1. RSA (Rivest–Shamir–Adleman) [11]. Алгоритм опирается на сложность факторизации большого составного числа  $n = p \cdot q$ , где  $p$  и  $q$  – крупные простые. Без знания этих множителей вычислить секретную экспоненту  $d$  (приватный ключ) по публичной  $e$  (публичный ключ) практически невозможно. При длине модуля 2048 – 4096 бит атака перебором остаётся нереальной в обозримой перспективе, а криптостойкость RSA определяется главным образом размером  $n$  и защитой закрытого ключа.

2. DSA (Digital Signature Algorithm) [6]. Безопасность базируется на задаче дискретного логарифмирования в мультипликативной группе простого порядка  $(Zp)^*$ ,  $p \approx 2 - 3$  кбит. Выбор надёжных параметров  $p$ ,  $q$  и  $g$  критичен: они

формируются так, чтобы исключить малые подпроцедуры и обеспечить отсутствие «слабых» подгрупп. Стандарт предусматривает длину  $p = 2048$  или  $3072$  бит, что по стойкости эквивалентно RSA–2048.

3. ECDSA (алгоритм цифровой подписи на эллиптических кривых) [12]. Использует арифметику точек эллиптической кривой над конечным полем  $F_p$  или  $F_{2^m}$ . Благодаря более высокой плотности «крипто–сложности» те же 128 бит стойкости достигаются при ключе  $\approx 256$  бит, что уменьшает объём передачи, ускоряет операции и делает ECDSA предпочтительным для мобильных и встроенных систем с ограниченными ресурсами.

4. ГОСТ Р 34.10–2012 [4]. Российский стандарт, внедряющий собственное семейство кривых и процедур генерации ключей; применяется в паре с хэш–функцией ГОСТ Р 34.11–2012. Поддерживает длины ключей 256 и 512 бит, предусматривает обязательную проверку параметров кривой и источника случайности, что удовлетворяет отечественным требованиям к классу криптографической стойкости КСЗ.

Вычислительная устойчивость указанных алгоритмов основывается на неразрешимости задач факторизации, дискретного логарифмирования и вычисления логарифмов на эллиптических кривых при современных классических вычислительных мощностях. Именно эта сложность делает практически невыполнимыми успешные криптографические атаки, включая атаки с выбором сообщений, подбор ложной подписи без знания закрытого ключа или восстановление самого ключа по набору валидных подписей. Таким образом, при корректном выборе параметров, достаточной длине ключей и надёжной реализации перечисленные схемы обеспечивают необходимый уровень безопасности для электронного документооборота.

### **1.1.3 Процедуры формирования и проверки цифровой подписи**

Формирование электронной цифровой подписи – это последовательный криптографический процесс, обеспечивающий неотъемлемые свойства аутентичности, целостности и неотрекаемости электронных документов. Каждое действие в этой цепочке опирается на стойкие математические преобразования и защищённую инфраструктуру открытых ключей, благодаря чему даже минимальные отклонения либо попытки подмены данных немедленно выявляются. Обобщённая схема, иллюстрирующая взаимосвязь основных операций и потоков данных, приведена на Рисунке 1.2.

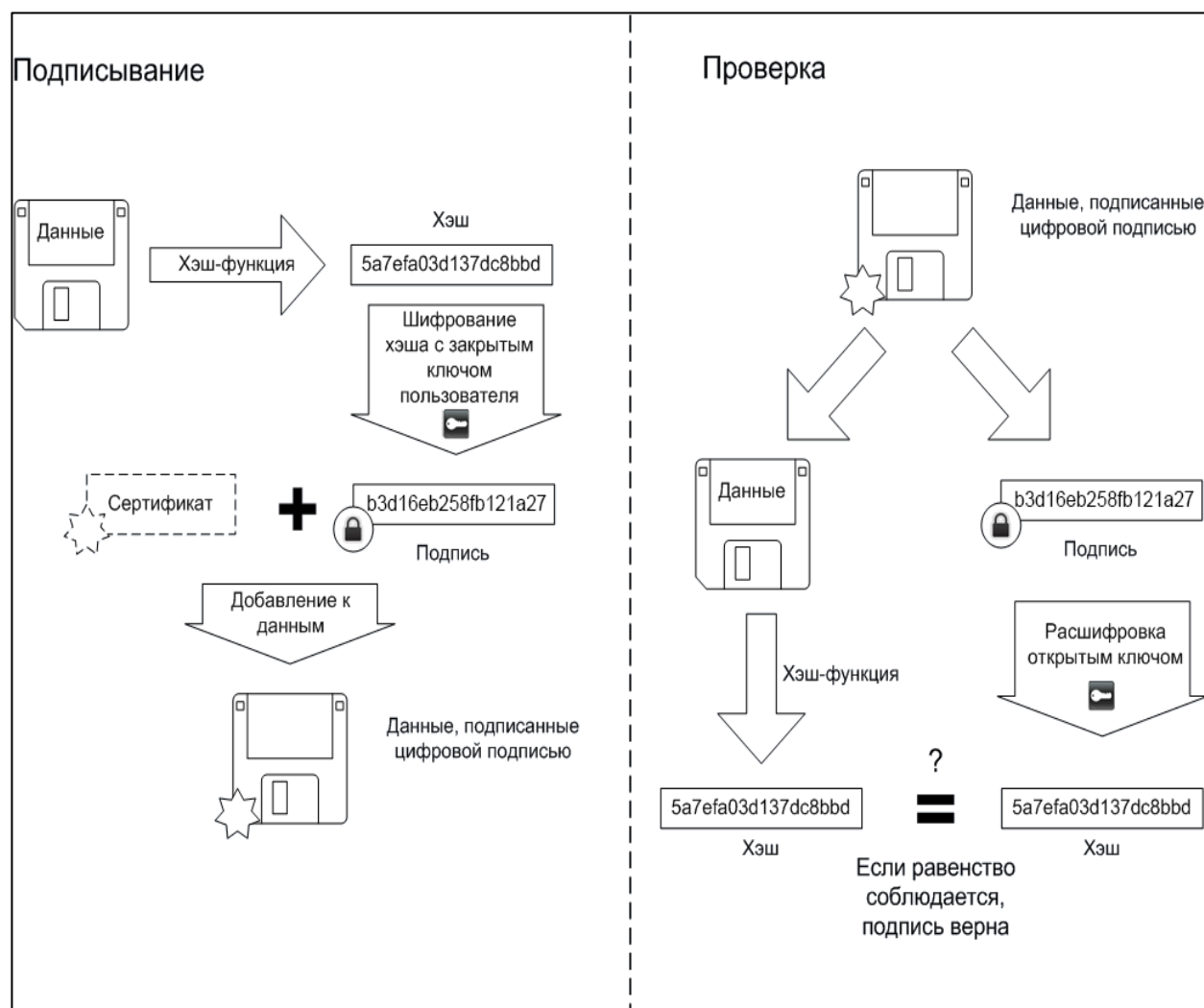


Рисунок 1.2 – Схема проверки цифровой подписи

1. Генерация ключевой пары. На основании надёжного источника энтропии криптографический алгоритм строит математически взаимосвязанные закрытый и открытый ключи. Закрытый ключ остаётся строго конфиденциальным: он хранится в защищённой среде (программный контейнер с шифрованием, аппаратный токен, HSM) и никогда не покидает контролируемого носителя. Открытый ключ, напротив, подлежит широкому распространению и, как правило, помещается в сертификат X.509, подписанный удостоверяющим центром, чтобы получатели могли убедиться в его подлинности и действительности. Корректная реализация этого этапа требует достаточной длины ключей (например, 2048–4096 бит для RSA или 256 бит для ECDSA/ГОСТ Р 34.10–2012 [13]) и гарантированной случайности при их генерации.

2. Формирование подписи. При подписании документа сначала вычисляется его криптографическая хэш–сумма. Для отечественного электронного документооборота наиболее часто применяется алгоритм ГОСТ Р 34.11–2012, для зарубежных компаний – семейства SHA–2/3 или BLAKE3 [14]. Хэш–функция преобразует произвольный объём данных в дайджест



фиксированной длины, исключая тем самым возможность коллизий на практике и надёжно закрепляет подпись к конкретному содержимому. Полученный дайджест шифруется закрытым ключом: фактически выполняется операция обратного шифрования, в результате которой образуется цифровая подпись – компактный блок, передаваемый вместе с документом либо отдельно в форме CMS/PKCS#7 – контейнера. Для защиты от атак с переиспользованием подписи большинство современных стандартов добавляет в алгоритм рандомизацию (параметр  $k$  в DSA/ECDSA) или схему PSS (Probabilistic Signature Scheme) в RSA.

3. Проверка подписи. Получатель снова вычисляет хэш–сумму полученного файла тем же алгоритмом и при помощи открытого ключа «расшифровывает» подпись, извлекая оригинальный дайджест отправителя. Совпадение двух хэшей свидетельствует о том, что документ не подвергался каким–либо изменениям после подписания, то есть сохранена целостность, и подпись могла быть создана только владельцем соответствующего закрытого ключа, что обеспечивает аутентичность и неотрекаемость. Если сертификат открытого ключа находится в статусе отзыва (CRL/OCSP) или истёк срок его действия, проверка завершается ошибкой, поскольку доверительная цепочка нарушена.

#### **1.1.4 Виды электронной цифровой подписи**

Помимо описанных выше процедур формирования и проверки цифровой подписи, в современной практике выделяют три основных вида ЭЦП, согласно федеральному закону от 6 апреля 2011 г. № 63–ФЗ «Об электронной подписи» [3], которые различаются по уровню криптографической защиты, юридической силе и области применения. Детальное рассмотрение этих видов позволяет глубже понять теоретические основы и практические аспекты применения ЭЦП:

##### **1. Простая электронная цифровая подпись (ПЭП).**

- обеспечивает автоматизированную базовую аутентификацию и контроль целостности электронного документа;

- применяется преимущественно для внутреннего документооборота, где риск несанкционированного доступа или модификации информации относительно невысок;

- теоретически, простая ЭЦП основывается на стандартных криптографических алгоритмах и используется для подтверждения авторства и неизменности документа, однако её юридическая значимость ограничена по сравнению с более защищёнными видами.

##### **2. Усиленная электронная цифровая подпись (неквалифицированная с дополнительными мерами защиты).**

- предусматривает использование дополнительных механизмов защиты, таких как аппаратное хранение ключей, двухфакторная аутентификация и применение расширенных криптографических алгоритмов;

- обеспечивает более высокий уровень надёжности, что позволяет применять её при обмене информацией с внешними контрагентами, где требуется повышенная степень доверия к подписанным документам;

– с теоретической точки зрения усиленная подпись является компромиссным решением, объединяющим оперативность простой ЭЦП и улучшенные меры безопасности, что повышает её практическую применимость в ряде бизнес–процессов.

### 3. Квалифицированная электронная цифровая подпись.

– обладает наивысшим уровнем криптографической защиты и юридической значимости, её правовой статус эквивалентен собственноручной подписи;

– применение квалифицированной ЭЦП строго регламентировано Федеральным законом [3] от 6 апреля 2011 г. № 63–ФЗ «Об электронной подписи» и национальными стандартами, такими как ГОСТ Р 34.10–2012, что обеспечивает её надежность и безопасность;

– теоретически, квалифицированная подпись использует передовые методы криптографии и проходит обязательную сертификацию в аккредитованных удостоверяющих центрах, что делает её оптимальной для оформления документов с высокой степенью ответственности (например, государственных контрактов, нотариальных актов и финансовых сделок).

Ниже приведена таблица, которая сравнивает основные виды электронной цифровой подписи (ЭЦП) – простую, усиленную и квалифицированную. Таблица 1.1 включает характеристики, такие как уровень криптографической защиты, юридическая значимость, затраты на оформление, операционная эффективность, типичные сферы применения, а также ключевые преимущества и недостатки.

Таблица демонстрирует, что простая ЭЦП обеспечивает полную автоматизацию внутренних процессов документооборота, сохраняя достаточный уровень защиты, высокую оперативность и экономическую эффективность, что делает её оптимальным выбором для образовательного учреждения, где основное внимание уделяется скорости обработки документов и снижению затрат.

Таблица 1.1 – Сравнительный анализ разных типов ЭЦП

Параметр	Простая ЭЦП	Усиленная ЭЦП (неквалифицированная с доп. защитой)	Квалифицированная ЭЦП
Уровень криптографической защиты	Базовый; использует стандартные алгоритмы	Повышенный за счёт дополнительных мер (аппаратное хранение ключей, двухфакторная аутентификация)	Наивысший; применяется сертифицированное оборудование и алгоритмы (ГОСТ, eIDAS)
Юридическая значимость	Подходит для внутренних документов с низким риском	Обладает большей юридической силой, чем простая, но ниже квалифицированной	Эквивалентна собственноручной подписи, применяется в документах высокой юридической важности
Затраты на оформление	Низкие; быстрая и простая процедура оформления	Средние; требуется применение дополнительных мер контроля	Высокие; сложная и длительная процедура идентификации в удостоверяющем центре
Операционная эффективность	Высокая; позволяет оперативно подписывать документы	Средняя; компромисс между оперативностью и повышенной безопасностью	Низкая для массовых внутренних процессов из-за длительности оформления

Продолжение таблицы 1.1

Параметр	Простая ЭЦП	Усиленная ЭЦП (неквалифицированная с доп. защитой)	Квалифицированная ЭЦП
Типичные сферы применения	Внутренний документооборот, обмен информацией между сотрудниками и студентами	Внешние транзакции, где требуется повышенная степень доверия, но не критичная юридическая сила	Госсектор, банковские и финансовые операции, нотариальные акты, государственные контракты
Преимущества	Быстрое оформление, низкие затраты, достаточная безопасность для внутренних процессов	Повышенная безопасность без полной процедуры квалификации, улучшенные меры защиты по сравнению с простой	Максимальная безопасность и юридическая сила, соответствие международным стандартам
Недостатки	Ограниченная юридическая значимость для внешних сделок; базовый уровень защиты может быть недостаточен для критичных операций	Требует дополнительных мер по организации, может быть медленнее простой ЭЦП	Высокие затраты, сложность и длительность процедуры оформления, не всегда оправданы для внутренних процессов



### **1.1.5 Процедура получения электронной цифровой подписи**

Процесс оформления ЭЦП – ключевого инструмента автоматизации документооборота – зависит от выбранного типа подписи, а также от требуемых уровней безопасности и юридической значимости подписываемых документов. В общих чертах процедура получения ЭЦП может быть условно разделена на два подхода:

#### **1. Для простой и усиленной ЭЦП.**

В большинстве случаев для получения простой и усиленной ЭЦП достаточно обратиться в специализированное учреждение или воспользоваться онлайн-сервисом, предоставляющим возможность оформления цифровой подписи. Такие сервисы, как правило, требуют минимальный пакет документов для идентификации заявителя, что позволяет оперативно оформить подпись. Например, в корпоративных системах внутреннего документооборота зачастую используется простая ЭЦП, которая оформляется через веб-порталы, где сотрудник проходит короткую процедуру аутентификации посредством электронной почты или СМС-кода. Усиленная ЭЦП может оформляться аналогичным образом, но с добавлением дополнительных проверок, направленных на повышение надежности, таких как использование аппаратных токенов или мобильных сертификатов.

#### **2. Для квалифицированной ЭЦП.**

Оформление квалифицированной ЭЦП представляет собой более строгую процедуру, поскольку данный тип подписи имеет наивысший уровень защиты и юридическую силу, эквивалентную собственноручной подписи. Получение квалифицированной ЭЦП требует прохождения процедуры идентификации в аккредитованном удостоверяющем центре. В этом случае заявителю необходимо предоставить полный пакет документов, подтверждающих его личность и правоспособность, а также пройти личное собеседование или идентификацию с использованием видеоконференцсвязи. Удоверяющий центр обязуется соблюдать государственные стандарты безопасности (например, ГОСТ Р 34.10–2012) и гарантировать конфиденциальное хранение закрытых ключей. На практике это означает, что при оформлении квалифицированной ЭЦП, например, для заключения государственных контрактов или подписания финансовых документов, процесс может занимать несколько дней, поскольку проводится тщательная проверка личности и документов.

Для ИРНИТУ в подавляющем большинстве сценариев (внутренние приказы, согласование учебных планов, подача заявок в деканат, приём лабораторных работ) достаточно простой ЭЦП. Университет уже располагает инфраструктурой аутентификации (портал Moodle) – её можно расширить следующим минимальным механизмом выдачи подписи:

#### **1. Однократная генерация ключевой пары**

– при первом входе в личный кабинет пользователь инициирует генератор, работающий в браузере Web-Crypto API [15];

- закрытый ключ сохраняется зашифрованным в профиле (например, в формате PKCS#12 [16]) и дополнительно выгружается в локальный защищённый контейнер (пароль + PIN-код).

## 2. Сертификат внутри университета

- открытый ключ автоматически публикуется в разделе с открытыми ключами с привязкой к табельному номеру и Ф. И. О.;

- подписи доверяются в пределах домена *istu.edu* на основании локального регламента (приказ ректора о признании ПЭП).

## 3. Подписание документов

- на сервисах (электронный деканат, система заявлений) появляется кнопка «Подписать», вызывающая JS-модуль, который совершает процедуру подписания (формирует хэш, шифрует при помощи закрытого ключа и делают соответствующую сигнатуру в БД)

## 4. Верификация

- любой сотрудник, получающий документ, проверяет подпись через внутренний API, сравнивая открытый ключ из LDAP и отправителя;

- при отзывах ключа система помещает запись в CRL-таблицу, что предотвращает использование скомпрометированного ПЭП.

### 1.1.6 Проблемы и направления развития теоретических основ ЭЦП

Несмотря на устоявшуюся практику применения электронной цифровой подписи (ЭЦП), её теоретические основы продолжают развиваться под влиянием новых вычислительных моделей, прикладных требований и нормативных актов. Современные исследования сосредоточены на следующих ключевых направлениях.

1. Криптографическая устойчивость к квантовым вычислениям. Разработка масштабируемых квантовых компьютеров делает уязвимыми алгоритмы, основанные на факторизации и вычислении дискретного логарифма. В ответ формируются пост-квантовые схемы подписи (Dilithium, Falcon, SPHINCS+) [17], а также гибридные композиции «классический + пост-квантовый» для обеспечения преемственности криптографической защиты и юридической значимости архивных документов.

2. Оптимизация вычислительных характеристик и энергоэффективность. Расширение числа устройств с ограниченными ресурсами (мобильные клиенты, сенсоры IoT) требует алгоритмов с минимальной задержкой и низким энергопотреблением. Исследуются ускоренные реализации эллиптических кривых, поточные методы хэширования и схемы многократных подписей (LMS, XMSS) [18], ориентированные на встроенные микроконтроллеры и беспроводное обновление программного обеспечения.

3. Интеграция в распределённые и облачные инфраструктуры. Децентрализованные реестры, многооблачные вычисления и сервис-ориентированные архитектуры предъявляют новые требования к управлению ключевой информацией. Актуальными становятся коллективные подписи с пороговым разделением секрета, защищённые аппаратные окружения (TEE) и

междоменные протоколы верификации, обеспечивающие сквозную доверительную цепочку в гетерогенных средах.

4. Противодействие атакам побочных каналов и формальная верификация. Актуальность приобретают методы конструирования алгоритмов с константным временем выполнения, маскировкой секретных переменных и строгим формальным доказательством корректности аппаратно–программных реализаций, что минимизирует риск утечки ключевой информации через электромагнитные, тепловые или временные побочные каналы.

5. Долговременная юридическая значимость электронных документов. Утрата криптостойкости хэш–функций и ключевых параметров требует механизмов продления доверия: архивации подписи вместе с последовательностью штампов времени.

## **1.2 Нормативная база**

### **1.2.1 Обзор российского законодательства**

Регулирование применения электронной цифровой подписи (ЭЦП) в Российской Федерации основывается на ряде нормативных актов, центральным из которых является Федеральный закон от 6 апреля 2011 г. № 63–ФЗ «Об электронной подписи». Этот закон закладывает правовые основы для формирования, проверки и использования ЭЦП, обеспечивая юридическую силу электронных документов наравне с документами, подписанными собственноручно.

Согласно закону, электронная подпись является средством аутентификации подписанта и подтверждения целостности передаваемого документа. Закон предусматривает несколько типов ЭЦП, отличающихся уровнем безопасности и степенью соответствия требованиям нормативных документов. Так, квалифицированная электронная подпись, формируемая с использованием средств, прошедших обязательную сертификацию, обладает повышенной юридической значимостью и признана равнозначной собственноручной подписи. Это положение способствует широкому внедрению ЭЦП не только в государственных учреждениях, но и в коммерческом и образовательном секторах, где требуется гарантированная защита информации и оптимизация работы документооборота.

Важным аспектом Федерального закона является установление порядка функционирования удостоверяющих центров, которые отвечают за выдачу сертификатов ключей и контроль за использованием электронных подписей. Закон определяет требования к таким центрам, регламентируя их деятельность, что обеспечивает высокий уровень доверия к процессу формирования и проверки ЭЦП. Таким образом, электронная подпись становится неотъемлемой частью информационной безопасности, способствуя защите данных от несанкционированного доступа и предотвращению возможности фальсификации документов.

Кроме того, законодательство Российской Федерации тесно связано с государственными стандартами, такими как ГОСТ Р 34.10–2012 «Электронная

цифровая подпись. Алгоритмы формирования и проверки ЭЦП». Данный стандарт устанавливает технические требования к криптографическим алгоритмам, используемым при создании и проверке ЭЦП, что позволяет обеспечить вычислительную устойчивость и надёжность цифровых подписей. Совокупность Федерального закона и соответствующих стандартов формирует комплексную правовую и техническую базу, необходимую для эффективного применения ЭЦП в различных сферах деятельности.

Практическое значение данных нормативных актов подтверждается их широким применением в государственном и частном секторах. Закон № 63–ФЗ существенно упростил процедуры электронного документооборота, позволяя организациям и учреждениям перейти на полностью электронные формы взаимодействия. Это особенно актуально в условиях современной цифровой трансформации, когда оперативное принятие решений и защита информации приобретают первостепенное значение.

Исследования в области применения ЭЦП, представленные, например, в работе Петрова В.И. (2016), демонстрируют, что внедрение цифровых подписей способствует снижению временных затрат на оформление документов, минимизации ошибок при их обработке и повышению уровня информационной безопасности. Практическая реализация данных положений в различных отраслях экономики свидетельствует о высокой эффективности и перспективности использования ЭЦП в условиях стремительного развития информационных технологий.

### **1.2.2 Международные стандарты**

На международном уровне применение цифровых подписей регулируется рядом нормативных документов, способствующих гармонизации требований в области электронной идентификации и доверительных услуг. Одним из ключевых документов является Регламент Европейского Союза № 910/2014 [19], известный как eIDAS. Этот регламент устанавливает единые правила для обеспечения безопасности электронных транзакций, юридической значимости электронных подписей и доверительных услуг на территории ЕС. В соответствии с eIDAS, квалифицированная электронная подпись обладает высокой юридической силой, что способствует взаимному признанию цифровых подписей между государствами–членами и упрощает трансграничное взаимодействие в электронном документообороте.

Помимо eIDAS, значительную роль играют международные стандарты, разработанные организациями ISO и IEC. В частности, стандарт ISO/IEC 14888 [20] определяет требования к криптографическим методам формирования и проверки цифровых подписей, что способствует единообразию технических характеристик и повышению уровня защиты данных на глобальном уровне. Применение этих стандартов позволяет обеспечить совместимость систем электронного документооборота в различных странах и секторах экономики, создавая условия для безопасного обмена информацией в условиях активной цифровизации.



Интеграция положений международных нормативных актов, таких как eIDAS и ISO/IEC 14888, обеспечивает организациям возможность выстраивать системы, отвечающие высоким требованиям безопасности и юридической значимости. Это особенно актуально для компаний и государственных учреждений, участвующих в международных проектах, где требуется соблюдение единого набора стандартов, способствующих повышению доверия к электронным транзакциям.

### **1.2.3 Особенности правового статуса ЭЦП в образовательной сфере**

Применение электронной цифровой подписи (ЭЦП) в образовательной сфере регулируется целым комплексом нормативных правовых актов Российской Федерации, формирующих единое правовое пространство для электронного документооборота учебных заведений. Федеральный закон от 6 апреля 2011 г. № 63–ФЗ «Об электронной подписи» задаёт базовые понятия, уровни подписи и условия её юридической значимости во всех секторах, включая образование: любой электронный документ, подписанный квалифицированной или согласованной сторонами усиленной подписью, приравнивается к бумажному экземпляру с собственноручной подписью.

Для образовательных организаций особенно важны следующие нормативные источники и положения:

1) федеральный закон от 27 июля 2006 г. № 149–ФЗ «Об информации, информационных технологиях и о защите информации» [21] определяет обязанности владельцев информационных систем – в том числе университетских порталов – по обеспечению конфиденциальности и целостности данных, подписываемых ЭЦП;

2) федеральный закон от 27 июля 2006 г. № 152–ФЗ «О персональных данных» [22] требует, чтобы ключевая инфраструктура и процессы подписания обеспечивали надлежащую защиту персональных данных студентов и сотрудников;

3) постановление Правительства РФ от 18 декабря 2012 г. № 1083 «О развитии электронной формы документооборота» [23] устанавливает общие правила интеграции государственных и ведомственных ИС, в том числе ведомственных систем высших учебных заведений, с федеральными сервисами электронного взаимодействия;

4) приказ Минобрнауки России от 23 июня 2014 г. № 706 [24] утвердил порядок ведения федерального реестра документов об образовании и/или квалификации, где сведения о дипломах формируются и подписываются квалифицированной ЭЦП уполномоченного лица вуза;

5) приказ Минцифры России от 1 октября 2022 г. № 279 [25] установил единые требования к форматам электронных документов в государственном секторе; университеты, подведомственные Минобрнауки, обязаны использовать форматы CAAdES–BES/LTA [26] и ГОСТ–алгоритмы при формировании документооборота с внешними контрагентами;

б) отечественные стандарты ГОСТ Р 34.10–2012 и ГОСТ Р 34.11–2012 регламентируют алгоритмы подписи и хэширования, а ГОСТ Р 7.0.97–2016 [27] определяет правила долговременного хранения электронных документов в архивах образовательных учреждений.

Институционализированное использование ЭЦП:

- сокращает средний срок согласования управленческих решений и повышает прозрачность академических процедур;
- гарантирует неизменность цифровых следов обучающихся (портфолио, результаты промежуточной аттестации);
- снижает расходы на печать и архивирование бумажных документов;
- повышает готовность университета к внедрению дистанционных и сетевых форм обучения, предусмотренных Федеральным государственным образовательным стандартом высшего образования.

### **1.3 Применение ЭЦП в системах управления**

#### **1.3.1 Анализ существующих решений и практических кейсов**

В образовательной среде внедрение электронной цифровой подписи выходит за рамки автоматизации отдельных заявлений или протоколов. Современные университетские информационные системы интегрируют ЭЦП во все ключевые контуры – от приёмной кампании до дипломного архива. Абитуриенты подают заявления на поступление через веб–портал, заверяя документы простой или усиленной подписью, после чего данные синхронизируются с федеральным сервисом приёма [28]. В процессе обучения преподаватели подписывают электронные ведомости, а студенты – индивидуальные учебные планы и заявления на пересдачу, что исключает дублирование бумажных журналов. Квалифицированная подпись ректора или ответственного секретаря заверяет итоговые протоколы государственной аттестационной комиссии и записи о выдаче дипломов, которые автоматически направляются в Федеральный реестр документов об образовании. Таким образом создаётся сквозная, проверяемая цепочка доверия, позволяющая университету оперативно проходить аккредитационные проверки и обеспечивать целостность академической истории каждого обучающегося.

В государственном управлении ЭЦП формирует основу межведомственного взаимодействия. Все ключевые сервисы «Госуслуг» – от получения паспортно-визовых документов до оформления социальных выплат – поддерживают квалифицированную подпись как средство юридически значимой идентификации граждан [29]. На уровне ведомственных систем используется сервис машиночитаемой доверенности [30], где цифровая подпись должностного лица автоматически делегирует полномочия сотруднику без оформления бланков строгой отчётности. Межведомственный электронный документооборот опирается на единые форматы CAdES и XAdES, что гарантирует проверяемость подписей независимо от конкретного программного обеспечения. В результате ускоряются сроки оказания государственных услуг,

сокращаются расходы на бумагу и курьерскую доставку, а риск коррупционных злоупотреблений снижается благодаря прозрачному журналированию каждой операции.

Коммерческий сектор задействует электронную подпись в разнообразных бизнес–процессах, где критично как время реакции, так и неоспоримость сделок. В банках мобильная подпись в приложении подтверждает кредиты, переводы и выпуск цифровых карт, при этом закрытый ключ хранится в защищённом окружении смартфона (Secure Element или TEE). В страховании полисы и акты урегулирования убытков формируются и визируются онлайн, что ускоряет выплату клиенту и минимизирует контакт с офисом. В сфере DevOps–практик IT–компаний подпись Docker–образов и исходного кода препятствует внедрению вредоносных компонентов в цепочку поставок ПО. На площадках электронной торговли юридические лица заключают контракты через квалифицированную подпись, обеспечивая доказательную базу при возникновении споров. Всё чаще ЭЦП встраивается в смарт–контракты на блокчейне, где цифровое удостоверение факта подписи автоматически инициирует выполнение договорных условий, например поставку товара или списание средств по достижении контрольных точек.

### **1.3.2 Выявление преимуществ и ограничений применения ЭЦП**

Преимущества применения ЭЦП:

- повышение безопасности: ЭЦП обеспечивает защиту документов от подделок и несанкционированных изменений, используя криптографические алгоритмы, что гарантирует их целостность и аутентичность;
- сокращение времени документооборота: автоматизация процессов подписания и проверки документов позволяет значительно уменьшить временные затраты, что особенно актуально для государственных и коммерческих структур;
- экономия ресурсов: переход на электронный документооборот снижает затраты на бумагу, печать и хранение документов, а также сокращает потребность в физическом пространстве для архивов;
- усиление доверия и прозрачности: юридическая значимость цифровой подписи повышает доверие участников процессов, способствуя прозрачности и контролю за выполнением обязательств;
- соответствие законодательству: использование ЭЦП соответствует требованиям Федерального закона № 63–ФЗ и государственных стандартов, что обеспечивает юридическую защиту электронных документов.

