Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ

Школа бакалавриата

ОТЧЕТ

По проекту
«Разработка системы управления доступом и выдачи ключей с NFC-идентификацией»

по дисциплине «Проектный практикум»

Заказчик:	Серков К. В.
Куратор: Фамилия И.О. ученая степень, ученое звание, должность	Серков К. В.
Студенты команды Hard skills, hard work Фамилия И.О.	Яковлев С. К.
Фамилия И.О.	Сабирова Е. А.
Фамилия И.О.	Казанцев Н. А.
Фамилия И.О	Колотов Н. А.

Екатеринбург, 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 3
1 Аналоги 5
2 Блок-схема процессов. 11
3 Требования к продукту 14
3.1 Функциональные требования 14
3.2 Нефункциональные требования 15
3.3 Интерфейсные требования 15
4 Архитектура продукта 17
4.1 Общая схема архитектуры 17
Детализация аппаратной части (клиентское устройство) 17
4.2 Серверная архитектура 17
4.3 Программная архитектура микроконтроллера 18
4.4 Диаграмма компонентов 19
4.5 Обоснование выбранных решений 19
5 Backlog 21
6 Проектирование и разработка 22
ЗАКЛЮЧЕНИЕ 25
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 27

ВВЕДЕНИЕ

Современные технологии управления доступом становятся неотъемлемой частью повседневной жизни, обеспечивая безопасность и удобство в различных сферах — от офисных помещений до производственных объектов. Однако во многих организациях до сих пор используется устаревший механический метод выдачи ключей, что крайне неудобно, повышает риск потери или несанкционированного доступа. В связи с этим актуальной задачей является разработка автоматизированной системы, которая позволит упростить и повысить безопасность процесса выдачи ключей.

Цель – упростить и автоматизировать процесс выдачи ключей, обеспечив удобный и безопасный доступ к механическим замкам для ограниченного круга пользователей.

Задачи:

– Разработка аппаратной части устройства, включающей NFC-считыватель, клавиатуру для ввода данных, дисплей и механизм выдачи ключей.

– Создание программного обеспечения для микроконтроллера, обеспечивающего идентификацию пользователя, обработку запросов и взаимодействие с сервером через HTTP-запросы.

– Реализация API для обмена данными между устройством и сервером.

– Тестирование и отладка системы.

– Разработка документации, включая чертежи, блок-схемы и инструкции по эксплуатации.

Актуальность проекта обусловлена необходимостью замены устаревших механических систем выдачи ключей на более безопасные и удобные автоматизированные решения. Использование NFC-технологии обеспечивает быструю и надежную идентификацию пользователей, а интеграция с сервером позволяет централизованно управлять доступом и вести учёт операций.

Данный продукт применим в различных местах, включая учебные заведения, офисные и административные здания, гостиницы, производства и так далее.

Планируется получить рабочий прототип устройства, способный идентифицировать пользователя по NFC метке, принимать запрос на выдачу и возврат ключа и взаимодействовать с сервером.

1 Аналоги

Для достижения результата, изначально нужно понять, как он реализован.

В данном проекте могут быть разные механики выдачи ключей, от которых зависит безопасность, оперативность, удобность процесса и автоматизированность.

Будущая ключница может иметь вид, например, вендингового аппарата.

Плюсы подхода:

– Достаточно безопасный вариант, так как пользователю недоступны все ключи сразу, что делает невозможным производство дубликатов ключей, к которым данный сотрудник доступа не имеет.

– Понятность: вендинговый аппарат понятен в механизмах работы пользователю и в изготовлении тоже.

– Быстрое получение ключа.

Минусы:

– Достаточно сложная реализация возврата ключа без человека, у которого есть возможность просто открыть и положить ключ обратно, что не вписывается в рамки автоматизированности и безопасности. Реализация возврата ключа механически потребует работы с манипуляторами и другие трудности

– Реализация выдачи ключа потребует доработки, для того чтобы не случалось ситуаций, когда ключ просто не выпал.

Также ключница может иметь вид индивидуальной для каждого ключа камеры хранения.

Рисунок 1 – Камера хранения

Если говорить конкретно про формат камеры хранения с отдельными ячейками, как почтомат, то:

Плюсы:

– Безусловно безопасно: нет доступа ко всем ключам и будет очень проблематично получить его, применив физическую силу.

