

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

Кафедра

вычислительной техники

«Вычислительная техника»

Специальность: 09.05.01 «Применение и эксплуатация
автоматизированных систем специального
назначения»

Специализация: «Эксплуатация вычислительных машин, комплексов,
систем и сетей специального назначения»

ДИПЛОМНЫЙ ПРОЕКТ

на тему

Система управления медиаданными

Студент Кис Киселев Александр Константинович
(подпись, дата) (ФИО полностью)

Руководитель Федюнин Р.Н.
(подпись, дата) (фамилия, инициалы)

Нормоконтролёр Кучин А.В.
(подпись, дата) (фамилия, инициалы)

Рецензент Васильков А.В.
(должность, место работы) (подпись, дата) (фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 10.06.2020 г. № 13)

Заведующий кафедрой Митрохин М.А.
(подпись) (фамилия, инициалы)

Работа защищена с отметкой хорошо (протокол заседания ГЭК от
18.06.2020 № 5)

Секретарь ГЭК Гурин Е.И.
(подпись) (фамилия, инициалы)

Пенза 2020

Федеральное государственное бюджетное образовательное учреждение
высшего образования
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет

Вычислительной техники

Кафедра

Вычислительная техника

«Утверждаю»

Заведующий кафедрой

 Митрохин М.А.

«30» апреля 2020г.

**ЗАДАНИЕ
НА ДИПЛОМНЫЙ ПРОЕКТ**

1. Студент Киселев Александр Константинович гр. 15BO2 факультета вычислительной техники.

2. Тема работы: «Система управления медиаданными».

Тема утверждена приказом ПГУ № 0335а/о от 30.04.2020 г.

3. Руководитель работы к.т.н., доцент каф. ВТ ПГУ Федюнин Р.Н.

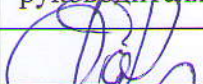
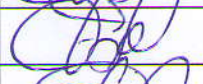
4. Задание на работу (назначение разработки, исходные данные и т.п.).

Назначение разработки – разработка сервиса, который содержит клиентское мобильное приложение и серверное web-приложение для взаимодействия с видео контентом на базе облачного хранилища Amazon. Перечень услуг разрабатываемого сервиса может регулярно обновляться. Для чего помимо клиентского приложения и облачного сервиса, приложение содержит прослойку веб-сервера. Основным функционалом приложения является обработка видео и аудио информации.

5. Перечень подлежащих разработке вопросов.

Требуется произвести анализ предметной области, выбрать функциональный состав системы, осуществить выбор стека технологий для реализации системы, разработать алгоритм работы системы, разработать структурную схему системы, разработать диаграммы: прецедентов, активностей, последовательностей, объектов, классов, с последующим написанием программы. Также выполнить задание по БЖД и экономике.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Анализ предметной области	10	9.05.20	
2.	Выбор компонентов для реализации принципиальной схемы	10	16.05.20	
3.	Составление принципиальной схемы устройства	20	16.05.20	
4.	Написание программы-прошивки для устройства	10	23.05.20	
5.	Отладка системы	10	23.05.20	
6.	Моделирование и поиск уязвимостей, исправление ошибок в работе системы	10	31.05.20	
7.	Оформление выпускной работы	30	6.06.20	

Дата выдачи задания 30 «апреля» 2020 года.

Руководитель дипломного проекта


(подпись, дата)

Федюнин Р.Н.

Задание к исполнению принял студент


(подпись, дата)

Киселев А.К.

Работу к защите допустить

Декан факультета


(подпись, дата)

Фионова Л.Р.

РЕФЕРАТ

Дипломная работа содержит 137 листов, 70 рисунков, 5 источников, 3 приложения.

ОБЛАЧНАЯ СИСТЕМА, КЛИЕНТСКОЕ ПРИЛОЖЕНИЕ, СЕРВИСНЫЕ ФУНКЦИИ, БЕССЕРВЕРНАЯ АРХИТЕКТУРА.

Цель работы – разработать масштабируемую, отказоустойчивую систему загрузки, воспроизведения и руководства видео контентом на базе Amazon.

Объектом разработки является программное приложение, обеспечивающее пользовательский доступ к каталогу видео и аудио контенту, администраторам доступ к настройкам системы, а программистам к инструментам видеопотока на базе облачного сервиса.

В результате проведенной работы должен быть разработан проект, удовлетворяющий поставленной задаче, а именно, создан программный комплекс – мобильное приложение клиент, web-сервис бизнес логики и облачное хранилище, и разработаны соответствующие алгоритмы, реализующие возможность хранить, обрабатывать и доставлять до потребителя информацию в виде видео потока.

Данное приложение рекомендуется использовать для корпоративного вещания. Система может быть внедрена в сегменте пресс службы и рекламы корпоративного сегмента.

					ПГУ 1.090501.17.001 ПЗ					
Изм.	Лист	№ докум.	Подпись	Дата	Система управления медиаданными Пояснительная записка			Лит.	Лист	Листов
Разраб.		Киселев А.К.								
Провер.		Федюнин Р.Н.							2	137
Н. Контр		Кучин А.В.						ФВТ, каф. ВТ гр. 15В02		
.										
Утверд.										

