

Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Национальный исследовательский университет ИТМО»  
Факультет Программной Инженерии и Компьютерной Техники

**Курсовая работа на тему**  
**«Создание системы для контроля кондиционеров»**  
по дисциплине «Информационные системы»

Выполнили:  
Денисова Алёна Александровна  
Калиев Александр Дмитриевич

Группа: Р3311

Проверил:  
Тюрин Иван Николаевич

Санкт-Петербург, 2025

# Contents

1. Задание .....	3
1.1. Предметная область, для которой будет разрабатываться информационная система .....	3
1.2. Подробное текстовое описание предметной области .....	3
1.3. Назначение и задачи системы .....	4
1.4. Функциональные и нефункциональные требования .....	4
1.5. Модели основных прецедентов .....	6
1.6. Архитектура будущей системы .....	8

# 1. Задание

## 1.1. Предметная область, для которой будет разрабатываться информационная система

Согласована предметная область: *система контроля кондиционеров.*

## 1.2. Подробное текстовое описание предметной области

Система контроля кондиционеров предназначена для централизованного управления кондиционерами, установленных в различных зданиях и помещениях.

*Основная цель:* обеспечить удобный мониторинг и настройку работы климатического оборудования через веб-приложение.

Каждый кондиционер закрепляется за конкретным зданием и комнатой в этом здании.

Для кондиционеров предусмотрены следующие возможности управления:

- *Ручное управление* – включение/выключение, выбор режима работы, задание температуры.
- *Расписание работы* – пользователь может настроить временные интервалы, в которые кондиционер будет автоматически включаться или выключаться, а также задавать разные режимы работы в зависимости от времени суток или дня недели.
- *Мониторинг состояния* – отображение текущего статуса (включен/выключен, активный режим, установленная температура).

Система также интегрируется с температурными датчиками, установленными в помещениях. Эти датчики позволяют получать информацию о текущей температуре воздуха в реальном времени.

Для удобства пользователей предусмотрены:

- просмотр списка зданий, комнат и установленных в них кондиционеров;
- отображение информации о каждом кондиционере и датчике;
- возможность управлять отдельными кондиционерами или сразу группами (например, всеми устройствами в комнате или здании);
- централизованное редактирование и хранение расписаний.

### 1.3. Назначение и задачи системы

Информационная система для контроля кондиционеров необходима для централизованного управления кондиционерами в зданиях и помещениях. Она упрощает процесс эксплуатации, позволяет эффективно организовать работу кондиционеров и обеспечивает экономию ресурсов за счет автоматизации.

#### Задачи, которые решает система:

- *Централизованное управление:* возможность управлять всеми кондиционерами через единое веб-приложение, без необходимости физического доступа к каждому устройству.
- *Автоматизация работы оборудования:* настройка расписаний включения и выключения кондиционеров.
- *Мониторинг состояния и параметров:* отображение текущего состояния кондиционеров; получение информации с температурных датчиков в реальном времени.
- *Оптимизация энергопотребления:* предотвращение работы кондиционеров вне необходимости.
- *Управление иерархией объектов:* структурированное хранение информации о зданиях, комнатах и установленных кондиционерах; групповые операции.

### 1.4. Функциональные и нефункциональные требования

#### *Функциональные требования*

- **FR01.** Система должна предоставлять возможность подачи заявки на получение аккаунта.
- **FR02.** Система должна предоставлять администраторам возможность одобрения нового аккаунта.
- **FR03.** Система должна предоставлять возможность регистрации с введением кода приглашения.
- **FR04.** Система должна предоставлять администраторам возможность добавлять здание и комнату.
- **FR05.** Система должна предоставлять администраторам возможность добавить кондиционер в комнату.
- **FR06.** Система должна предоставлять администраторам возможность включить или выключить кондиционер.

- **FR07.** Система должна предоставлять администраторам возможность выбрать режим работы кондиционера.
- **FR08.** Система должна предоставлять администраторам возможность создать расписание работы кондиционера.
- **FR09.** Система должна предоставлять администраторам возможность просматривать список зданий, комнат и кондиционеров.
- **FR10.** Система должна предоставлять администраторам возможность просматривать текущее состояние кондиционера.
- **FR11.** Система должна предоставлять администраторам возможность редактировать и удалять расписания кондиционера.

### ***Нефункциональные требования***

#### **Usability (удобство использования)**

- **U01.** Система должна обеспечивать единообразие дизайна и расположения элементов на всех страницах сайта.
- **U02.** Система должна предоставлять пользователям возможность быстро находить нужное здание, комнату или кондиционер через поиск/фильтры.