– Не нуждается в присутствии третьего лица для возврата ключа на место. Процессы полностью автоматизированы.

– Для пользователя удобная система, простая и быстрая.

Минусы:

– Затратно для производства при большом объеме ключей.

– Достаточно громоздко, не подходит для огромного количества ключей.

Ключница в виде простого ящика с ключами.

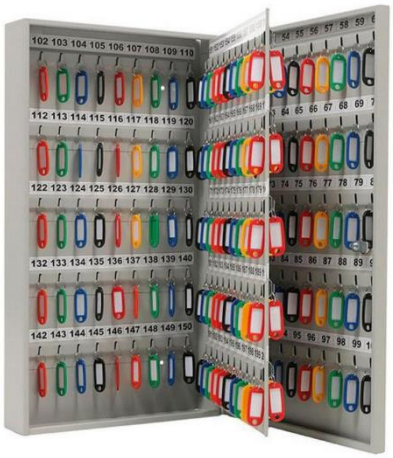

Рисунок 2 – Ключница с общим пространством под все ключи.

Плюсы:

– Очень в реализации и не затратен.

– Может вместить много ключей с использованием слоев.

Минусы:

– Самый небезопасный вариант - при верификации пользователя, которому могут быть не положено иметь доступ к некоторым ключам, он может их получить.

– Выдача ключа не автоматизирована - сотруднику придется вручную искать ключ

– Ключи не гарантированно попадают в свою ячейку, слишком сильный человеческий фактор.

Еще существуют ключницы с фиксатором ключа. (Так же есть вариации с подсветкой, что упрощает процесс для пользователя в поиске и возврате)

Рисунок 3 – Ключница с фиксатором.

Плюсы:

– Невозможно взять любой ключ при открытии ящика. Этим она лучше, чем вариант 3.

Минусы:

– Все еще есть доступ к самим ключам, да их нельзя снять, но можно сделать замеры и сделать дубль - небезопасно.

– Есть возможность срезать трос.

Ключница с фиксатором и брелком с меткой.

Рисунок 4 – Ключница с фиксатором и брелком меткой.

По сути, абсолютно то же самое, но плюсом отслеживание ситуаций, когда возвращают неправильный ключ в слот.

Но тут минус – дополнительные затраты на метки.

Более подходящий вариант - ключница с боксами.

Рисунок 5 – Ключница с боксами.

Плюсы:

– Система безопасна – нет ни визуального доступа к неразрешенным ключам, так и физического.

– Система простая и понятна пользователю.

– Не сильно затратна в ресурсах.

– Полностью автоматизирована

2 Блок-схема процессов.

Для наглядного представления логики работы системы управления доступом и выдачи ключей была разработана блок-схема процессов, которая детализирует последовательность действий устройства от момента идентификации пользователя до выдачи ключа.