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	7
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	6
1.1 Существующие решения	6
1.2 Сравнение площадок для размещения видео-контента	7
1.3 Сравнение архитектурных решений реализации приложения	9
1.4 Сравнение программных решений реализации приложения	13
1.5 Сравнение языков программирования	19
2 РАЗРАБОТКА СИСТЕМЫ	23
2.1 Реализация интерфейса регистрации и входа в систему	24
2.2 Реализация функций видеоплеера	36
2.3 Реализация интерфейса видеоплеера	45
2.4 Описание работы интерфейса системы	52
3 ТЕХНОЛОГИЯ ХРАНЕНИЯ МЕДИАКОНТЕНТА В БЕССЕРВЕРНЫХ ПРИЛОЖЕНИЯХ	62
3.1 Создание Bucket	62
3.2 Добавление объектов	Error! Bookmark not defined.
3.3 Права доступа	65
4 ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ЭФФЕКТИВНОСТИ ИНВЕСТИЦИЙ ВО ВНЕДРЕНИЕ СИСТЕМЫ УПРАВЛЕНИЯ МЕДИАДААННЫМИ	69
4.1 Введение	69
4.2 Расчет затрат на разработку программного обеспечения	71
4.3 Расчет цены разработанной программы	Error! Bookmark not defined.
4.4 Расчет капитальных вложений	Error! Bookmark not defined.
4.5 Расчет эксплуатационных расходов	Error! Bookmark not defined.
4.6 Определение расходов на интернет трафик	Error! Bookmark not defined.
4.7 Заключение	78

5 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ В ПОМЕЩЕНИЯХ ВЫЧИСЛИТЕЛЬНЫХ ЦЕНТРОВ	Error! Bookmark not defined.
5.1 Обеспечение электро-безопасности	79
5.1.1 Классификация помещений по степени опасности поражения людей электрическим током	Error! Bookmark not defined.
5.1.2 Типы заземляющих устройств	Error! Bookmark not defined.
5.1.3 Требования к величине сопротивления заземляющих устройств	Error! Bookmark not defined.
5.1.4 Оборудование, подлежащее защитному заземлению	Error! Bookmark not defined.
5.1.5 Пример электроустановки, подключенной к сети с изолированной нейтралью.....	Error! Bookmark not defined.
5.2 Расчёт заземляющего устройства	Error! Bookmark not defined.
5.3 Расчёт сопротивления заземляющего устройства	88
5.4 Обеспечение противопожарной безопасности.....	91
5.4.1 План эвакуации.....	91
5.4.2 Пожарные краны.....	92
5.4.3 Огнетушители.....	92
5.5 Обеспечение санитарно-бытовых условий.....	99
5.6 Освещение.....	100
5.6.1 Расчёт.....	101
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	104
ПРИЛОЖЕНИЕ А	108
ПРИЛОЖЕНИЕ В	Error! Bookmark not defined.
ПРИЛОЖЕНИЕ С	Error! Bookmark not defined.
ПРИЛОЖЕНИЕ D	Error! Bookmark not defined.
ПРИЛОЖЕНИЕ E.....	Error! Bookmark not defined.
ПРИЛОЖЕНИЕ F.....	Error! Bookmark not defined.

ВВЕДЕНИЕ

Потоковое вещание - это мультимедийный контент, который пользователи непрерывно получают от провайдера услуг.

В настоящий момент существует множество интернет площадок Youtube, Facebook, с мощной поддержкой систем хранения информации, гибким API для разработки приложений и современным интерфейсом для хранения, обработки и передачи видеoinформации.

Основным недостатком вышеперечисленных решений является то, что все они узко специализированы и до конца не могут обеспечить и гарантировать гибкости в обработке информации.

Эта причина вынуждает корпоративных потребителей выбирать комплексные решения.

В частности одним из таких решений является использование связки облачное решение Amazon, собственный стриминговый сервер и мобильное клиентское приложение.

В данном отчете представлена реализация системы потоковой передачи видео информации корпоративным сотрудникам.

Необходимость данного решения вызвана тем, что порой требуется информирование сотрудников крупных компаний в виде видео инструкции. Для чего основные каналы передачи корпоративной информации e-mail, SMS рассылка, Slack или чат-боты не подходят.

Этот прибор значительно упростит задачу сотрудникам ДПС и поможет бороться с нарушителями.

					ПГУ 1.090501.17.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		7

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

В первой главе будет проведен анализ предметной области и инструментов реализации данного проекта.

1.1. Существующие решения

Те, кому необходимо универсальное решение, способное работать с разными объектами и охватывать как одно технологическое подразделение, так и весь производственно-технологический комплекс, приобретают корпоративные системы управления медиаресурсами A-MAM (Automated Media Asset Management), разработанные компанией BRAM Technologies.

A-MAM обеспечивает обмен файлами не только в сетях с высокоскоростными хранилищами, но и с ленточными LTO-библиотеками. Система способна самостоятельно управлять роботом библиотеки. Полуавтоматическая система обработки электронных заявок обеспечивает учет внешних носителей и перемещения данных.

Шведская компания Cantemo выпустила систему управления медиаактивами, которая состоит из ядра Cantemo Portal и инфраструктуры Cantemo Portal Apps, которая содержит соответствующие API и открытые интерфейсы для интеграции с решениями сторонних производителей.

Cantemo Portal это своего рода склад, постоянно наполняемый видео- и аудиоклипами, статичными изображениями и другими цифровыми медиаданными. Он обеспечивает поиск и воспроизведение материала вне зависимости от того, где именно в многоуровневой среде хранения этот материал находится. Кроме того, Cantemo Portal легко интегрируется с распространенными инструментами монтажа, транскодирования, распространения и архивирования контента.

					ПГУ 1.090501.17.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

Платформа Dalet Galaxy объединяет технологическую цепочку работы с контентом и делает ее более эффективной благодаря управлению активами, метаданными, рабочими процессами и процедурами.

Dalet Galaxy обладает важными функциональными особенностями, которые обогащают работу с медиаданными и открывают возможность формирования бизнес-моделей.