#### **Reliability (надёжность)**

- **R01.** Система должна сохранять все данные о кондиционерах и расписаниях при сбоях.
- **R02.** Система должна предоставлять пользователям возможность безопасного входа в систему даже при высоких нагрузках.
- **R03.** Система должна предоставлять возможность отмены или подтверждения ключевых операций редактирования/удаления.
- **R04.** Система должна хешировать пароли при помощи bcrypt.

#### **Performance (производительность)**

- **P01.** Система должна обеспечивать обработку пользовательских запросов не более чем за 2 секунды.
- **P02.** Система должна предоставлять возможность одновременной работы не менее чем 50 пользователей.
- **P03.** Система должна обеспечивать время загрузки главной страницы < 5 секунд для 95% пользователей.
- **P04.** Система должна предоставлять возможность обновления данных с датчиков температуры в реальном времени.

#### **Design Constraints (ограничения разработки)**

- **D01.** Для разработки бэкенда использовать Java, Spring Framework с необходимыми модулями.

- **D02.** В качестве СУБД использовать PostgreSQL.
- **D03.** Для разработки фронтенда использовать TypeScript, библиотеку React, Tailwind CSS.
- **D04.** Для связи бэкенда и фронтенда использовать протокол HTTP, спецификация REST.

## 1.5. Модели основных прецедентов

<b>Прецедент:</b> Регистрация пользователя в системе
<b>ID:</b> 1
<b>Краткое описание:</b> Пользователь создаёт новый аккаунт для работы в системе.
<b>Главные акторы:</b> Пользователь
<b>Предусловия:</b> Пользователь не зарегистрирован в системе.
<b>Основной поток:</b> <ol style="list-style-type: none"> <li>1. Пользователь отправляет заявку на регистрацию, вводя свой email.</li> <li>2. Администратор одобряет создание аккаунта.</li> <li>3. Система отправляет пользователю электронное письмо с кодом приглашения.</li> <li>4. Пользователь регистрируется в системе, вводя код приглашения и создавая пароль.</li> <li>5. Система сохраняет пользователя в базе данных.</li> </ol>
<b>Постусловия:</b> Новый аккаунт создан и готов к использованию.

<b>Прецедент:</b> Управление кондиционером
<b>ID:</b> 2
<b>Краткое описание:</b> Администратор вручную управляет состоянием кондиционера.
<b>Главные акторы:</b> Администратор
<b>Предусловия:</b> Администратор вошёл в систему и имеет доступ к комнате с кондиционером.

<b>Основной поток:</b> <ol style="list-style-type: none"> <li>1. Администратор выбирает здание и комнату.</li> <li>2. Администратор выбирает конкретный кондиционер.</li> <li>3. Администратор задаёт действие: включить/выключить, изменить режим или температуру.</li> <li>4. Система отправляет команду на устройство.</li> <li>5. Система обновляет интерфейс и отображает новое состояние.</li> </ol>
<b>Постусловия:</b> Параметры кондиционера изменены в соответствии с действиями администратора.

<b>Прецедент:</b> Создание расписания работы кондиционера
<b>ID:</b> 3
<b>Краткое описание:</b> Администратор автоматизирует работу кондиционера через расписание.
<b>Главные акторы:</b> Администратор
<b>Предусловия:</b> Администратор аутентифицирован, кондиционер добавлен в систему.
<b>Основной поток:</b> <ol style="list-style-type: none"> <li>1. Администратор выбирает необходимый кондиционер из списка.</li> <li>2. Администратор нажимает кнопку “Добавить расписание”.</li> <li>3. Администратор выбирает параметры расписания.</li> <li>4. Система сохраняет расписание для кондиционера.</li> </ol>
<b>Постусловия:</b> В системе создано расписание для выбранного кондиционера.

<b>Прецедент:</b> Получение данных с датчиков температуры
<b>ID:</b> 4
<b>Краткое описание:</b> Система получает и отображает актуальные показания датчиков.
<b>Главные акторы:</b> Система, Датчик температуры

<b>Предусловия:</b> Датчики подключены и передают данные.
<b>Основной поток:</b> <ol style="list-style-type: none"> <li>1. Датчик фиксирует текущую температуру.</li> <li>2. Датчик отправляет данные в систему.</li> <li>3. Система принимает данные и обновляет значения в базе.</li> <li>4. Система отображает данные пользователю в интерфейсе.</li> </ol>
<b>Постусловия:</b> Пользователь видит актуальную температуру в выбранной комнате.

## 1.6. Архитектура будущей системы

### Стек технологий:

- **Backend:** Java 21, Spring Boot, Gradle
- **Frontend:** React, Redux, TypeScript
- **СУБД:** PostgreSQL

### Диаграмма развертывания:

