



МИНОБРНАУКИ РОССИИ
федеральное государственное автономное образовательное учреждение
высшего образования
«Московский государственный технологический университет «СТАНКИН»
(ФГАОУ ВО «МГТУ «СТАНКИН»)

**Институт
информационных
технологий**

**Кафедра
информационных технологий
и вычислительных систем**

КУРСОВОЙ ПРОЕКТ
ПО ДИСЦИПЛИНЕ
«ОСНОВЫ НОВЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»

СТУДЕНТА 4 КУРСА бакалавриата ГРУППЫ ИДБ-22-05

ЗАМОСТЬЯНОВ НИКОЛАЙ ВЛАДИСЛАВОВИЧ

НА ТЕМУ
Реализация функции дистанционного заказа справок на базе личного кабинета
университета

Направление: 09.03.01 Информатика и вычислительная техника
Профиль подготовки: Разработка программных комплексов в рамках цифровой трансформации деятельности предприятий

Отчет сдан « » 20 г.

Оценка

Преподаватель Бибииков О.Д., старший преподаватель

(подпись)

МОСКВА 2025

ОГЛАВЛЕНИЕ

АННОТАЦИЯ	3
Проблематика.....	3
Актуальность.....	3
Основная цель	3
Описание конечного продукта	4
Use-case.....	Error! Bookmark not defined.
ОПИСАНИЕ ТЕРМИНОВ.....	4
ОБЩЕЕ ОПИСАНИЕ СИСТЕМЫ.....	5
Основные варианты использования (Use Cases)	6
Ограничения и допущения	7
ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ (FR)	8
Требования к подсистеме «Студент».....	8
Требования к подсистеме «Администратор»	9
Системные функции	10
НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ (NFR)	10
Требования к производительности	11
Требования к надежности.....	11
Требования к безопасности	11
Требования к удобству использования (Usability)	11
Требования к совместимости	12
ТРЕБОВАНИЯ К АРХИТЕКТУРЕ И ТЕХНОЛОГИЯМ.....	12
ТРЕБОВАНИЯ К ДОКУМЕНТАЦИИ.....	13
ПРИЕМКА И ТЕСТИРОВАНИЕ	13

АННОТАЦИЯ

Система предназначена для использования в рамках существующей IT-инфраструктуры университета. Областью применения является личный кабинет (ЛК) университета, в который данный сервис будет интегрирован в качестве нового модуля. Сервис автоматизирует процесс заказа и получения справок, затрагивая взаимодействие между студентами и сотрудниками административных подразделений.

Проблематика. Существующий процесс получения справок об обучении требует от студентов личного посещения управления персоналом, подачи бумажного заявления и ожидания в очередях. Данная процедура характеризуется значительными временными затратами как для студентов, так и для сотрудников. Отсутствует единый инструмент для заказа документа, что увеличивает время ожидания для конечного пользователя (студента).

Актуальность. Разработка сервиса является актуальной задачей в контексте стратегии цифровой трансформации университета. Автоматизация рутинных административных процедур позволяет:

- Повысить качество и скорость предоставления университетских услуг.
- Снизить нецелевые трудозатраты сотрудников, позволяя им сконцентрироваться на более сложных задачах.
- Соответствовать современным ожиданиям студентов, привыкших к получению услуг в цифровом формате.
- Создать централизованную систему учета и хранения данных о выданных справках.

Основная цель - автоматизировать и упростить процесс заказа, обработки и получения справок об обучении для студентов и сотрудников университета. **Повысить качество** предоставления услуги «получение справки для студентов и сотрудников университета» за счет автоматизации процесса получения справок.

Для достижения основной цели система должна решать следующие задачи:

- Реализовать для студентов возможность подачи заявки на получение справки через интерфейс личного кабинета.
- Обеспечить сотрудников управления персоналом инструментом для централизованного приема, обработки и выполнения заявок.
- Внедрить систему статусов заявок («Новая», «В работе», «Завершена», «Отклонена») для обеспечения прозрачности процесса.

- Предоставить функциональность для загрузки сотрудником и скачивания студентом готовой справки в электронном виде (формат PDF).
- Реализовать механизм автоматических уведомлений в личном кабинете об изменении статуса заявки.