Первое из таких улучшений использование искусственного интеллекта во всем рабочем процессе.

Второе обеспечивает гибридное применение, когда в одном комплексе присутствуют локальная, частная и облачная составляющие.

А третье новшество заключается в помещении социальных сетей в центр всей работы.

Все вышеперечисленные решения являются решениями класса Enterprise, а значит стоят больших денег, далее, их универсальность является как преимуществом, так и недостатком, который лишает данные решения гибкости. В частности если заказчику потребуется реализовать свой проект на базе бессерверной архитектуры, большинство вышеописанных решений не сможет это сделать быстро.

Исходя из выше изложенного делаем вывод, что решения управления медиаконтентом являются актуальными. Существующие решения слишком громоздкие и для решения задач субподрядных организаций и/или бригад корпоративного сегмента требуются более легкие и гибкие решения.

1.2 Сравнение площадок для размещения видео-контента

Вторая задача данного проекта – выбрать публичную площадку для реализации основных функций проекта: загрузка, хранение, трансляция видео контента. Для решения задачи сравним существующие площадки по

критериям, необходимым для выполнения задачи (стоимость размещения информации, наличие API для реализации бизнес логики, степень доступности)

Таблица 1.1. – Характеристики площадок для размещения видео-контента

№	Название	Характеристики	Недостатки
1	Youtube	стоимость размещения: бесплатное; коммерческий контент: допускается; редактирование видео онлайн: присутствует; система авторизации Google OAuth API; API YouTube использует протокол на базе архитектуры REST, поддерживающий; представления данных Atom, JSON и JSON-C.	отображение рекламы поверх и вокруг ваших роликов; доступ в хранилище только через профиль пользователя; ограничения по содержанию контента.

Продолжение таблицы 1.1

№	Название	Характеристики	Недостатки
2	Facebook	стоимость размещения: бесплатное; нет ограничений на размер видео файла; Система авторизации OAuth API; Поддерживает Graph API Rest API	сбои в работе проигрывателя ограничения по содержанию контента; доступ в хранилище только через профиль пользователя; редактирование видео онлайн: нет;
3	Amazon	стоимость размещения: бесплатное; нет ограничений на размер видео файла; Система авторизации OAuth API; API состоит из мультязычной платформы (поддерживает Node.js, Ruby, Python, PHP, Go, C++, AWS Mobile); поддерживает JSON, Soap форматы.	коммерческий контент в целом, не допускается. редактирование видео онлайн: присутствует; доступ в хранилище через профиль пользователя и непосредственно через файловую систему (DinamoDB); коммерческая платформа со строгими правилами размещения и пользования контентом.

В результате анализа данных из таблицы 1.1. для реализации проекта выбрана платформа Amazon, т.к. платформа бесплатно предоставляет

возможность использовать полноценное API, систему управления базой данных – DinamoDB и систему авторизации Auth0 (инструмент реализует функции, связанные с проверкой подлинности для вашего приложения или сайта).

1.3 Сравнение архитектурных решений реализации приложения

Основой системы на базе Amazon является бессерверный подход. Суть которого в том, что приложение находится на облачном сервисе и выполняется только по мере необходимости. В нашем случае облачный сервис будет выполнять роли: хранилища медиа информации, бизнес логика второго эшелона (первый эшелон бизнес логики будет расположен на сервере разработчиков). На рисунке 1.1 представлены инструменты, которые станут частью реализуемого приложения.

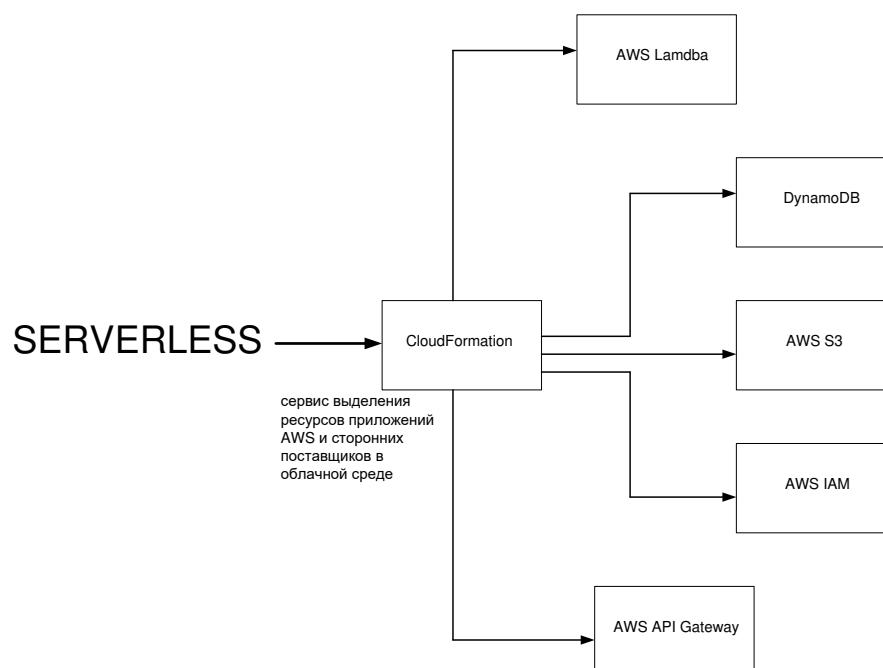


Рисунок 1.1. Инструменты платформы Amazon

ClaudFormation - язык для моделирования и выделения ресурсов приложений AWS и сторонних поставщиков в облачной среде.