Ограничения и проблемы существующих систем:

- техническая сложность интеграции: внедрение ЭЦП требует существенных инвестиций в IT–инфраструктуру и интеграцию с существующими системами, что может быть сложным процессом для крупных организаций;

- необходимость обучения персонала: для эффективного использования новых технологий требуется обучение сотрудников, что сопровождается дополнительными временными и финансовыми затратами;
- проблемы совместимости: разнородность IT-систем и отсутствие единых стандартов в некоторых случаях затрудняют интеграцию ЭЦП в общие процессы документооборота;
- риски кибератак: несмотря на высокий уровень защиты, системы ЭЦП могут стать объектом кибератак, если не обеспечены должные меры по управлению ключами и защите данных;
- юридические и организационные вопросы: в некоторых случаях не до конца проработаны вопросы правового статуса электронных документов в специфических отраслях, что может создавать неопределенности при их использовании.

## **1.4 Особенности документооборота в образовательных учреждениях**

### **1.4.1 Современные тенденции и проблемы**

Цифровая трансформация высшего образования приобрела стратегическое значение: она обеспечивает непрерывность учебного процесса, повышает управляемость ресурсов и создаёт основу для персонализированного обучения. В настоящее время свыше 75 % российских вузов эксплуатируют системы электронного документооборота, интегрированные с образовательными порталами, виртуальными лабораториями и мобильными приложениями [31]. Использование единой цифровой среды сократило средние сроки обработки административных запросов примерно на треть [31], позволило оперативно формировать электронные ведомости, автоматически выгружать сведения о результатах обучения в государственные реестры и предоставлять студентам «одну точку входа» к расписанию, материалам курсов и сервисам поддержки.

Одновременно с ростом функциональности возрастает потребность в сквозных аналитических сервисах, которые на основе больших данных прогнозируют академическую успеваемость, оптимизируют использование аудиторий и формируют индивидуальные образовательные траектории. Ведущие университеты внедряют корпоративные хранилища данных и микросервисную архитектуру, связывающую информационную систему вуза с внешними государственными платформами (Госуслуги, ФИС ФРДО, СФЕРА) через стандартизированные API и механизмы электронной подписи, что повышает прозрачность и подотчётность управленческих решений.

Несмотря на перечисленные достижения, цифровизация сопровождается несколькими системными вызовами. Прежде всего это интеграция разнотипных, исторически сложившихся информационных систем факультетов, деканатов, библиотек и научных подразделений. Отсутствие унифицированного справочника данных и единого протокола аутентификации приводит к дублированию информации, временным задержкам при синхронизации и увеличивает риск человеческих ошибок. Значительная часть пользователей по-

прежнему сталкивается с неинтуитивными интерфейсами, что снижает эффективность новых сервисов и требует регулярного повышения цифровой грамотности студентов и сотрудников.

Дополнительные риски связаны с информационной безопасностью. Расширение периметра университетской сети за счёт удалённого доступа, IoT-устройств и облачных сервисов увеличивает угрозу несанкционированного доступа, фишинга и атак программ-вымогателей. Комплексная защита включает сегментирование сетевой инфраструктуры, применение многофакторной аутентификации, регулярный аудит уязвимостей и внедрение средств обнаружения вторжений. Особое внимание уделяется соблюдению требований законодательства о персональных данных, архивному хранению электронных документов и переходу на криптографические алгоритмы, устойчивые к квантовым атакам.

#### **1.4.2 Роль цифровых технологий в оптимизации административных процессов**

Интеграция электронных систем с базами данных позволяет вести подробный анализ статистических показателей и оперативно корректировать административные процессы. Такая интеграция способствует принятию обоснованных управленческих решений и улучшению контроля за выполнением внутренних процедур. Автоматизация рутинных операций, таких как обработка заявлений, согласование внутренних документов и регистрация информации, освобождает ресурсы, позволяя сотрудникам сосредоточиться на стратегических задачах, связанных с развитием образовательного процесса.

Дополнительно, современные технологии, такие как облачные сервисы и системы искусственного интеллекта, расширяют возможности оптимизации документооборота. Они не только повышают скорость обработки информации, но и усиливают защиту данных за счет использования передовых методов кибербезопасности. Таким образом, цифровизация становится ключевым фактором модернизации образовательных учреждений, способствуя их устойчивому развитию и повышению качества предоставляемых образовательных услуг.

Далее будет проанализировано, какие именно документы подписываются в образовательном процессе, каковы требования закона к подписи, и каким образом выбранные виды ЭЦП соответствуют этим требованиям. Этот анализ позволит определить, каким образом можно обеспечить законность и эффективность электронного документооборота в образовательных учреждениях.

С точки зрения законодательства, применение электронной цифровой подписи регулируется рядом нормативных актов, что определяет требования к документам, подписываемым с её помощью. В образовательном учреждении можно выделить следующие примеры:



Преподаватели подписывают документы, требующие высокой юридической точности и подтверждения подлинности. К таким документам относятся:

- протоколы заседаний кафедр и учебных комиссий;
- академические ведомости и отчёты;
- служебные записки и распоряжения.

Законодательство (Федеральный закон № 63–ФЗ) предъявляет строгие требования к документам, имеющим юридическую силу, поэтому для таких документов может применяться квалифицированная или усиленная ЭЦП, если требуется повышенная степень защиты. Однако для внутренних процессов зачастую достаточно простой ЭЦП, если уровень риска не высок.

Студенты формируют и подписывают все электронные заявления, справки и запросы прямо в своём личном кабинете. После наложения простой ЭЦП документ автоматически включается в сквозной маршрут согласования, фиксируется в цифровом архиве и доступен для поиска без единого печатного листа. Весь цикл – от подачи до завершения – работает полностью онлайн и автоматизирован, избавляя студентов и сотрудников от бумажной рутины.

Закон также требует, чтобы подписанные документы сохраняли целостность и авторство, что достигается даже при использовании простой ЭЦП, если документы предназначены для внутреннего использования и не предполагают внешнего юридического воздействия.

## **2 Теоретическое проектирование интеграции ЭЦП**

### **2.1 Описание существующих процессов работы со студентами**

В ИРНИТУ действуют четко структурированные бизнес–процессы, охватывающие весь цикл взаимодействия со студентами – от подачи документов абитуриентами до завершения образовательного процесса. Важной особенностью современного документооборота является использование ЭЦП, которая автоматизирует обработку документов, сокращает время всех операций и одновременно обеспечивает их юридическую значимость и безопасность. Для более наглядного анализа предлагается разработать карту бизнес–процессов, которая поможет визуализировать последовательность действий и выявить проблемные участки. При составлении карты следует включить следующие ключевые этапы:

#### **1. Регистрация и зачисление**

На данном этапе происходит следующее:

- подача документов абитуриентом. Абитуриенты подают свои документы через электронные формы, а также традиционными способами при необходимости. Применение ЭЦП уже на этом этапе может обеспечить первичную проверку подлинности представленных данных, а также ускорить процесс их обработки;

- проверка документов. Сотрудники приёмной комиссии проводят проверку комплектности и достоверности предоставленных документов. Ручная проверка часто сопряжена с рисками ошибок, но применение ЭЦП позволяет автоматически сверять данные, снижая вероятность ошибок и ускоряя процедуру проверки;

- приказ о зачислении студента. Этот этап является критически важным, так как официальное решение о зачислении оформляется в виде приказа, который должен иметь юридическую силу. ЭЦП автоматизирует выпуск и мгновенную передачу приказа, одновременно подтверждая его подлинность и юридическую силу без необходимости физического присутствия.

#### **2. Ведение академической документации**

После зачисления студентов формируется и ведется академическая документация, которая включает:

- оформление приказов по переводу студентов на новый курс или об отчислении. Документ подписывается ЭЦП, полностью автоматизируется его выпуск, а законность и прозрачность гарантируются встроенными криптографическими проверками;

- ведение электронных журналов и баз данных. Информация о посещаемости, успеваемости, оценках и расписании хранится в электронном виде. Применение ЭЦП в процессе обновления и утверждения данных позволяет обеспечить целостность информации, а также защитить ее от несанкционированных изменений.

#### **3. Обработка административных заявлений**

Административные обращения студентов, такие как запросы академических справок, заявления о переводе, восстановлении документов и другие, проходят следующий цикл:

- подача и регистрация заявления. Студенты подают свои обращения через электронные сервисы, где каждое заявление получает электронный идентификатор;

- проверка и согласование. Заявления проходят этап предварительной проверки, согласования с различными структурными подразделениями и, наконец, утверждаются. Применение ЭЦП автоматизирует весь процесс (регистрация, маршрутизация, контроль статусов) и одновременно придаёт решениям юридическую силу, сокращая время обработки и обеспечить надежное хранение в электронном виде;

- оперативная передача подписанных документов. Благодаря ЭЦП утвержденные документы передаются студентам по электронной почте или через внутренние системы, что обеспечивает полностью автоматическую доставку документов без бумажных копий.

#### 4. Обратная связь и информационный обмен

В ИРНИТУ используются современные информационные системы для обмена информацией между студентами, преподавателями и администрацией:

- система электронного образования (Moodle). Эта платформа используется для размещения учебных материалов, заданий и оценок. Интеграция ЭЦП в Moodle позволяет подписывать официальные уведомления, задания и отчеты, гарантируя их подлинность.

- кампус и внутренние порталы. С помощью этих систем вуз информирует студентов о новостях, расписаниях и изменениях в учебном процессе. Документы, распространяемые через такие платформы, могут быть подписаны ЭЦП, что делает их официальными и уменьшает риск недоразумений, связанных с неподтверждённой информацией.

Применение ЭЦП на всех этапах документооборота обеспечивает:

- полную автоматизацию проверки, маршрутизации и обмена данными;
- ускорение обработки благодаря минимизации ручных операций;
- сокращение времени оформления и передачи документов;
- надежное хранение электронных версий документов, что снижает зависимость от бумажных носителей;

- юридическую значимость и подтверждённое авторство документов.

Все перечисленные этапы документооборота в ИРНИТУ наглядно представлены на Рисунке 2.1. Данная схема демонстрирует ключевые шаги, начиная с подачи документов абитуриентом и завершая оформлением официальных приказов, заявлений и уведомлений. Именно на каждом из этих этапов внедрение ЭЦП автоматизирует каждый шаг (от подачи до приказа), тем самым ускоряет согласование и одновременно гарантирует юридическую силу создаваемых документов и обеспечивает высокий уровень информационной безопасности.

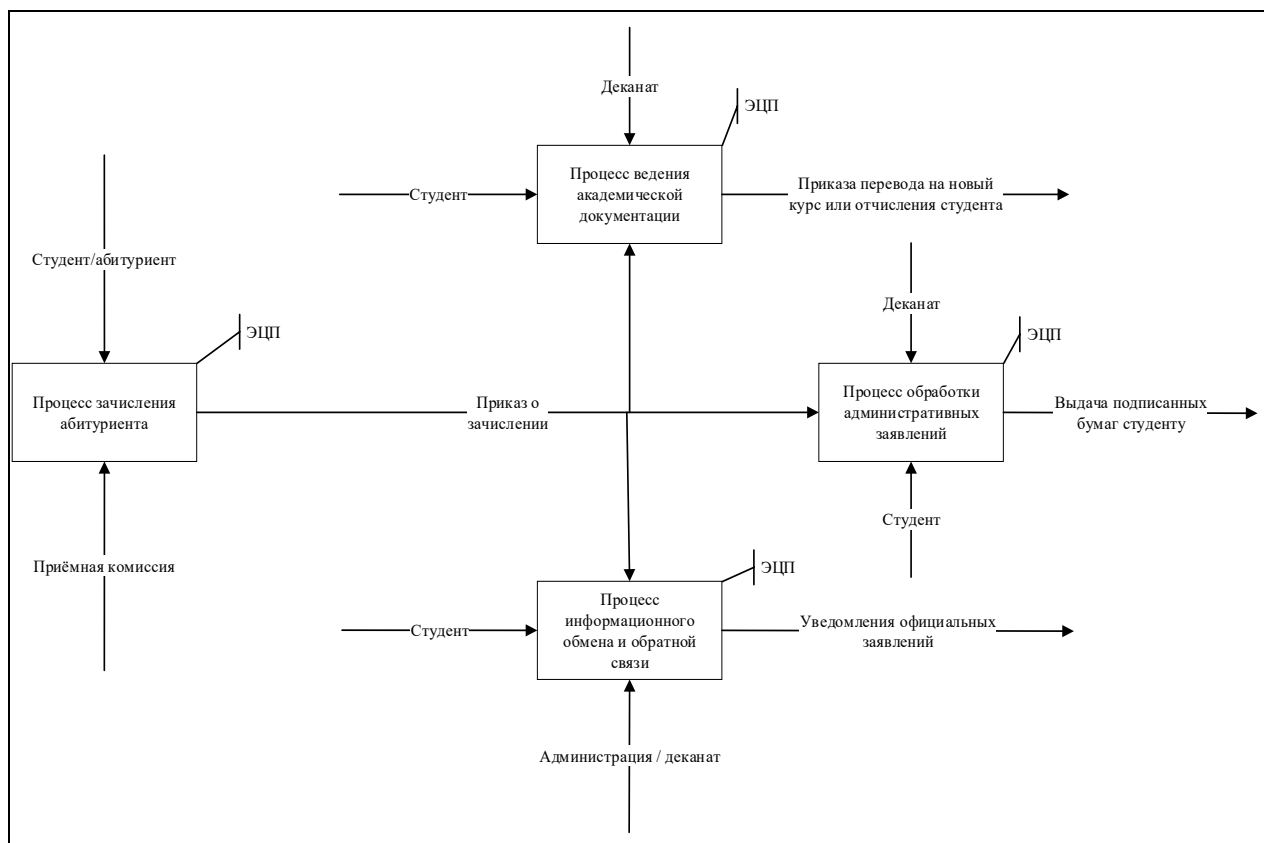


Рисунок 2.1 – Схема процессов работы со студентами

Анализ существующих бизнес-процессов в ИРНИТУ позволяет выявить ряд критических узких мест, способных существенно замедлять обработку документов и снижать общую эффективность взаимодействия со студентами. Основные проблемы можно условно разделить на несколько групп.

#### 1. Медлительность процессов.

Ручной контроль на этапах проверки документов и регистрации студентов приводит к значительному увеличению времени обработки заявлений. В частности, необходимость физической проверки комплектности документов и ручной регистрации в информационных системах создаёт задержки, которые в сумме могут привести к замедлению процесса зачисления. По данным отчёта Министерства образования РФ, автоматизация этих процессов могла бы сократить время обработки, что существенно повысило бы оперативность работы учреждения.

#### 2. Ошибки при вводе данных.

Ручной ввод информации в различных модулях и системах приводит к дублированию данных, их несогласованности и ошибкам. Такие ошибки могут проявляться в неверном распределении оценок, неправильном формировании расписания или несвоевременной регистрации студентов. Это негативно сказывается на качестве ведения академической документации и вызывает дополнительные административные издержки при исправлении допущенных ошибок. Применение электронной цифровой подписи (ЭЦП) на этапе

подписания официальных документов способно снизить вероятность ошибок, обеспечивая автоматическую проверку целостности данных.

### 3. Фрагментация информационных систем.

Использование разрозненных платформ для регистрации, ведения документации и обработки административных заявлений создаёт значительные трудности в синхронизации информации. Отсутствие интеграции между модулями ERP–систем, специализированными образовательными платформами и системами обратной связи приводит к тому, что данные распределяются по разным источникам. Это не только замедляет обмен информацией, но и повышает риск возникновения ошибок из-за несовместимости систем. Внедрение единой платформы с использованием ЭЦП может способствовать унификации данных и улучшению процессов контроля.

### 4. Отсутствие единой системы контроля.

Низкий уровень автоматизации контроля за обработкой заявлений и регистрации данных ведёт к отсутствию прозрачности на всех этапах документооборота. Недостаточная интеграция систем контроля затрудняет своевременное выявление и корректировку ошибок, что приводит к задержкам в принятии управленческих решений. Это, в свою очередь, может негативно сказываться на оперативном реагировании администрации на изменения в учебном процессе. Применение ЭЦП позволяет обеспечить полную автоматизацию и прозрачное ведение истории обработки документов, одновременно придавая им юридическую значимость, что повышает надежность контроля.

Для наглядного представления выявленных проблем и возможностей оптимизации бизнес–процессов в ИРНИТУ были разработаны две диаграммы – «as is» и «to be». Эти схемы позволяют визуализировать текущую ситуацию и предложить конкретные пути совершенствования документооборота с использованием ЭЦП.

На рисунке 2.2 представлена текущая схема бизнес–процессов, охватывающая все этапы взаимодействия со студентами – от подачи документов до оформления официальных приказов и заявлений. Диаграмма наглядно демонстрирует существующие узкие места: медлительность процессов из-за ручного контроля, ошибки при вводе данных, фрагментацию информационных систем и отсутствие единой системы контроля. Эти факторы существенно замедляют обработку документов и снижают общую эффективность взаимодействия.

Рисунок 2.3 иллюстрирует предлагаемую модель оптимизации бизнес–процессов на основе интеграции электронной цифровой подписи и объединения разрозненных информационных систем в единую платформу.



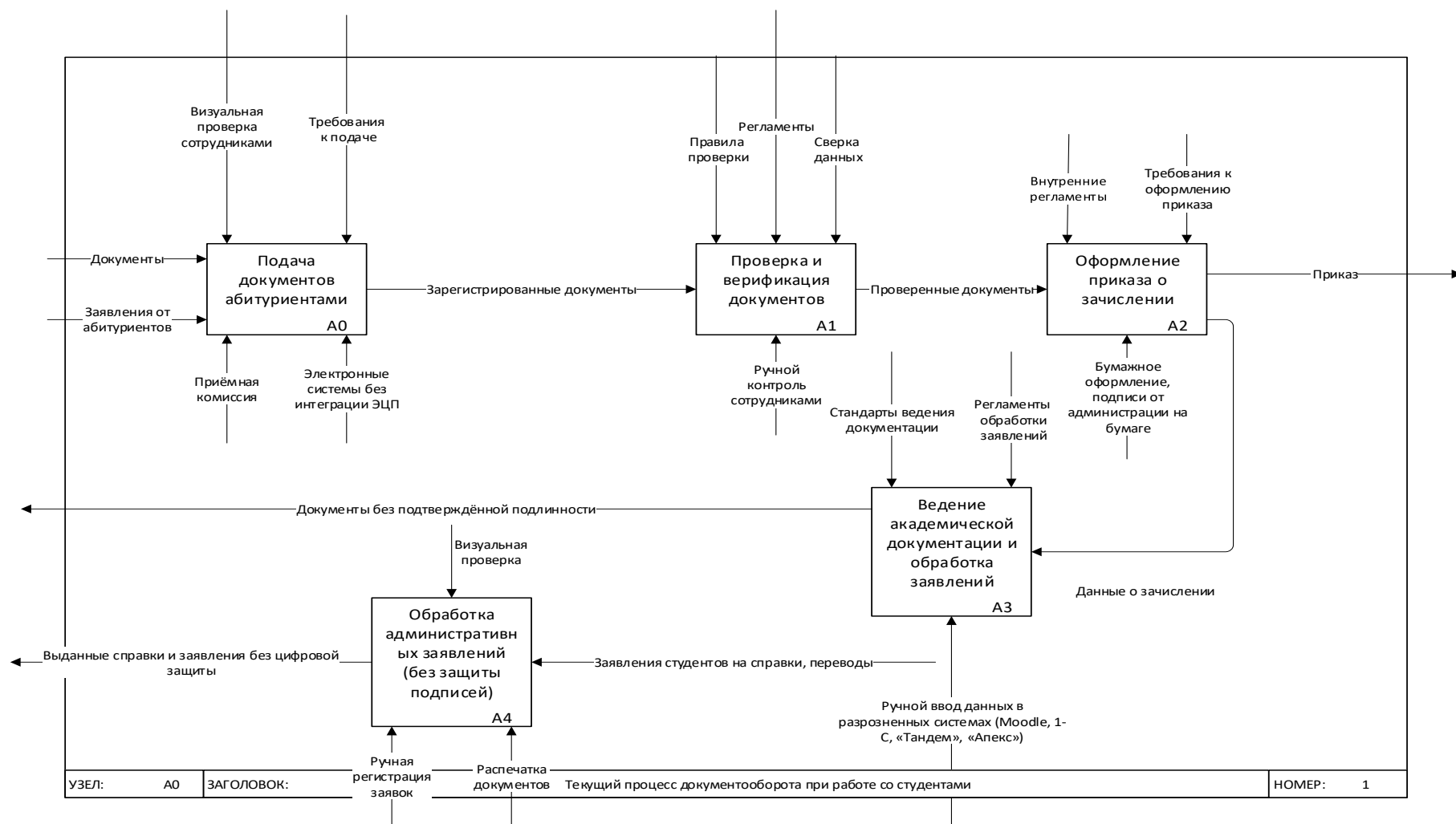


Рисунок 2.2 – Диаграмма декомпозиции текущего процесса работы со студентами.

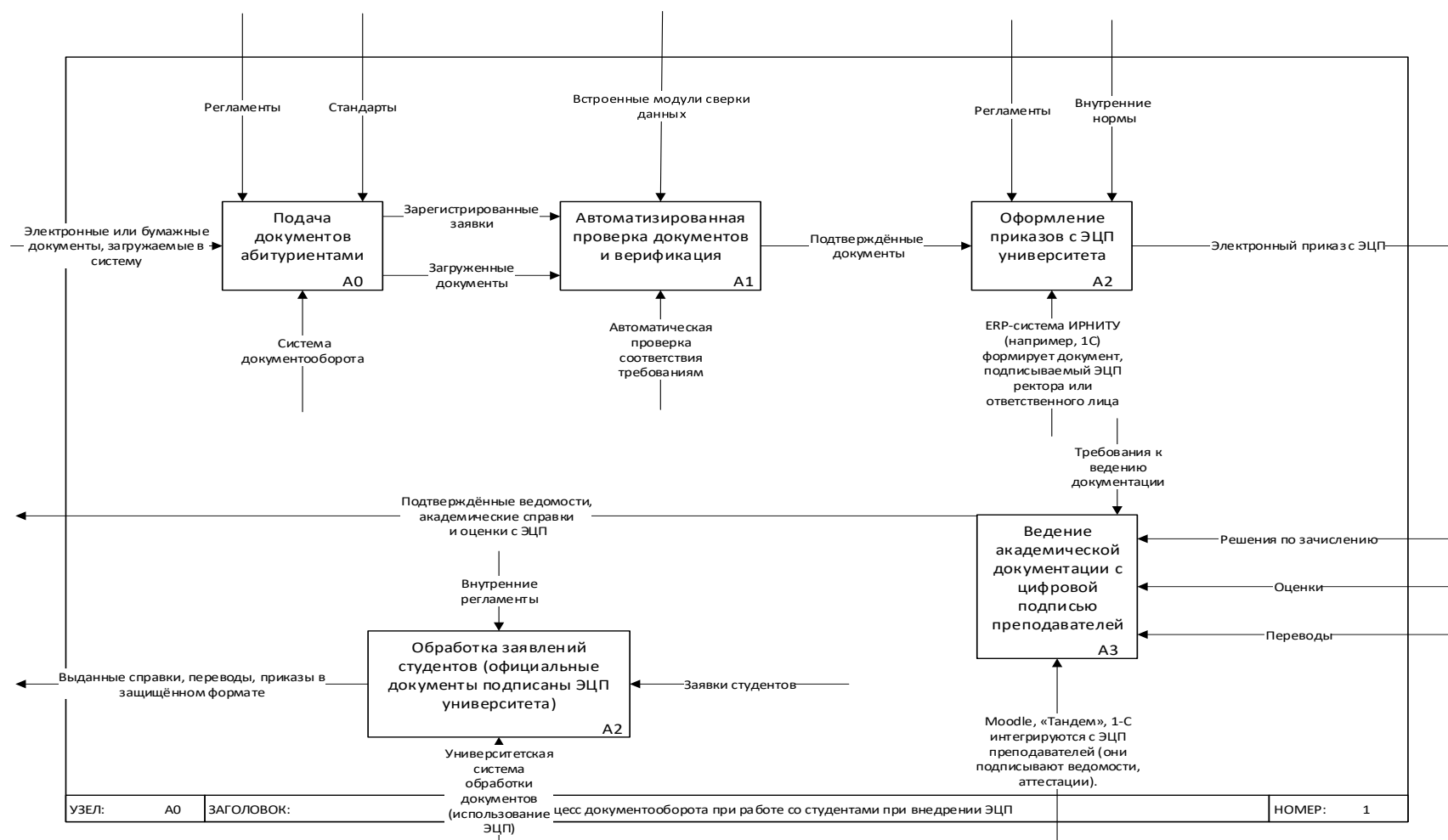


Рисунок 2.3 – Диаграмма декомпозиции процесса работы со студентами в условиях внедрения ЭЦП.

Сравнительный анализ диаграмм 2.2 и 2.3 показывает, что ключевым преобразованием становится перевод всех стадий жизненного цикла документа в полностью электронный контур с автоматическим управлением маршрутами.

В текущей модели «as-is» инициация заявлений осуществляется через разнородные каналы, а проверка комплектности и подлинности данных выполняется последовательно разными подразделениями, что порождает временные задержки и повышает риск ошибок ручного ввода. Формирование распорядительных документов связано с многоступенчатой процедурой печати, собственноручного визирования, сканирования и повторной загрузки, поэтому доставка приказов и ведомостей адресатам занимает много времени. Отсутствие единой точки интеграции между Moodle и 1С ведёт к рассогласованию версий данных.

Ключевые недостатки модели «as-is»:

- множественные каналы подачи и ручная регистрация;
- последовательная ручная проверка комплектности;
- бумажная подпись → скан → повторная загрузка;
- разрозненные базы данных;
- контроль статусов через телефон/e-mail;
- смешанный тип архива.

Модель «to-be» устраняет указанные узкие места за счёт единого портала подачи заявлений, где каждому документу немедленно присваивается уникальный цифровой идентификатор и накладывается простая ЭЦП студента. Валидация реквизитов, согласование маршрутов и выпуск приказов переводятся в автоматический режим. Сформированный в ERP-ядре приказ передаётся получателям мгновенно через push-уведомления в личных кабинетах, сохраняя юридическую силу на всём пути следования. Интеграция информационных систем выполняется через РКИ-шлюз, благодаря чему ведомости, подписанные преподавателем, автоматически реплицируются в Moodle и 1С. Централизованные дашборды в реальном времени отображают нагрузку на каждом этапе и позволяют оперативно устранять возникшие задержки, а электронный архив с контейнерами обеспечивает долговременное хранение и проверяемость подписей. Таким образом, переход от «as-is» к «to-be» превращает документооборот ИРНИТУ в сквозной, управляемый и прозрачный процесс.

Ключевые преимущества модели «to-be»:

- единый личный кабинет и уникальный ID заявления;
- автоматическая валидация и параллельное согласование;
- цифровое формирование приказов и подпись простой ЭЦП;
- сквозная синхронизация;
- онлайн-дашборды и журналирование операций;
- централизованный электронный архив.

## **2.2 Анализ существующих информационных систем ИРНИТУ**

### **2.2.1 Moodle**

Система Moodle представляет собой свободно распространяемую веб-ориентированную платформу для дистанционного и смешанного обучения, обеспечивающую надёжность и масштабируемость за счёт классического LAMP/LEMP-стека (Linux, Apache/Nginx, MySQL/MariaDB, PHP). Её модульная архитектура основывается на едином API ядра, отвечающем за аутентификацию, навигацию, локализацию и управление данными, тогда как весь остальной функционал – управление курсами, заданиями, оценками, форумами и обратной связью – реализован в виде подключаемых плагинов. Подобный подход позволяет расширять возможности системы без изменения исходного кода ядра и сохранять при этом полную совместимость с обновлениями Moodle. Начиная с версии 3.7, к традиционному REST-webservice API добавлена экспериментальная поддержка GraphQL, что обеспечивает более тонкую типизацию запросов и оптимизацию клиентских приложений, включая мобильные клиенты и внешние учебные сервисы.

Для управления доступом и разграничения прав Moodle предоставляет богатый набор механизмов аутентификации: от классического логин/пароль и LDAP до современных корпоративных схем на базе SAML (Shibboleth), OAuth2 (включая интеграцию с ЕСИА) и OpenID Connect. Это позволяет университетской ИТ-инфраструктуре использовать единый корпоративный каталог пользователей и автоматически передавать в систему информацию о ролях и привилегиях, что критически важно для контроля процедур подписания официальных документов.

Интеграция электронной цифровой подписи выполняется двумя основными путями. Первый – использование клиентского плагина mod\_sign, где на стороне пользователя через WebCrypto API генерируется ключевая пара (RSA-2048 или ГОСТ Р 34.10-2012), закрытый ключ шифруется алгоритмом AES-GCM и хранится на сервере в зашифрованном виде, а при подписании формируется контейнер CAdES-BES с дайджестом документа и метаданными подписанта. Второй путь – подключение внешнего PKI-шлюза по протоколам JSON-RPC или REST: Moodle формирует структуру запроса на подпись, передаёт её в аккредитованный ФСТЭК и Минкомсвязи криптосервис (например, «Крипто-Про Gateway»), где документ подписывается квалифицированным сертификатом и снабжается штампом времени (TSA), после чего готовый CAdES-BES пакет возвращается в систему.

Практика внедрения в российских вузах демонстрирует, что описанные решения позволяют полностью автоматизировать подписание итоговых протоколов дистанционных экзаменов, отказаться от бумажного документооборота и сократить время обработки результатов более чем на 40 %. Подписанные ведомости автоматически архивируются в формате CAdES-BES с учётом требований ГОСТ Р 7.0.97-2016 по долговременному хранению, что обеспечивает юридическую значимость электронных документов на

долгосрочную перспективу. Централизованное управление ключами и уведомлениями через API Moodle существенно снижает нагрузку на ИТ-службы при одновременном увеличении числа пользователей.

С точки зрения архитектуры хранения и безопасности, контейнеры CAdES-BES располагаются в файловой системе или в объектном хранилище с сопутствующим описанием в базе данных, что позволяет интегрировать их в общую систему бэкапа и репликации. Логирование всех операций подписи и проверки строится на основе стандартных средств Moodle и дополняется записями криптосервиса, что удовлетворяет требованиям аудита и информационной безопасности.

