Приложение а

Техническое задание

**Техническое задание**

«проектирование, моделирование и разработка автоматизированной информационной системы система учета успеваемости студентов учебных групп для гбпоу кст».

разработчик:

Студент группы сип 313/23

Барышников владимир сергеевич

На\_\_\_\_\_листах

Согласованно

Руководитель

Курсовой(дипломной) работы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_а.в,павлов

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2025 г.

Москва

2025

Оглавление

[**1.** **Общие сведени** 3](#_Toc208782314)

[**1.1.** **Наименование системы** 3](#_Toc208782315)

[**1.2.** **Основания для разработки** 4](#_Toc208782316)

[**1.3.** **Участники проекта** 4](#_Toc208782317)

[**1.4.** **Сроки выполнения** 4](#_Toc208782318)

[**1.5.** **Этапы разработки:** 4](#_Toc208782319)

[**1.6.** **Источники финансирования** 4](#_Toc208782320)

[**1.7.** **Порядок сдачи результатов** 4](#_Toc208782321)

[**2.** **Назначение и цели создания системы** 4](#_Toc208782322)

[**2.1.** **Назначение системы** 4](#_Toc208782323)

[**2.2 цели создания** 5](#_Toc208782324)

[**3.** **Требования к системе** 5](#_Toc208782325)

[**3.1.** **Функциональные требования** 5](#_Toc208782326)

[**3.2.** **Требования к пользовательскому интерфейсу** 5](#_Toc208782327)

[**3.3.** **Технические требования** 7](#_Toc208782328)

[**3.4.** **Требования к производительности** 8](#_Toc208782329)

[**3.5.** **Требования к интеграции** 8](#_Toc208782330)

[**4.** **Этапы разработки** 8](#_Toc208782331)

[**4.1.** **Анализ и проектирование:** 8](#_Toc208782332)

[**4.2.** **Разработка:** 9](#_Toc208782333)

[**4.3.** **Тестирование:** 9](#_Toc208782334)

[**4.4.** **Внедрение:** 9](#_Toc208782335)

[**4.5.** **Сопровождение (гарантийный период):** 9](#_Toc208782336)

[**5.** **Порядок приемки** 9](#_Toc208782337)

[**5.1.** **Предварительные испытания:** 9](#_Toc208782338)

[**5.2.** **Опытная эксплуатация:** 9](#_Toc208782339)

[**5.3.** **Приемочные испытания:** 9](#_Toc208782340)

[6. **Требования к документации** 9](#_Toc208782341)

[**7.** **Источники разработки** 10](#_Toc208782342)

Таблица 1 – обозначения и сокращения

|  |  |
| --- | --- |
| api (application programming interface | Это набор правил, протоколов и инструментов, который позволяет различным программам и системам обмениваться данными и функциями, |
| Бд | База данных |
| Гбпоу кст. | Государственное бюджетное профессиональное образовательное учреждение города москвы "колледж современных технологий имени героя советского союза м.ф. Панова" |
| Клиент-серверная | Это модель организации вычислительных систем, где клиент запрашивает услуги или данные, а сервер обрабатывает эти запросы и предоставляет ответ. |
| Язык программирования rust | Это современный компилируемый язык программирования, созданный для обеспечения высокой скорости, безопасности и надежности, особенно в системном программировании и высоконагруженных приложениях. |
| Субд | Система управления базой данных |
| Sqlite | Это лёгкая, встраиваемая (внутрипроцессная) реляционная база данных, которая не требует отдельного сервера и хранит всю базу данных в одном файле на диске. |
| Рендеринг | Процесс создания двумерного изображения или анимации из компьютерной модели или сцены |
| Role-based access control | Это метод управления правами доступа к ресурсам, при котором привилегии назначаются не отдельным пользователям, а ролям, соответствующим их должностным обязанностям в организации. |
| **Rest api** | **Архитектурный стиль, или набор правил, для создания веб-сервисов, позволяющий приложениям обмениваться данными по сети, используя стандартные протоколы http.** |
| Json | **Это лёгкий текстовый формат для хранения и обмена структурированными данными, который легко читается человеком и обрабатывается машинами.** |
| Csv | **Это текстовый формат файла для представления табличных данных, где каждая строка представляет запись, а значения в столбцах разделены запятыми или другими разделителями** |
| Ui/ux | **Это комплексное направление в создании цифровых продуктов (сайтов, приложений, сервисов), которое фокусируется на том, чтобы сделать взаимодействие пользователя с продуктом удобным, понятным и привлекательным.** |
| Tauri | **Это фреймворк для создания кросс-платформенных настольных приложений с использованием веб-технологий (html, css, javascript) для фронтенда и языка rust для бэкенда** |
| Фио | Фамилия имя отчество |