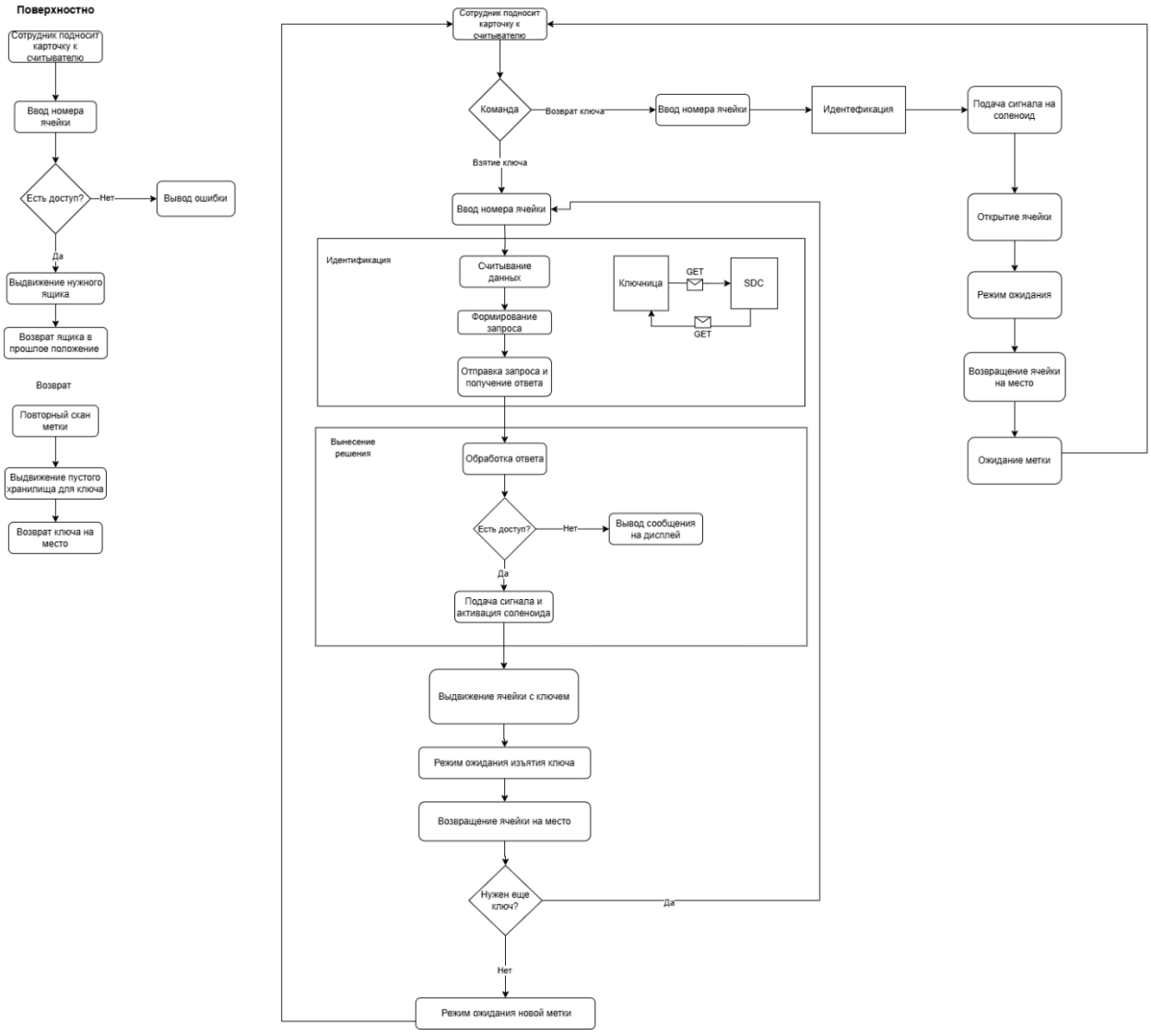

Рисунок 6 – Блок-схема процессов.

Первое - выбор команды.

1) Пользователь прикладывает метку к считывателю. Считыватель достает UID и нужную информацию;

2) Устройство ожидает команду;

3) Пользователь в зависимости нужды нажимает определенную команду: «1» - взять ключ или «2» вернуть.

Теперь распишем процессы.

Кейс «1» - Взять ключ.

1) Устройство ожидает номер ящика;

2) Пользователь набирает номер ячейки;

3) Устройство обрабатывает команду и формирует запрос на сервер;

4) Устройство отправляет запрос на доступ и получает ответ;

Устройство обрабатывает ответ и выносит решение: если нет доступа, то вывод ошибки на дисплей.

Если есть доступ:

1) Подача сигнала на соленоид;

2) Соленоид активирован - ящик доступен для открытия;

3) Пользователь достает ящик и забирает ключ;

4) Пользователь задвигает ящик на место.

Устройство предлагает пользователю взять еще один ключ.

«1» - Да, «2» - Нет, завершить работу.

Кейс «1» - Возвращаемся к началу пункта 2.1

Кейс «2» - Система возвращается к начальному положению - ожиданию метки.

Кейс “2” - Возврат ключа.

1) Устройство ждет ввода номера ячейки;

2) Пользователь вводит номер;

3) Устройство обрабатывает команду и формирует запрос на сервер;

4) Устройство отправляет запрос на доступ и получает ответ.

Устройство обрабатывает ответ и выносит решение:

Нет доступа: Вывод ошибки на дисплей.

Есть доступ:

1) Подача сигнала на соленоид;

2) Соленоид активирован - ящик доступен для открытия;

3) Пользователь достает ящик и возвращает ключ;

4) Пользователь задвигает ящик на место;

5) Система возвращается в режим ожидании метки.