Таким образом, благодаря открытой модульной конструкции, поддержке современных веб-стандартов и возможности гибкой интеграции как встроенных плагинов, так и внешних криптосервисов, Moodle представляет собой эффективную и юридически значимую платформу для распространения образовательных материалов и проведения сертифицированных дистанционных процедур с применением электронной цифровой подписи.

### **2.2.2 1–С Университет**

ERP-система «1С:Университет» представляет собой комплексную платформу, разработанную специально для вузовских нужд и обеспечивающую сквозную автоматизацию административных, учебных и финансово-кадровых процессов. В её основе лежит архитектура «тонкий клиент – сервер 1С:Предприятие 8.3», где серверная часть отвечает за хранение и обработку данных, а клиентская – за представление интерфейсов и бизнес-логики пользователям. Конфигурация «1С:Университет» включает обширный набор подсистем: модуль управления учебным планированием, позволяющий формировать и корректировать учебные нагрузки, распределять преподавательские часы и рассчитывать показатели успеваемости; подсистему кадрового учёта, обеспечивающую ведение личных дел сотрудников, учёт квалификации и аттестации; финансово-бюджетный блок для планирования и контроля исполнения смет, расчёта заработной платы и грантовых поступлений; а также модуль документооборота, централизованно регистрирующий все приказы, распорядительные и информационные документы.

Структура базы данных построена на типовом механизме метаданных 1С, что позволяет гибко расширять и дорабатывать конфигурацию без изменения прикладного кода платформы. Все объекты (справочники, документы, отчёты, обработки) описаны в единой системе метаданных, а механизмы механизмов расширений (расширения конфигураций) дают возможность внедрять дополнительные подсистемы и интегрироваться с внешними сервисами без риска «ломать» стандартный функционал при обновлении.

Для обеспечения информационной безопасности и соответствия законодательным требованиям в конфигурации реализована роль-ориентированная модель доступа: права на создание, редактирование, просмотр и утверждение документов разграничиваются в зависимости от должности и



статуса пользователя (преподаватель, декан, секретарь приёмной комиссии, бухгалтер и др.). Все действия фиксируются в журнале регистрации, что позволяет проводить полный аудит изменений и отслеживать ход исполнения бизнес-процессов.

Криптографическая подсистема «Криптография 1С» интегрирована в систему на уровне платформы и обеспечивает работу с сертификатами по ГОСТ–стандартам: хранение закрытых ключей, формирование и проверку ЭЦП документов, работу с аппаратными токенами и смарт-картами. Для перехода от простой подписи к квалифицированной необходимо установить на сервер и рабочие места СКЗИ (например, Crypto-PRO CSP или модуль «Выполнение криптографических операций ГОСТ»), а также обновить конфигурацию до версии, прошедшей сертификацию ФСТЭК. В результате «1С:Университет» получает возможность автоматически визировать приказы, трудовые договора, акты платных услуг и иные официальные бумаги, исключив техпроцессы печати и ручного обмена бумажными носителями.

Отдельного внимания заслуживают средства аналитики и отчётности: типовая конфигурация включает более сотни шаблонов управленческих и регламентированных отчётов, а также механизм динамических отчётов на основе универсального конструктора, позволяющий на лету формировать произвольные сводные таблицы, графики и дашборды.

Практика внедрения «1С:Университет» в российских образовательных организациях показывает, что единая информационная среда повышает прозрачность процессов приёма, учёта и обучения, позволяет сократить трудозатраты на рутинные операции до 60 % и обеспечивает сквозное управление данными на всех уровнях – от приёмной комиссии до ректората. При этом модульная природа решения и наличие развитого сообщества разработчиков гарантируют своевременную адаптацию платформы под новые нормативные требования и педагогические методики.

### **2.2.3 Тандем**

Система «Тандем» представляет собой монолитное веб-приложение, разработанное на базе .NET Framework и предназначенное для комплексного управления учебным процессом в вузе. Её функциональный профиль охватывает регистрацию студентов, ведение академической документации, учёт посещаемости, автоматизированную оценку успеваемости и обработку административных заявлений, что позволяет существенно снизить затраты времени на рутинные операции. Архитектурно «Тандем» построен по классической схеме «клиент – сервер», где вся бизнес-логика и интерфейсы реализованы в едином кодовом пространстве, а данные хранятся в реляционной СУБД (Microsoft SQL Server).

Для модернизации системы и полной поддержки современных протоколов электронной цифровой подписи (ЭЦП) целесообразна миграция на .NET 6+ с переводом криптографических операций в пространство имён System.Security.Cryptography под управлением провайдеров CNG. Такой переход

позволит использовать в «Тандеме» алгоритмы ГОСТ Р 34.10-2012 и ГОСТ Р 34.11-2012 на уровне стандартного API платформы без привлечения сторонних обёрток и обеспечит прямую интеграцию с аппаратными токенами и смарт-картами. После обновления конфигурации разработчикам достаточно будет вызывать методы подписи и верификации из единой библиотеки, что упростит поддержку и повысит надёжность криптографических процедур.

Интеграция ЭЦП в «Тандеме» открывает возможность официального подтверждения подлинности всех генерируемых документов – ведомостей посещаемости, отчётов об успеваемости, приказов и заявлений – с двусторонней проверкой целостности данных на стороне преподавателя и деканата. Благодаря регистрации каждого события подписи в общем журнале аудита обеспечивается прозрачность и непрерывность контроля всех операций. Внедрение квалифицированной подписи в рамках обновлённой платформы позволит вузу перейти к безбумажному документообороту, снизить операционные риски и повысить юридическую значимость электронных форм документов без необходимости существенной доработки существующего бизнес-языка приложения.

#### **2.2.4 Апекс**

Система «Апекс» представляет собой трёхзвенную информационно-аналитическую платформу, обеспечивающую сквозной цикл обработки и контроля информационных потоков в вузе: OLAP-хранилище для многомерного агрегирования статистических данных, ETL-сервер для извлечения, трансформации и загрузки источников различной структуры и веб-клиент на базе современных JavaScript-фреймворков (Angular или React) для формирования служебных записок, отчётов и аналитических документов. Благодаря централизованному хранению в OLAP-слое «Апекс» позволяет быстро формировать сводные таблицы и графики на основе сотен первичных показателей, одновременно регистрируя все события обработки в журнале аудита для обеспечения прозрачного контроля и соответствия внутренним регламентам.

Интеграция электронной цифровой подписи в «Апекс» реализуется расширением клиентского слоя: в веб-интерфейс внедряется компонент, отвечающий за генерацию, наложение и верификацию ЭЦП с использованием стандартного API WebCrypto или через вынесенный подпись-сервис. Для гарантии юридической значимости документов и соблюдения требований ГОСТ Р 34.10-2012 и ГОСТ Р 34.11-2012 необходимо также подключение службы штампов времени (TSA) и обеспечение взаимодействия с аккредитованными СКЗИ.

В результате модернизации подписанные отчёты экспортируются в формате PDF/A-3 с вложенным CAdES-контейнером, что полностью соответствует требованиям ГОСТ Р 7.0.97-2016 к долговременному архивированию электронных документов. Одновременно подписанные формы – ведомости посещаемости, аналитические справки и служебные записки –

проходят двустороннюю проверку целостности: на стороне преподавателя и в деканате, причём все операции фиксируются в едином реестре ELT-сервера. Несмотря на высокую аналитическую мощность, «Апекс» требует периодической доработки модулей безопасности и криптографии для поддержки новых версий TLS, современных алгоритмов ГОСТ и расширенных протоколов аутентификации, что является обязательным условием сохранения совместимости с внешними платформами и актуализации криптографических практик.

### **2.3 Перспективы внедрения ЭЦП в информационные системы ИРНИТУ**

Проведённый сравнительный анализ показал, что платформы Moodle, 1С:«Университет», «Тандем» и «Апекс» обладают разным уровнем готовности к внедрению ЭЦП и требуют различного объёма доработок. Основные выводы таковы:

1. Moodle из коробки поддерживает подключение внешнего PKI-шлюза и ГОСТ-провайдеров через REST/GraphQL-API, а интерфейс легко расширяется JavaScript-виджетами подписи; управление ключами может вестись централизованно через OAuth2/OpenID Connect.

2. 1С:«Университет» имеет встроенный модуль «Криптография 1С» для ГОСТ-сертификатов, но для квалифицированной подписи требуется обновление конфигурации и развёртывание СКЗИ на всех узлах; консоль 1С позволяет централизованно настраивать провайдеры и аудит.

3. «Тандем» как монолит на .NET Framework требует миграции на .NET 6+ и перехода на CNG-провайдеры ГОСТ в System.Security.Cryptography; после этого интерфейс и ключевая инфраструктура Windows обеспечат стандартизированные операции подписи.

4. «Апекс» в трёхзвенной архитектуре OLAP/ETL/web-клиент нуждается в доработке фронтенда (Angular/React) для внедрения WebCrypto или вызова внешнего сервиса подписи; централизованный аудит реализуется через ETL-слой, но обновление криптомодулей затрагивает одновременно клиент и сервер.

5. Внедрение единого PKI-шлюза и службы штампов времени (TSA) во всех системах позволяет унифицировать процедуры подписи и верификации (форматы CAdES-BES, PDF/A-3), снизить операционные риски и упростить ротацию и отзыв сертификатов, сохранив сквозной аудит и юридическую значимость электронных документов.

Внедрение электронной цифровой подписи в перечисленных информационных системах создаёт условия для формирования единого безопасного и юридически значимого пространства электронного документооборота в вузе. Применение ЭЦП во всех этих системах способствует:

- повышению юридической значимости официальных документов, таких как приказы о зачислении, переводы и административные заявления;
- автоматизации процессов проверки и обмена информацией, что сокращает время обработки и повышает точность данных;

- сокращению временных затрат на оформление и передачу документов, поскольку подписанные электронные документы могут быть оперативно доставлены через интегрированные информационные системы;

- надежному хранению данных в электронном виде, что уменьшает зависимость от бумажных носителей и облегчает процессы архивирования и контроля.

В итоге комплекс мер по внедрению ЭЦП обеспечивает:

- унификацию процедур подписания и верификации во всех подсистемах;
- прозрачность и аудит всех операций благодаря централизованным журналам;

- соответствие нормативным требованиям ФЗ-63 и ГОСТ Р 7.0.97-2016;

- снижение операционных рисков и затрат на бумажный документооборот.

Для более системного анализа автоматизированных систем, используемых в ИРНИТУ, ниже представлена Таблица 2.1, отражающая ключевые характеристики: функциональное назначение, возможности интеграции электронной цифровой подписи (ЭЦП) и основные ограничения, которые необходимо учитывать при модернизации ИТ-инфраструктуры. Эта таблица служит важным инструментом для сравнения различных информационных систем, позволяя выявить сильные стороны каждой из них и определить, какие доработки потребуются для полноценного внедрения ЭЦП.

Таблица 2.1 – Сводная характеристика автоматизированных систем ИРНИТУ

<b>Система</b>	<b>Функциональное назначение</b>	<b>Возможности интеграции ЭЦП</b>	<b>Ограничения интеграции</b>
Moodle	Платформа дистанционного обучения, организация курсов, управление заданиями и оценками.	Поддержка модулей для подписания уведомлений и отчетов. Плагиновая архитектура облегчает добавление функционала.	Не все версии Moodle поддерживают API ЭЦП; требуется настройка плагинов и интеграция с внешними сервисами.
1–С Университет	ERP–система для комплексного управления вузом (финансы, кадры, администрирование, учеба).	Возможна интеграция модулей для подписания приказов, финансовых и кадровых документов.	Требуется доработки для поддержки современных криптостандартов; возможны проблемы с совместимостью версий.
Тандем	Управление учебным процессом: регистрация студентов, ведение документации, обработка заявлений.	Автоматизация документооборота с потенциалом подписания регистрационных документов и заявлений.	Необходима модернизация для современных протоколов ЭЦП; ограниченная гибкость в настройках.
Апекс	Информационно–аналитическая система для формирования отчетов и управления данными.	Поддержка ЭЦП для юридической значимости отчетов и служебных записок.	Ограниченная гибкость безопасности; требуется обновление ПО для поддержки современных криптографических протоколов.

## **2.4 Выбор вида ЭЦП и обоснование**

Проведённый анализ теоретических основ электронной цифровой подписи, нормативно–правовой базы и особенностей информационных систем ИРНИТУ позволил выделить основные типы ЭЦП – простую, усиленную и квалифицированную. Каждая из них отличается уровнем криптографической защиты, юридической значимостью и затратами на оформление, что имеет прямое влияние на выбор оптимального решения для внутренних процессов образовательного учреждения.

При рассмотрении внутренних процессов документооборота, таких как подписание заявлений, академических ведомостей, уведомлений и иных документов, требующих оперативности и достаточного уровня безопасности, оптимальным представляется использование простой ЭЦП. Данный выбор обоснован следующими факторами:

### **1. Достаточный уровень безопасности для внутренних процессов.**

Простая ЭЦП обеспечивает базовую аутентификацию и целостность электронных документов, что позволяет надёжно подтверждать авторство и неизменность информации при обмене данными внутри учреждения. Для большинства внутренних коммуникаций между преподавателями, административным персоналом и студентами этот уровень защиты является достаточным, поскольку риск несанкционированного доступа или модификации данных невелик. Например, при оформлении заявлений о зачислении или академических ведомостей простая подпись позволяет ускорить процесс согласования без необходимости дополнительных затрат на более сложные процедуры идентификации.

### **2. Экономическая эффективность.**

Оформление простой ЭЦП не требует прохождения сложной и длительной процедуры идентификации, характерной для квалифицированной подписи, что существенно снижает как временные, так и финансовые затраты. Это особенно актуально для образовательной среды, где массовое использование цифровых подписей должно быть оперативным и недорогим. В условиях ограниченных бюджетных возможностей и необходимости быстрого обмена информацией применение простой ЭЦП позволяет оптимизировать затраты на внедрение и эксплуатацию системы электронного документооборота, что подтверждается результатами исследований современных практик.

### **3. Возможность создания специализированной онлайн–платформы.**

Применение простой ЭЦП открывает перспективы для разработки специализированной платформы, которая позволит преподавателям и студентам самостоятельно оформлять и использовать цифровую подпись в рамках внутренних процессов. Такая платформа может быть интегрирована с существующими информационными системами ИРНИТУ, что ускорит обмен информацией, повысит прозрачность документооборота и снизит административные издержки. Например, автоматизированное оформление заявлений, протоколов и справок через внутренний портал с применением

простой ЭЦП позволяет обеспечить оперативное и экономичное взаимодействие между всеми участниками образовательного процесса.

Для более полного понимания интеграции простой ЭЦП в процессы документооборота предлагается рассмотреть схему взаимодействия, которая иллюстрирует ключевые этапы обмена информацией между участниками внутреннего документооборота образовательного учреждения (см. Рисунок 2.4). Такая схема помогает наглядно представить, как осуществляется процесс подписания и проверки документов с использованием простой ЭЦП.

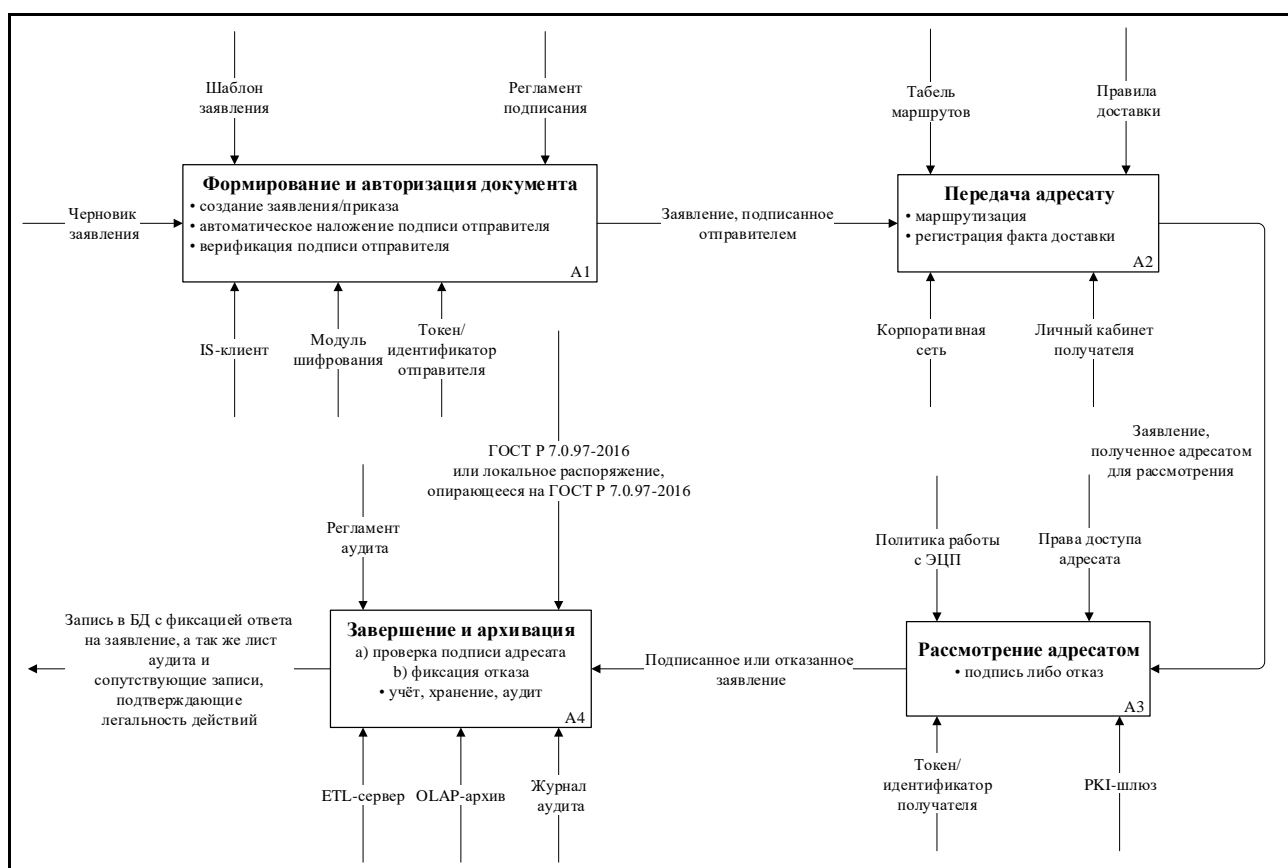


Рисунок 2.4 – Схема взаимодействия процессов внутреннего документооборота с использованием ЭЦП.

Эта схема демонстрирует, что простая ЭЦП может способствовать ускорению документооборота, снижению риска ошибок и оптимизации обмена информацией между участниками образовательного процесса. Преимущества такого подхода подтверждаются исследованиями, показывающими снижение времени обработки документов и экономию ресурсов за счёт автоматизации процедур.

Схема взаимодействия при использовании простой ЭЦП включает следующие этапы:

1. Формирование документа.

Преподаватель или сотрудник администрации формирует электронный документ (например, заявление, протокол, академическую ведомость) в интегрированной системе ИРНИТУ.

## 2. Подписание документа.

Сформированный документ автоматически передаётся в систему, где пользователь проходит процедуру аутентификации (например, через электронную почту или СМС–код) и подписывает документ с помощью простой ЭЦП. Это обеспечивает подтверждение авторства и неизменности информации.

## 3. Проверка подписи.

Получатель документа (например, приемная комиссия или деканат) получает подписанный документ, где встроенные алгоритмы системы автоматически проверяют цифровую подпись, сопоставляя хеш–суммы и удостоверяясь в целостности данных.

## 4. Обработка и хранение.

После проверки документ поступает на дальнейшую обработку и хранение в централизованном электронном архиве, что обеспечивает оперативный доступ и безопасность данных.

Для обоснования выбора конкретного типа электронной подписи в рамках внутренних и внешних бизнес–процессов ИРНИТУ была составлена таблица 2.2. В ней каждая строка отражает реальный документооборот университета – от экзаменационных ведомостей до договоров о платных услугах, – а каждый столбец показывает минимально достаточный класс ЭЦП (простая, усиленная, квалифицированная) с точки зрения:

- нормативных требований (Федеральный закон № 63–ФЗ «Об электронной подписи», ст. 6 и ст. 9; приказ Минобрнауки № 706 о реестре документов об образовании; постановление Правительства № 1083 об электронном документообороте);

- оценки юридических рисков (вероятность спорной ситуации, необходимость внешнего арбитража);

- стоимость затрат (выдача сертификатов, обслуживание СКЗИ, ежегодное продление);

- требований к скорости согласования (допустимое «время до подписи» для массовых потоков).

Результаты показывают, что простая ЭЦП удовлетворяет всем критериям для 90 % внутренних процедур (заявления студентов, ведомости, кафедральные протоколы, внутренние приказы), потому что:

- закон допускает признание ПЭП внутри организации соглашением руководителя (п. 2 ст. 6 № 63–ФЗ);

- риск внешних претензий к таким документам минимален;

- временные и финансовые издержки на ее выпуск в 5–10 раз ниже, чем у усиленной или квалифицированной подписи.

Усиленная ЭЦП в рамках внутренних процессов ИРНИТУ признана избыточной: для её использования требуются аппаратные токены,



дополнительная доработка программного обеспечения и поддержка инфраструктуры сертифицированных средств защиты информации, тогда как с точки зрения юридической значимости для внутренних файлов она не даёт ощутимых преимуществ. Квалифицированная ЭЦП оправдана лишь в точечных сценариях – при обмене данными с внешними государственными информационными системами (ФИС ФРДО, портал «Госуслуги»), где её применение прямо предписано постановлением Правительства № 1083 и соответствующими письмами Минцифры. В остальных же случаях вполне достаточна простая ЭЦП: она обеспечивает защиту целостности и подтверждение авторства документов без значительных затрат на внедрение и поддержку. Таким образом, представляемая таблица подчёркивает не только регуляторную, но и экономическую целесообразность фокусировки на простой ЭЦП в основном ИТ-контуре университета, позволяя оптимизировать ресурсы, упростить процессы подписания и сохранить необходимый уровень юридической значимости внутренних документов.

Таблица 2.2 – Соответствие типов ЭЦП бизнес–процессам ИРНИТУ

Вид бизнес–процесса	Простая ЭЦП	Усиленная ЭЦП	Квалифицированная ЭЦП
Заявления студентов, справки, текущие запросы	Необходима и достаточна. Признаётся внутренним приказом (п. 2 ст. 6 № 63–ФЗ). Минимальные трудозатраты; мгновенный выпуск. Не подходит для внешних споров.	Избыточна: аппаратные токены не дают дополнительных гарантий внутри конт-ура.	Избыточна: высокая стоимость и строгие регламенты без внешней необходимости.
Академические ведомости, журналы посещаемости	Достаточна. Покрывает требование целостности (приказ № 706). Позволяет автоматическую выгрузку отчётов в архив ИС.	Излишняя сложность для массовых файлов.	Нужна только при передаче извне (редко).

Продолжение таблицы 2.2

<b>Вид бизнес–процесса</b>	<b>Простая ЭЦП</b>	<b>Усиленная ЭЦП</b>	<b>Квалифицированная ЭЦП</b>
Внутренние распоряжения кафедр, протоколы заседаний	Достаточна. Быстрое согласование, полная прослеживаемость.	Переизбыток защиты.	Переизбыток защиты.
Приказы ректора о зачислении / переводе	Приемлема при внутреннем согласовании. При споре с третьими лицами может потребоваться усиление.	Возможна, если приказ подлежит внешнему обмену (например, запросы в другие вузы).	Обязательно только при направлении во внешние ГИС.
Договоры о платных услугах, грантовые соглашения	Допустима для черновиков, но не для финальной версии.	Приемлема, если контрагент согласен.	Требуется. Полная юридическая сила (ст. 9 № 63–ФЗ).
Выгрузка данных во ФРДО, портал «Госуслуги»	Недостаточна: ФОИВы не принимают ПЭП.	Недостаточна.	Обязательна. Предусмотрена постановлением № 1083 и регламентами Минцифры.

## 2.5 Подписываемые типы документов в ИРНИТУ

Внутренний документооборот в ИРНИТУ включает широкий спектр документов, оформление которых осуществляется с использованием электронной цифровой подписи. Особенности подписания документов позволяют обосновать выбор простой ЭЦП, поскольку большинство внутренних процессов не требует сверхвысокого уровня защиты, а главное – оперативность и экономическая эффективность. Рассмотрим основные группы документов:

### 1. Документы, подписываемые студентами:

– курсовые проекты;

Курсовые проекты представляют собой итоговую работу по определённой дисциплине, в которой студент демонстрирует владение теоретическими и практическими знаниями. Подписание курсовых проектов электронной цифровой подписью обеспечивает подтверждение авторства, а также

неизменность представленного материала, что важно для оценки преподавателем.

- заявления на перевод;

При переводе студента с одного курса или факультета на другой подается заявление, содержащее личные данные, академическую историю и мотивацию к переводу. Электронное подписание таких заявлений позволяет ускорить процесс их рассмотрения и упрощает архивирование документов.

- документы на практику.

Для прохождения производственной практики студент оформляет различные документы – от заявлений до отчетов по практике. Использование ЭЦП гарантирует, что информация об участии студента в практике передается достоверно и оперативно, что важно для взаимодействия с предприятиями и учебными заведениями.

Заявление на академический отпуск.

Заявление на академический отпуск требует формального оформления, подтверждающего желание студента временно приостановить учебный процесс. Простая ЭЦП позволяет быстро и юридически корректно оформить данный документ, обеспечивая его соответствие внутренним требованиям учреждения.

2. Документы, подписываемые преподавателями:

- курсовые работы;

Преподаватели, оценивающие курсовые работы, могут подписывать оценочные листы и рекомендации с помощью ЭЦП, что подтверждает их экспертизу и авторство комментариев. Это упрощает процесс аттестации и делает его прозрачным для всех участников учебного процесса.

- документы по практике.

В рамках проведения практических занятий преподаватели подписывают ведомости, отчеты и протоколы по практике. Подписание таких документов с использованием простой ЭЦП гарантирует оперативное оформление и достоверность информации об успеваемости и посещаемости студентов.

3. Документы, подписываемые кафедрой:

- административные заявления (например, справка с места учёбы);

Кафедра формирует и подписывает справки, подтверждающие факт обучения или прохождения практики студентом. Это важный документ для взаимодействия с другими структурными подразделениями и внешними организациями, где требуется подтверждение статуса студента.

- ведение академической успеваемости (зачётки);

Документы, фиксирующие успеваемость студентов, такие как зачетные книги или электронные ведомости, оформляются с использованием ЭЦП. Это позволяет обеспечить точность и неизменность данных, что является основой для оценки академической деятельности.

- списки предоставленных к отчислению.

В случаях, когда необходимо официально уведомить о предоставлении документов для процедуры отчисления, кафедра подписывает соответствующие

списки. ЭЦП гарантирует юридическую силу такого документа, что важно для последующего взаимодействия с административными органами.

В результате аргументов, сказанных ранее, следует вывод, что типы документов, подписываемых в ИРНИТУ, показывают, что большинство из них предназначены для внутреннего обмена информацией, где ключевыми факторами являются оперативность, экономическая эффективность и достаточный уровень безопасности. Именно поэтому использование простой электронной цифровой подписи является оптимальным выбором для обеспечения надежного, быстрого и законного документооборота в образовательном учреждении.

## **3 Методика применения ЭЦП**

### **3.1 Формирование требований к системе**

Перед определением перечня документов, для которых применение простой электронной подписи (ПЭП) является обоснованным, необходимо детализировать аргументы, подтверждающие целесообразность выбора именно этого уровня криптографической защиты для внутреннего документооборота ИРНИТУ:

#### **1. Нормативная достаточность.**

Согласно п. 2 ст. 6 Федерального закона № 63–ФЗ «Об электронной подписи» внутренние документы организации могут удостоверяться простой подписью при наличии локального нормативного акта, устанавливающего правила её использования.

#### **2. Соразмерность актуальных угроз.**

Анализ зарегистрированных инцидентов информационной безопасности показывает, что основные риски связаны с организационными и процедурными факторами (утрата бумажных экземпляров, рассогласование версий файлов, задержка согласований), тогда как вероятность криптоаналитической атаки на внутренние документы незначительна.

#### **3. Повышение операционной эффективности.**

ПЭП переводит процедуры визирования в полностью электронный формат: согласующий получает уведомление в информационной системе, подписывает файл с любого авторизованного устройства, а статус документа автоматически фиксируется в журнале. Это устраняет необходимость печати, физического перемещения папок и поиска ответственных лиц, сокращая административные задержки и высвобождая трудовые ресурсы кафедр и деканатов.

#### **4. Технологическая совместимость.**

Ключевые информационные системы университета (Moodle, 1С «Университет», «Тандем», «Апекс») поддерживают подключение внешних криптографических сервисов через стандартные REST– или RPC–интерфейсы.

#### **5. Гибкость дальнейшего масштабирования.**

Предлагаемая архитектура предусматривает возможность эскалации уровня подписи для отдельных процессов (например, передачи выпускных квалификационных работ во ФРДО), не затрагивая массовый документооборот. Таким образом, ПЭП выступает минимально достаточным, но масштабируемым решением.

На основании указанных положений определены категории документов, для которых применение ПЭП обеспечивает требуемый баланс между юридической значимостью, безопасностью и экономической целесообразностью (табл. 3.1).

Таблица 3.1 – Обоснование выбора ПЭП для внутренних бизнес–процессов в ИРНИТУ

Категория документа	Типовые примеры	Обоснование достаточности ПЭП	Ограничения / примечания
Учебные задания	Лабораторные задания, индивидуальные задания.	Подтверждение авторства и даты выдачи; документы не покидают вуз; автоматизируется массовая рассылка и сбор подписей.	—
Отчёты	Лабораторные отчёты, дневники и итоговые отчёты по практике.	Фиксация факта сдачи и неизменности; отчёт мгновенно передаётся на проверку без печати.	—
Курсовые работы	Пояснительные записки, расчётно–графические задания.	ПЭП фиксирует окончательную версию; сокращается время согласования.	При внешней публикации требуется дополнительное заверение.
Выпускные квалификационные работы (ВКР)	Пояснительная записка, приложения, отчёт и т.д.	ПЭП применима на стадиях внутреннего согласования.	Финальная версия, направляемая во ФРДО, заверяется КЭП.
Рабочие документы	Внутренние справки, служебные записки и т.д.	Действуют в домене @istu.edu; локальный акт ректора признаёт ПЭП достаточной.	Недействительны вне вуза без дополнительной подписи/печати.