# **Общие сведения**

* 1. **Наименование системы** 
     + Полное наименование: проектирование, моделирование и разработка автоматизированной информационной системы «система учета успеваемости студентов учебных групп для ГБПОУ КСТ».
     + Краткое наименование: ИС «учет успеваемости КСТ».
  2. **Основания для разработки**

Работа выполняется на основании:

учебного плана специальности 09.02.07 – информационные системы и программирование – курсовая работа.

* 1. **Участники проекта** 
     + Заказчик:
       - Наименование: колледж современных технологий имени героя советского
       - Союза м. Ф. Панова, учебный корпус ярославский.
       - Адрес: г. Москва, Хибинский проезд 10
       - Контакты: +7 (499) 182-67-62
     + Разработчик:
       - Разработчик: студент группы сип-312/23 Барышников Владимир Сергеевич
       - Адрес фактический: Лубянская площадь, дом 2
       - Контакты: +7 (977) 782-11-27
  2. **Сроки выполнения** 
     + Начало работ: 1 сентября 2025 года.
     + Окончание работ: 21 октября 2025 года.
  3. **Этапы разработки:** 
     + Проектирование – 15.09.25 г.
     + Моделирование – 21.09.25 г.
     + Разработка – 01.10.25 г.
     + Тестирование – 07.10.25 г.
     + Предпроектная защита – 10.10.25 г.
  4. **Источники финансирования**

задание выполняется в рамках курсовой работы. Финансирование осуществляется за счёт средств разработчика.

* 1. **Порядок сдачи результатов**

Результаты работ предоставляются поэтапно. По завершении каждого этапа заказчик получает:

* 1. Отчет о выполнении.
  2. Демонстрацию функционала.

1. **Назначение и цели создания системы** 
   1. **Назначение системы**

Информационная система предназначена для:

* + - Автоматизации учет успеваемости студентов, также обеспечивается контроль над посещаемостью.
    - Обеспечения хранение и удобный доступ к информации.
    - Облегчения выполнения своих обязанностей преподавательского состава ГБПОУ КСТ.
    - Создание отчётов об успеваемости студентов как, в частности, так и в формате группы.
    - Обеспечения разграничения прав доступа к данным в зависимости от роли пользователя.

## **2.2 цели создания**

* Повышение эффективности работы сотрудников учебного учреждения.
* Снижение трудозатрат за счет автоматизации.
* Улучшение качества данных и скорости их обработки.
* Обеспечение безопасности информации внутри учебного заведения.
* Контроль качества обучения в удобном формате информационной системе
* Возможность создание и формирование электронных отчетов, которые внутри информационной системе могут быть доставлены руководителям учебного заведения.

1. **Требования к системе** 
   1. **Функциональные требования** 
      * Основные функции:
        + Выставление отметок в информационную систему.
        + Просмотр выставленных ранее отметок
        + Управление посещаемостью студентов каждой группы.
      * Дополнительные функции**:**
        + Создание отчетов за месяц.
        + Просмотр статистики выставленных отметок для каждой группы отдельно.
   2. **Требования к пользовательскому интерфейсу** 
      * Тип интерфейса: десктоп-клиент, адаптируемый под стандартные размеры мониторов.
      * Дизайн: строгий, лаконичный интерфейс, соответствующий корпоративному стилю образовательного учреждения.
      * Удобство: интуитивно понятный интерфейс с элементами управления, ориентированными на неподготовленного пользователя.
      * Общие сведения
        + Система: «информационная система учёта успеваемости студентов группы ГБПОУ КСТ».
        + Развёртывание: локально на каждом отдельном пк (нет клиент-серверной архитектуры, данные хранятся на локальной бд, например sqlite/ms access).
        + Интерфейс: настольное приложение (windows forms / wpf / qt / javafx — зависит от твоего выбора).
      * Структура интерфейса (основные окна)
        + Главное окно (меню навигации).
        + Окно учёта студентов.
        + Окно учёта посещаемости.
        + Окно учёта оценок.
        + Окно формирования отчётов.
        + Окно «о программе» / справка.
      * Главное окно
        + Кнопка "студенты"
        + Назначение: открывает окно управления списком студентов.
        + Расположение: верхняя панель главного окна, слева.
        + Событие при нажатии: вызывает форму «список студентов».
      * Кнопка "посещаемость"
        + Назначение: переход к учёту посещаемости.
        + Действие: открывает форму «посещаемость».
      * Кнопка "успеваемость"
        + Назначение: переход к окну выставления и просмотра оценок.
        + Действие: открывает форму «оценки».
      * Кнопка "отчёты"
        + Назначение: генерация отчётных документов (word/excel/печать).
        + Действие: открывает форму «формирование отчётов».
      * Кнопка "выход"
        + Назначение: завершение работы программы.
        + Действие: закрывает приложение с подтверждением.