3 Требования к продукту

Требования к разрабатываемой системе управления доступом и выдачи ключей с NFC-идентификацией можно разделить на функциональные, нефункциональные и интерфейсные.

3.1 Функциональные требования

Система должна обеспечивать:

1) Идентификация пользователя

– Считывание NFC-метки/карты для авторизации пользователя.

– Проверка прав доступа пользователя в системе.

2) Ввод и обработка запроса

– Возможность ввода номера ключа/помещения с помощью клавиатуры.

– Отображение информации на дисплее (статус операции, ошибки).

3) Взаимодействие с сервером

– Отправка HTTP-запроса на сервер с данными (ID пользователя, номер ключа).

– Получение и обработка ответа от сервера (разрешение/запрет выдачи ключа).

4) Выдача ключа

– Механическое извлечение запрошенного ключа при успешной авторизации.

– Блокировка выдачи при отказе сервера или ошибке доступа.

5) Возврат ключа

– Отправка HTTP-запроса на сервер с данными (ID пользователя, номер ключа), обработка запроса и возвращение ответа (ключ не в пользовании, отказано в доступе).

– Возврат ключа в ячейку после выдвижения, и возвращение в ячейки в стандартное положение.

6) Обработка ошибок

– Информирование пользователя о проблемах (неверная карта, отсутствие прав, ошибка связи с сервером).

– Возможность повторного ввода данных или отмены операции.

3.2 Нефункциональные требования

Надежность и безопасность

– Защита от несанкционированного доступа (NFC-аутентификация).

Производительность

– Время отклика системы (идентификация + выдача ключа) — не более 3 секунд.

Масштабируемость

– Возможность подключения дополнительных модулей (например, журналирование операций, интеграция с СКУД).

– Поддержка различных типов NFC-карт (Mifare, NFC-F, NFC-A).

Эргономика и дизайн

– Удобный интерфейс (четкий дисплей, интуитивная клавиатура).

– Компактный и прочный корпус, устойчивый к внешним воздействиям.

3.3 Интерфейсные требования

Аппаратные интерфейсы:

– NFC-модуль (RC522).

– Микроконтроллер (ESP32-S3 Графит, Arduino Nano 3 с поддержкой Wi-Fi/HTTP).

– Дисплей (TFT для вывода информации).

– Клавиатура.

– Механизм выдачи ключей (соленоид).

Программные интерфейсы:

– HTTP API для обмена данными с сервером.

– Протоколы связи (SPI/I2C для периферии, Wi-Fi/Ethernet для подключения к сети).

4 Архитектура продукта

4.1 Общая схема архитектуры

Система построена по модульному принципу с четким разделением функций между аппаратной частью (клиентское устройство) и программной составляющей (сервер). Взаимодействие между компонентами осуществляется через REST API с использованием HTTP-запросов.

Детализация аппаратной части (клиентское устройство)

Основные компоненты:

Микроконтроллер ESP32-S3 (выбран по критериям):

– Встроенные модули Wi-Fi и Bluetooth

– Поддержка SPI/I2C/UART

– Достаточное количество GPIO

– Низкое энергопотребление

– Большое количество памяти

NFC-модуль RC522:

– Частота работы: 13.56 МГц

– Поддержка протоколов: ISO 14443A/MIFARE

– Интерфейс связи: SPI

– Дальность считывания: до 3 см

Механизм выдачи ключей: соленоид.

Периферия:

– TFT-дисплей 320×240 (SPI)

– Матричная клавиатура 4×4

4.2 Серверная архитектура

Компоненты сервера:

Backend написан на Python 3.12 с помощью Flask. Содержит два эндпоинта – для запросов получения и возврата ключа.

Использовалась база данных SQLite, с таблицей пользователей с полями: id, full_name, serial, access, taken_keys.