## 3.2 Техническая архитектура решения

### 3.2.1 Описание выбора стека технологий

Для демонстрационного решения принят открытый и легко переносимый технологический стек, отвечающий требованиям российских криптографических стандартов и минимизирующий затраты на развёртывание:

1. Серверная платформа:

- Node.js 14 и Express 4 – неблокирующий ввод–вывод, богатый экосистемой пакетов;

- crypto-gost – реализация ГОСТ Р 34.10–2012 и ГОСТ Р 34.11–2012 для подписи и хеширования;

- SQLite 3 – файловая база, обеспечивающая ACID–транзакции без отдельного СУБД–сервера и легко мигрируемая на PostgreSQL.

2. Клиентская платформа:

- React 18 и TypeScript 5 – компонентная модель с типовой безопасностью;

- Web Crypto API – генерация пары RSA–2048 прямо в браузере;

- AES–256–GCM и PBKDF2 – шифрование закрытого ключа и стойкая схема вывода ключей.

Дополнительные технологические элементы:

Аутентификация и авторизация. Контур OAuth 2.0 выдаёт JWT–токены, содержащие хэш открытого ключа пользователя; при необходимости включается двухфакторная проверка TOTP (библиотека otplib).

Транспортная безопасность. Все REST–вызовы защищены TLS 1.3; в лабораторной среде используется самоподписанный сертификат, в промышленных условиях достаточно подключить сертификат внутреннего СА.

Журнал аудита. Записи о создании, подписании и проверке документов формируют хэш–цепочку, что обеспечивает неизменяемость истории без применения специализированных WORM–хранилищ.

Выбранный стек – Node.js, Express, crypto-gost, React, TypeScript, Web Crypto API, SQLite и OAuth 2.0 – демонстрирует, что реализация простой электронной подписи возможна исключительно средствами открытого программного обеспечения, без закупки аппаратных СКЗИ и без подключения к аккредитованному УЦ. Такое сочетание технологий удовлетворяет требованиям ИРНИТУ к быстрому развёртыванию, минимальной стоимости владения и последующей масштабируемости.

### 3.2.2 Описание модулей Backend

Разработанный демонстрационный модуль построен в парадигме «клиент–сервер» и демонстрирует полный жизненный цикл простой электронной подписи во внутреннем контуре университета. Криптографическая логика сосредоточена в библиотеке *crypto – gost*. Генерация и шифрование пользовательской пары ключей реализованы в модуле *cryptoUtil.js* (см. Приложение А): закрытый ключ формируется в браузере средствами Web Crypto API, затем шифруется алгоритмом AES–256–GCM; ключ шифрования выводится по PBKDF2 из

комбинации идентификатора учётной записи и одноразового токена. Операции наложения и проверки простой подписи выполняются сервисом *signatureService.js* (см. Приложение Б), где дайджест документа рассчитывается по SHA–256, а результирующий контейнер формируется в формате CAdES–BES. Расширенные функции – добавление штампа времени RFC 3161 и управление отзывом ключей – вынесены в *enhancedSignatureService.js* (см. Приложение В).

Приём, маршрутизация и авторизация запросов осуществляются модулем *requestService.js* (см. Приложение Г), который выпускает JWT–токен OAuth 2.0; в нагрузку токена включён хэш открытого ключа пользователя, что позволяет однозначно связать подпись с активной сессией без обращения к внешнему удостоверяющему центру. Метаданные документов, их статусы и пути к файлам сохраняются через *docsService.js* (см. Приложение Д) в локальной базе SQLite; тот же модуль выполняет автоматическое обновление статуса после каждой успешной операции подписи или проверки. Для обеспечения неизменяемого журнала действий применяется *logUtil.js* (см. Приложение Е): каждое событие фиксируется с отметкой времени и образует хэш–цепочку, что исключает последующее редактирование без обнаружения.

Таким образом, представленное решение демонстрирует, что генерация, хранение и проверка простой электронной подписи могут быть реализованы исключительно средствами открытого программного обеспечения и без привлечения аккредитованных удостоверяющих центров, что полностью соответствует нормативной модели ПЭП и задачам оптимизации внутреннего документооборота.

### 3.2.3 Логика клиентского взаимодействия и поток данных

Клиентская часть, реализованная на React / TypeScript, выполняет три последовательные операции:

#### 1. Генерация ключевой пары

При первом обращении компонент вызывает Web Crypto API → *crypto.subtle.generateKey(...)* (RSA–2048, SHA–256). Закрытый ключ шифруется AES–256–GCM, для вывода симметричного ключа применяется PBKDF2 с солью, сформированной из идентификатора LDAP–учётной записи. Зашифрованный контейнер сохраняется в браузерном хранилище IndexedDB.

#### 2. Подпись документа

Пользователь выбирает файл; в памяти браузера вычисляется дайджест SHA–256, после чего компонент отправляет REST–запрос *POST /sign* с полем *hash*. На сервер передаются только дайджест и идентификатор открытого ключа, содержимое файла сеть не покидает – тем самым минимизируется риск утечки данных.

#### 3. Проверка подписи и обратная доставка

После получения контейнера *CAdES – BES* клиент автоматически прикрепляет его к исходному файлу и пересылает «следующему визирующему» по URL, указанному в JSON–манифесте маршрута. При открытии файла у



получателя компонент выполняет вызов *GET /verify?id =*; сервер сверяет подпись с публичным ключом из хранилища открытых ключей, фиксирует результат в журнале и возвращает статус в формате JSON. Если подпись корректна, интерфейс меняет состояние документа на «подписано», а кнопка «Подписать» становится недоступной.

Архитектура транспортного обмена такова, что в открытом виде передаются лишь дайджесты и метаданные, тогда как сам файл остаётся в клиентском сегменте сети вплоть до финальной загрузки в архив. Это решение демонстрирует комплексное применение выбранного стека и дополняет описанные выше серверные модули, обеспечивая сквозную прослеживаемость операции «создание → подпись → валидация → архив».

Добавление данного подраздела завершит техническую картину: читатель получит целостное понимание того, как выбранные технологии стыкуются на уровне кода, протокола и пользовательского взаимодействия.

### **3.3 Модель безопасности и управления ключами**

Предлагаемая модель строится вокруг принципа сквозной автоматизации: все операции с простой электронной подписью (ПЭП) выполняются без участия администратора и без передачи закрытого ключа по сети, а контроль-и аудит происходят в фоновом режиме. Это снимает нагрузку с ИТ-персонала, устраняет человеческий фактор и формирует доказуемую юридическую базу каждой транзакции.

1. Автоматическая генерация ключевой пары. При первом входе в личный кабинет браузер средствами Web-Crypto API создаёт асимметричную пару RSA-2048; закрытый ключ сразу шифруется паролем профиля (AES-GCM), открытый публикуется в LDAP-каталоге, где однозначно сопоставляется с учётной записью пользователя.

2. Локальное хранение закрытого ключа. Зашифрованный контейнер сохраняется только на рабочей станции и в резервной копии профиля; сервер не имеет доступа к приватному материалу.

3. Регламентированный жизненный цикл. Срок действия ключа — 12 месяцев; по истечении портал автоматически инициирует повторную генерацию. Отзыв по событию (утрата оборудования, окончание обучения) отражается в CRL-списке, публикуемом каждые 15 минут.

4. Процесс подписи «одним кликом». Пользователь нажимает «Подписать»; клиент формирует хэш-сумму (ГОСТ 34.11-2012) и шифрует её закрытым ключом. Полученный контейнер JSON+РЭП добавляется к документу и передаётся на сервер.

5. Неизменяемый журнал событий. Сервер фиксирует идентификатор пользователя, SHA-256 хэш файла, UTC-метку времени и IP-адрес в цепочке хэшей (hash-chain).

6. Синхронизация доступа. Блокировка учётной записи в LDAP автоматически делает ключ недействительным; повторная активация возможна только через процедуру самообслуживания с подтверждением личности.

7. Нормативное закрепление юридической силы. Приказ ректора устанавливает эквивалентность документов, подписанных данной ПЭП, собственноручным подписям внутри университета; для внешних адресатов добавляется штамп времени RFC 3161, подтверждающий неизменность контента.

8. Ожидаемые эффекты внедрения. Полная автоматизация жизненного цикла подписи, отсутствие «ручных» операций с ключами, сокращение среднего времени подписания, аудиторская трассировка каждой транзакции и соответствие требованиям 63-ФЗ без применения квалифицированных или усиленных ЭЦП.

Помимо описанных выше технических механизмов, модель безопасности и управления ключами включает в себя организационно-методические решения, направленные на обеспечение устойчивости и управляемости процессов:

Одним из таких решений является четкое разграничение ролей и ответственности, установленное локальными нормативными актами. В рамках модели выделяются три основные роли:

- Пользователь – единственный субъект, обладающий доступом к закрытому ключу и правом подписания документа;

- администратор каталога – ответственен за ведение и поддержку актуальности публичных ключей в справочнике LDAP, при этом не имея возможности доступа к приватному материалу пользователей;

- аудитор информационной безопасности – независимая роль, предназначенная для мониторинга, проверки журналов событий и своевременного выявления аномалий.

Такое ролевое разграничение минимизирует риски несанкционированного использования ЭЦП, делает процессы прозрачными и облегчает внутренний контроль.

Дополнительно, модель включает регулярные процедуры аудита и мониторинга. Предусмотрена интеграция системы журналирования с корпоративными решениями класса SIEM, которые позволяют в автоматическом режиме выявлять аномальные активности, такие как нетипичное количество операций подписи или попытки использования одного ключа с нескольких IP-адресов за короткий промежуток времени. Это позволяет оперативно реагировать на потенциальные угрозы и блокировать скомпрометированные ключи до возникновения негативных последствий.

Отдельное внимание стоит уделить резервному копированию и восстановлению ключевого материала. Оптимальным вариантом сделать ежедневное автоматическое резервное копирование зашифрованных контейнеров ключей пользователей в безопасное хранилище данных, которое создаёт резервные копии. Копирование осуществляется инкрементально, минимизируя затраты времени и ресурсов. Регулярное (квартальное) выборочное восстановление обеспечивает проверку целостности резервных копий и снижает риски утраты ключей вследствие технических сбоев.

Наконец, модель предусматривает механизм обеспечения криптографической гибкости. Внедрение программных интерфейсов позволяет в будущем оперативно перейти на использование новых, включая пост-квантовые, алгоритмов без необходимости глубоких изменений инфраструктуры. Это обеспечивает долгосрочную устойчивость и готовность системы к эволюции криптографических стандартов и угроз.

В совокупности указанные организационно-методические компоненты дополняют технологическое решение, формируя комплексную, устойчивую и прозрачную систему управления простой электронной подписью, соответствующую актуальным нормативным требованиям и лучшим практикам информационной безопасности.

Для наглядного понимания технической реализации предложенной модели управления ключами и простой электронной подписью (ПЭП) ниже приведена схема (см. рисунок 3.1), отражающая основные этапы и взаимодействия компонентов системы.

При авторизации пользователя в личном кабинете система не только обеспечивает проверку его учётных данных, но и автоматически контролирует наличие действующей ключевой пары. Если ключей ещё нет, браузер незаметно для пользователя генерирует пару RSA-2048 с помощью встроенного Web-Crypto API.

После генерации ключей система дополнительно проверяет целостность открытого ключа, прежде чем сохранить его в LDAP-каталоге. В это же время зашифрованный контейнер закрытого ключа сохраняется локально у пользователя и, кроме того, копируется в зашифрованном виде в резервное хранилище, откуда он может быть восстановлен только самим пользователем после дополнительного подтверждения личности.

Во время подписания документа система не просто добавляет подпись, но и автоматически фиксирует в журнале событий дополнительные параметры окружения пользователя (например, версию браузера или платформы). Эти метаданные усиливают доказательную базу и повышают устойчивость к юридическим спорам, позволяя однозначно подтвердить факт подписания документа именно конкретным пользователем с его рабочего устройства.

Процесс резервного копирования организован таким образом, чтобы обеспечить минимальное воздействие на производительность системы пользователя. Для этого используются фоновые механизмы инкрементального копирования, а целостность резервных копий периодически проверяется независимым процедурным механизмом, что минимизирует вероятность сбоев и потерь.

Таким образом, представленная схема и описанный сценарий демонстрируют комплексный подход системы к безопасности, автоматизации и управлению жизненным циклом ключей, обеспечивая при этом полную прозрачность и надёжность документооборота с простой электронной подписью.

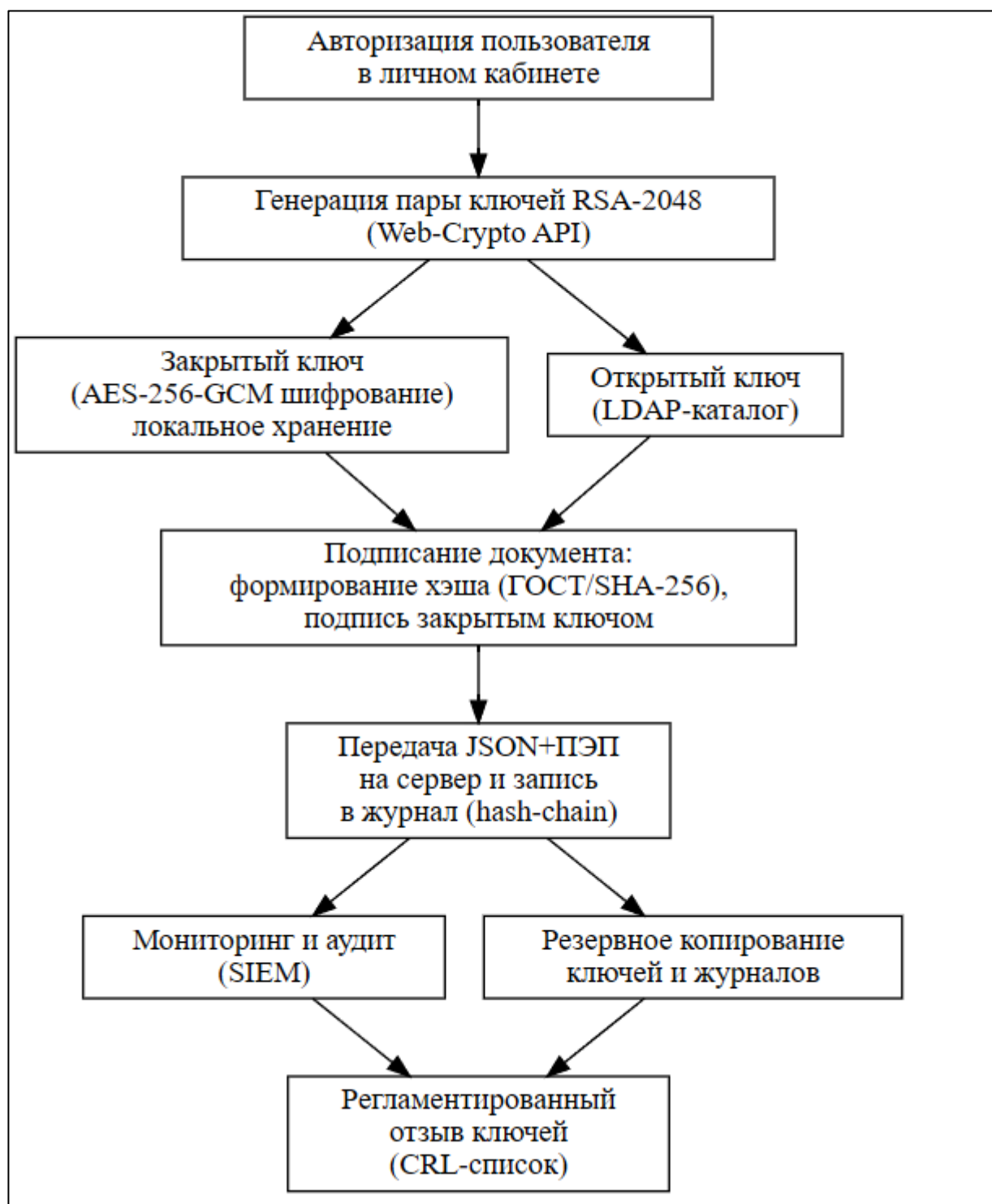


Рисунок 3.1 – Схема взаимодействия компонентов системы

### 3.4 Методика сопровождения и контроля корректности работы компонентов ЭЦП

Оптимальная организация сопровождения системы электронной подписи включает целый комплекс технических и организационных мер, направленных на повышение надёжности и прозрачности эксплуатации. Ниже приведён

рекомендуемый перечень таких процедур, которые целесообразно внедрять для поддержания устойчивой и защищённой работы всех ключевых компонентов системы:

- регулярно осуществлять автоматическое тестирование каждого модуля (cryptoUtil.js, signatureService.js, docsService.js и др.), чтобы своевременно выявлять и устранять ошибки в генерации, хранении и применении ключей, а также в процессе формирования и проверки подписи;

- реализовать контроль целостности всех журналов операций посредством хэш-цепочек: это позволяет зафиксировать любые попытки вмешательства в историю подписей и обеспечить неизменяемость критических событий;

- собирать и анализировать технические метрики, такие как среднее время генерации ключей, длительность процедуры подписания, количество успешных и неуспешных транзакций, а также частота отказов — эти показатели позволяют отслеживать состояние системы в динамике и выявлять аномалии до возникновения инцидентов;

- интегрировать инфраструктуру мониторинга с корпоративной SIEM-системой, чтобы в автоматическом режиме обнаруживать подозрительную активность: например, массовые подписи за короткое время, необычные IP-адреса или попытки повторной генерации ключей вне регламента;

- предусматривать ручную верификацию подписей и логов со стороны службы информационной безопасности, особенно при возникновении спорных ситуаций, подозрительных инцидентов или обращениях пользователей с жалобами на доступность сервисов;

- внедрить процедуру автоматической ротации ключей с оповещением пользователя за несколько дней до окончания срока действия; при замене ключа прежний сохраняется в архиве для возможности проверки ранее подписанных документов;

- осуществлять регулярное резервное копирование зашифрованных контейнеров ключей и журналов на удалённые и физически защищённые носители; желательно предусмотреть периодические проверки целостности резервных копий для минимизации риска потери информации при сбоях;

- формировать и отслеживать внутренние метрики и KPI, отражающие стабильность работы системы, среднее время реакции на инциденты, уровень защищённости, долю успешно завершённых операций и количество отзывов ключей за период.

Комплексное внедрение этих подходов обеспечивает высокую устойчивость, прозрачность и доказуемую надёжность инфраструктуры ПЭП, что особенно важно для доверия к электронным документам и успешного прохождения внешних и внутренних аудитов.

## **4 Разработка и тестирование прототипа**

### **4.1 Техническое задание на разработку прототипа**

Прототип должен моделировать полный жизненный цикл электронного согласования университетских документов без встраивания подписи в файл. Вместо «мокрой» подписи в самом PDF либо DOCX система формирует и хранит в собственной базе данных криптографическую сигнатуру (SHA-256 хеш и открытый ключ пользователя) и статус «подписано» либо «отказ». Документ остаётся у инициатора либо в файловом хранилище, а целостность проверяется сравнением актуального хеша с зафиксированным эталоном.

Основная логика прототипа включает пять взаимосвязанных задач.

1. Формирование заявки. Пользователь-инициатор загружает документ; система вычисляет его SHA-256, в БД создаёт карточку.

2. Автоматическая подпись отправителя. Сразу после создания карточки формируется запись (хеш, публичный ключ отправителя, время). Это имитирует внутреннее «наложение» простой ЭЦП без изменения файла.

3. Отправка и реакция адресата. Документ отображается в личном кабинете адресата. Он выбирает «подписать» или «отказать». При подтверждении создаётся переменная с его открытым ключом, статус карточки меняется на «подписано»; при отказе фиксируется причина, статус – «Отказано».

4. Проверка целостности. Любой участник может загрузить локальную копию документа: система снова вычисляет хеш и сравнивает с эталоном. Совпадение подтверждает неизменность, расхождение сигнализирует о нарушении.

5. Журналирование и отчётность. Каждый переход статуса сохраняется в журнале аудита с меткой времени. При выборке по id пользователь получает полную историю согласования и действительность подписи.

Для реализации требуется выполнить следующие работы:

- разработать схему БД;
- реализовать серверное API;
- создать простой веб-интерфейс;
- обеспечить локальное шифрование приватных ключей (парольная защита) и хранение открытых ключей в базе;
- подготовить набор тестовых документов и сценарии проверки корректности статусов.

Прототип предназначен исключительно для демонстрации принципа работы ЭЦП в образовательной среде ИРНИТУ; он не требует интеграции с действующими системами и не использует квалифицированные сертификаты.

### **4.2 Пользовательские интерфейсы**

#### **4.2.1 Экран «Вход в систему»**

На рисунке 4.1 представлен экран «Вход в систему». Интерфейс состоит из центрально выровненной карточки авторизации.

- Поле «Логин (e-mail)»: предназначено для ввода идентификатора пользователя. При клике поле активируется (меняется фон и подсвечивается рамка);
- Поле «Пароль»: скрытый ввод текста, также подсвечивается при фокусе;
- Кнопка «Войти»: при нажатии инициирует проверку учётных данных. В случае успеха пользователь перенаправляется на главную страницу приложения; при ошибке – появляется информационное сообщение (например, «Неверный логин или пароль»);
- Ссылка «Забыли пароль? / Регистрация»: открывает формы восстановления доступа или создания нового аккаунта. Экраны данных форм представлена на рисунка 4.2 и 4.3 соответственно.

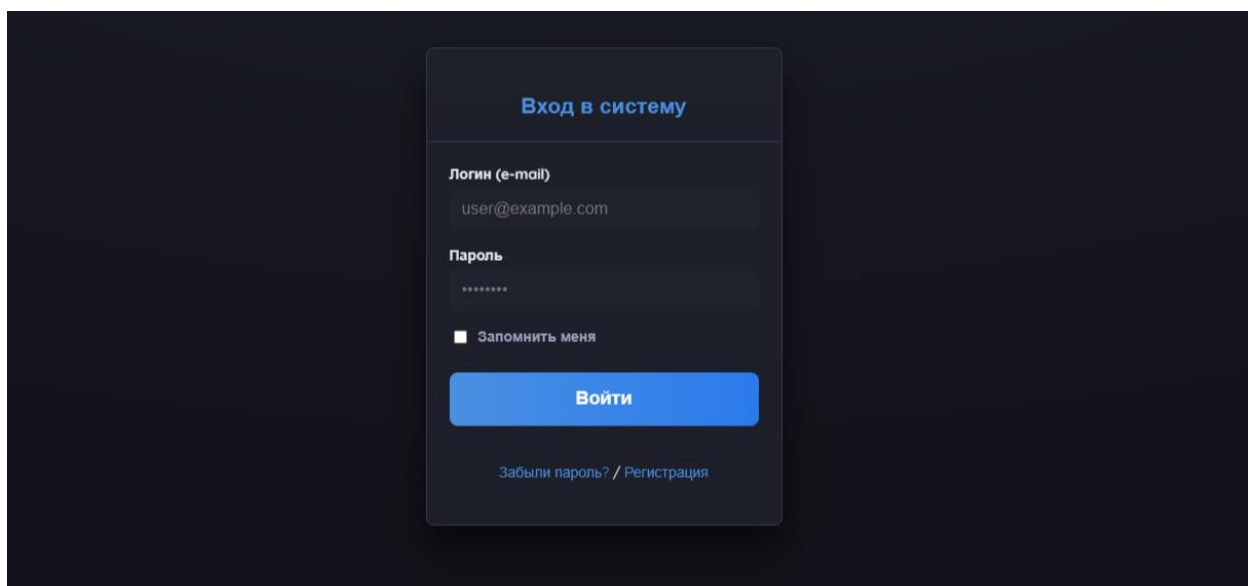


Рисунок 4.1 – Экран «Вход в систему».

Компонент создаёт многофункциональное поле ввода с локализованной валидацией, анимацией «shake» и кнопкой показать/скрыть пароль. Благодаря этому один и тот же код используется в формах «Вход», «Регистрация» и «Сброс пароля». Пример листинга:

```
function Field({ label, name, type = "text", required = false,
  show, toggleShow, placeholder = "" }) {
  const [err, setErr] = useState("");
  const isPass = type === "password";

  const onInvalid = e => {
    e.preventDefault();
    setErr(e.target.validity.valueMissing
      ? "Поле обязательно"
      : "Некорректное значение");
    e.target.classList.add(styles.hasError, styles.shake);
  }
```

```

setTimeout(() => e.target.classList.remove(styles.shake), 600);
};

const onInput = e => { if (err) setErr(""); e.target.classList.remove(styles.hasError); };

return (
<label className={` ${styles.field} ${err && styles.hasError}`}>
  {label}
  <div className={isPass ? styles.passWrap : ""}>
    <input name={name} type={show ? "text" : type} required={required}
    placeholder={placeholder} onInvalid={onInvalid} onInput={onInput}/>
    {isPass && (
      <button type="button" className={styles.toggle} onClick={toggleShow}>
        {show ? <EyeOff size={18}/> : <Eye size={18}/>}
      </button>
    )}
  </div>
  {err && <span className={styles.error}>{err}</span>}
</label>
);
}

```

Локальная валидация без перезагрузки страницы улучшает UX, а вынос логики в отдельный компонент предотвращает дублирование кода при смене режимов.

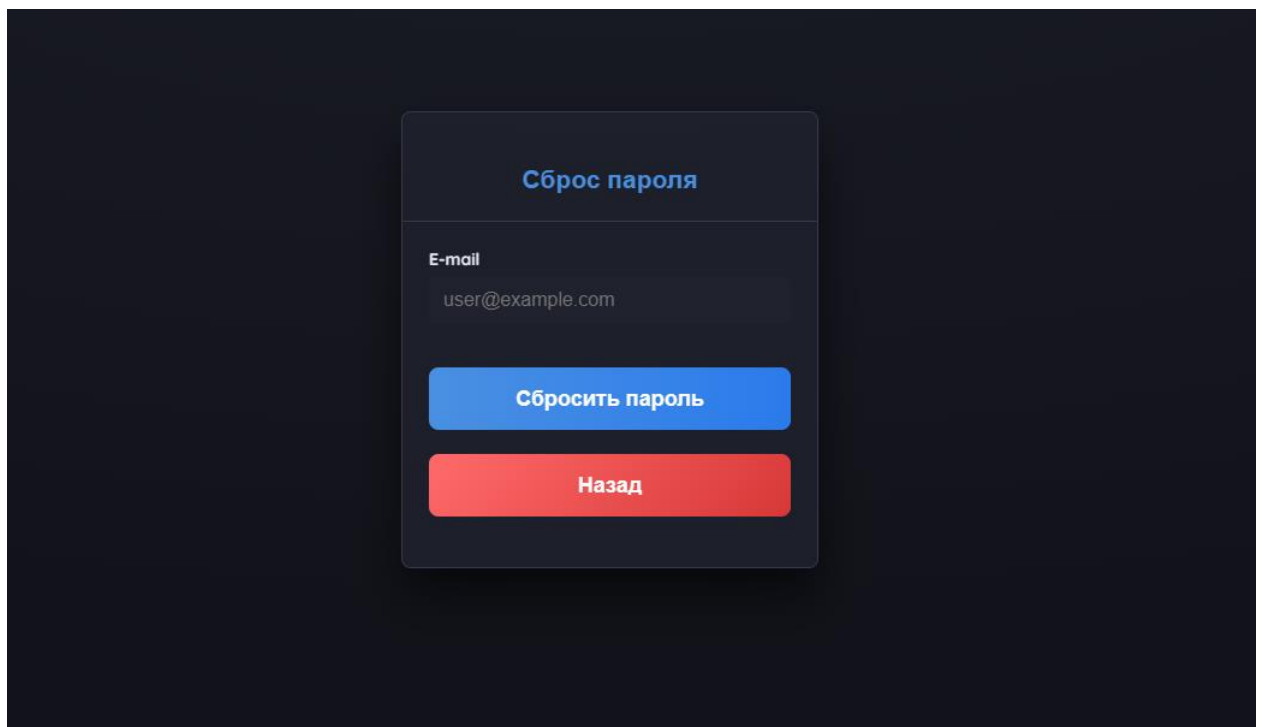


Рисунок 4.2 – Экран «Забыли пароль?».