Окно «студенты»

* + - * Кнопка "добавить студента"
      * Назначение: добавить запись о новом студенте.
      * Действие: открывает модальное окно «новый студент» с полями (фио, дата рождения, группа).
    - Кнопка "удалить студента"
      * Назначение: удалить выбранного студента из списка.
      * Действие: при нажатии появляется окно подтверждения удаления.
    - Кнопка "редактировать"
      * Назначение: изменить данные выбранного студента.
      * Действие: открывает форму редактирования с уже заполненными полями.
    - Кнопка "поиск" (рядом с полем ввода)
      * Назначение: фильтрация списка студентов по фио/id.
      * Действие: в таблице остаются только подходящие записи.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* + - Окно «посещаемость»
      * Календарь
      * Назначение: выбор даты занятия.
      * Действие: отображает список студентов и чекбоксы «присутствовал/отсутствовал».
    - Кнопка "сохранить"
      * Назначение: записать данные посещаемости в локальную БД.
    - Кнопка "экспорт"
      * Назначение: выгрузить список посещаемости за период в Excel.

Окно «успеваемость»

* + - * Таблица с фамилиями студентов и колонками по дисциплинам.
      * Кнопка "выставить оценку"
      * Назначение: открыть форму выставления оценки.
      * Поля: студент, предмет, дата, оценка, комментарий.
    - Кнопка "сохранить изменения"
      * Назначение: записать изменения в локальную БД.

Окно «отчёты»

* + - * Кнопка "сформировать отчёт по успеваемости"
      * Назначение: построение сводной таблицы/графика по оценкам студентов.
      * Кнопка "сформировать отчёт по посещаемости"
      * Назначение: построение отчёта за период.
      * Кнопка "экспорт в pdf/excel"
      * Назначение: выгрузка отчёта в файл.

Цветовая палитра:

* Основной фон
  + Название: белый
  + Код цвета: #ffffff
  + Назначение: фон всех основных экранов и форм.
* Вторичный фон
  + Название: дымчато-белый
  + Код цвета: #f5f5f5
  + Назначение: фон карточек, таблиц, панелей.
* Акцентный цвет
  + Название: темный пурпурно-синий
  + Код цвета: #0f172a
  + Назначение: основные кнопки, заголовки, акцентные элементы.
* Вторичный акцент
  + Название: голубо-синий
  + Код цвета: #64748b
  + Назначение: вспомогательные кнопки, подписи.
* Успех
  + Название цвета: зеленый крайола
  + Код цвета: #22c55e
  + Назначение: сообщения об успешном выполнении, зелёные индикаторы.
* Предупреждение
  + Название цвета: рапсово-желтый
  + Код цвета: #f59e0b
  + Назначение: уведомления, требующие внимания.
* Ошибка
  + Название цвета: глубокий желто-розовый
  + Код цвета: #ef4444
  + Назначение: ошибки, кнопки «удалить».
* Текст (основной)
  + Название: тёмно- серо-синий
  + Код цвета: #0f172a
  + Назначение: основной цвет текста.
* Текст (вторичный)
  + Название: перламутрово-ежевичный
  + Код цвета: #6b7280
  + Назначение: комментарии, подписи и менее важный текст.
  1. **Технические требования**
     + Архитектура: клиент-серверная.
     + Серверная часть:
       - Языки программирования: rust.
       - Субд: sqlite.
     + Клиентская часть:
       - Необходимые характеристики: windows 10, доступ к интернету.
       - Окружение: рендеринг осуществляется в веб-вью, предоставляемом операционной системой (webview2 на windows, wkwebview на macos, webkitgtk на linux). Это обеспечивает совместимость и высокую производительность.
     + Безопасность:
       - Аутентификация: по логину и паролю. Пароли хранятся в хэшированном виде с использованием соли (например, с помощью алгоритма bcrypt или argon2).
       - **Авторизация:** на основе ролей (role-based access control), реализованная на уровне ядра (rust).
       - **Защита от уязвимостей:** благодаря использованию rust (память-безопасный язык) и подготовленных запросов к sqlite риски sql-инъекций и ошибок, связанных с памятью, сведены к минимуму. Защита от xss обеспечивается санитизацией данных на стороне ядра перед их возвратом в ui.
       - **Шифрование данных:** шифрование соединения (https) не требуется, так как коммуникация между клиентом и ядром происходит локально через высокопроизводительный ipc-канал. Для шифрования файла базы данных по желанию пользователя могут быть использованы сторонние sqlite-расширения (например, sqlcipher).