API-эндпоинты:

POST /return_key – запрос на возврат ключа

– Вход: {serial, key}

– Выход: {access: true/false, message}

POST /issue_key - запрос на выдачу ключа

– Вход: {serial, key}

– Выход: {access: true/false, message}

4.3 Программная архитектура микроконтроллера

Слои приложения:

Драйверы оборудования:

– SPI для RC522

– SPI для TFT

– GPIO для клавиатуры/сервопривода

Библиотеки:

– MFRC522.h - работа с NFC

– Ethernet - подключение к сети

– HTTPClient.h - отправка запросов

Поток данных:

1) Инициализация периферии

2) Подключение к Ethernet

3) Основной цикл:

– Ожидание NFC-карты

– Ввод нужного ключа

– Отправка запроса на сервер

– Обработка ответа

– Активация механизма выдачи

4.4 Диаграмма компонентов

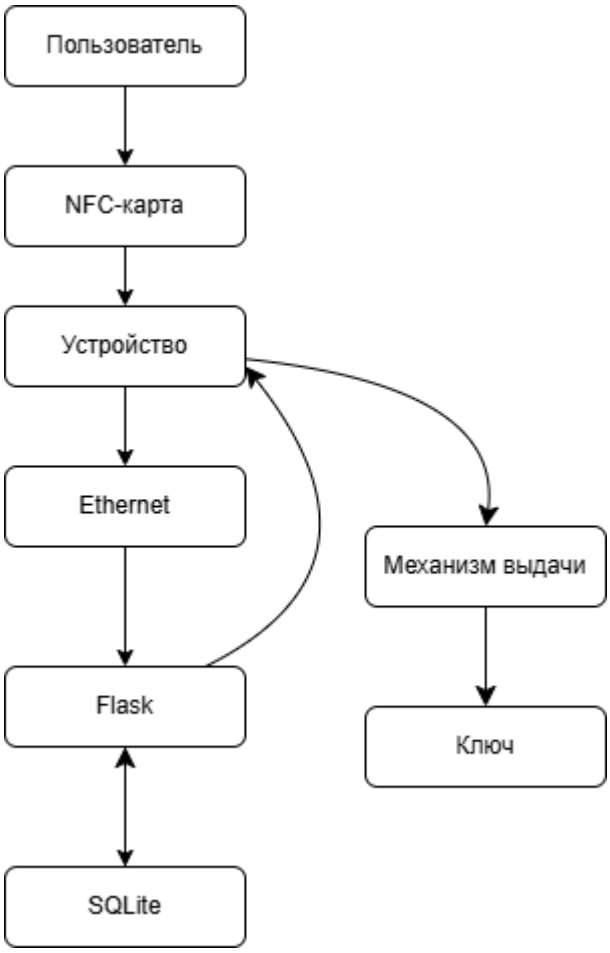

Рисунок 7 – Диаграмма компонентов.

Рабочий цикл системы начинается с идентификации пользователя по NFC-карте. Контроллер формирует HTTP-запрос к серверу, который проверяет права доступа в базе данных SQLite. В зависимости от результата проверки сервер либо разрешает выдачу ключа через активацию соленоида, либо возвращает сообщение об отказе.

4.5 Обоснование выбранных решений

1) ESP32-S3 вместе с Arduino:

– Встроенный Wi-Fi на ESP32-S3

– Больше памяти (512KB SRAM)

– Поддержка большего количества пинов

2) Flask для сервера:

– Большое комьюнити, много информации

– Легкая разработка

3) SPI для NFC:

– Высокая скорость (до 10 МГц)

– Минимальная загрузка процессора

– Стабильная работа библиотек

Архитектура обеспечивает безопасность через шифрование данных и валидацию запросов, устойчивость к сбоям за счет механизмов автоматического восстановления, а также масштабируемость для будущего расширения функционала. Система допускает интеграцию дополнительных модулей аутентификации, подключение к корпоративным системам и развитие интерфейсов управления.

5 Backlog

Таблица 1 – Backlog проекта

Приоритет	Задача	Ответственный	Статус
Высокий	Формирование требований к системе	Все	Выполнено
Высокий	Выбор компонентов и составление закупочного листа	Яковлев Савелий	Выполнено
Высокий	Отрисовка 3D модели ящика	Сабирова Екатерина	Выполнено
Высокий	Написание кодов подключения составляющих	Колотов Никита	Выполнено
Высокий	Составление чертежей подключения	Казанцев Никита	Выполнено
Высокий	Написание сервера, обрабатывающего запросы	Казанцев Никита	Выполнено
Высокий	Сборка компонентов устройства управления	Яковлев Савелий	Выполнено
Высокий	Написание ПО обработки и связи с сервером	Колотов Никита	В процессе
Высокий	Пайка мосфета и соленоида	Яковлев Савелий	Выполнено
Высокий	Тестовая сборка ячейки и отладка	Яковлев Савелий	В процессе
Средний	Отрисовка 3D модели полного устройства	Сабирова Екатерина	Выполнено
Средний	Логирование запросов на сервер	Казанцев Никита	В процессе
Средний	Масштабирование на 2 ячейки	Яковлев Савелий	To Do
Низкий	Финальная сборка и тестирование	Яковлев Савелий	To Do

