**Содержание отчета**

**1. Постановка задачи**

Система видеонаблюдения должна обеспечивать эффективное управление видеопотоками, их запись и хранение. Основные архитектурные требования включают:

* Высокая производительность для обработки нескольких видеопотоков.
* Надежность и доступность системы.
* Удалённый доступ к видеозаписям через сеть.

**2. Описание программных средств**

Для проектирования и построения моделей используется:

* **UML-диаграммы**: инструмент для визуализации систем.
* **StarUML** (версия 3.1): разработчик StarUML Team, доступен на [официальном сайте].
  + **Режим использования**: настольное приложение.
  + **Платформы**: Windows, macOS, Linux.
  + **Модели**: поддержка UML 2.x, включая диаграммы развертывания и компонентов.

**3. Описание практического задания**

**Список компонентов подсистемы**

1. **Камера**
   * **Технология реализации**: IP-камера с поддержкой H.264.
2. **Рекордер**
   * **Технология реализации**: NVR (Network Video Recorder), программное обеспечение на базе Linux.
3. **Сервер управления**
   * **Технология реализации**: веб-сервер, разработанный на Python с использованием Flask.
4. **Клиентское приложение**
   * **Технология реализации**: веб-приложение на React.

**Существующие связи между компонентами подсистемы**

* **Камера** ---> **Рекордер** (видеопоток)
* **Рекордер** ---> **Сервер управления** (данные о записях)
* **Сервер управления** <--- **Клиентское приложение** (интерфейс пользователя)

**Интерфейсы для взаимодействия**

* REST API для связи между клиентским приложением и сервером управления.
* RTSP (Real-Time Streaming Protocol) для передачи видеопотока от камер к рекордеру.

**Список физических устройств**

1. **Камера**
   * **Характеристики**: разрешение 1080p, поддержка Wi-Fi.
2. **Рекордер**
   * **Характеристики**: 4 канала, 2 ТБ HDD.
3. **Сервер**
   * **Характеристики**: 16 ГБ RAM, 4 ядра CPU, 500 ГБ SSD.
4. **Клиентское устройство**
   * **Характеристики**: ПК или ноутбук с браузером.

**Соединения – протоколы связи между устройствами**

* **Камера** <--> **Рекордер**: RTSP
* **Рекордер** <--> **Сервер управления**: HTTP/HTTPS
* **Клиентское приложение** <--> **Сервер управления**: REST API через HTTP/HTTPS

**Среды выполнения**

* **Рекордер**: Linux
* **Сервер управления**: Linux с установленным Flask
* **Клиент**: любой браузер

**Размещение компонентов на устройствах**

* **Камера** размещается на месте наблюдения.
* **Рекордер** устанавливается в серверной комнате.
* **Сервер управления** размещается на выделенном сервере.
* **Клиентское приложение** доступно на любом клиентском устройстве.

**Диаграммы**