## **Требования к производительности**

* + - Время отклика системы: не более 2 секунд.
    - Количество одновременных пользователей: до 40.
    - Объем обрабатываемых данных: до 20 ГБ.
  1. **Требования к интеграции** 
     + **Внешние системы:** на первом этапе интеграция с внешними системами не предусмотрена. Система должна быть спроектирована с возможностью последующего расширения функционала. В будущем интеграция (например, выгрузка данных в 1c) может быть реализована через:
     + **Rest api:** развертывание отдельного минимального http-сервера в фоновом режиме для обслуживания запросов.
     + **Локальные файлы:** экспорт данных в стандартные форматы (csv, json) для последующей обработки внешними системами.
     + **Форматы данных:** для импорта/экспорта данных система должна поддерживать форматы csv и json. Генерация отчетов также может быть реализована в этих форматах.

1. **Этапы разработки** 
   1. **Анализ и проектирование:**
      * Детальный сбор и формализация требований.
      * Проектирование логической и физической структуры базы данных.
      * Создание технического проекта и макетов пользовательского интерфейса (UI/UX).
      * Согласование тз и макетов с заказчиком.
   2. **Разработка:**
      * **Реализация серверной части (ядро / backend)**на rust с использованием фреймворка tauri. Создание api для безопасного взаимодействия с фронтендом.
      * **Проектирование и миграция базы данных** sqlite.
      * **Разработка клиентской части (frontend)** на javascript.
      * **Реализация механизмов аутентификации и авторизации** на стороне rust.
      * **Написание модулей отчётности,** генерирующих данные для последующего отображения во фронтенде или экспорта в файлы.
   3. **Тестирование:**
      * Функциональное тестирование всех модулей системы.
      * Нагрузочное тестирование (в рамках заявленных требований).
      * Тестирование безопасности.
      * Исправление выявленных ошибок.
   4. **Внедрение:**
      * Развертывание системы на тестовом стенде заказчика.
      * Презентация системы и обучение ключевых пользователей (администраторов).
      * Перенос данных из существующих учетных систем (при наличии и возможности).
      * Развертывание на рабочем сервере.
   5. **Сопровождение (гарантийный период):**
      * Техническая поддержка пользователей.
      * Мониторинг работы системы.
      * Исправление критических ошибок, выявленных в процессе эксплуатации.
2. **Порядок приемки** 
   1. **Предварительные испытания:**

Проверка функционала на тестовом стенде разработчика по заранее согласованному с заказчиком тест-плану. Устранение всех критических и существенных замечаний.

* 1. **Опытная эксплуатация:**

Тестирование системы ограниченным кругом пользователей заказчика на тестовом стенде в течение [, например, 14] календарных дней. Фиксация всех замечаний.

* 1. **Приемочные испытания:**

Окончательная проверка системы на рабочем сервере заказчика по полному тест-плану. Подписание акта сдачи-приемки работ после успешного прохождения испытаний и устранения всех замечаний.

1. **Требования к документации**

* Техническая документация:
  + Руководство системного администратора (инструкция по установке, настройке и развертыванию).
  + Описание архитектуры системы и API (при наличии).
  + Описание базы данных (er-диаграмма).
* Пользовательская документация:
  + Руководство пользователя для каждой роли (администратор, преподаватель, лаборант) с пошаговыми инструкциями.
* Отчеты
  + Предоставление отчетов по итогам каждого этапа работ.

1. **Источники разработки**
   * + Федеральный закон № 152-фз «о персональных данных».
     + Внутренние регламенты и документы по учету материально-технических ценностей ГБПОУ КСТ.
     + Настоящее техническое задание.

Приложения:

• макеты интерфейсов (обязательно).

• дополнительные технические спецификации.

Подписи сторон:

Заказчик: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / [фио]

Разработчик: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / [фио]

Дата: \_\_\_\_\_\_\_\_\_\_\_