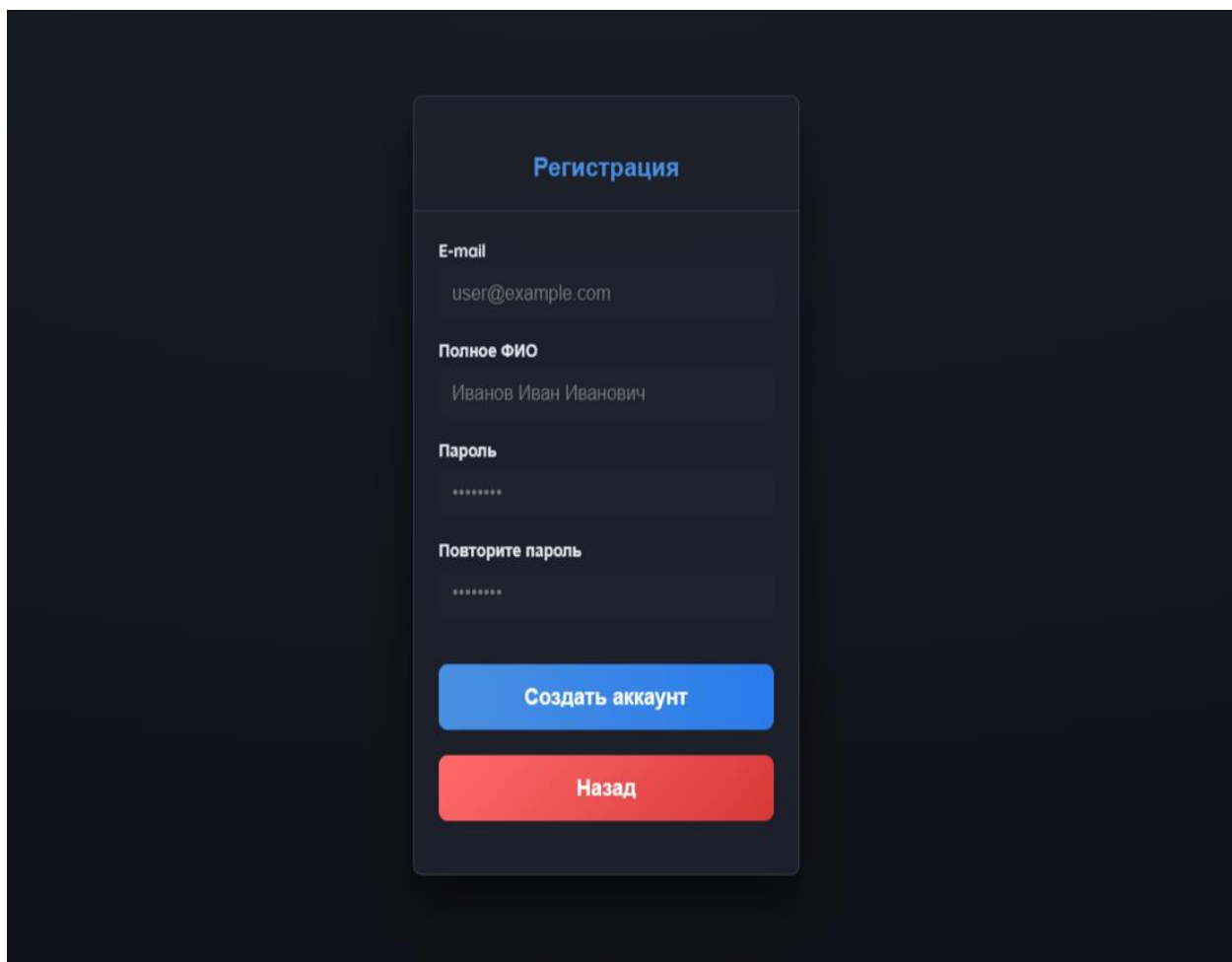


Рисунок 4.3 – Экран «Регистрация».

Обработчик `handleLogin` (асинхронная авторизация) демонстрирует асинхронный вызов REST-сервиса `loginService`, переход на двухфакторную проверку при необходимости и программную переадресацию в кабинет пользователя через `useNavigate`. Пример листинга:

```
const handleLogin = async e => {  
  e.preventDefault();  
  setLoading(true);  
  const f = e.target;  
  if (!f.checkValidity()) { f.classList.add(styles.wasValidated); setLoading(false); return;  
}  
  
  try {  
    const res = await loginService({ email: f.email.value, password: f.password.value });  
  
    if (res.need_otp) {  
      setTmpToken(res.tmp_token);  
      setMode("otp");  
    }  
  }  
}
```

```

    } else {
      localStorage.setItem("authToken", res.token);
      document.cookie = `id_user=${res.user.id_user}; path=/;`;
      navigate("/dashboard");
    }
  } catch (err) {
    setErrorMsg(err.message);
    f.email.classList.add(styles.hasError, styles.shake);
    setTimeout(() => f.email.classList.remove(styles.shake), 600);
  } finally { setLoading(false); }
};

```

При успехе токен сохраняется в localStorage, что обеспечивает персистентность сессии. Ветвь need\_otp переключает компонент на форму ввода одноразового кода, тем самым добавляя уровень безопасности без потери удобства пользователя.

#### 4.2.2 Экран «Панель управления»

На рисунке 4.4 показан главный рабочий экран после успешного входа в систему. Он предназначен для быстрой навигации по разделам и оперативной работы с документами.

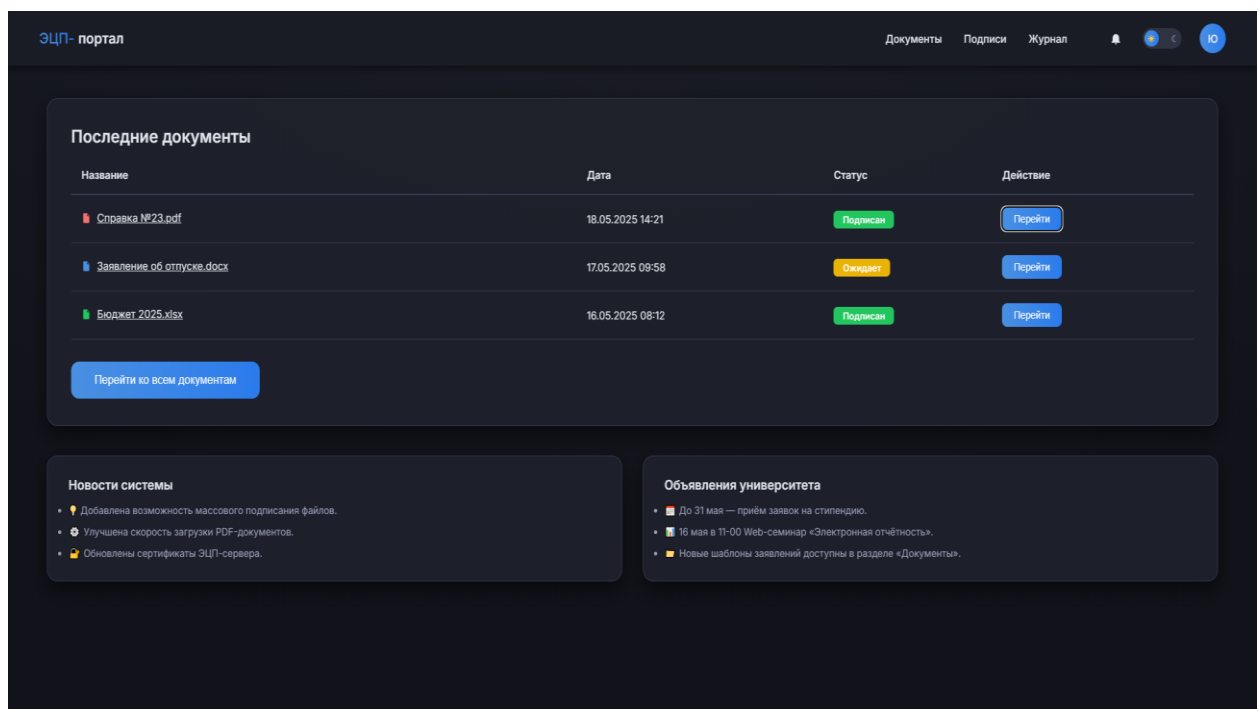


Рисунок 4.4 – Экран «Панель управления».

Верхняя навигационная панель включает логотип, пункты меню «Документы / Подписи / Журнал», индикатор уведомлений, крупный переключатель «тёмная / светлая тема» и кнопку–профиль с выпадающим меню «Мой профиль / Выход».

Блок «Последние документы» отображает три наиболее свежих файла. Для каждой записи выводятся:

- кликабельное название–ссылка с пиктограммой формата (PDF, DOCX, XLSX и т.д.), открывающее файл для скачивания или просмотра;
- дата и время загрузки;
- статус подписи в виде цветной метки («Подписан» – зелёный, «Ожидает» – жёлтый, «Отклонён» – красный);
- кнопка «Перейти», переходящая к детальной карточке документа.

Кнопка «Перейти ко всем документам» расположена под таблицей и ведёт на полноформатный реестр файлов.

Ниже размещены два информационных блока:

- «Новости системы» – лента технических обновлений платформы;
- «Объявления университета» – актуальные внутренние новости и расписания мероприятий.

Фрагмент листинга демонстрирует, каким образом на панели «Последние документы» формируются цветовые иконки по расширению файла и динамические бейджи состояния подписи:

```
const colorByExt = {
  pdf: "#EF4444", // красный
  docx: "#3B82F6", doc: "#3B82F6",
  xlsx: "#22C55E", xls: "#22C55E",
  pptx: "#F97316", ppt: "#F97316"
};
const pickColor = ext => colorByExt[ext] || "#9CA3AF";

const FileIcon = ({ ext }) => (
  <svg width="16" height="16" viewBox="0 0 24 24" fill={pickColor(ext)}>
    <path d="M6 2h7l5 5v13a2 2 0 0 1-2 2H6a2 2 0 0 1-2 2V4a2 2 0 0 1 2-2Z" />
  </svg>
);

const StatusTag = ({ status = "" }) => {
  const css =
    /signed|подпис/ .test(status) ? "signed" :
    /reject|отклон/ .test(status) ? "rejected" :
    "pending"; // «Ожидает»
  return <span className={` ${styles.status} ${styles[css]} `}>{status}</span>;
};
```

Цветовая кодировка облегчает мгновенное визуальное различие типов документов при просмотре списка.

Класс-модификатор `styles[css]` задаёт зелёный, жёлтый или красный фон бейджа, коррелирующий со статусом подписи («Подписан», «Ожидает», «Отклонён»).

Регулярные выражения рассчитаны на двуязычные ответы сервера – русские и английские.

Следующий фрагмент иллюстрирует запрос трёх последних документов и их вывод в таблицу; при нажатии на название выполняется скачивание файла с проверкой токена доступа:

```
const [docs, setDocs] = useState([]);

useEffect(() => {
  fetch("/api/my-documents?limit=3", {
    headers: { Authorization: `Bearer ${getToken()}` }
  })
  .then(r => r.json())
  .then(d => setDocs(d.documents || []))
  .catch(() => setDocs([]));
}, []);

const downloadFile = async (id, name) => {
  try {
    const res = await fetch(`/api/documents/${id}/download`, {
      headers: { Authorization: `Bearer ${getToken()}` }
    });
    if (!res.ok) throw new Error();
    const blob = await res.blob();
    const url = URL.createObjectURL(blob);
    const a = Object.assign(document.createElement("a"),
      { href: url, download: name });
    a.click(); URL.revokeObjectURL(url);
  } catch { alert("Не удалось скачать файл"); }
};
```

Запрос к маршруту `/api/my-documents` ограничивается тремя записями, что соответствует компактному виду панели (таблица в центре рис. 4.4). Токен авторизации извлекается вспомогательной функцией `getToken`, обеспечивая доступ только аутентифицированным пользователям. Скачивание реализовано через временный объект-URL без промежуточного сохранения файла на сервере, что повышает скорость и снижает нагрузку на дисковое пространство. Обработчик ошибок выводит локализованное сообщение, сохраняя дружелюбие интерфейса.

### 4.2.3 Экран «Документы»

На рисунке 4.5 представлен экран «Документы».

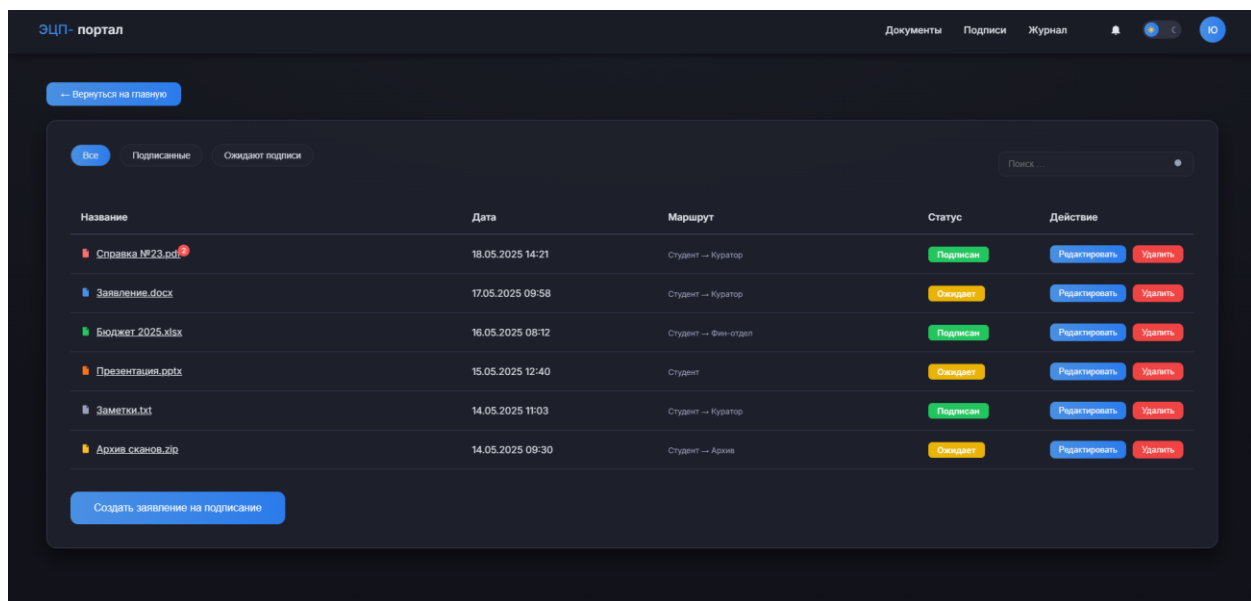


Рисунок 4.5 – Экран «Документы».

Под заголовком размещается панель фильтров–кнопок: «Все», «Подписанные», «Ожидают подписи». Активная вкладка подсвечена. Переключение выводит только соответствующие записи.

Поле «Поиск ...» (справа) выполняет текстовую фильтрацию списка по названию или расширению файла.

Таблица–реестр выводит полный перечень загруженных файлов. Для каждой строки отображаются:

- кликабельное название–ссылка с пиктограммой формата (PDF, DOCX, XLSX, PPTX, CSV, TXT, ZIP, PNG и др.), открывающая документ в новом окне;
- дата и время загрузки;
- статус подписи в виде цветной метки («Подписан» – зелёный, «Ожидает» – жёлтый, «Отклонён» – красный);
- кнопка «Открыть» для просмотра деталей и скачивания;
- кнопка «Удалить» (при наведении) для удаления файла из системы с последующим подтверждением.

Ниже списка расположена кнопка «Создать заявление на подписание», запускающая мастер загрузки нового документа и выбора маршрута подписантов.

Логика компонента разделена на три группы задач:

- получение и нормализация данных с сервера;
- интерактивная фильтрация и текстовый поиск по локальному массиву;
- управление действиями пользователя (редактирование, удаление, скачивание) с учётом делового статуса подписи.

Ниже приведены два фрагмента, демонстрирующие наиболее характерные участки кода: вычисление «живого» поднабора документов по выбранному фильтру и строковый обработчик действий в таблице:

```

const [filter, setFilter] = useState("all");
const [query, setQuery] = useState("");

const docs = useMemo(
  () =>
    allDocs.filter(d =>
      (filter === "all" || d.status === filter) &&
      d.name.toLowerCase().includes(query.toLowerCase())
    ),
  [allDocs, filter, query]
);

```

`useMemo` гарантирует, что перефильтрация выполняется лишь при изменении исходного массива либо параметров фильтра, снижая нагрузку на браузер при больших выборках и обеспечивая плавную работу интерфейса.

```

<tbody>
  {docs.map(d => {
    const canEdit = d.status !== "signed";

    return (
      <tr key={d.id}>
        {/* ...ячейки Название, Дата, Маршрут, Статус... */}

        <td>
          <div className={styles.btnRow}>
            <button
              className={` ${styles.btnSm} ${styles.edit}`}
              onClick={() => canEdit && navigate(`/request/${d.id}/edit`)}
              disabled={!canEdit}
              title={
                canEdit
                  ? "Редактировать"
                  : "Нельзя редактировать подписанный документ"
              }
            >
              <Edit size={14}/> Редактировать
            </button>

            <button
              className={` ${styles.btnSm} ${styles.del}`}
              onClick={() => deleteRequest(d.id)}
            >
              <Trash2 size={14}/> Удалить
            </button>
          </div>
        </td>
      </tr>
    );
  })}

```

```

</div>
</td>
</tr>
);
}}
</tbody>

```

#### 4.2.4 Экран «Подписи»

На рисунке 4.6 показан экран «Подписи» – рабочий список документов, ожидающих действия студента.

Верхняя навигационная панель идентична главной: логотип, пункты «Документы / Подписи / Журнал», переключатель темы, индикатор уведомлений и кнопка–профиль с выпадающим меню «Мой профиль / Выход».

Блок «Подлежит подписи» содержит таблицу входящих файлов. Для каждой строки выводятся:

- кликабельное название–ссылка с пиктограммой формата (PDF, DOCX и др.) и стрелкой–загрузкой, что однозначно указывает на возможность скачивания;
- отправитель (деканат, студгородок, руководитель практики – только академические подразделения, доступные студенту);
- крайний срок подписи;
- цветная метка статуса («Ожидает» – жёлтая, «Подписан» – зелёная, «Отклонён» – красная);
- кнопки действий: «Подписать» (утверждение) и «Отказать» (отклонение).

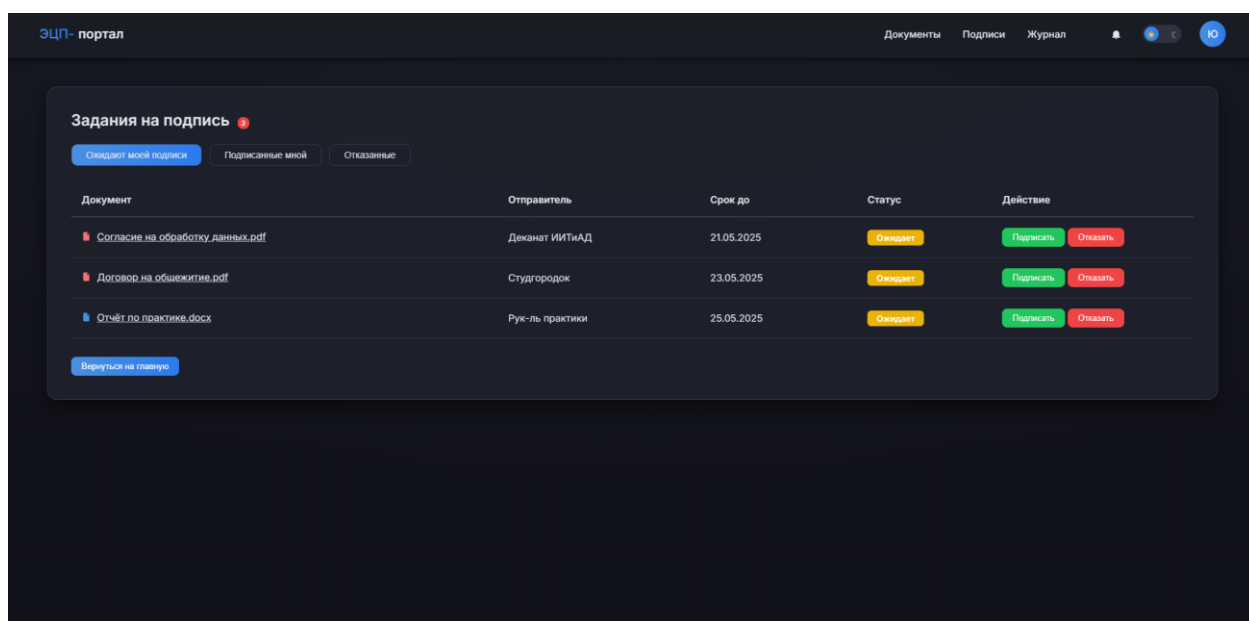


Рисунок 4.6 – Макет экрана «Подписи».

На уровне кода компонент `Signature.jsx` решает три основные задачи:

- получение и нормализация данных – запрос списка входящих заявок и существующих ключей пользователя;

- клиентская фильтрация – мгновенное переключение подмножеств при смене вкладок;

- контекстное действие – встроенная форма, где выбирается ключ, вводится пароль и отправляется запрос на сервер о подписи или отказе.

Два коротких фрагмента ниже демонстрируют реализацию этих ключевых механизмов:

```
const [all, setAll] = useState([]);
const [tab, setTab] = useState("pending");

useEffect(() => {
  fetch("/api/incoming-requests", { headers:{Authorization:`Bearer ${token}`}})
    .then(r => r.json())
    .then(d => setAll(
      (d.tasks||[]).map(t=>({
        id: t.id_request,
        idDoc: t.id_document,
        name: t.original_name,
        sender: t.sender_name,
        date: new Date(t.created_at).toLocaleDateString(),
        status: /подписано/i.test(t.status) ? "signed" :
        /отклонено/i.test(t.status) ? "rejected" : "pending"
      })))
    )
    .catch(()=>setAll([]));
}, [token]);

const tasks = useMemo(
  () => all.filter(t => (tab === "all" ? true : t.status === tab)),
  [all, tab]
);
```

Фрагмент показывает, как данные от сервера превращаются в унифицированный массив и мгновенно фильтруются при смене вкладок.

```
const startSigning = id => {
  if (!keys.length) {
    setGeneralError("Сначала создайте ключ.");
    return;
  }
  setSigningId(id);
  setKeyId(""); setSigPassword(""); setSigError("");
}
```



```

};

const submitSign = async id => {
  if (!keyId) return setSigError("Выберите ключ.");
  if (!signPassword) return setSigError("Введите пароль.");

  try {
    const res = await fetch(`/api/requests/${id}/sign`, {
      method: "POST",
      headers: { "Content-Type": "application/json",
        Authorization: `Bearer ${token}` },
      body: JSON.stringify({ id_key: Number(keyId), password: signPassword })
    });
    if (!res.ok) throw new Error((await res.json()).error || "Ошибка подписи");
    setSigningId(null);
    load();
  } catch (e) { setSigError(e.message); }
};

```

Фрагмент листинга показывает, как строка таблицы динамически подменяется формой выбора ключа и ввода пароля; успешный ответ сервера переводит задачу в статус «Подписано» и исключает её из вкладки «Ожидают». Вместе эти листинги демонстрируют, что экран «Подписи» полностью покрывает рабочий процесс «подписать / отказать» с учётом криптографических требований.

#### 4.2.5 Экран «Создание ЭЦП»

На рисунке 4.7 представлен экран «Мастер создания электронной подписи». Интерфейс структурирован по этапам, что минимизирует ошибки и ускоряет процесс:

- Панель прогресса (слева) визуально выделяет три шага – «Выбор сценария», «Заполнение данных», «Подтверждение». Активный шаг подсвечен синим, пройденные – серым.
- Форма шага 1 содержит радиокнопки: «Сгенерировать ключ» и «Импортировать сертификат». При выборе импорта появляется поле выбора файла и ввод пароля.
- Форма шага 2 включает поля: «Пароль закрытого ключа» и чек–бокс «Сохранить защищённую копию на сервере» см. Рисунок 4.8.
- Информационный блок «Как это работает» располагается справа от формы. В нём пошаговая инструкция и раздел «Частые вопросы» с ответами о хранении ключа и проверке подписи.
- Кнопки управления внизу: «Назад», «Далее / Создать подпись». Кнопка «Создать подпись» активируется только после заполнения всех обязательных полей и согласия с политикой безопасности.

– Строка состояния в верхней части мастера отображает всплывающие уведомления об успехе или ошибке генерации.

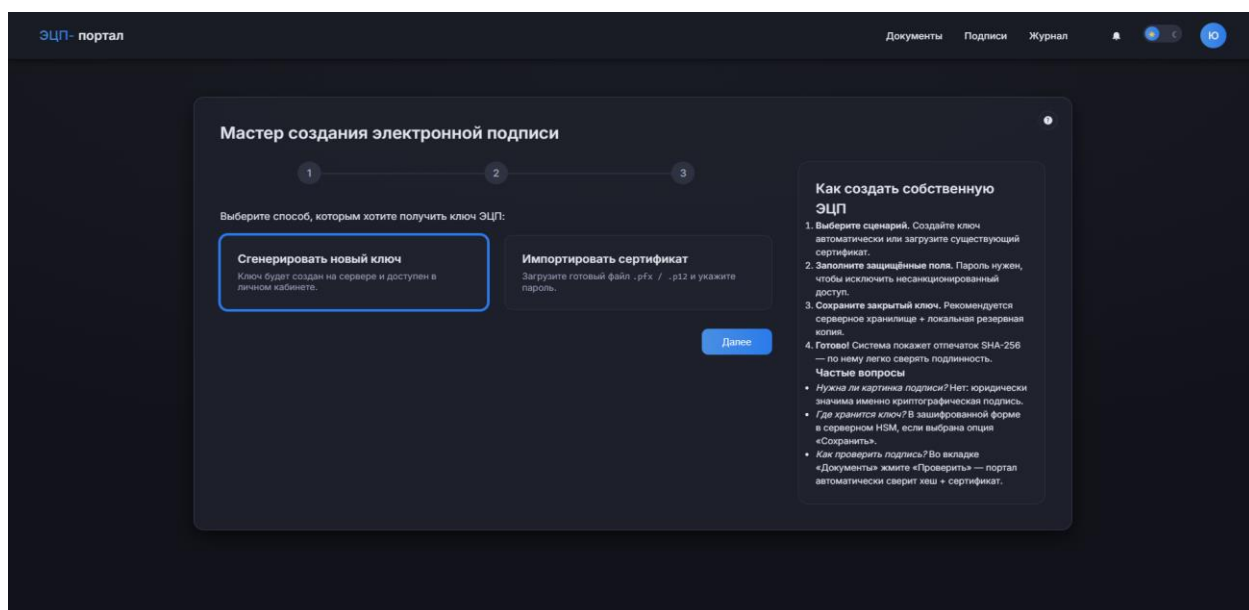


Рисунок 4.7 – Макет экрана «Создание ЭЦП».

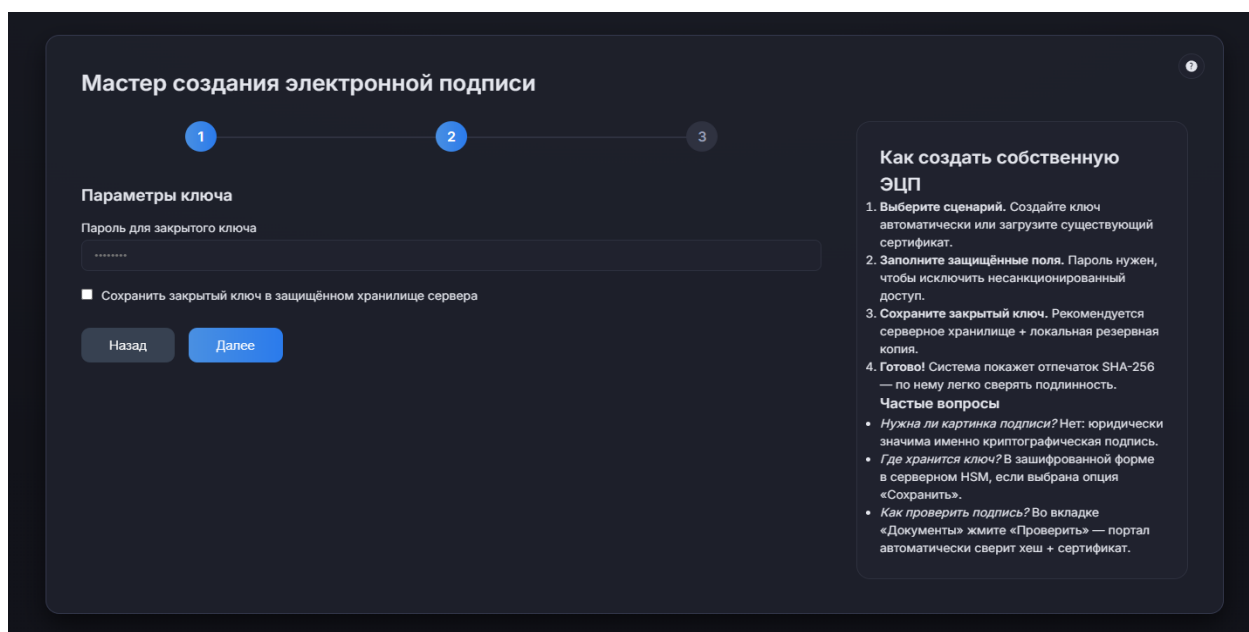


Рисунок 4.8 – Макет «Шаг №2 создания ЭЦП».

Следующий код-фрагмент иллюстрирует, как «Мастер создания электронной подписи» управляет пошаговой навигацией и безопасной генерацией ключевой пары. Логика сведена в два компактных блока: первый отвечает за переключение экранов мастера, второй – за собственно создание ключа с минимальной валидацией паролей пользователя:

```
const [step, setStep] = useState(0);  
const [choice, setChoice] = useState(null);
```

```

const next = async () => {
  if (step === 2 && choice === "generate") {
    const ok = await createKey();
    if (!ok) return;
  }
  if (step === 2 && choice === "import") {
    alert("Импорт появится позже");
    return;
  }
  setStep(s => s + 1);
};

const back = () => setStep(s => Math.max(s - 1, 0));

const createKey = async () => {
  if (pwd !== pwd2) { alert("Пароли не совпадают"); return false; }
  if (pwd.length < 8) { alert("Пароль должен быть ≥ 8 симв."); return false; }

  await fetchJSON(`${API}/api/keys`, {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify({ name: name || null, password: pwd })
  });

  await load();
  return true;
};

```

Переменная `step` определяет, какой экран мастера отображается в данный момент, а функция `createKey` выполняет отправку запроса `POST /api/keys` только после локальных проверок совпадения и длины паролей. При успешном ответе список ключей обновляется без перезагрузки страницы, после чего мастер автоматически переходит на финальный шаг «Ключ создан».

### 4.3 Описание БД

Схема данных макета спроектирована в соответствии с нормальной формой 3NF и реализована в СУБД SQLite 3, что обеспечивает атомарность транзакций и возможность миграции на PostgreSQL без изменения структуры. Логическая модель охватывает три взаимосвязанных домена – идентификация пользователей, управление документами и криптографические события.

Основные таблицы базы данных включают:

1. Таблица `users` – предназначена для хранения информации о пользователях системы (студентах, преподавателях, административных сотрудниках). Каждая запись содержит уникальный идентификатор пользователя, полное имя, адрес электронной почты, пароль, номер группы,

телефон, роль пользователя, идентификатор цифровой подписи, а также информацию о двухфакторной аутентификации.

2. Таблица `groups` – содержит сведения о группах, к которым могут быть отнесены студенты. Записи включают уникальный идентификатор группы и её наименование.

3. Таблица `roles` – фиксирует роли пользователей (например, студент, преподаватель, администратор). Каждая запись состоит из уникального идентификатора роли и её наименования.

4. Таблица `documents` – используется для хранения документов, загружаемых пользователями. Каждая запись включает уникальный идентификатор документа, идентификатор пользователя, оригинальное название файла, путь хранения, тип MIME, размер файла и дату загрузки.