6 Проектирование и разработка

В процессе разработки системы была применена гибкая методология с динамическим уточнением требований. Планирование задач поверхностно было произведено в самом начале проекта, а их детализация происходила походу разработки. Это позволило совместить тщательное планирование с гибкостью внесения изменений.

Разработка велась поэтапно, начиная с аппаратного прототипа, где последовательно подключались и тестировались: экран, модуль NFC, дисплей, клавиатура, модуль Ethernet. В процессе выявились несколько ключевых проблем, такие как недостаточность пинов, которая была решена увеличением количества микропроцессоров – образованием кластера, нестабильная работа отечественной версии ESP32-S3 «Графит».

Разработка ПО проводилась на языке C++ в Arduino IDE. Использовались такие библиотеки как Wire, SPI, MFRC522, Adafruit_GFX, Adafruit_ST7789, KeyPad и так далее.

Параллельно этому велись отрисовки 3D моделей в Blender.

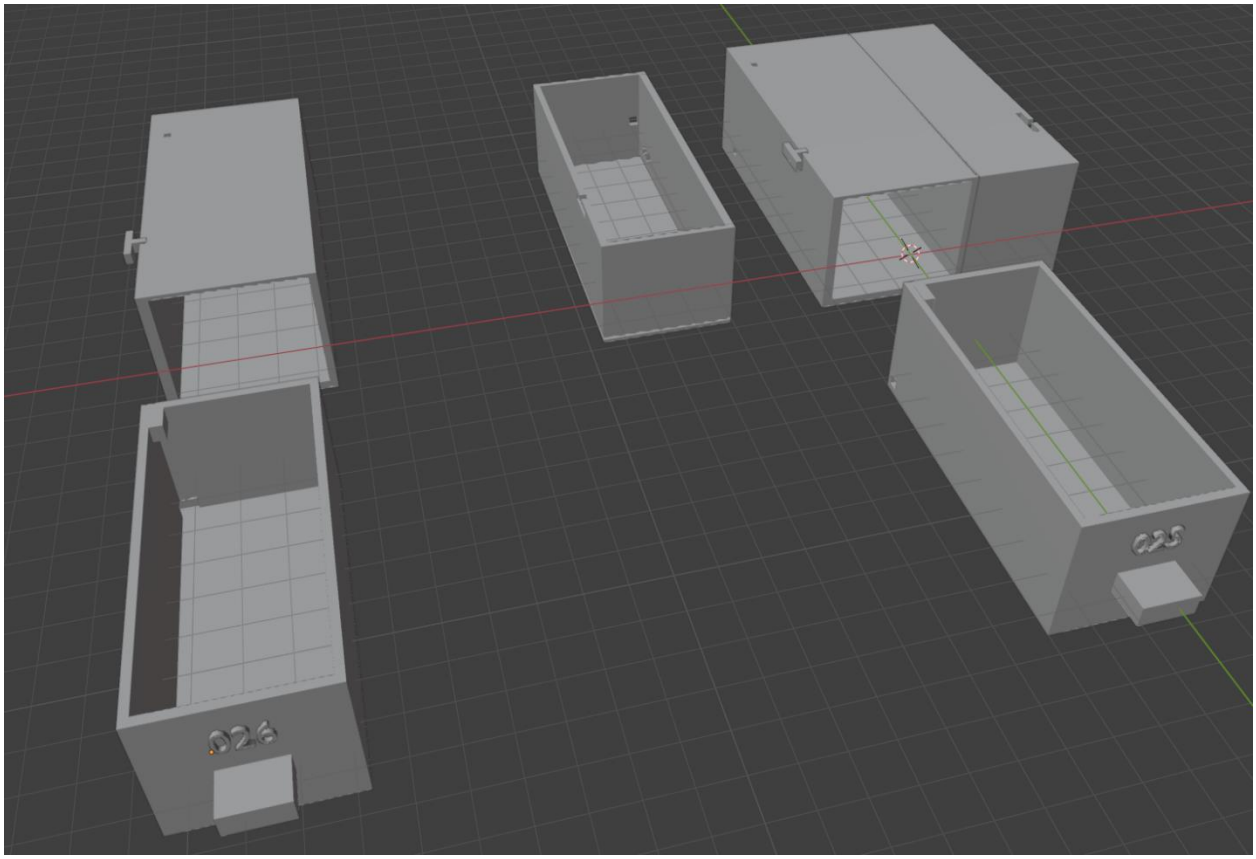

Рисунок 8 – 3D модель ящика и модульного соединения.