AWS Lambda позволяет запускать программный код без выделения серверов и управления ими. Оплате подлежит только фактическое время выполнения вычислений.

Amazon S3 - онлайн-веб-служба, предоставляющая возможность для хранения и получения любого объёма данных, в любое время из любой точки сети, так называемый файловый хостинг.

AWS IAM служба, позволяющая создавать пользователей AWS, группы, управлять ими, а также использовать разрешения, чтобы предоставлять или запрещать доступ к ресурсам AWS.

Amazon API Gateway – это управляемый сервис для разработчиков, предназначенный для создания, публикации, обслуживания, мониторинга и обеспечения безопасности API. API приложения получают доступ к данным, бизнес-логике сервисов. Позволяет создавать API RESTful и WebSocket, которые являются главным компонентом приложений для двусторонней связи в режиме реального времени.

В отличие от классических фреймворков на базе модели Модель-Вид-Контроллер (рисунок 1.2), приложение будет реализовано как бессерверная распределенная модель (рисунок 1.3). Данный выбор связан с тем, что AWS имеет в своем составе инструменты работы с медиа данными. В частности сервис для работы с мультимедиа в облаке - Amazon Elastic Transcoder. Сервис Elastic масштабируемый, простой в использовании и экономичный способ обработки медиафайлов в формат пригодный для воспроизведения на различных устройствах: смартфоны, планшеты и т.д.

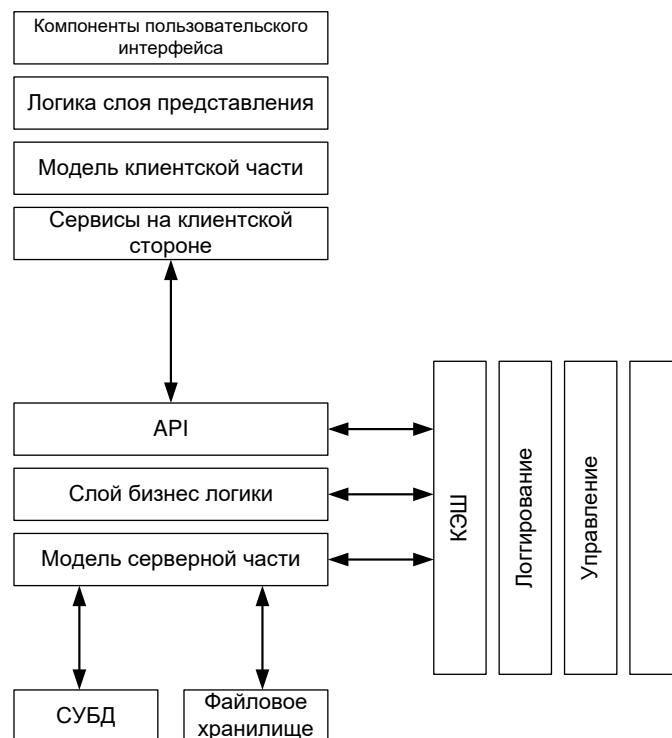


Рисунок 1.2. Фреймворк на базе архитектуры модель вид контроллер.

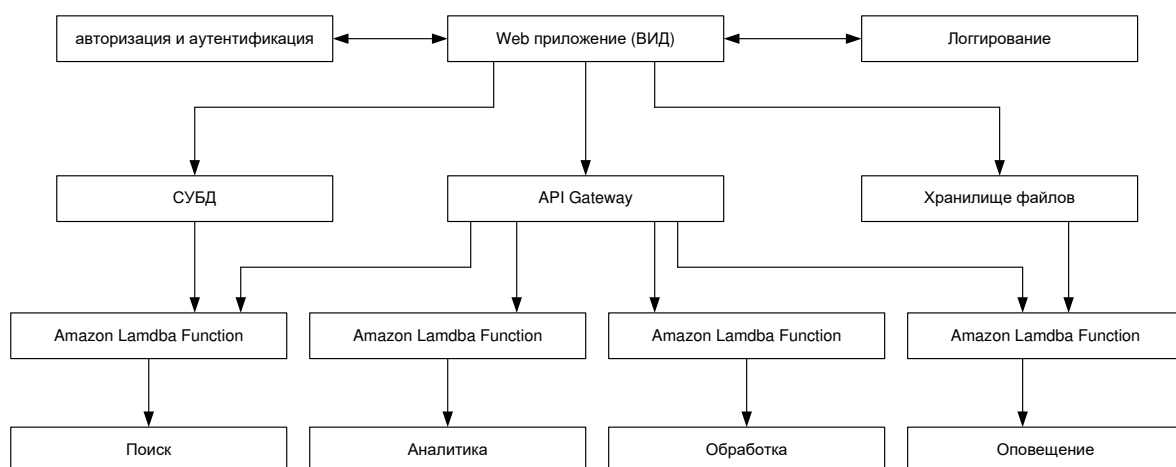


Рисунок 1.3. Бессерверного распределенного приложения

Основными функциями реализуемого приложения являются: загрузка видео в хранилище, кодирование и сжатие видео контента, выдача пользователю списка файлов, напоминание пользователю о новых файлах, авторизация пользователей в приложении. Так как приложение реализуется в

формате бессерверного приложения, то обобщенная архитектура приложения выглядит следующим образом (рисунок 1.4).