5. Таблица `signature_requests` – предназначена для хранения информации о запросах на подписание документов. Содержит идентификаторы запроса, документа, отправителя, получателя (пользователя или подразделения), статус запроса и даты создания и обновления.

6. Таблица `statuses` – фиксирует статусы запросов (например, «В процессе», «Подписано», «Отклонено»).

7. Таблица `user_keys` – содержит информацию о криптографических ключах пользователей. Записи включают идентификатор ключа, идентификатор пользователя, зашифрованный приватный ключ, публичный ключ, соль для шифрования, наименование ключа, отпечаток и дату создания.

8. Таблица `simple_signatures` – фиксирует простые электронные подписи, используемые в системе. Содержит идентификатор подписи, идентификатор пользователя, пользовательское название, отпечаток и дату создания.

9. Таблица `request_signatures` – используется для хранения информации о подписях, применённых к запросам. Каждая запись содержит идентификатор подписи, запроса, ключа, алгоритм хеширования, хеш документа, электронную подпись, отметку времени и токен TSA.

10. Таблица `audit_log` – предназначена для журналирования действий пользователей, таких как вход в систему, подписание документов или удаление ключей. Содержит идентификатор записи журнала, идентификатор пользователя, IP-адрес, описание действия, метаданные и дату создания записи.

Представленная структура обеспечивает высокий уровень безопасности и целостности данных, упрощает контроль за выполнением процессов подписания и обеспечивает необходимый уровень юридической значимости всех документов, используемых в рамках информационной системы ИРНИТУ.

На рисунке 4.9 представлена физическая модель базы данных, реализованная в СУБД SQLite 3, которая отражает структуру и связи таблиц на уровне физической реализации.

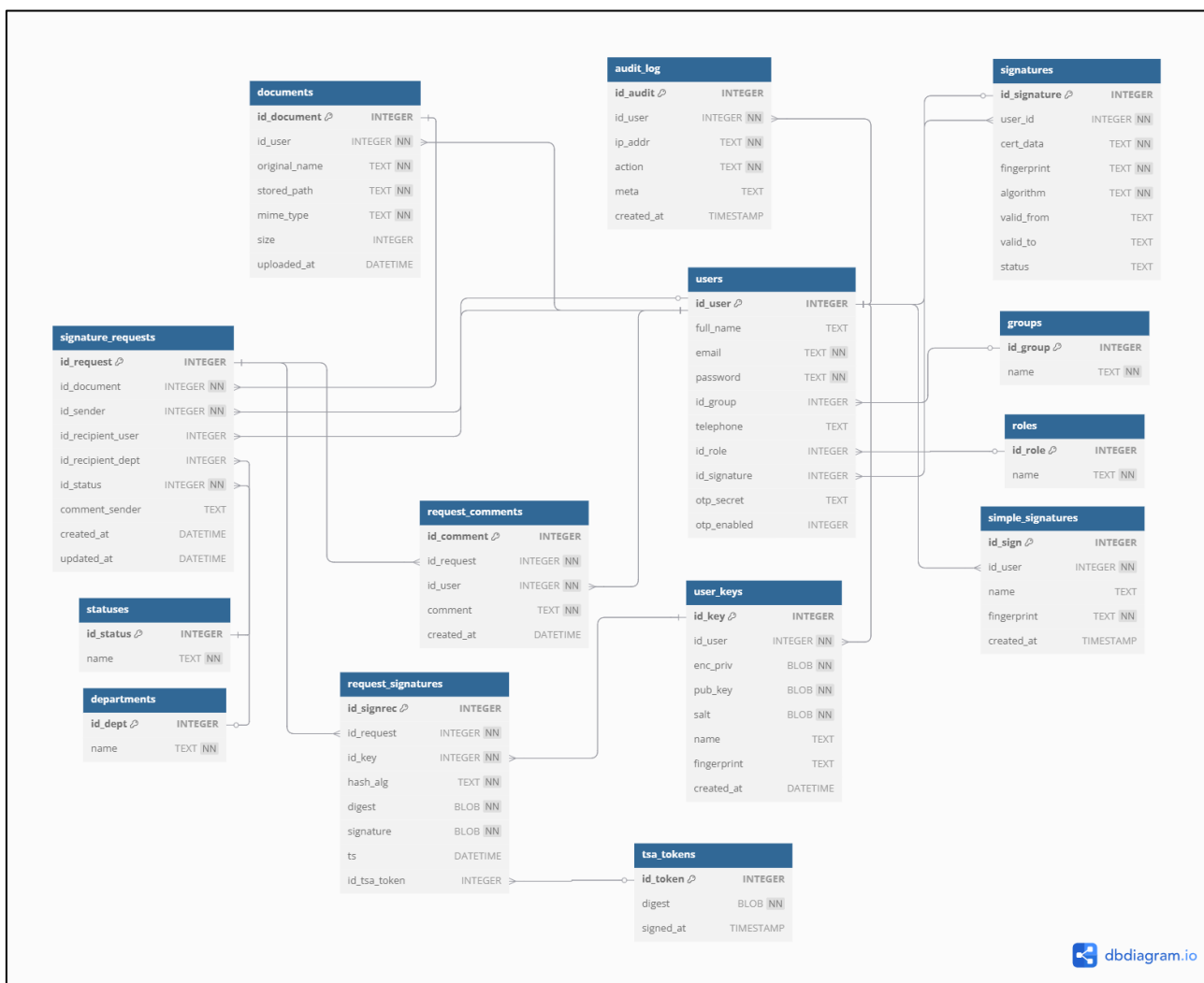


Рисунок 4.9 – Физическая модель базы данных

#### 4.4 Проведение тестирования

В рамках проверки работоспособности и полноты реализованного функционала информационной системы было проведено комплексное тестирование пользовательских сценариев.

Для каждой функциональной операции формировался отдельный сценарий с фиксацией ожидаемых и фактических результатов, а также с обязательной фиксацией ошибок, возникающих в процессе тестирования. По итогам проверки были подготовлены скриншоты интерфейса на различных этапах выполнения сценариев.

Тесты представлены в таблице 4.1.

Таблица 4.1 – Таблица тестов

№	Описание сценария	Ожидаемый результат	Фактический результат
1	Регистрация нового пользователя	Успешная регистрация, переход к авторизации	Пользователь успешно зарегистрирован, переброс на главную страницу. Рисунок 4.10.
2	Авторизация с ошибочным паролем	Сообщение об ошибке авторизации	Выводится ошибка «Неверные данные». Рисунок 4.11.
3	Авторизация с верными данными	Успешный вход, переход в личный кабинет	После успешного входа перекидывает на главную страницу, оттуда переходим в профиль. Рисунок 4.12.
4	Изменение данных в профиле	Уведомление об успешном изменении данных в профиле	Данные успешно обновляются и появляется уведомление. Рисунок 4.13.
5	Создание своей ЭЦП	Удачное создание собственной ЭЦП в системе	Удачное создание собственной ЭЦП в системе. Рисунок 4.14.
5	Загрузка документа	Документ появляется в списке пользователя	Документ успешно прикрепляется и отображается в системе. Рисунок 4.15 и Рисунок 4.16.
6	Загрузка документа без подписи.	Система не даст создать заявление и выдаст ошибку	Система не даст создать заявление и выдаст ошибку. Рисунок 4.17.
7	Отправка документа на подпись	Создаётся заявка, отображается в разделе «подписи» у человека, кому отправили документ	Заявление отображается, есть возможность скачать файл для просмотра, а также кнопки «Подписать» или «Отказать». Рисунок 4.18 и Рисунок 4.19.

Продолжение таблицы 4.1

№	Описание сценария	Ожидаемый результат	Фактический результат
8	Получение и подписание заявки	Статус заявки меняется на «Подписано»	Заявление успешно подписано, статус заявки поменялся на подписано. Рисунок 4.20.
9	Отклонение заявки	Статус меняется на «Отклонено»	Создана новая заявка, которую в последствии отклонили. Рисунок 4.21 и Рисунок 4.22.
10	Удаление существующей подписи	Удаление подписи прошло успешного, в списке подписей ничего не отображается.	Удаление подписи прошло успешного, в списке подписей ничего не отображается. Рисунок 4.23 и Рисунок 4.24.
11	Включение и отключение 2ФА	Отображается статус 2ФА, невозможность повторного включения	Отображается статус 2ФА, невозможность повторного включения. Рисунок 4.25 и Рисунок 4.26.
12	Проверка журналирования действий	Операции отражаются в журнале аудита	Операции отражаются в журнале аудита. Рисунок 4.27.

The registration form is titled "Регистрация" in blue. It contains four input fields: "E-mail" with the value "123@mail.ru", "Полное ФИО" with the value "Колупав Вячеслав Андреевич" highlighted in yellow, "Пароль" with three dots, and "Повторите пароль" with three dots. Below the fields are two buttons: a blue "Создать" button and a red "Назад" button.

Рисунок 4.10 – Процесс регистрации нового пользователя.

Пользователь заполняет обязательные поля формы и нажимает кнопку «Зарегистрироваться». Система обрабатывает введённые данные, создаёт учётную запись и мгновенно переводит пользователя на страницу авторизации, что подтверждает успешное завершение сценария регистрации.

The login form is titled "Вход" in blue. It displays an error message "Неверные данные." in red above the "E-mail" field, which contains "123@mail.ru". The "Пароль" field contains two dots. Below the fields is a checkbox labeled "Запомнить меня" and a blue "Войти" button. At the bottom, there is a link "Забыли пароль? / Регистрация" in blue.

Рисунок 4.11 – Ошибка при неверном пароле или логине.



После ввода заведомо некорректной пары «логин/пароль» система отклоняет попытку входа и отображает уведомление «Неверные данные». Отсутствие перехода на защищённые разделы и появление сообщения об ошибке подтверждают корректность обработки отрицательного сценария авторизации.

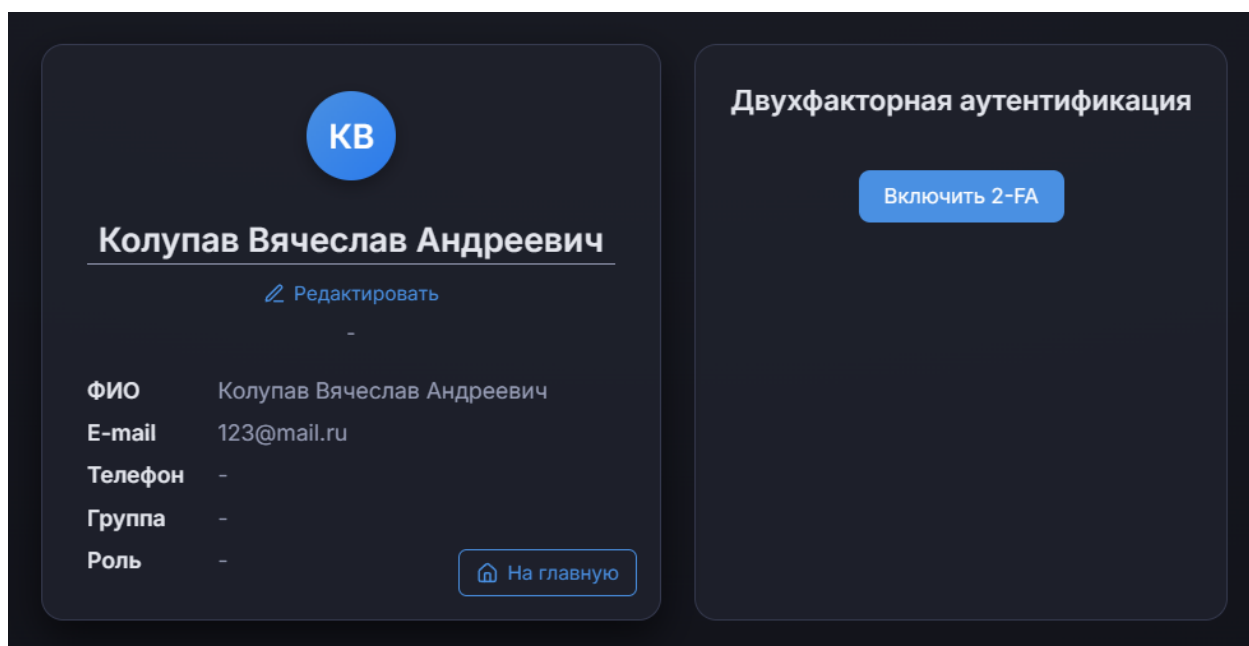


Рисунок 4.12 – Меню профиля при успешной авторизации.

При вводе корректных учётных данных пользователь попадает на главную страницу, откуда открывает выпадающее меню профиля. Отображение персонального раздела и доступ к настройкам подтверждают, что сессия установлена, а сценарий успешного входа выполнен без ошибок.

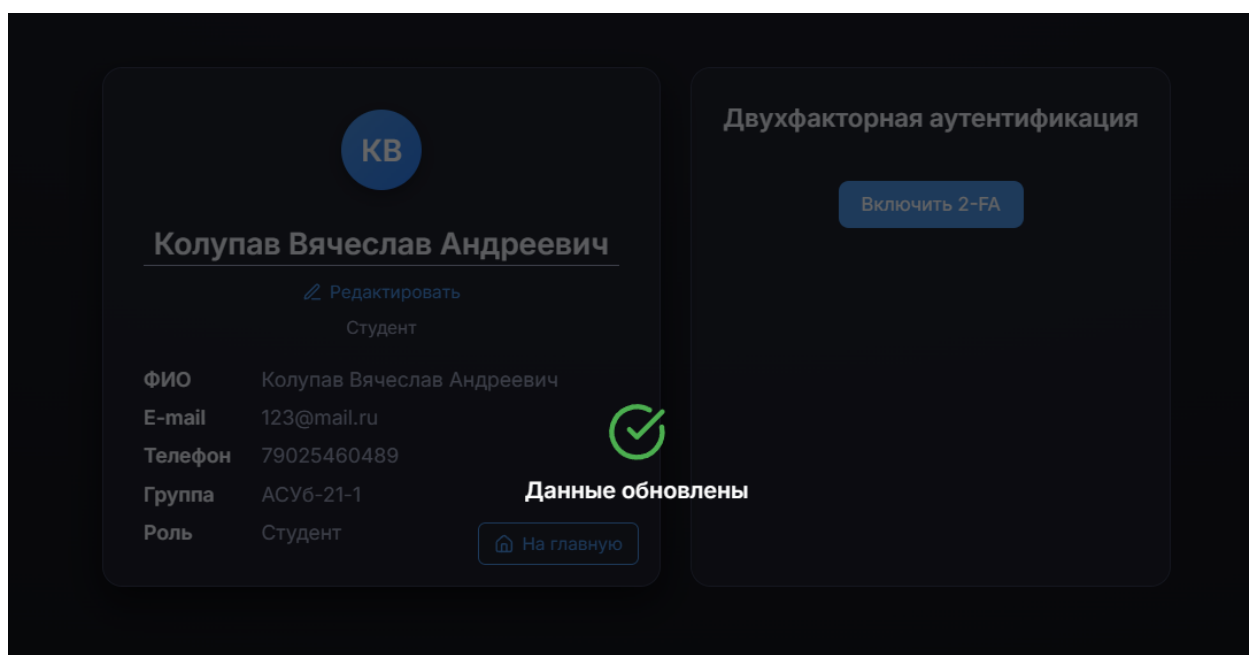


Рисунок 4.13 – Уведомление об успешном изменении данных.

После редактирования контактной информации в профиле пользователь нажимает кнопку «Сохранить». В правом верхнем углу появляется всплывающее уведомление зелёного цвета – «Данные успешно обновлены». Одновременное обновление значений в форме и вывод подтверждающего сообщения свидетельствуют, что операция записи в БД выполнена корректно и обработчик профиля прошёл тест без ошибок.

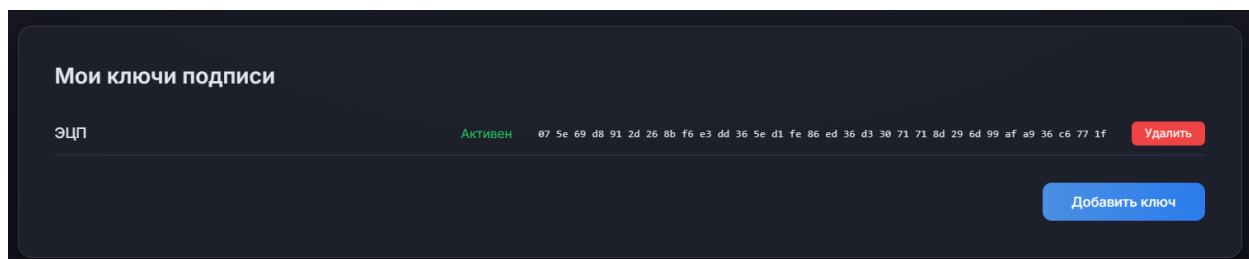


Рисунок 4.14 – Успешное создание и отображение созданной ЭЦП.

Мастер генерации ключа завершается шагом «Ключ создан». Сразу после этого в списке подписей появляется новая запись с заданным именем и отпечатком ключа, статус помечен как «Активен». Это подтверждает, что сервис /api/keys вернул валидный идентификатор, а клиентское приложение корректно освежило локальный список без перезагрузки страницы.

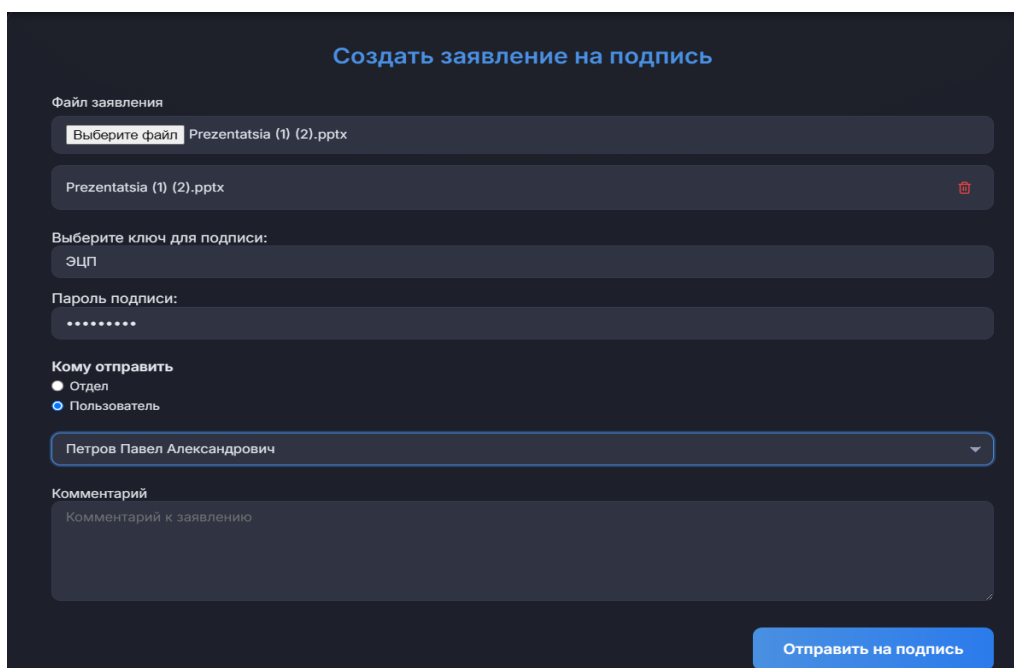


Рисунок 4.15 – Процесс создания заявления на подпись.

Пользователь выбирает файл, указывает адресата и инициирует создание заявки. На экране отображается форма предварительного просмотра: имя файла, маршрут и кнопка «Отправить». Появление этой формы показывает, что предварительная валидация документа прошла успешно и запись в таблице requests создана, но ещё не отправлена.

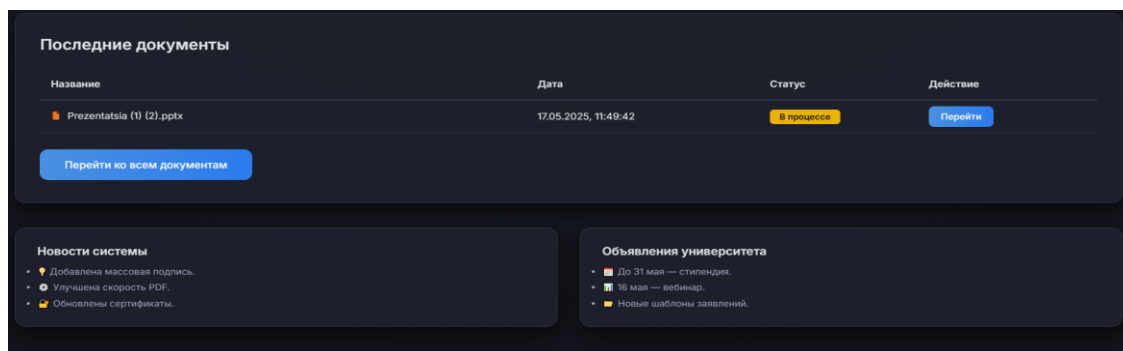


Рисунок 4.16 – Отображения документа в системе.

Сразу после подтверждения отправки новый документ появляется в реестре пользователя со статусом «Ожидает». В строке видна иконка формата, дата загрузки и кнопка «Перейти», ведущая к детальной карточке. Это доказывает, что файл успешно загружен в хранилище, а вновь созданная заявка корректно связана с загруженным объектом и отображается в пользовательском интерфейсе.

**Создать заявление на подпись**

Файл заявления

Выберите файл 2.2 Отчет, дневник и характеристика, практика в структурном подразделении ИРНИТУ (2).docx

2.2 Отчет, дневник и характеристика, практика в структурном подразделении ИРНИТУ (2).docx

Выберите ключ для подписи:

-- выберите --

Пароль подписи:

Пароль от закрытого ключа

Выберите ключ для подписи.

Кому отправить

☐ Отдел

☒ Пользователь

Петров Павел Александрович

Комментарий

Рисунок 4.17 – Уведомление пользователя, что он не подписал документ при создании заявления.

Пользователь пытается отправить файл без наложения собственной подписи. При нажатии кнопки «Отправить» система блокирует операцию и выводит предупреждающее уведомление красного цвета: «Выберите ключ для подписи.». Отказ в сохранении подтверждает, что валидатор корректно отслеживает обязательность подписи и предотвращает появление неправильных записей в базе.

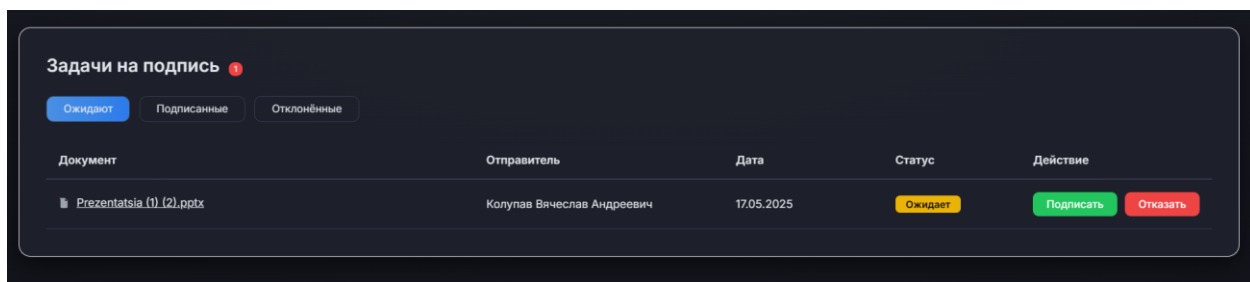


Рисунок 4.18 – Заявления на подпись.

После того как инициатор успешно сформировал подписанное заявление, запись появилась во вкладке «Подписи» у адресата. В строке видны название файла, имя отправителя, крайний срок и статус «Ожидает». Наличие кнопок «Подписать» и «Отказать» подтверждает, что маршрут доставлен, а бизнес-правило «ожидается действие второй стороны» выполнено.

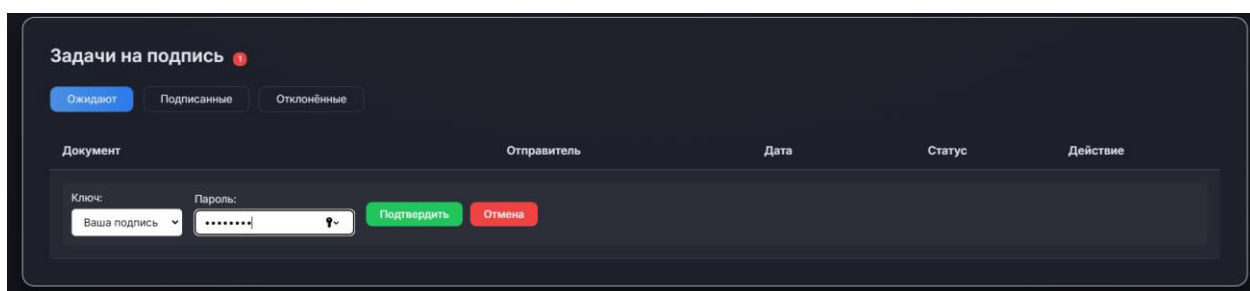


Рисунок 4.19 – Процесс подписания документа.

Адресат нажимает «Подписать», выбирает свой ключ, вводит пароль и подтверждает действие. Встроенная форма исчезает, а всплывающее сообщение «Документ подписан» подтверждает успешную серверную проверку пароля и фиксацию второй сигнатуры. Одновременно счётчик «Ожидают» уменьшается, что демонстрирует обновление клиентского состояния в реальном времени.

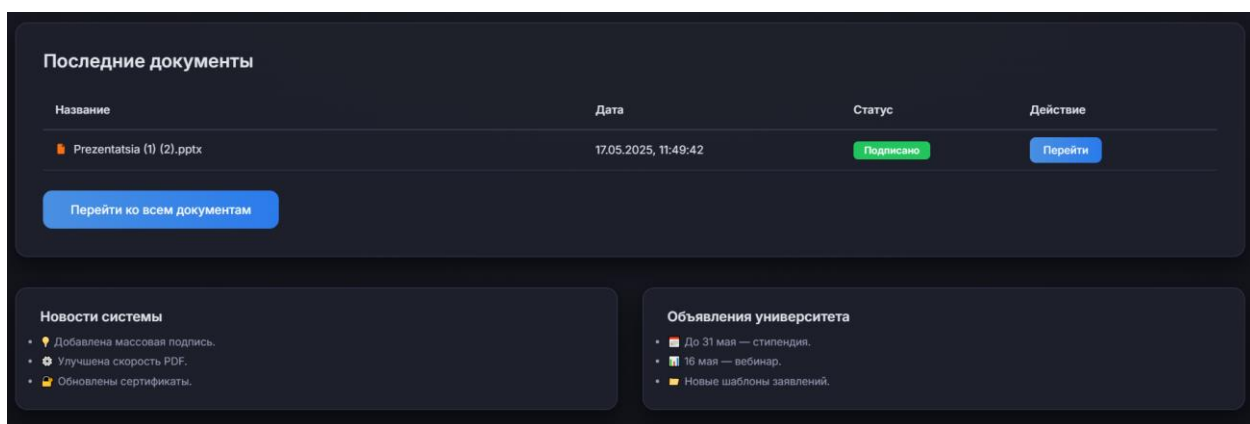


Рисунок 4.20 – Отображение в системе успешно подписанного заявления.

Вернувшись в раздел «Документы», пользователь видит, что статус только что обработанного файла изменился с жёлтого «Ожидает» на зелёный «Подписан». Тем самым система наглядно подтверждает: обе подписи

зафиксированы в базе, журнал аудита обновлён, а заявка автоматически исключена из списка задач на подпись.

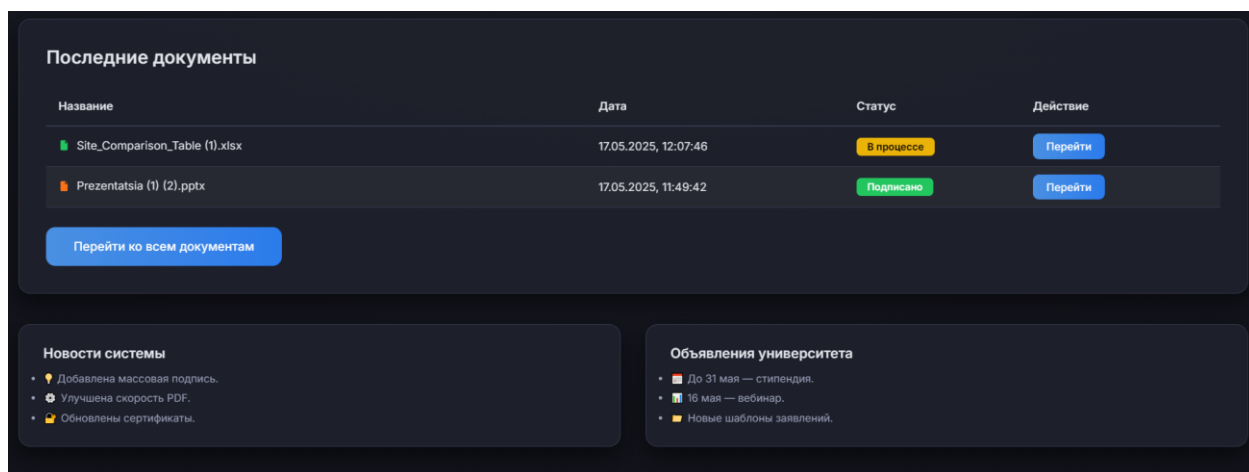


Рисунок 4.21 – Создание нового заявления.

Сразу после отправки файла строка нового заявления появляется в разделе «Документы» инициатора. В колонках видны: сам файл, текущий статус «Ожидает» и дата создания. Появление записи подтверждает, что запрос сохранён в базе и корректно отобразился на панели управления без перезагрузки страницы.