Описание конечного продукта. Результатом проекта станет новый программный модуль «Электронные справки», интегрированный в существующий личный кабинет университета. Данный модуль будет представлять собой веб-приложение, разработанное с использованием современного стека технологий (Frontend: Angular 14+, Backend: .NET 8 WebAPI, СУБД: PostgreSQL) и готовое к внедрению в реальные бизнес-процессы университета.

Система будет использовать базу данных PostgreSQL для надежного хранения информации о заявках, их статусах и прикрепленных документах. Конечный продукт должен полностью соответствовать требованиям, изложенным в настоящем Техническом задании, и успешно пройти приемочное тестирование.

Заказчик: Университет в лице Департамента Цифровой Трансформации и Управления персоналом. Заказчик определяет требования к системе, предоставляет необходимую информацию и осуществляет приемку готового продукта.

Разработчик: Студент Замостьянов Николай Владиславович + должность + отдел, выполняющий курсовую работу. Разработчик несет ответственность за проектирование, реализацию и тестирование системы в соответствии с настоящим Техническим заданием.

Конечные пользователи:

- Студенты - основная аудитория сервиса, использующая его для заказа и получения справок.
- Администратор системы (сотрудник Управления персоналом) - ответственный сотрудник, который обрабатывает поступающие заявки, готовит и загружает справки.

ОПИСАНИЕ ТЕРМИНОВ

Система - разрабатываемый программный модуль «Электронные справки», предназначенный для интеграции в Личный кабинет университета.

Техническое задание (ТЗ) - настоящий документ, описывающий требования к Системе.

Личный кабинет (ЛК) - существующая информационная система университета, предоставляющая персонализированный доступ к академическим и административным сервисам.

Пользователь - любое лицо (Студент или Администратор), взаимодействующее с Системой после прохождения процедуры аутентификации.

Студент - роль пользователя, обладающая правами на создание заявок на получение справок и просмотр их статуса.

Администратор - роль пользователя (сотрудник Управления персоналом), обладающая правами на просмотр и обработку заявок от студентов.

Заявка - электронный запрос, создаваемый Студентом в Системе для получения справки установленного образца.

Справка об обучении - официальный документ установленной формы, подтверждающий факт обучения студента в университете.

Статус заявки - атрибут заявки, отражающий текущий этап ее жизненного цикла. Возможные значения: «Новая», «В работе», «Завершена», «Отклонена».

Уведомление - автоматически генерируемое системное сообщение в интерфейсе ЛК, информирующее Студента об изменении статуса его Заявки.

API (Application Programming Interface) - программный интерфейс приложения; в контексте проекта – набор методов для взаимодействия между frontend- и backend-частями Системы.

СУБД (Система Управления Базами Данных) - программное обеспечение для создания и управления базами данных (в данном проекте – PostgreSQL).

ОБЩЕЕ ОПИСАНИЕ СИСТЕМЫ

Назначение системы. Система «Электронные справки» предназначена для автоматизации бизнес-процесса заказа справки об обучении. Она обеспечивает централизованный сбор, обработку и хранение заявок, а также упрощает взаимодействие между студентами и сотрудниками управления персоналом.

Ключевые функции системы включают:

- Формирование студентом заявки на справку через веб-интерфейс.
- Обработка заявки ответственным сотрудником.
- Отслеживание статуса заявки в режиме реального времени.

- Предоставление готовой справки в электронном виде.

Целевая аудитория (роли пользователей). В системе предусмотрены две основные роли пользователей:

- Студент. Конечный пользователь системы, обучающийся университета. Студент использует систему для создания заявок на справки, отслеживания их статуса и получения готового документа.
- Администратор. Сотрудник Управления персоналом университета. Администратор использует систему для просмотра списка всех поступающих заявок, их обработки (принятия в работу, завершения, отклонения) и загрузки электронных версий готовых справок.

Основные варианты использования (Use Cases)

Сценарии для роли «Студент»:

- Студент авторизуется в ЛК и переходит в раздел «Электронные справки».
- Студент создает новую заявку на получение справки об обучении, выбирая один из доступных форматов получения (очно, скан в ЛК).
- Студент просматривает список своих ранее поданных заявок и их текущие статусы.
- При изменении статуса заявки на «Завершена» или «Отклонена», студент получает уведомление в ЛК.
- В случае готовности электронной версии справки, студент скачивает прикрепленный PDF-файл из деталей завершенной заявки.