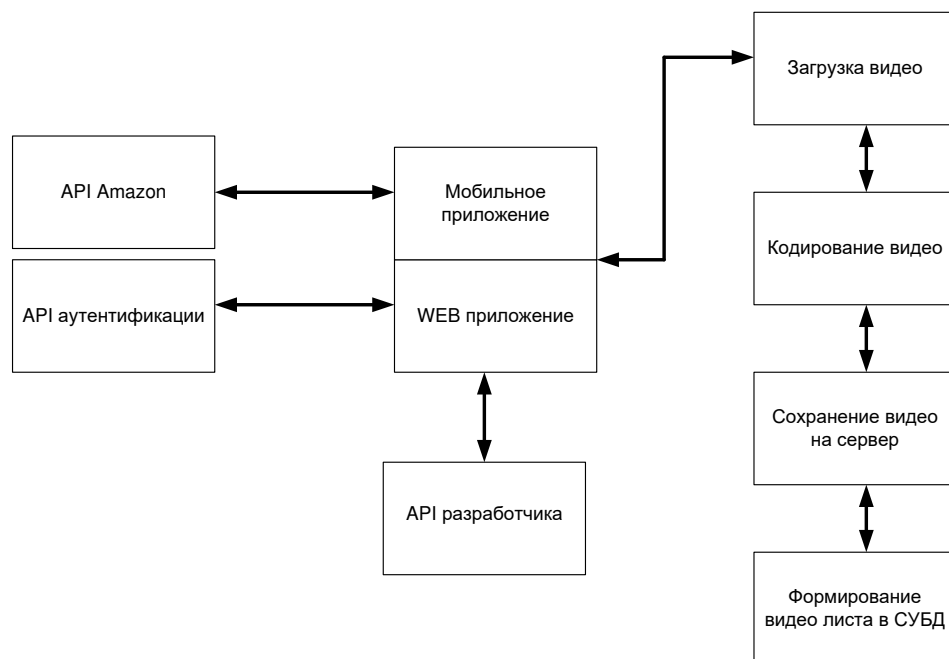


Рисунок 1.4 Классическая реализация приложения на базе AWS

Как видно из описания выше, бессерверная архитектура имеет инструментарий позволяющий реализовать функционал бизнес логики приложения и функционал хранения данных.

Далее следует определиться с инструментами реализации интерфейсной части приложения.

1.4 Сравнение программных решений реализации приложения

Первым делом проведем анализ и выбор инструментария работы с медиаконтентом на уровне фронтенда (html и js библиотеки и/или плагины).

Для чего сделаем статистическую выборку с сайта github.com, который предоставляет объективную статистику использования различных библиотек и плагинов (рисунок 1.6).

Downloads in past 6 Months ▾

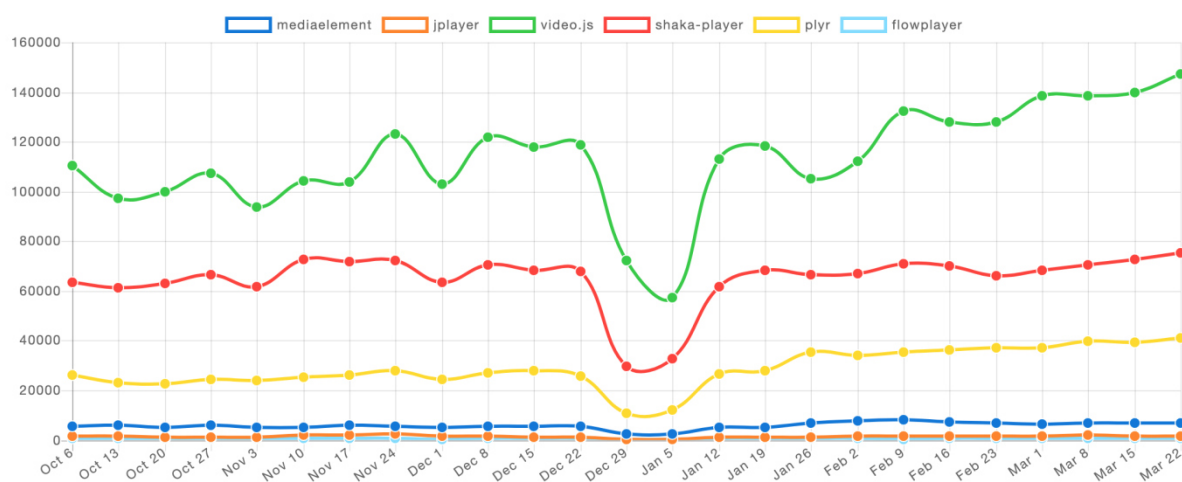


Рисунок 1.5 Количество скачиваний библиотек для работы с видео контентом

На основании данной статистики были выбраны самые популярные видео web плееры с открытым исходным кодом, которые могут быть полезны в реализации данного проекта.

Библиотека Video.js - библиотека JavaScript и CSS, которая облегчает работу по созданию HTML5-видео. Библиотеку можно также назвать HTML5-видеоплеером. Video.js обеспечивает управление и создание медиа контейнеров в HTML/CSS, а также исправляет несоответствия в разных браузерах и добавляет дополнительные функции, такие как управление видео. Video.js позволяет переключаться на Flash тогда, когда недоступен HTML5, а также обеспечивает JavaScript API для взаимодействия с видео.