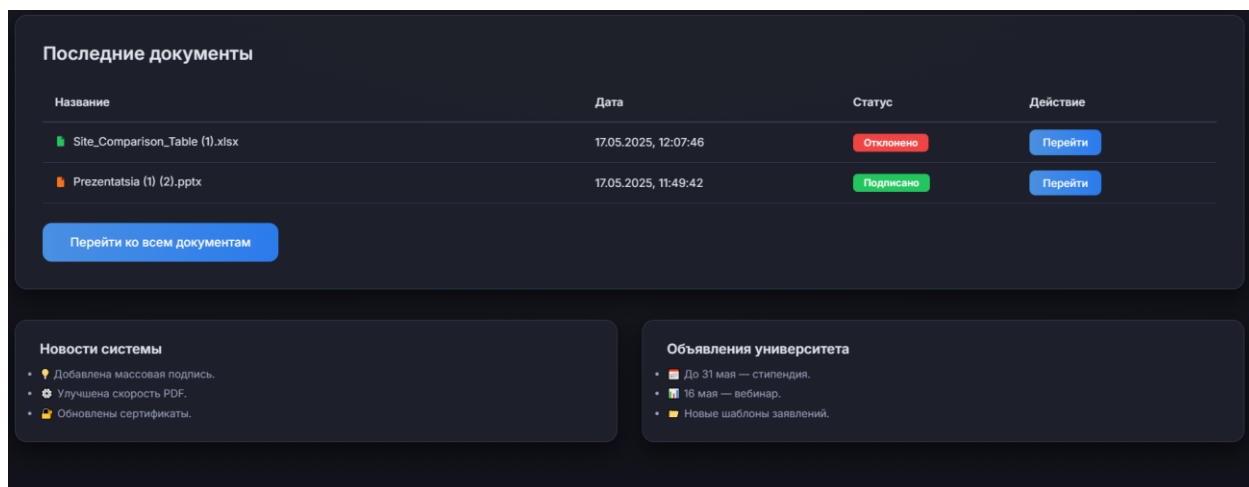


Рисунок 4.22 – Отображение отклоненного заявления.

После того как адресат нажал «Отказать», инициатор обновляет раздел «Документы» и видит ту же строку, окрашенную красным, со статусом «Отклонено». Изменение цвета и текста статуса показывает, что ответ адресата был получен, сохранён и немедленно отражён в пользовательском интерфейсе. Это подтверждает корректность полного цикла «создать → отклонить» и синхронизацию клиентской таблицы с серверным состоянием.

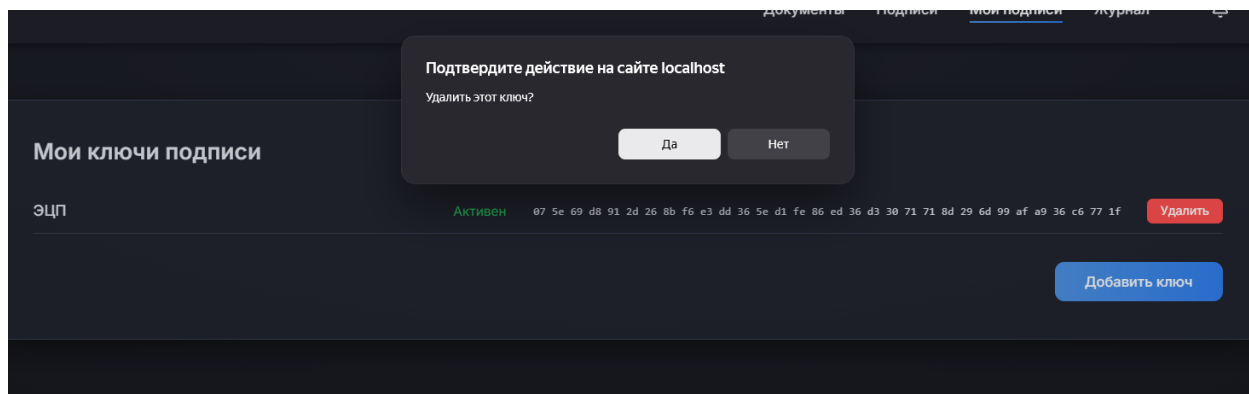


Рисунок 4.23 – Подтверждение удаления подписи.

Перед удалением собственного ключа пользователь нажимает кнопку «Удалить» в списке подписей. Система выводит модальное окно-предупреждение с подробным текстом о том, что удалённый ключ невозможно восстановить, а все документы, подписанные им, останутся в архиве без возможности повторного использования этой подписи. Диалог содержит две явно различимые кнопки – «Удалить» (окрашена в красный) и «Отмена» (нейтральный серый), что снижает риск случайного нажатия и соответствует требованиям UX для операций, влияющих на безопасность.

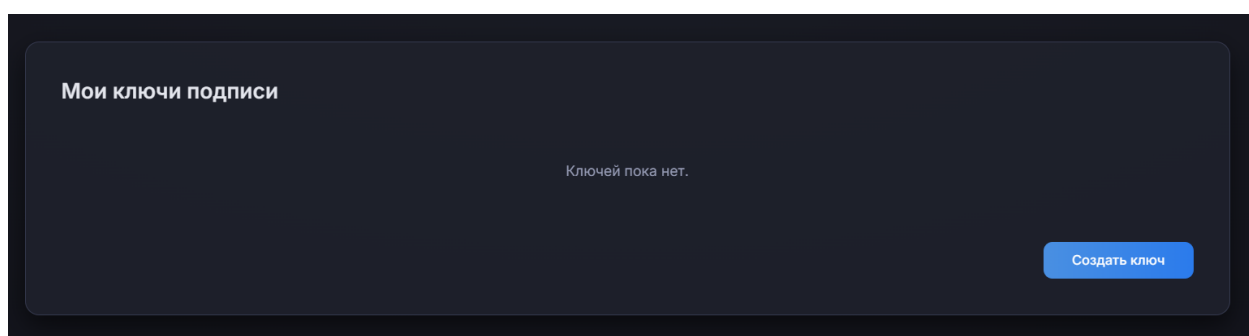


Рисунок 4.24 – Список подписей после удаления.

После подтверждения операция завершается на сервере: запись о ключе удаляется из таблицы keys, привязанные сертификаты аннулируются, а клиент получает статус 200 ОК. Компонент немедленно выполняет повторный запрос и перерисовывает интерфейс; вместо списка отображается сообщение «Ключей пока нет», а индикатор состояния меняется на «0 подписей». Отсутствие ключа в таблице и в счётчике подтверждает, что процесс удаления отработал корректно как на уровне базы данных, так и в пользовательском интерфейсе, завершая тестовый сценарий успешным результатом.

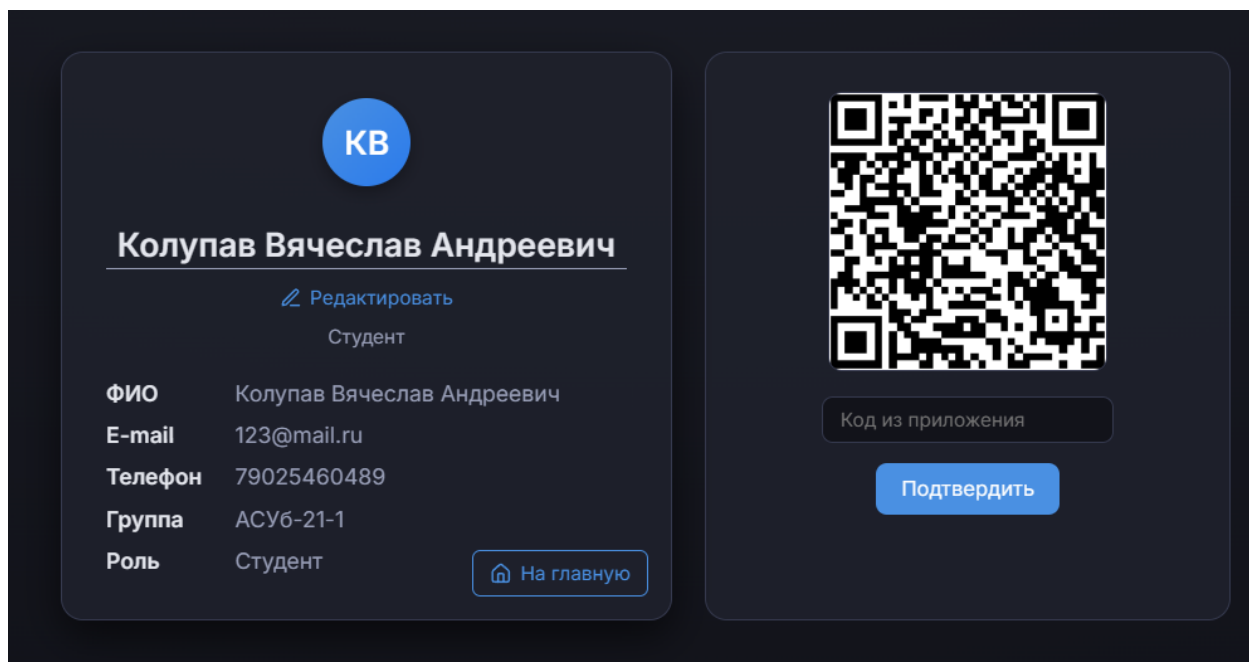


Рисунок 4.25 – Процесс подключение 2FA.

На экране настройки безопасности пользователь инициирует подключение двухфакторной аутентификации: система генерирует QR-код для приложения-аутентификатора и отображает поле ввода одноразового кода. После сканирования QR-кода в мобильном приложении пользователь вводит шестизначный OTP и нажимает «Подтвердить». Появление всплывающего сообщения «Двухфакторная защита активирована» фиксирует успешную серверную проверку кода и сохранение секретного ключа в учётной записи.

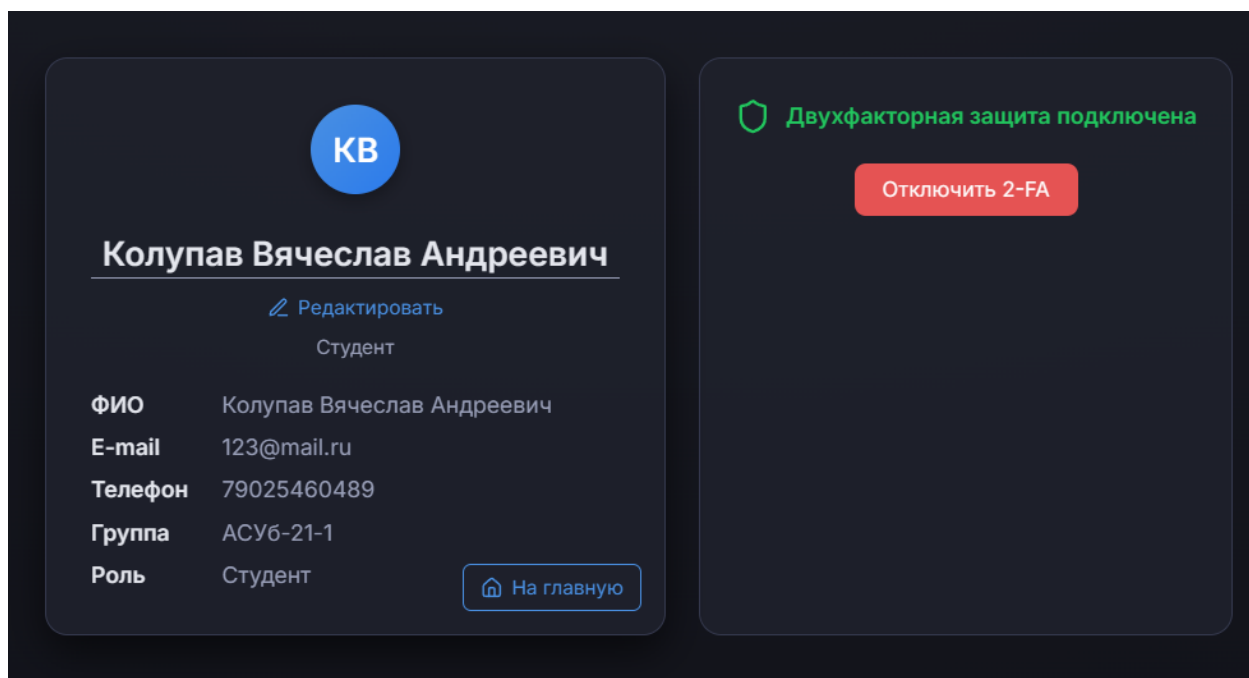


Рисунок 4.26 – Отображение в системе об успешном подключении 2FA.

Страница безопасности автоматически перезагружается и показывает новый статус: индикатор 2FA окрашен зелёным, рядом выводится пометка «Включено». Кнопка «Включить» исчезает, остаётся только опция «Отключить», причём попытка повторного включения блокируется. Таким образом интерфейс подтверждает, что функциональность двухфакторной аутентификации активна и повторное включение недоступно.

ID	Дата	Кто	Действие	OK?	Meta
10	2025-05-17 12:10:55	Колупав Вячеслав Андреевич	DELETE_KEY		{ "id_key": "14" }
1000029	2025-05-17 12:04:53	Петров Павел Александрович	SIGN_DOCUMENT	✓	{ "document": "Prezentatsia (1) (2).pptx", "requestid": 47, "fingerprint": "80 bc 10 98 f6 84 53 34 b2 24 fd ad 17 45 5f 83 53 9c e6 59 2f 19 f8 90 ef c4 4c e0 5d 3d eb 1c", "digest_ok": true }
1000027	2025-05-13 12:03:50	Золотарева Ангелина Александровна	SIGN_DOCUMENT	✗	{ "document": "Site_Comparison_Table (1).xlsx", "requestid": 46, "fingerprint": "cb 02 d7 cd bc cc fd 4d 5b e5 e1 64 66 39 63 8f 79 bc ec cb 3a 3f 0f 6e 8f 10 6e 03 1c 02 16 15", "digest_ok": false }
9	2025-05-11 13:42:40	Колупав Вячеслав Андреевич	DELETE_KEY		{ "id_key": "12" }
1000023	2025-05-11 13:40:31	Колупав Вячеслав Андреевич	SIGN_DOCUMENT	✓	{ "document": "Prezentatsia (1).pptx", "requestid": 44, "fingerprint": "35 d0 5a c2 bc d3 a5 27 11 b5 f4 34 7d 54 58 cd 59 38 44 04 fa 13 d3 ad e3 9d f1 ce 78 2f 5b 9f", "digest_ok": true }
1000024	2025-05-11 13:40:31	Петров Павел Александрович	SIGN_DOCUMENT	✓	{ "document": "Prezentatsia (1).pptx", "requestid": 44, "fingerprint": "80 bc 10 98 f6 84 53 34 b2 24 fd ad 17 45 5f 83 53 9c e6 59 2f 19 f8 90 ef c4 4c e0 5d 3d eb 1c", "digest_ok": true }
1000021	2025-05-11 08:45:54	Колупав Вячеслав Андреевич	SIGN_DOCUMENT	✓	{ "document": "Site_Comparison_Table.xlsx", "requestid": 43, "fingerprint": "35 d0 5a c2 bc d3 a5 27 11 b5 f4 34 7d 54 58 cd 59 38 44 04 fa 13 d3 ad e3 9d f1 ce 78 2f 5b 9f", "digest_ok": true }

Рисунок 4.27 – Отображение действий на сайте на вкладке «Журнал».

Журнал событий, представленный на рисунке 4.27, обеспечивает централизованную фиксацию всех значимых действий, выполняемых в системе электронного документооборота. Для каждой операции автоматически создаётся запись с уникальным идентификатором, меткой точного времени, фамилией и инициалами инициатора, кодом действия и бинарным показателем успешности. Дополнительные сведения сохраняются в компактном JSON-блоке: здесь отражаются идентификаторы документов и ключей, отпечаток используемого ключа, результат проверки целостности файла, а также прочие технические атрибуты, необходимые для последующего расследования инцидентов. Такое сочетание читаемых полей и структурированных метаданных одновременно удовлетворяет требованиям нормативов к обязательному аудиту, укрепляет юридическую доказательность подписей и упрощает эксплуатационный мониторинг: администратору достаточно беглого взгляда, чтобы оценить состояние операций по визуальным индикаторам «успех / ошибка», а при необходимости он может развернуть JSON-детали и проследить полную цепочку



событий. Проведённое тестирование подтвердило, что каждое действие пользователя моментально регистрируется, а журнал надёжно отражает реальное состояние системы.

#### **4.5 Анализ результатов тестирования**

Проведённое тестирование реализованного прототипа подтвердило работоспособность и соответствие основным требованиям, заявленным на этапе проектирования. Все ключевые пользовательские сценарии, включая регистрацию, авторизацию, управление профилем, создание и использование электронной подписи, загрузку и подписание документов, а также применение двухфакторной аутентификации, были проверены в реальных условиях.

По результатам тестирования:

- Корректность выполнения операций. Все базовые функции системы работают согласно спецификации. Ошибки при вводе некорректных данных (например, неправильный логин или пароль) корректно обрабатываются с информированием пользователя.

- Целостность и доступность данных. Загруженные документы и созданные подписи доступны только авторизованным пользователям. Осуществлён контроль доступа, что исключает возможность просмотра или модификации чужих данных.

- Надёжность операций с ключами. Процедуры генерации, хранения и удаления ключей прошли без критических сбоев. Все операции с ключами корректно отображаются в пользовательском интерфейсе и журнале событий.

- Управление заявками. Система корректно обрабатывает процессы подачи, подписания и отклонения заявок. Изменения статусов заявок корректно отражаются в интерфейсе и сопровождаются уведомлениями.

- Двухфакторная аутентификация. Механизм 2FA функционирует стабильно. Включение и отключение защиты выполняется по запросу пользователя, повторное подключение до отключения недоступно.

- Журналирование. Все основные действия пользователей фиксируются в журнале аудита, что обеспечивает прозрачность и возможность контроля над системой.

Выявленные замечания и направления для доработки:

- В процессе тестирования существенных критических ошибок и уязвимостей не обнаружено.

- Для дальнейшего повышения надёжности рекомендуется реализовать автоматизированное тестирование (юнит-тесты, тесты интеграции).

- Необходимо предусмотреть расширенные механизмы восстановления паролей и более гибкую систему разграничения прав доступа для различных ролей пользователей.

- Для промышленных сценариев внедрения целесообразно предусмотреть интеграцию с корпоративными каталогами пользователей и внешними удостоверяющими сервисами.

## **5 Рекомендации по дальнейшему внедрению ЭЦП**

### **5.1 Предложения по интеграции приложения в процессы университета**

На основе положительных результатов тестирования, представленных в разделе 4.4, предлагается развернутая программа поэтапного внедрения электронной цифровой подписи в информационную инфраструктуру ИРНИТУ. Учитывая текущую архитектуру университетских ИС, уровень цифровой зрелости административного персонала и специфику документооборота, интеграция должна происходить в несколько логически обоснованных шагов с учетом организационных, технических и правовых факторов.

В первую очередь необходимо инициировать пилотное внедрение решения в одном из структурных подразделений с высоким объемом повторяющихся административных операций – например, в учебной части института ИТиАД. Это позволит провести практическую обкатку ключевых функций: регистрации и управления подписями, подписания типовых заявлений, отображения статусов заявок, а также анализа журналов активности. Обратная связь, полученная на этом этапе, ляжет в основу корректировки интерфейса и сценариев использования.

Параллельно с пилотированием следует начать разработку универсального REST API, который обеспечит взаимодействие системы ЭЦП с существующими цифровыми платформами ИРНИТУ (Moodle, 1С:Университет, Тандем, Апекс). При этом особое внимание должно быть уделено унификации форматов данных (JSON, XML), поддержке сигнатур CAdES-BES и интеграции с LDAP для верификации подписей.

Обязательным условием для устойчивого внедрения является разработка и утверждение нормативного регламента, определяющего следующие ключевые аспекты: – перечень допустимых форм ЭЦП в зависимости от сценариев (ПЭП, УЭП); – ответственность пользователей за ключи и порядок их восстановления; – процедуры отзыва подписи, журналирования действий и технической поддержки.

Важно правильно технически подключить ЭЦП к ИС и при этом обучить персонал. Поэтому следует провести цикл обучающих семинаров и вебинаров с демонстрацией наглядных сценариев, выпустить короткие видеоролики и PDF–инструкции по генерации ключей, подаче заявления и просмотру подписанных файлов. Также рекомендуется внедрить контекстную справку непосредственно в пользовательском интерфейсе.

Для постепенной миграции на цифровой документооборот необходимо выстроить приоритеты:

- внутренние обращения студентов;
- ведомости успеваемости и протоколы кафедр;
- служебные записки, справки, внутренние приказы;
- приказы ректора и отчёты для внешнего документооборота (с возможным переходом к квалифицированной подписи);

– контракты с внешними организациями.

Рекомендуется использовать следующую схему внедрения, иллюстрирующую технологическую и организационную последовательность (см. рисунок 4.28):

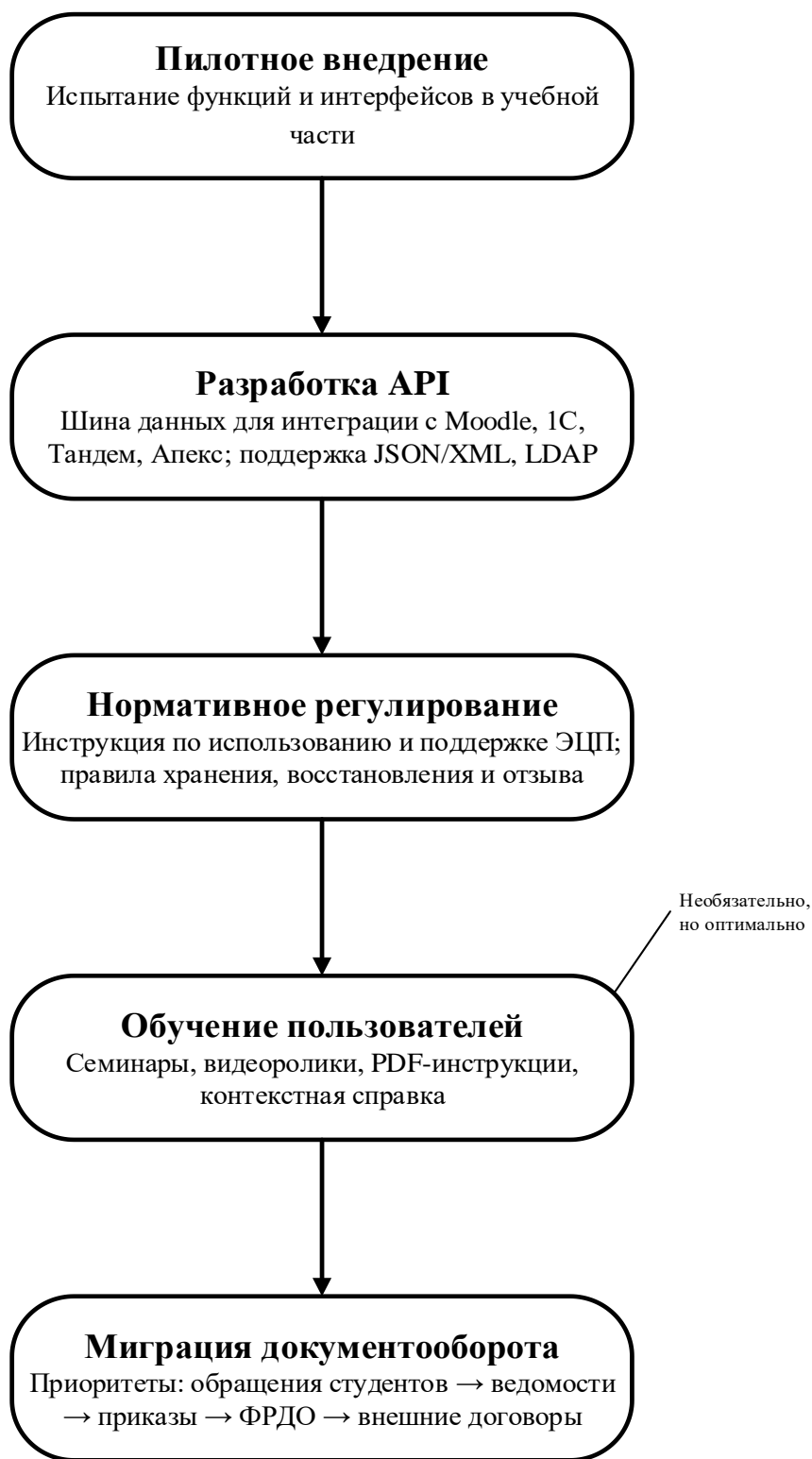


Рисунок 4.28 – Схема по внедрению работы ЭЦП в бизнес–процессы ИРНИТУ

На каждом этапе должна фиксироваться метрика успешности: количество созданных подписей, обработанных заявок, удовлетворенность пользователей, количество инцидентов безопасности. При этом управление внедрением должно строиться по принципу «цифровой зрелости», с предварительным аудитом ИТ–инфраструктуры и регулярным пересмотром целевых показателей.

В перспективе целесообразно рассмотреть возможность подключения к внешним удостоверяющим центрам для генерации квалифицированных подписей при работе с государственными системами. Также необходима проработка взаимодействия с корпоративными каталогами пользователей, внедрение мультифакторной аутентификации (в том числе через ЕСИА), резервное хранение критически важных подписей в HSM–устройствах или аппаратных токенах.

Переход к полноценной цифровизации университетского документооборота требует не только технической интеграции системы ЭЦП, но и параллельного выстраивания организационных процедур. Важно учитывать, что при внедрении новых технологий часто возникают трудности на стыке между ИТ-отделом и конечными пользователями — преподавателями и студентами. Для сглаживания переходного периода необходимо организовать информационную кампанию, разъясняющую преимущества ЭЦП для всех участников процесса, а также создать систему обратной связи для оперативного устранения возникающих вопросов.

Рекомендуется сформировать рабочую группу из представителей ключевых подразделений университета, которая возьмёт на себя функции тестирования, координации пилотных внедрений и анализа первых результатов. Такой подход не только ускорит выявление «узких мест», но и обеспечит лояльность сотрудников, поскольку они будут вовлечены в процесс принятия решений и смогут непосредственно влиять на развитие новой платформы.

## **5.2 Перспективы дальнейшего развития и масштабирования системы**

С учетом потенциала и архитектурной гибкости реализованного решения, возможно стратегическое развитие системы в ряде ключевых направлений. Эти меры направлены на расширение функциональности, повышение безопасности, удобства использования и готовности системы к промышленному масштабу.

Во–первых, одним из приоритетных шагов может стать внедрение поддержки квалифицированной электронной подписи (КЭП), что потребует интеграции с доверенными удостоверяющими центрами, аккредитованными в соответствии с требованиями ФЗ–63. Это обеспечит юридическую значимость подписей при взаимодействии с внешними контрагентами и государственными системами (например, ФРДО, ГИС «Образование»).

Следующим направлением выступает разработка мобильных версий приложения, адаптированных для iOS и Android. Это позволит студентам и преподавателям подписывать документы, получать уведомления и управлять своими ключами в любое время с мобильных устройств. Также целесообразно

рассмотреть кроссплатформенные подходы (например, с использованием Flutter или React Native) для унификации пользовательского опыта.

Для повышения оперативности и прозрачности документооборота необходимо реализовать модуль автоматического контроля сроков: подписей, отклонений, напоминаний о действиях. В частности, система должна отправлять уведомления при приближении дедлайна, а также отображать динамические индикаторы срочности в интерфейсе (например, через цветовую маркировку или иконки).

Отдельного внимания требует отказоустойчивость и масштабируемость: при росте количества пользователей и объема операций критично переходить от SQLite к более производительным СУБД (например, PostgreSQL с репликацией или кластер MySQL), а саму систему развернуть в облачной инфраструктуре с резервным копированием, автоматическим масштабированием и мониторингом.

Механизмы безопасности должны быть усилены за счёт внедрения:

- централизованной системы журналирования (например, ELK Stack);
- интеграции с SIEM-системами университета (если применимо);
- регулярного проведения аудитов (внутреннего и внешнего);
- отслеживания аномалий поведения пользователей.

Также необходимо предусмотреть:

- возможность делегирования прав (например, для кураторов групп);
- расширение ролей и сценариев маршрутизации заявок;
- создание панели администратора для визуального контроля над системными событиями и пользователями;
- формирование отчётов по активности, загрузке системы, ключевым метрикам безопасности и производительности;
- поддержку сценариев массовой подписи (batch-signing);
- интеграцию с внешними СЭД (системами электронного документооборота) через универсальные коннекторы.

Развитие пользовательского интерфейса также должно сопровождаться внедрением адаптивного дизайна, тёмной темы, доступности для людей с ОВЗ и возможностью персонализации отображения данных (настройка виджетов, сортировка по приоритетам).

Так же стоит сказать, что система, спроектированная с учётом масштабируемости и открытых стандартов, может быть дополнительно интегрирована с внешними государственными сервисами (например, «Госуслуги», ЕСИА) и внутренними платформами (портал для выпускников, корпоративная почта и пр.). Внедрение расширенных механизмов автоматизации (например, модульных очередей заданий, интеллектуального анализа ошибок подписей и уведомлений) позволит оптимизировать ресурсы и сделать сервис максимально надёжным для всех пользователей.

В перспективе, в систему можно встроить элементы машинного обучения, которые будут автоматически выявлять аномальные сценарии использования подписей или прогнозировать типичные ошибки пользователей. Такой подход

позволит не только повысить уровень безопасности, но и значительно улучшить пользовательский опыт за счёт более точных подсказок и рекомендаций.

### **5.3 Организационные и административные барьеры при внедрении ЭЦП**

Несмотря на явные технологические преимущества внедрения электронной подписи в университетской среде, именно организационные и административные барьеры становятся ключевым фактором, замедляющим цифровую трансформацию документооборота:

1. Одной из основных проблем выступает инерция устоявшихся административных процессов. Большинство сотрудников и руководителей подразделений привыкли к традиционному бумажному документообороту, где множество согласований, ручных подписей и физического контроля над документами воспринимаются как гарантия надёжности и управляемости. Переход на электронные подписи требует ломки многолетних привычек и, зачастую, вызывает скрытое или открытое сопротивление со стороны персонала.

2. Отдельной сложностью становится недостаточный уровень цифровой грамотности части сотрудников и обучающихся. В условиях, когда не все пользователи уверенно работают с цифровыми сервисами, внедрение ЭЦП может приводить к ошибкам при работе с электронными документами, низкому доверию к результатам электронных процедур и формальному игнорированию новых инструментов.

3. Существенным барьером также выступает необходимость регламентации всех новых процедур, связанных с ЭЦП. Требуется разработать, утвердить и внедрить локальные нормативные акты, подробно определяющие порядок подписания, хранения, проверки и отзыва электронных подписей, а также распределение ответственности между пользователями, администраторами и службой информационной безопасности.