Сценарии для роли «Администратор»:

- Администратор авторизуется в ЛК и переходит в интерфейс управления заявками.
- Администратор видит список всех заявок с возможностью их фильтрации по статусу (например, отобразить только «Новые»).
- Администратор открывает конкретную заявку, принимает ее в работу, изменяя ее статус на «В работе».
- После подготовки документа Администратор завершает заявку:
- Если был выбран электронный формат, он прикрепляет готовый PDF-файл и устанавливает статус «Завершена».

- Если был выбран очный формат, он изменяет статус на «Завершена» с системным сообщением о том, что справка готова к выдаче.
- В случае невозможности выполнения заявки, Администратор устанавливает статус «Отклонена» с обязательным указанием причины.
- Сроки и ресурсы: разработка системы осуществляется в рамках учебного семестра силами одного студента-разработчика.

Ограничения и допущения

Ограничения:

- Система является модулем существующего ЛК, а не самостоятельным приложением.
- Аутентификация и авторизация пользователей осуществляются исключительно средствами ЛК. Система не реализует собственные механизмы входа.
- В рамках текущей версии системы для заказа доступен только один тип документа – «Справка об обучении».
- Система уведомлений работает только в рамках интерфейса ЛК. Интеграция с внешними сервисами (e-mail, мессенджеры) не предусмотрена.
- +все ограничения идут от лк. Документ с ограничениями лк

Допущения:

- Предполагается, что все студенты и ответственные сотрудники имеют активные учетные записи и стабильный доступ к ЛК университета.
- Допускается, что необходимая для работы модуля серверная инфраструктура (включая СУБД PostgreSQL и брокер сообщений RabbitMQ) уже развернута и сконфигурирована.
- Предполагается, что пользователи обладают базовыми навыками работы с веб-интерфейсами и не нуждаются в дополнительном обучении для использования основных функций системы.

Ограничения по языкам разработки и технологиям:

- Клиентская часть (Frontend) должна быть реализована как одностраничное веб-приложение (SPA) с использованием фреймворка Angular (версии 14 или выше) и языка TypeScript.
- Серверная часть (Backend) должна быть реализована на платформе .NET 8 с использованием технологии ASP.NET Core

Web API. Взаимодействие с базой данных (PostgreSQL) осуществляется через Entity Framework Core.

- Для управления очередью уведомлений используется брокер сообщений RabbitMQ.

Стандарты и нормативы:

- Пользовательский интерфейс системы должен соответствовать современным принципам UI/UX-дизайна (интуитивная понятность, отзывчивость, консистентность).
- Необходимо учитывать гайдлайны и фирменный стиль (brandbook) университета для обеспечения визуального единства с основным порталом ЛК.

ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ (FR)

Этот раздел содержит конкретные функциональные требования к системе. Каждое требование имеет уникальный идентификатор для удобства ссылок, отслеживания и тестирования.

Общие требования

FR-1: Аутентификация и авторизация пользователей.

Описание: Система не должна иметь собственной страницы входа. Доступ к функциям системы должен предоставляться только пользователям, прошедшим аутентификацию через центральный сервис Личного кабинета (ЛК) университета. После успешной аутентификации система должна определить роль пользователя (Студент или Администратор) и предоставить доступ к соответствующему интерфейсу и набору функций.

Предусловие: Пользователь не авторизован в ЛК.

Постусловие: Пользователь успешно вошел в ЛК и перенаправлен в интерфейс сервиса «Электронные справки» с правами, соответствующими его роли. **Критерии приемки:** При переходе по прямой ссылке на сервис неавторизованный пользователь перенаправляется на страницу входа ЛК. После успешного входа студент видит интерфейс для заказа справок, а администратор — интерфейс для управления заявками.