Популярность данного проета подтверждает статистика на сайте github.com

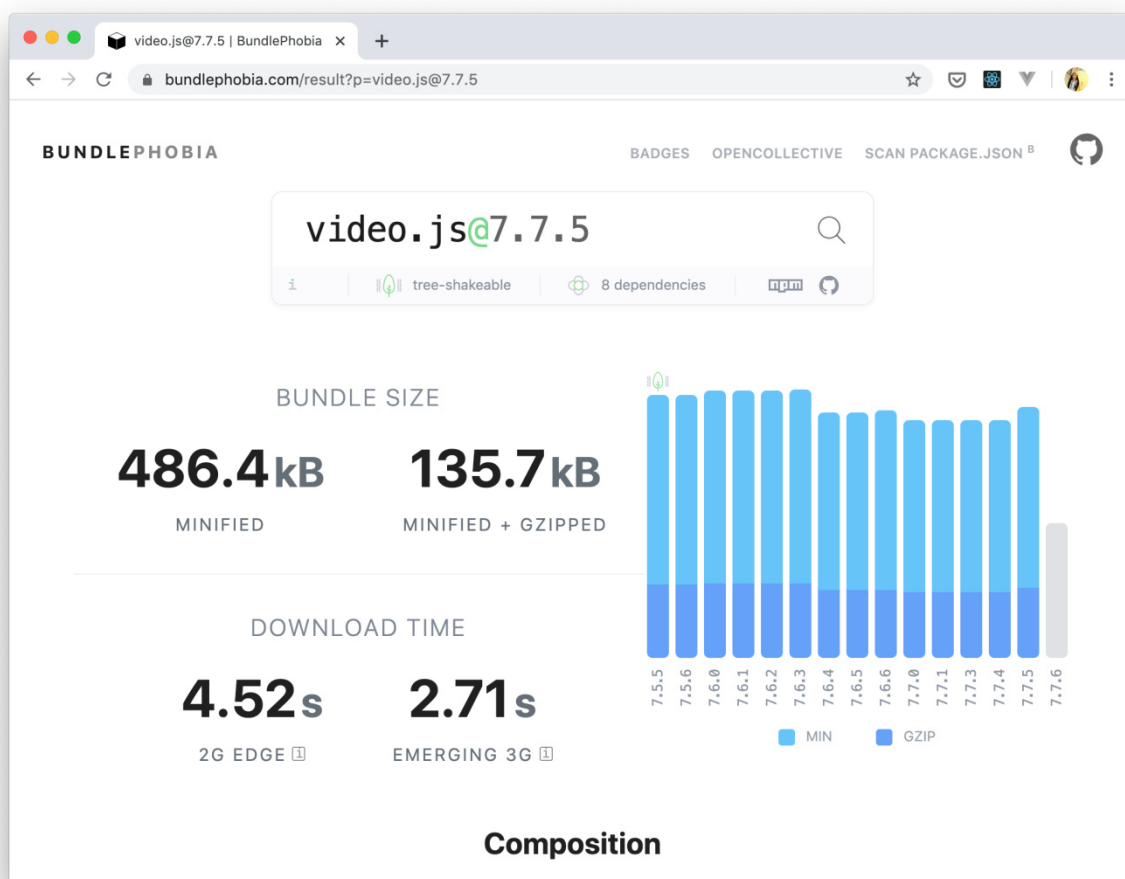


Рисунок 1.6 Статистика библиотеки Video.js

Shaka Player - библиотека JavaScript с открытым исходным кодом для адаптивных медиа. Он воспроизводит адаптивные медиа-форматы (такие как DASH и HLS) в браузере, без использования плагинов или Flash. Вместо этого Shaka Player использует открытые веб-стандарты MediaSource Extensions и Encrypted Media Extensions. Shaka Player также поддерживает автономное хранение и воспроизведение мультимедиа с использованием IndexedDB.