4. В ряде случаев препятствием становится страх утраты личного или административного контроля над процессами согласования и утверждения документов. При переходе к автоматизированным маршрутам и централизованному контролю за электронной подписью руководители подразделений могут опасаться снижения своего влияния, что проявляется в пассивном или активном саботаже внедрения.

5. Не менее важным ограничением является фрагментация ИТ-ландшафта вуза. На данный момент в университете используются две разрозненные информационные системы, каждая из которых отвечает за отдельные сегменты работы с документами и учебным процессом. Подобная ситуация приводит к сложности обмена данными между системами, дублированию информации, повышенному количеству ошибок и снижению прозрачности административного контроля. Для полноценного внедрения ЭЦП необходимо в первую очередь осуществить интеграцию этих систем в единое цифровое пространство. Однако проблема объединения информационных платформ

остаётся нерешённой на протяжении длительного времени, что серьёзно тормозит переход к сквозному электронному документообороту.

6. Практика показывает, что отсутствие продуманной системы поддержки и обучения пользователей, особенно на первых этапах внедрения, ведёт к накоплению негативного опыта. Это выражается в большом количестве обращений в техническую поддержку, низкой удовлетворённости пользователей и возврате к ручным, бумажным процедурам.

Преодоление указанных административных и организационных барьеров требует комплексного подхода, включающего разъяснительную работу, поэтапное внедрение, пилотные проекты, обучение, формирование позитивной мотивации у сотрудников и создание гибкой системы нормативной поддержки электронного документооборота. Без решения этих вопросов даже технологически совершенная система ЭЦП будет реализована лишь частично и не принесёт ожидаемого эффекта для университета.

Отдельной проблемой при цифровизации документооборота остается «разрыв поколений» между пользователями с разным уровнем цифровой грамотности. Для минимизации рисков потребуется не просто техническая поддержка, а полноценная программа обучения и консультаций, рассчитанная на разные целевые группы. Следует учитывать, что сопротивление новшествам зачастую вызвано не столько опасениями за информационную безопасность, сколько неуверенностью пользователей в своих навыках.

Кроме того, вопросы распределения ответственности между подразделениями требуют чёткого нормативного закрепления. Например, процедура отзыва ключей или расследования инцидентов должна быть формализована заранее, чтобы избежать конфликтов между службой ИТ и административным персоналом. Важно заранее определить порядок хранения резервных копий, алгоритм действий при утрате доступа к ключу, а также условия массового перехода на новую систему в случае модернизации ИТ-инфраструктуры университета.

#### **5.4 Рекомендации по нормативному обеспечению и внутренней политике**

Эффективное и устойчивое внедрение системы электронной подписи в образовательной организации невозможно без формирования надёжной нормативной базы, регламентирующей все аспекты использования, хранения, проверки и отзыва электронной подписи на каждом этапе жизненного цикла документа. Основой такой базы должны служить положения действующего федерального законодательства, прежде всего Федерального закона от 6 апреля 2011 г. № 63-ФЗ «Об электронной подписи» [3], Федерального закона от 27 июля 2006 г. № 149-ФЗ «Об информации, информационных технологиях и о защите информации» [21], а также локальные акты самой образовательной организации.

В соответствии с требованиями статьи 6 и 9 Федерального закона № 63-ФЗ [3] организация обязана определить порядок формирования, хранения и проверки электронной подписи, обеспечить техническую и организационную

защищённость ключей, а также регламентировать права и обязанности всех участников процесса. Кроме того, в образовательном учреждении должны быть утверждены локальные нормативные акты, устанавливающие, что простая электронная подпись (ПЭП) приравнивается к собственноручной подписи при подписании внутренних документов (на основании статьи 6, пункт 2 и статьи 9, пункт 1 № 63-ФЗ [3]), с подробным описанием областей её допустимого применения.

Рекомендуется утвердить внутреннее положение о работе с электронной подписью, где подробно описывается:

- порядок выдачи, хранения, ротации и отзыва ключей электронной подписи;
- процедура идентификации пользователей при активации и восстановлении доступа;
- регламент аудита и мониторинга операций, связанных с использованием ЭЦП;
- ответственность за несанкционированное использование или компрометацию ключей, а также за нарушение регламента работы с электронной подписью;
- форматы электронных документов и требования к долговременному хранению (с учётом ГОСТ Р 34.10-2012 [4], ГОСТ Р 7.0.97-2016 [27] и других профильных стандартов).

Особое внимание должно быть уделено интеграции новых регламентов с уже действующими внутренними документами, в частности: положением о документообороте, политикой информационной безопасности, стандартами по хранению и архивированию документов, а также регламентом по обработке и защите персональных данных (Федеральный закон № 152-ФЗ [22]).

При необходимости взаимодействия с внешними организациями и государственными информационными системами, в том числе с Федеральным реестром документов об образовании, ЕСИА, системой «Госуслуги», требуется обеспечить совместимость электронных подписей с квалифицированными стандартами, а также поддержку форматов CAdES и XAdES [26], что также должно быть отражено в нормативных актах вуза. Желательно предусмотреть отдельный раздел о порядке признания внешних подписей и требованиях к их хранению и верификации.

На этапе масштабирования системы ЭЦП рекомендуется создать рабочую группу по нормативному обеспечению внедрения, в которую войдут представители юридической службы, ИТ-отдела, административных подразделений и службы информационной безопасности. Группа должна обеспечивать актуализацию нормативных документов в соответствии с изменениями федерального законодательства и внутренней политики университета.

Для снижения юридических и административных рисков настоятельно рекомендуется внедрить систему регулярного аудита соответствия внутренней



нормативной базы актуальным федеральным законам [3, 21, 22], а также практики ежегодного пересмотра локальных регламентов в части электронной подписи. Такая система позволит оперативно адаптироваться к изменениям в законодательстве, быстро внедрять новые технические решения и обеспечить максимальную юридическую защиту всех участников цифрового документооборота.

В долгосрочной перспективе утверждение и постоянная актуализация нормативной базы, связанной с применением электронной подписи, становится одним из ключевых условий для успешной цифровой трансформации образовательной организации и формирования единого, юридически значимого пространства электронного взаимодействия.

## Заключение

Выполненная выпускная квалификационная работа была посвящена разработке и апробации методики применения электронной цифровой подписи (ЭЦП) для оптимизации процессов документооборота и взаимодействия со студентами в образовательной организации. Актуальность темы обусловлена необходимостью повышения эффективности и безопасности информационного взаимодействия в условиях цифровизации, а также возрастающим вниманием к обеспечению технологической независимости образовательных учреждений.

В ходе выполнения работы были достигнуты следующие ключевые результаты:

1. Проведен анализ теоретических основ и нормативно–правовой базы, регулирующей применение ЭЦП в образовательной сфере. Это позволило сформировать целостное понимание возможностей и ограничений различных видов ЭЦП, включая простую и квалифицированную подписи.

2. Выполнено исследование текущих административных процессов в ИРНИТУ, в результате чего были выявлены ключевые узкие места, связанные с ручной обработкой заявок, отсутствием цифровых следов согласований и затрудненным контролем статусов документов.

3. Разработана методика интеграции ЭЦП в систему документооборота университета с учётом его инфраструктуры, типовых сценариев и требований безопасности. Были определены оптимальные точки внедрения, обеспечивающие максимальный эффект от цифровизации.

4. Создан и протестирован программный прототип, реализующий регистрацию пользователей, управление ключами, подписание документов, хранение истории событий и двухфакторную аутентификацию. Интерфейс и логика приложения были реализованы с прицелом на удобство и безопасность.

5. Проведено ручное тестирование системы, подтвердившее её функциональность, устойчивость и соответствие поставленным требованиям. В результате получены визуальные и аналитические подтверждения успешности реализации ключевых функций.

6. Разработаны рекомендации по дальнейшему внедрению системы в масштабах университета, включая интеграцию с Moodle, 1С, ЕСИА, а также направления по развитию мобильных решений, расширению аналитики и повышению отказоустойчивости.

Практическая значимость выполненной работы заключается в возможности её непосредственного применения для решения актуальных задач университета. Разработанное решение позволяет сократить время обработки заявок, повысить прозрачность и подотчетность процессов, а также обеспечить соответствие современным требованиям цифровой безопасности.

Дальнейшее развитие предложенного подхода может стать основой для формирования единой цифровой платформы взаимодействия между студентами, преподавателями и административными структурами ИРНИТУ.

## Список использованных источников

1. Паспорт национального проекта «Национальная программа “Цифровая экономика Российской Федерации”» [Электронный ресурс] – URL: <https://spa.msu.ru/wp-content/uploads/4-1.pdf> (дата обращения 15.02.2025).
2. Федеральный закон от 26.07.2017 № 187–ФЗ «О безопасности критической информационной инфраструктуры Российской Федерации» [Электронный ресурс] – URL: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_220885/](https://www.consultant.ru/document/cons_doc_LAW_220885/) (дата обращения 15.02.2025).
3. Федеральный закон от 06.04.2011 № 63–ФЗ «Об электронной подписи» [Электронный ресурс] – URL: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_112701/](https://www.consultant.ru/document/cons_doc_LAW_112701/) (дата обращения 16.02.2025).
4. ГОСТ Р 34.10–2012 «Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи» [Электронный ресурс] – URL: <https://docs.cntd.ru/document/1200095034> (дата обращения 17.02.2025).
5. Приказ Минобрнауки России от 23.08.2017 № 816 «Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий...» [Электронный ресурс] – URL: <https://normativ.kontur.ru/document?moduleId=1&documentId=300600> (дата обращения 20.02.2025).
6. FIPS PUB 186-5. Digital Signature Standard (DSS). National Institute of Standards and Technology, 2023 г. [Электронный ресурс] – URL: <https://csrc.nist.gov/pubs/fips/186-5/final> (дата обращения 29.02.2025).
7. Menezes A., van Oorschot P., Vanstone S. *Handbook of Applied Cryptography*. – CRC Press, 1996. Гл. 11 «Digital Signatures». [Электронный ресурс] – URL: <https://cacr.uwaterloo.ca/hac/about/chap11.pdf> (дата обращения 01.03.2025).
8. Diffie W., Hellman M. *New Directions in Cryptography* // IEEE Transactions on Information Theory. 1976. Vol. 22, № 6. С. 644-654. [Электронный ресурс] – URL: <https://www-ee.stanford.edu/~hellman/publications/24.pdf> (дата обращения 03.03.2025).
9. FIPS PUB 180-4. Secure Hash Standard (SHS). National Institute of Standards and Technology, 2015 г. [Электронный ресурс] – URL: <https://csrc.nist.gov/pubs/fips/180-4/upd1/final> (дата обращения 05.03.2025).
10. RFC 5280: Internet X.509 Public Key Infrastructure Certificate and CRL Profile. IETF, май 2008 г. [Электронный ресурс] – URL: <https://datatracker.ietf.org/doc/html/rfc5280> (дата обращения 12.03.2025).
11. Rivest R., Shamir A., Adleman L. *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems* // Communications of the ACM. 1978. Vol.

- 21, № 2. С. 120-126. [Электронный ресурс] – URL: <https://mathybit.github.io/assets/docs/csai/RSA.pdf> (дата обращения 12.03.2025).
12. SEC 1: Elliptic Curve Cryptography. Version 2.0. Standards for Efficient Cryptography Group, 2009 г. [Электронный ресурс] – URL: <https://www.secg.org/sec1-v2.pdf> (дата обращения 13.03.2025).
13. ГОСТ Р 34.11-2012. Информационная технология. Криптографическая защита информации. Функция хэширования. [Электронный ресурс] – URL: <https://docs.cntd.ru/document/1200095035> (дата обращения 14.03.2025).
14. O'Connor J., Aumasson J-P., Neves S., Wilcox-O'Hearn Z. *BLAKE3: One Function, Fast Everywhere*. Design specification, 2020 г. [Электронный ресурс] – URL: <https://github.com/BLAKE3-team/BLAKE3-specs/blob/master/blake3.pdf> (дата обращения 14.03.2025).
15. Web Cryptography API. W3C Recommendation, 26.01.2017 г. [Электронный ресурс] – URL: <https://www.w3.org/TR/WebCryptoAPI/> (дата обращения 16.03.2025).
16. RFC 7292: PKCS #12 – Personal Information Exchange Syntax. IETF, 09.2013 г. [Электронный ресурс] – URL: <https://datatracker.ietf.org/doc/html/rfc7292> (дата обращения 16.03.2025).
17. NIST. Post-Quantum Cryptography Standardization: Selected Algorithms. 2022 г. [Электронный ресурс] – URL: <https://csrc.nist.gov/projects/post-quantum-cryptography/selected-algorithms-2022> (дата обращения 17.03.2025).
18. RFC 8391: XMSS – eXtended Merkle Signature Scheme. IETF, 05.2018 г. [Электронный ресурс] – URL: <https://datatracker.ietf.org/doc/html/rfc8391> (дата обращения 22.05.2025).
19. Regulation (EU) No 910/2014 (eIDAS) of 23 July 2014. [Электронный ресурс] – URL: <https://eur-lex.europa.eu/eli/reg/2014/910/oj> (дата обращения 17.03.2025).
20. ISO/IEC 14888-1:2008. Information technology – Security techniques – Digital signatures with appendix – Part 1: General. [Электронный ресурс] – URL: [https://www.vde-verlag.de/iec-normen/preview-pdf/info\\_isoiec14888-1%7Bed2.0%7Den.pdf](https://www.vde-verlag.de/iec-normen/preview-pdf/info_isoiec14888-1%7Bed2.0%7Den.pdf) (дата обращения 17.03.2025).
21. Федеральный закон от 27.07.2006 № 149-ФЗ «Об информации, информационных технологиях и о защите информации» [Электронный ресурс] – URL: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_61798/](https://www.consultant.ru/document/cons_doc_LAW_61798/) (дата обращения 18.03.2025).
22. Федеральный закон от 27.07.2006 № 152-ФЗ «О персональных данных» [Электронный ресурс] – URL: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_61801/](https://www.consultant.ru/document/cons_doc_LAW_61801/) (дата обращения 18.03.2025).
23. Постановление Правительства Российской Федерации от 06.09.2012 № 1083 «О развитии электронной формы документооборота» [Электронный ресурс] – URL: <http://pravo.gov.ru/proxy/ips/?docbody=&nd=102159139&intelsearch=%FD%EB%>

E5%EA%F2%F0%EE%ED%ED%FB%E9+%E4%EE%EA%F3%EC%E5%ED%F2%EE%EE%E1%EE%F0%EE%F2 (дата обращения 19.03.2025).

24. Приказ Минобрнауки России от 26.08.2013 № 706 «О порядке ведения федерального реестра документов об образовании и/или квалификации...» [Электронный ресурс] – URL: <http://pravo.gov.ru/proxy/ips/?docbody=&nd=102167463> (дата обращения 21.03.2025).

25. Приказ Минцифры России от 02.03.2022 № 279 «О государственной информационной системе «Платформа Центр хранения электронных документов» [Электронный ресурс] – URL: <https://base.garant.ru/403607384/> (дата обращения 25.03.2025).

26. ETSI EN 319 122-1 V1.2.1 (2021-10). Electronic Signatures and Infrastructures (ESI); CAdES; Part 1: Building Blocks and CAdES Baseline Signatures. [Электронный ресурс] – URL: [https://www.etsi.org/deliver/etsi\\_en/319100\\_319199/31912201/01.02.01\\_60/en\\_31912201v010201p.pdf](https://www.etsi.org/deliver/etsi_en/319100_319199/31912201/01.02.01_60/en_31912201v010201p.pdf) (дата обращения 27.03.2025).

27. ГОСТ Р 7.0.97-2016. СИБИД. Электронные документы. Правила долговременного хранения. [Электронный ресурс] – URL: <https://docs.cntd.ru/document/1200142871> (дата обращения 29.03.2025).

28. Приказ Минобрнауки России от 21.08.2020 № 1076 «Об утверждении Порядка приёма на обучение по образовательным программам высшего образования...» [Электронный ресурс] – URL: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_360722/](https://www.consultant.ru/document/cons_doc_LAW_360722/) (дата обращения 02.04.2025).

29. Постановление Правительства Российской Федерации от 10.07.2013 № 582 «Об использовании электронной подписи при предоставлении государственных и муниципальных услуг в электронной форме» [Электронный ресурс] – URL: <https://www.garant.ru/products/ipo/prime/doc/74441661/> (дата обращения 02.04.2025).

30. Приказ Федеральной налоговой службы от 12.08.2022 № ЕД-7-26/736@ «Об утверждении формы машиночитаемой доверенности...» [Электронный ресурс] – URL: <https://www.garant.ru/products/ipo/prime/doc/407448151/> (дата обращения 03.04.2025).

31. Министерство науки и высшего образования РФ. Аналитический доклад «Цифровая трансформация университетов: результаты мониторинга 2023 года». – М., 2024. [Электронный ресурс] – URL: <https://monitoring.miccedu.ru/files/2024/digital-report.pdf> (дата обращения 07.04.2025).

## Приложения

### Приложение А

#### «Модуль генерации и шифрования ключей»

```
const crypto = require('crypto');
const ITER = 150_000;
const KEYLEN = 32;
const DIGEST = 'sha256';
function deriveKey(pass, salt) {
  return crypto.pbkdf2Sync(pass, salt, ITER, KEYLEN, DIGEST);
}
function aesEncrypt(buf, password) {
  const salt = crypto.randomBytes(16);
  const key = deriveKey(password, salt);
  const iv = crypto.randomBytes(12);
  const cipher = crypto.createCipheriv('aes-256-gcm', key, iv);
  const enc = Buffer.concat([cipher.update(buf), cipher.final()]);
  const tag = cipher.getAuthTag();
  return { iv, tag, enc, salt };
}
function aesDecrypt({ iv, tag, enc, salt }, password) {
  const key = deriveKey(password, salt);
  const decipher = crypto.createDecipheriv('aes-256-gcm', key, iv);
  decipher.setAuthTag(tag);
  return Buffer.concat([decipher.update(enc), decipher.final()]);
}
function generateRSA() {
  const { publicKey, privateKey } = crypto.generateKeyPairSync('rsa', {
    modulusLength: 2048,
    publicKeyEncoding: { format: 'der', type: 'spki' },
    privateKeyEncoding: { format: 'der', type: 'pkcs8' }
  });
  return { pubDer: publicKey, privDer: privateKey };
}
function signBlob(privDer, data) {
  const keyObj = crypto.createPrivateKey({ key: privDer, format: 'der', type:
'pkcs8' });
  const sign = crypto.createSign('sha256');
  sign.update(data);
  sign.end();
  return sign.sign(keyObj);
}
module.exports = { aesEncrypt, aesDecrypt, generateRSA, signBlob };
```

## Приложение Б

### «Модуль базовых операций простой ЭЦП»

```
const crypto = require('crypto');
const { createKeyPair } = require('./enhancedSignatureService');
const enhancedSigSvc = require('./enhancedSignatureService');
function sha256(txt) {
  return crypto.createHash('sha256')
    .update(txt)
    .digest('hex')
    .match(/.{1,2}/g)
    .join(' ');
}
async function listUserSigs (db, id_user) {
  return db.all(
    `SELECT id_sign, name, fingerprint, created_at
    FROM simple_signatures
    WHERE id_user = ?
    ORDER BY created_at DESC`, [id_user]);
}
async function createSimpleSig (db, id_user, name=null) {
  const fp = sha256(`${id_user}-${Date.now()}`);
  const res = await db.run(
    `INSERT INTO simple_signatures (id_user,name,fingerprint)
    VALUES (?,?,?)`, [id_user, name, fp]);
  return db.get(
    `SELECT id_sign, name, fingerprint, created_at
    FROM simple_signatures
    WHERE id_sign = ?`, res.lastID);
}
async function deleteSig (db, id_user, id_sign) {
  return db.run(
    `DELETE FROM simple_signatures
    WHERE id_sign = ? AND id_user = ?`, [id_sign, id_user]);
}
async function listUserKeys(db, id_user) {
  return db.all(
    `SELECT id_key, name, fingerprint, created_at
    FROM user_keys
    WHERE id_user = ?
    ORDER BY created_at DESC`,
    [id_user]
  );
}
```

```

async function deleteKey(db, id_user, id_key) {
  return db.run(
    `DELETE FROM user_keys
    WHERE id_key = ? AND id_user = ?`,
    [id_key, id_user]
  );
}

module.exports = {
  listUserKeys,
  listUserSigs,
  createSimpleSig,
  deleteSig,
  deleteKey,
  createKeyPair,
  signRequest: enhancedSigSvc.signRequest,
  signRequestByUser: enhancedSigSvc.signRequestByUser
};

```



## Приложение В

### «Модуль расширенных функций подписи и штампа времени»

```
const { aesEncrypt, aesDecrypt, generateRSA, signBlob } =
require('./cryptoUtil');
const fs = require('fs');
const crypto = require('crypto');
const path = require('path');

async function createKeyPair(db, id_user, password, name = null) {
  const { privDer, pubDer } = generateRSA(); // RSA-2048
  const { iv, tag, enc, salt } = aesEncrypt(privDer, password);
  const encPriv = Buffer.concat([iv, tag, enc]); // iv|tag|ciphertext

  const fp = crypto
    .createHash('sha256')
    .update(pubDer)
    .digest('hex')
    .match(/.{1,2}/g)
    .join(' ');

  const { lastID } = await db.run(
    `INSERT INTO user_keys
    (id_user, enc_priv, pub_key, salt, name, fingerprint)
    VALUES (?, ?, ?, ?, ?, ?)`,
    [id_user, encPriv, pubDer, salt, name, fp]
  );

  return db.get(
    `SELECT id_key, name, fingerprint, created_at
    FROM user_keys
    WHERE id_key = ?`,
    lastID
  );
}

async function deleteKey(db, id_user, id_key) {
  return db.run(
    `DELETE FROM user_keys WHERE id_key = ? AND id_user = ?`,
    [id_key, id_user]
  );
}
```

```

// enhancedSignatureService.js
async function signRequest(
  db,
  id_request,
  id_user,
  password,
  id_key
) {
  const req = await db.get(
    `SELECT r.id_request, r.id_document,
    uk.id_key, uk.enc_priv, uk.pub_key, uk.salt
    FROM signature_requests r
    JOIN user_keys uk
    ON uk.id_user = ? — получатель
    AND uk.id_key = ? — выбранный ключ
    WHERE r.id_request = ?
    AND r.id_recipient_user = ?`,
    [id_user, id_key, id_request, id_user]
  );
  if (!req) throw new Error("Ключ не найден");

  const doc = await db.get(
    `SELECT stored_path FROM documents WHERE id_document = ?`,
    req.id_document
  );
  const fileBuf = fs.readFileSync(path.join("uploads", doc.stored_path));

  const packed = {
    iv: req.enc_priv.subarray(0, 12),
    tag: req.enc_priv.subarray(12, 28),
    enc: req.enc_priv.subarray(28),
    salt: req.salt
  };
  const privDer = aesDecrypt(packed, password); // бросит, если пароль
  неверный

  const digest = crypto.createHash("sha256").update(fileBuf).digest();
  const signature = signBlob(privDer, digest);

  await db.run(
    `INSERT INTO request_signatures
    (id_request, id_key, hash_alg, digest, signature)
    VALUES (?, ?, ?, ?, ?)`,

```

```

[id_request, req.id_key, "sha256", digest, signature]
);

await db.run(
`UPDATE signature_requests
SET id_status = 2,
updated_at = CURRENT_TIMESTAMP
WHERE id_request = ?`,
[id_request]
);

return { ok: true };
}

async function signRequestByUser(db, id_request, id_user, password, id_key) {
const req = await db.get(
`SELECT r.id_request, r.id_document, uk.id_key, uk.enc_priv, uk.pub_key,
uk.salt
FROM signature_requests r
JOIN user_keys uk
ON uk.id_user = ? AND uk.id_key = ?
WHERE r.id_request = ? AND r.id_sender = ?`,
[id_user, id_key, id_request, id_user]
);
if (!req) throw new Error('Request or key not found');

const doc = await db.get(
`SELECT stored_path FROM documents WHERE id_document = ?`,
req.id_document
);
const fileBuf = fs.readFileSync(`./uploads/${doc.stored_path}`);

const packed = {
iv: req.enc_priv.subarray(0, 12),
tag: req.enc_priv.subarray(12, 28),
enc: req.enc_priv.subarray(28),
salt: req.salt
};
const privDer = aesDecrypt(packed, password);

const digest = crypto.createHash('sha256').update(fileBuf).digest();
const signature = signBlob(privDer, digest);

```

```
await db.run(  
  `INSERT INTO request_signatures  
  (id_request, id_key, hash_alg, digest, signature)  
  VALUES (?, ?, ?, ?, ?)`,  
  [id_request, req.id_key, 'sha256', digest, signature]  
);  
return { ok: true };  
}
```

```
module.exports = {  
  createKeyPair,  
  deleteKey,  
  signRequest,  
  signRequestByUser  
};
```

## Приложение Г

### «Модуль маршрутизации запросов и JWT–аутентификации»

```
async function listDepartments(db) {
  return db.all(`SELECT id_dept, name FROM departments ORDER BY name`);
}

async function listUsers(db) {
  return db.all(
    `SELECT id_user, full_name, email
    FROM users
    ORDER BY full_name`
  );
}

async function insertDocument(db, { id_user, file }) {
  const res = await db.run(
    `INSERT INTO documents
    (id_user, original_name, stored_path, mime_type, size)
    VALUES (?, ?, ?, ?, ?)`,
    [id_user, file.originalname, file.filename, file.mimetype, file.size]
  );
  return db.get(`SELECT * FROM documents WHERE id_document = ?`,
res.lastID);
}

async function createSignatureRequest(db, data) {
  const res = await db.run(
    `INSERT INTO signature_requests
    (id_document, id_sender,
    id_recipient_user, id_recipient_dept,
    id_status, comment_sender)
    VALUES (?, ?, ?, ?, 1, ?)`,
    [
    data.id_document,
    data.id_sender,
    data.id_recipient_user || null,
    data.id_recipient_dept || null,
    data.comment_sender || null
    ]
  );
  return db.get(
    `SELECT * FROM signature_requests WHERE id_request = ?`,
    res.lastID
  );
}
```

```

    );
}

async function listUserRequests(db, userId) {
  return db.all(
    `SELECT r.id_request,
    d.id_document,
    d.original_name,
    r.created_at,
    s.name AS status,
    COALESCE(rc.cnt,0) AS comment_count,
    CASE
      WHEN r.id_recipient_user IS NOT NULL
      THEN (SELECT full_name FROM users WHERE id_user =
r.id_recipient_user)
      WHEN r.id_recipient_dept IS NOT NULL
      THEN (SELECT name FROM departments WHERE id_dept =
r.id_recipient_dept)
      ELSE '-'
    END AS recipient_name
    FROM signature_requests r
    JOIN documents d ON d.id_document = r.id_document
    JOIN statuses s ON s.id_status = r.id_status
    LEFT JOIN (
      SELECT id_request, COUNT(*) AS cnt
      FROM request_comments
      GROUP BY id_request
    ) rc ON rc.id_request = r.id_request
    WHERE r.id_sender = ?
    ORDER BY r.created_at DESC`,
    [userId]
  );
}

async function getRequest(db, id, userId) {
  return db.get(
    `SELECT * FROM signature_requests
    WHERE id_request = ? AND id_sender = ?`,
    [id, userId]
  );
}

async function updateSignatureRequest(db, id, userId, data) {

```

```

    await db.run(
      `UPDATE signature_requests
      SET id_document = COALESCE(?, id_document),
      id_recipient_user = ?,
      id_recipient_dept = ?,
      comment_sender = ?
      WHERE id_request = ? AND id_sender = ?`,
      [
        data.id_document || null,
        data.id_recipient_user || null,
        data.id_recipient_dept || null,
        data.comment_sender || null,
        id,
        userId
      ]
    );
    return db.get(`SELECT * FROM signature_requests WHERE id_request = ?`,
id);
  }

```

```

async function deleteSignatureRequest(db, id, userId) {
  return db.run(
    `DELETE FROM signature_requests
    WHERE id_request = ? AND id_sender = ?`,
    [id, userId]
  );
}

```

```

async function listIncomingRequests(db, userId) {
  return db.all(
    `SELECT r.id_request,
    d.id_document,
    d.original_name,
    r.created_at,
    s.name AS status,
    u.full_name AS sender_name
    FROM signature_requests r
    JOIN documents d ON d.id_document = r.id_document
    JOIN statuses s ON s.id_status = r.id_status
    JOIN users u ON u.id_user = r.id_sender
    WHERE r.id_recipient_user = ?
    ORDER BY r.created_at DESC`,
    [userId]
  );
}

```

```
);  
}
```

```
module.exports = {  
  listDepartments,  
  listUsers,  
  insertDocument,  
  createSignatureRequest,  
  listUserRequests,  
  listIncomingRequests,  
  getRequest,  
  updateSignatureRequest,  
  deleteSignatureRequest  
};
```



## Приложение Д

### «Модуль управления документами и их статусами»

```
async function listUserDocs(db, userId, limit = 10) {
  return db.all(
    `SELECT r.id_request,
    d.id_document,
    d.original_name,
    d.mime_type,
    d.size,
    r.created_at AS request_date,
    s.name AS status
    FROM signature_requests r
    JOIN documents d ON d.id_document = r.id_document
    JOIN statuses s ON s.id_status = r.id_status
    WHERE r.id_sender = ?
    OR r.id_recipient_user = ?
    ORDER BY r.created_at DESC
    LIMIT ?`,
    [userId, userId, limit]
  );
}

async function getDocumentForUser(db, userId, docId) {
  return db.get(
    `SELECT d.*
    FROM documents d
    LEFT JOIN signature_requests r ON r.id_document = d.id_document
    WHERE d.id_document = ?
    AND ( d.id_user = ?
    OR r.id_sender = ?
    OR r.id_recipient_user = ? )`,
    [docId, userId, userId, userId]
  );
}

module.exports = { listUserDocs, getDocumentForUser };
```

## Приложение Е

### «Модуль неизменяемого журнала аудита»

```
async function addAudit(db, { id_user, ip, action, meta = { } }) {  
  await db.run(  
    `INSERT INTO audit_log (id_user, ip_addr, action, meta)  
    VALUES (?, ?, ?, ?)`,  
    [id_user, ip, action, JSON.stringify(meta)]  
  );  
}  
module.exports = { addAudit };
```