Требования к подсистеме «Студент»

FR-2: Создание заявки на справку. **Описание:** Система должна предоставлять студенту возможность создать новую заявку на получение справки об обучении. Пользователь должен выбрать формат получения справки из предопределенного списка: «Забрать очно» или «Получить скан в ЛК». После подтверждения заявка создается в системе с начальным статусом «Новая». **Предусловие:** Пользователь авторизован с ролью «Студент» и находится на главной странице сервиса. **Постусловие:** В системе создана новая запись о заявке со статусом «Новая», которая отображается в списке заявок

студента. **Критерии приемки:** Когда студент выбирает формат справки и нажимает кнопку «Отправить заявку», в его списке заявок появляется новая запись со статусом «Новая».

FR-3: Просмотр списка своих заявок и их статусов. **Описание:** Система должна отображать студенту список всех созданных им ранее заявок в виде таблицы или списка. Для каждой заявки должна отображаться ключевая информация: уникальный номер (ID), дата создания и текущий статус («Новая», «В работе», «Завершена», «Отклонена»). Список должен быть отсортирован по дате создания в порядке убывания (от новых к старым). **Предусловие:** Пользователь авторизован с ролью «Студент». **Постусловие:** На экране отображен полный и актуальный список заявок пользователя. **Критерии приемки:** На главной странице сервиса студент видит таблицу со всеми своими заявками и их актуальными статусами.

FR-4: Скачивание готовой справки. **Описание:** Если заявка имеет статус «Завершена» и для нее был выбран формат «Получить скан в ЛК», система должна предоставить студенту возможность скачать прикрепленный файл. Рядом с такой заявкой должна отображаться кнопка или ссылка «Скачать». **Предусловие:** Заявка студента имеет статус «Завершена» и содержит прикрепленный администратором файл. **Постусловие:** Файл справки в формате PDF загружен на устройство пользователя. **Критерии приемки:** При нажатии на кнопку «Скачать» напротив соответствующей заявки начинается загрузка PDF-файла.

Требования к подсистеме «Администратор»

FR-5: Просмотр и фильтрация списка всех заявок. **Описание:** Администратор должен видеть полный список всех заявок от всех студентов. Список должен содержать информацию: ID заявки, ФИО студента, дата создания, статус. Система должна предоставлять инструменты для фильтрации заявок по статусу (например, отобразить только «Новые»). **Предусловие:** Пользователь авторизован с ролью «Администратор». **Постусловие:** На экране отображен отфильтрованный список заявок. **Критерии приемки:** Администратор может успешно применить фильтр по статусу «Новая» и увидеть в списке только те заявки, которые еще не были взяты в работу.

FR-6: Принятие заявки в работу. **Описание:** Администратор должен иметь возможность изменить статус заявки с «Новая» на «В работе». Эта функция необходима для фиксации того, что заявка принята и находится на исполнении, что предотвращает ее повторную обработку другим сотрудником. **Предусловие:** Заявка имеет статус «Новая». **Постусловие:** Статус заявки изменен на «В работе». **Критерии**

приемки: Когда администратор нажимает кнопку «Взять в работу» для новой заявки, ее статус в общем списке обновляется на «В работе».

FR-7: Завершение заявки. **Описание:** Система должна позволять администратору завершить обработку заявки. Если формат получения — «Получить скан в ЛК», администратор должен прикрепить файл в формате PDF и подтвердить действие, после чего статус заявки изменится на «Завершена». Если формат — «Забрать очно», администратор изменяет статус на «Завершена» без прикрепления файла. **Предусловие:** Заявка имеет статус «В работе». **Постусловие:** Статус заявки изменен на «Завершена». Система инициирует отправку уведомления студенту. **Критерии приемки:** После того как администратор прикрепляет PDF-файл к заявке и подтверждает ее завершение, статус заявки меняется на «Завершена», и студент получает уведомление о готовности.

FR-8: Отклонение заявки. **Описание:** В случае невозможности выполнить заявку, администратор должен иметь возможность отклонить ее. При этом система должна требовать обязательного заполнения текстового поля с указанием причины отклонения. После подтверждения статус заявки меняется на «Отклонена». **Предусловие:** Заявка имеет статус «Новая» или «В работе». **Постусловие:** Статус заявки изменен на «Отклонена». Система инициирует отправку уведомления студенту с указанием причины. **Критерии приемки:** Когда администратор вводит причину и нажимает «Отклонить», статус заявки обновляется на «Отклонена», и студент получает соответствующее уведомление.