Статистика использования данной библиотеки приведена ниже на рисунке 1.7

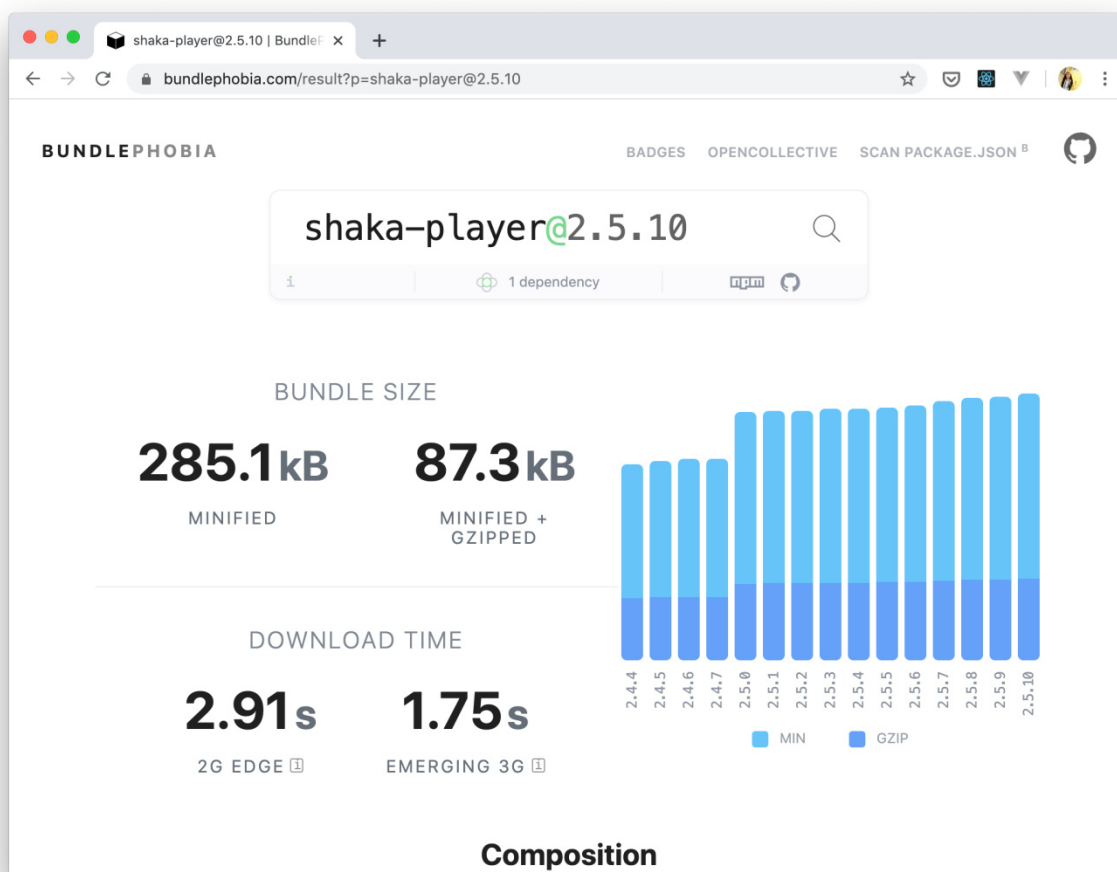


Рисунок 1.7 Статистика библиотеки Shaka Player

Библиотека JavaScript Plyr имеет большое количество настроек, а также поддерживает субтитры в формате WebVTT (Web Video Text Tracks). Инструмент имеет небольшой размер (в сжатом виде всего 6.4KB), настраивается и поддерживается всеми современными браузерами. У него много конкурентов в HTML5. Плеер Plyr сложнее в обращении, чем приведенные выше конкуренты, к тому же использует в своей работе только семантические HTML-элементы.

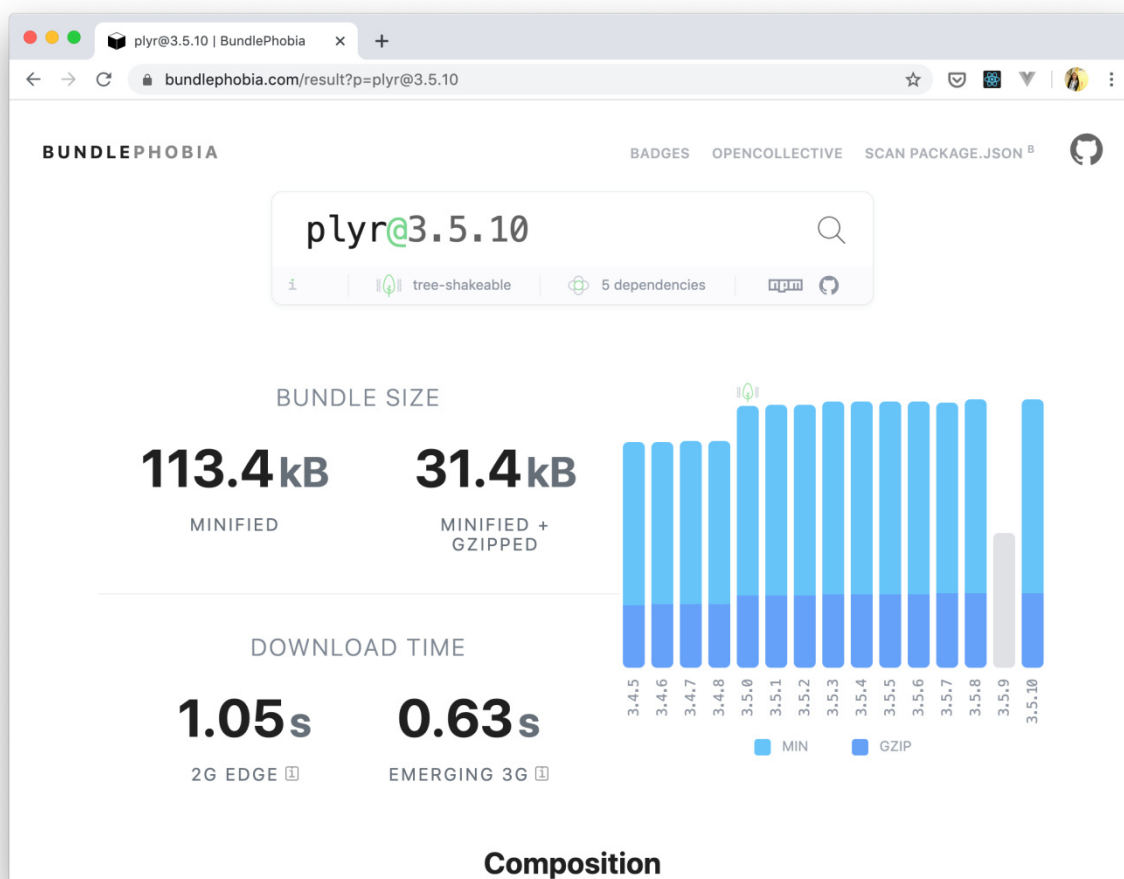


Рисунок 1.8 Статистика библиотеки Plyr

MediaElement.js это HTML5 аудио и видео плеер, который работает в старых браузерах имитируя MediaElement HTML5 API с помощью Flash и Silverlight. От выше перечисленных библиотека отличается отсутствием поддержки, что отражено на графиках использования данной библиотеки.

Далее в таблице 1.2 приведем сравнительный анализ библиотек работы с медиа данными. Отметим, что все выбранные инструменты являются свободно распространяемыми, имеют как стандартный, так и дополнительный функционал. Как показало сравнение Video.js подходит для реализации данного проекта так как содержит функционал, требуемый для

реализации проекта и зарекомендовал себя как инструмент адекватно реализующий инструментарий в работе с инструментами AWS.