Также параллельно этому велась разработка и тестирование сервера с помощью Insomnia.

Изначально код мог только обработать запрос на проверку доступа, и вернуть {access: true/false, message: «Доступ разрешен»/ «Доступ запрещен»}, в базе данных не было поля taken_keys. В ходе процесса разработки была выявлена проблема несовместимых версий библиотек, которая решилась установкой подходящей, и проблема несопоставимых типов данных, решившаяся изменением в коде.

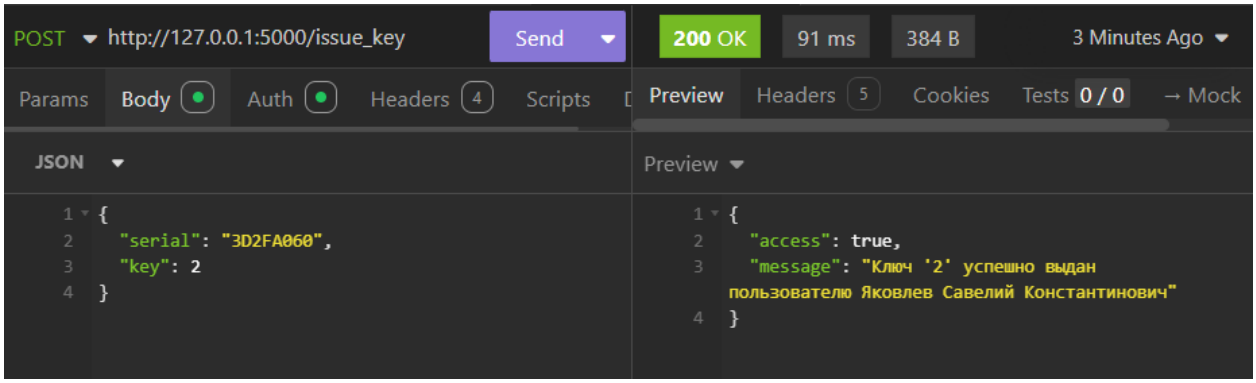

Рисунок 9 – Тест обработки запроса на изъятие ключа.

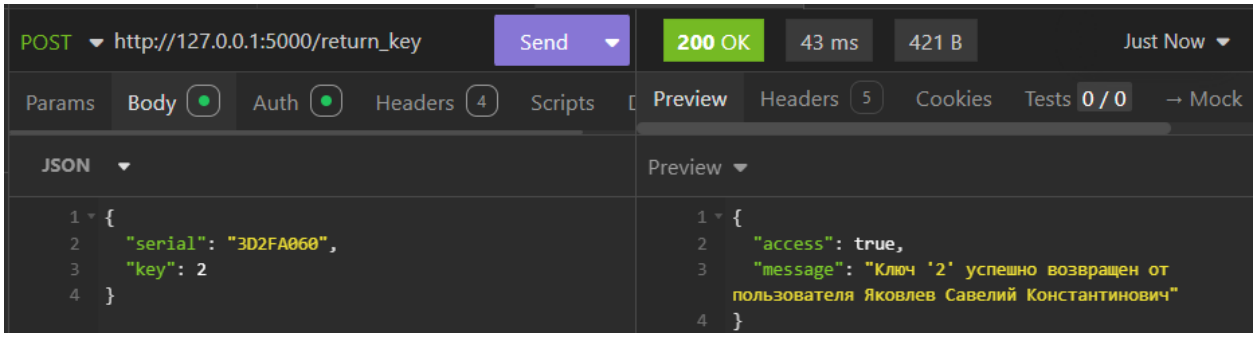

Рисунок 10 – Тест обработки запроса на возврат ключа.