Системные функции

FR-9: Система уведомлений. **Описание:** Система должна автоматически генерировать и отображать уведомления для студента в интерфейсе ЛК при изменении статуса его заявки на «Завершена» или «Отклонена». Уведомление должно содержать номер заявки и ее новый статус. **Предусловие:** Администратор выполняет действие, изменяющее статус заявки на «Завершена» или «Отклонена». **Постусловие:** В интерфейсе ЛК студента появляется новое уведомление. **Критерии приемки:** Сразу после завершения заявки администратором студент видит в своем ЛК уведомление: «Ваша заявка №[ID] завершена».

НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ (NFR)

Наряду с функциональными, система должна соответствовать ряду требований к качеству, производительности, безопасности и другим свойствам, которые описывают, как система выполняет свои функции.

Требования к производительности

- Основные пользовательские операции (открытие списка заявок, создание новой заявки, изменение статуса) должны выполняться со временем отклика не более 3 секунд при нормальной нагрузке.
- Система должна обеспечивать стабильную работу при одновременном доступе не менее 50 пользователей без заметной деградации производительности.

Требования к надежности

- Система должна быть доступна для пользователей 99% времени в течение учебного года. Допускаются плановые перерывы в работе для технического обслуживания, о которых пользователи должны быть уведомлены заранее средствами ЛК.
- Операции, изменяющие данные (создание заявки, смена статуса), должны быть транзакционными, чтобы исключить потерю или повреждение данных в случае сбоя сервера или сети.

Требования к безопасности

- Все данные, передаваемые между клиентским браузером и сервером, должны быть зашифрованы с использованием протокола HTTPS.
- Система должна строго контролировать доступ к данным и функциям на основе ролей. Студент должен иметь доступ только к своим заявкам. Администратор не должен иметь доступа к функциям, предназначенным для студентов (например, создание заявки от своего имени).
- Загружаемые администратором файлы справок должны храниться в защищенной директории вне корневой папки веб-приложения. Прямой доступ к файлам по ссылке должен быть невозможен без предварительной авторизации и проверки прав доступа.

Требования к удобству использования (Usability)

- Интерфейс системы должен быть интуитивно понятным. Процесс заказа справки для нового пользователя не должен требовать изучения инструкций и занимать более 1-2 минут.
- Визуальный стиль (шрифты, цветовая схема, расположение элементов) модуля «Электронные справки» должен соответствовать общему дизайну Личного кабинета университета для обеспечения бесшовного пользовательского опыта.
- Система должна предоставлять пользователям однозначно интерпретируемые сообщения об успехе или ошибках при выполнении ключевых операций (например, «Ваша заявка успешно отправлена» или «Ошибка: не удалось загрузить файл»).

Требования к совместимости

- Система должна корректно отображаться и функционировать в последних двух стабильных версиях следующих веб-браузеров: Google Chrome, Mozilla Firefox, Microsoft Edge, Yandex Browser.

ТРЕБОВАНИЯ К АРХИТЕКТУРЕ И ТЕХНОЛОГИЯМ

Система разрабатывается по клиент-серверной архитектуре. Она будет состоять из двух основных компонентов:

- Клиентское приложение (Frontend). Одностраничное приложение (SPA), работающее в браузере пользователя. Оно отвечает за пользовательский интерфейс, отображение данных и взаимодействие с пользователем.
- Серверное приложение (Backend). Реализует всю бизнес-логику, обработку данных, взаимодействие с базой данных и другими системами. Взаимодействие между клиентской и серверной частями осуществляется через RESTful API.

Backend будет спроектирован с использованием многослойной архитектуры, включающей следующие логические слои:

- Слой представления (Presentation Layer), который отвечает за прием HTTP-запросов и отправку ответов (контроллеры Web API).
- Слой бизнес-логики (Business Logic Layer), который содержит основную логику обработки заявок и управления их жизненным циклом (сервисы).
- Слой доступа к данным (Data Access Layer), который обеспечивает взаимодействие с базой данных (репозитории, Entity Framework Core).

Такая архитектура обеспечивает слабую связанность компонентов, что упрощает разработку, тестирование и дальнейшую поддержку системы.