Тестирование показало, что запросы обрабатываются успешно и достаточно быстро. Так же есть обработка неправильного типа данных, отсутствия серийного номера в базе данных, кейса, когда ключ уже в пользовании, и ошибок сервера и базы данных.

ЗАКЛЮЧЕНИЕ

В ходе разработки проекта был создан комплекс программных модулей, интегрированных в единую систему управления доступом и выдачи ключей с NFC-идентификацией.

При реализации проекта были выявлены и успешно решены следующие основные проблемы:

1) Идея реализации: были выдвинуты множество идей реализации проекта, подходящие под общие требования, но в итоге выбран способ.
2) API: был разработан API для взаимодействия с сервером.
3) 3D Модели: отрисованы итоговые модели и одобрены для печати.
4) Сборка: огромное количество работы было проведено и попыток предпринято, но в итоге была собрана система из микроконтроллеров и других составляющих, система работает.
5) Разработка ПО: было написано по на языке C++, которое выполняет требуемые функции.

Разработанная система управления доступом и выдачи ключей с NFC-идентификацией в целом соответствует ключевым требованиям заказчика и пользователей:

– Автоматизация выдачи ключей достигнута за счет интеграции NFC-считывателя, микроконтроллера и серверного API.

– Контроль доступа реализован через проверку прав пользователя в базе данных перед выдачей ключа.

– Удобство использования обеспечено интуитивным интерфейсом (дисплей, клавиатура) и быстрой обработкой запросов.

Однако не все требования выполнены в полной мере:

– Журналирование операций (дополнительный функционал) пока не реализовано из-за ограничений по срокам.

– Не выполнено масштабирование на 10 ячеек, так же вследствие сжатых сроков.

Вывод: Система готова к пилотному внедрению, но требует оптимизации для промышленной эксплуатации.

Анализ качества:

– Надежность: Система стабильно работает в нормальных условиях, но требует доработки для работы в неидеальных средах (плохой интернет, механические нагрузки).

– Безопасность: NFC-аутентификация обеспечивают базовый уровень защиты, но необходимо добавить защиту от подбора карт (например, лимит попыток) и шифрование перейдаваемых данных.

– Юзабилити: Интерфейс понятен пользователям, но требует улучшения визуальной обратной связи.

Проект еще может быть улучшен и развит:

– Оптимизация связи: например, очередь запросов с повторной отправкой при сбоях или использование других более мощных составляющих.

– Доработка механизма выдачи, то есть замена соленоида на что-то более надежное и удобное.

– Масштабирование: увеличение масштабов ключницы на десятки или сотни ячеек.

– Расширение функционала: мобильное приложение для удаленного управления, интеграция с другими корпоративными системами.

– Повышение безопасности: усложнение аутентификации, шифрование данных.

Проект успешно решает задачу автоматизации выдачи ключей, но требует доработок для повышения надежности и безопасности. Дальнейшее развитие системы должно включать как исправление текущих недостатков, так и добавление новых функций для расширения области применения. Полученный опыт показывает важность тестирования в реальных условиях и модульного подхода к проектированию.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Random Nerd Tutorials [Электронный ресурс]. URL: https://randomnerdtutorials.com/esp32-s3-devkitc-pinout-guide/ (дата обращения: 26.05.2025).

2. RobotClass [Электронный ресурс]. URL: https://robotclass.ru/articles/graphites3/ (дата обращения: 26.05.2025).

3. RoboCraft [Электронный ресурс]. URL: https://robocraft.ru/arduino/518 (дата обращения: 26.05.2025).

4. Espressif Systems Documentation [Электронный ресурс]. URL: https://docs.espressif.com/projects/esp-idf/en/stable/esp32s3/api-reference/peripherals/spi_master.html (дата обращения: 26.05.2025).

5. AlexGyver [Электронный ресурс]. – URL: https://alexgyver.ru/arduino-rfid/ (дата обращения: 26.05.2025).

6. Bilibili [Электронный ресурс]. – URL: https://www.bilibili.com/video/BV1Z14y1b7M4/ (дата обращения: 26.05.2025).

7. Flask Documentation [Электронный ресурс]. URL: https://flask.palletsprojects.com/en/stable/ (дата обращения: 26.05.2025).

8. Blender.org [Электронный ресурс]. URL: https://www.blender.org/ (дата обращения: 26.05.2025).