Для реализации системы должен быть использован следующий стек технологий:

- Язык программирования (Backend): C# 12
- Платформа (Backend): .NET 8
- Фреймворк (Backend): ASP.NET Core Web API 8
- ORM: Entity Framework Core 8
- Фреймворк (Frontend): Angular 14+
- Язык программирования (Frontend): TypeScript
- Система управления базами данных (СУБД): PostgreSQL (версии 15 или выше)

- Брокер сообщений: RabbitMQ (для реализации асинхронных уведомлений)
- Контейнеризация: Docker (для обеспечения унифицированной среды развертывания и разработки)

Система будет взаимодействовать со следующими внешними и смежными системами:

- Личный кабинет университета (ЛК):
 - Интерфейс: Интеграция на уровне аутентификации.
 - Описание: Система не реализует собственную логику входа. Она получает информацию об аутентифицированном пользователе и его роли от основного приложения ЛК.
- Брокер сообщений (RabbitMQ):
 - Интерфейс: Протокол AMQP.
 - Описание: Серверная часть системы будет отправлять сообщения в очередь RabbitMQ при изменении статуса заявки. Другой сервис (или подписчик в рамках этого же приложения) будет обрабатывать эти сообщения для создания уведомлений в интерфейсе ЛК.
- База данных (PostgreSQL):
 - Интерфейс: Драйвер Npgsql.
 - Описание: Взаимодействие с СУБД для выполнения CRUD-операций (Create, Read, Update, Delete) с данными о заявках, пользователях и связанных сущностях.

ТРЕБОВАНИЯ К ДОКУМЕНТАЦИИ

Помимо самого программного продукта, в рамках проекта должна быть подготовлена следующая сопроводительная документация:

- Техническое задание (ТЗ) - настоящий документ, который после утверждения является основным документом, определяющим требования к системе.
- Руководство пользователя - документ, предназначенный для конечных пользователей системы (студентов и администраторов). Он должен содержать пошаговые инструкции с иллюстрациями (скриншотами) для выполнения всех основных операций

ПРИЕМКА И ТЕСТИРОВАНИЕ

Приемка системы осуществляется для проверки ее соответствия требованиям, изложенным в настоящем ТЗ.

Проект считается успешно завершенным и готовым к приемке при выполнении следующих условий:

- Все функциональные требования (FR), перечисленные в разделе 4, полностью реализованы и функционируют корректно.
- Система соответствует всем нефункциональным требованиям (NFR), описанным в разделе 5.
- Отсутствуют критические дефекты, приводящие к аварийному завершению работы системы, потере данных или нарушению безопасности.
- Вся сопроводительная документация, указанная в разделе 7, предоставлена в полном объеме.

Сценарии приемочного тестирования:

ID Тест-кейса	Связанные требования	Краткое описание и шаги	Ожидаемый результат
ТС-1	FR-2, FR-3	Создание заявки студентом. 1. Пользователь с ролью «Студент» входит в ЛК. 2. Переходит в сервис «Электронные справки». 3. Нажимает «Заказать справку» и выбирает формат «Получить скан в ЛК». 4. Подтверждает создание заявки.	В списке заявок студента появляется новая запись со статусом «Новая». В общем списке у администратора также появляется эта заявка.
ТС-2	FR-5, FR-6, FR-7, FR-9	Обработка и завершение заявки. 1. Пользователь с ролью «Администратор» входит в ЛК. 2. Находит заявку из ТС-1. 3. Изменяет ее статус на «В работе». 4. Прикрепляет тестовый PDF-файл и изменяет статус на «Завершена».	Статус заявки корректно изменяется на каждом шаге. После завершения студент получает в ЛК уведомление о готовности справки.
ТС-3	FR-4	Скачивание справки студентом. 1. Студент входит в ЛК после выполнения ТС-2. 2. Находит завершенную заявку. 3. Нажимает кнопку «Скачать».	Начинается загрузка PDF-файла, который был прикреплен администратором.
ТС-4	FR-8, FR-9	Отклонение заявки.	Статус заявки изменяется на

		1. Администратор находит новую заявку. 2. Нажимает «Отклонить». 3. Вводит причину (например, «Тестовое отклонение») 4. Подтверждает действие.	«Отклонена». Студент получает в ЛК уведомление об отклонении с указанием причины.
--	--	--	--