Jul 25, 2010 – Mar 26, 2020

Contributions: Commits ▾

Contributions to master, excluding merge commits

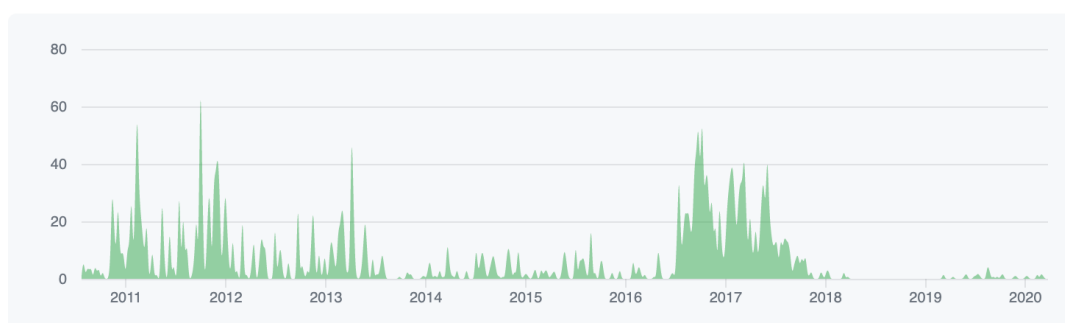


Рисунок 1.9 Статистика скачиваний библиотеки MediaElement.js

Таблица 1.2 Сравнительный анализ библиотек работы с медиа данными

Имя	API	встраиваемость	аудио	платформа	JS	лицензия
Video.js	есть	да	да	YouTube, Vimeo, AWS	да	MIT
Shaka Player	-	да	да	YouTube, Vimeo	да	MIT
Plyr	есть	да	да	YouTube, Vimeo	да	MIT
MediaElement.js	-		да	YouTube, Vimeo	да	GNU GPL

1.5 Сравнение языков программирования

Node.js программная платформа, основанная на ядре транслирующем JavaScript в машинный код, превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через собственный API, подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода.

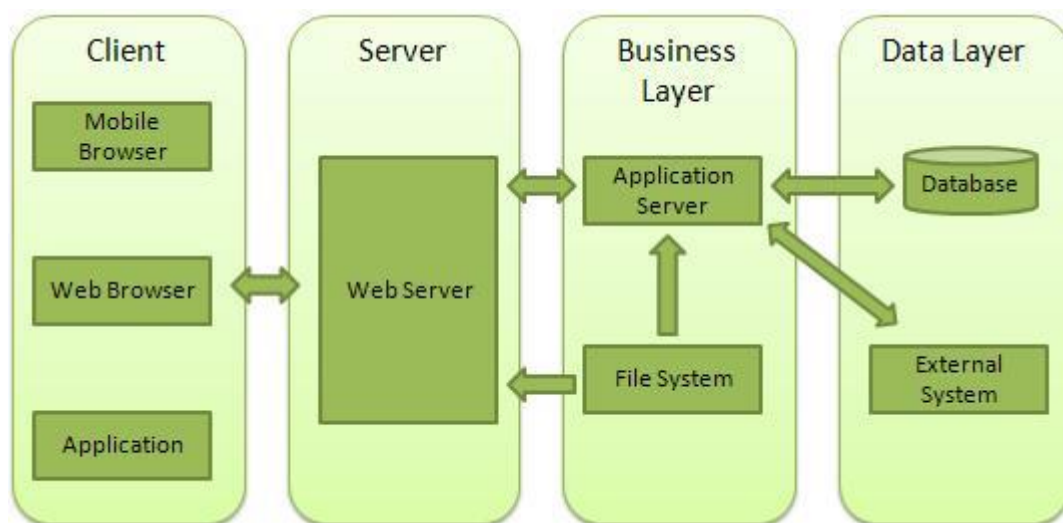


Рисунок 1.10 Архитектура Node.js приложения.

Плюсы Node.js:

1) Быстрое серверное решение

Node.js позволяет используя очередь событий JavaScript создавать приложения с неблокирующим вводом/выводом которые способны обрабатывать несколько запросов одновременно. Используя встроенную в JavaScript асинхронность, можно создавать высокомасштабируемые серверные приложения, которые максимизируют использование одного CPU и памяти компьютера при одновременной обработке большего количества запросов, чем обычные многопоточные серверы. Такая функциональность делает Node.js прекрасным выбором для приложений реального времени.

2) Один язык на фронт-енде и бекенде

Много популярных JS фреймворков таких как React или Vue написаны на JavaScript, который является основным языком всех современных браузеров. Используя Node.js на сервере разработчик получает все преимущества скриптового языка на обеих платформах. Иметь один язык на фронт-енде и бекенде очень хорошо для обеспечения поддержки приложения и координации между членами команды.

3) Гибкость

Node.js не имеет строгих правил или жестких зависимостей, что оставляет простор для созидания и креативности при разработке приложений. Разработчики сами выбирают архитектуру, зависимости.

Недостатки Node.js

1) Малая эффективность в операциях интенсивно использующих CPU. Событийно-ориентированная архитектура Node.js имеет некоторые ограничения а именно низкую эффективность при большой нагрузки на CPU.

2) Незрелость платформы

Вместе со стабильными стандартными библиотеками такие как HTTP or Crypto, доступно множество библиотек от сторонних разработчиков. Эта система еще не достаточно сформировалась, и приходится зависеть от не протестированных модулей.

PHP скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов.

Плюсы PHP

1) Большая кодовая база

PHP имеет базу для всевозможных решений от систем управления контентом до мощных фреймворков таких как Laravel, Symfony.

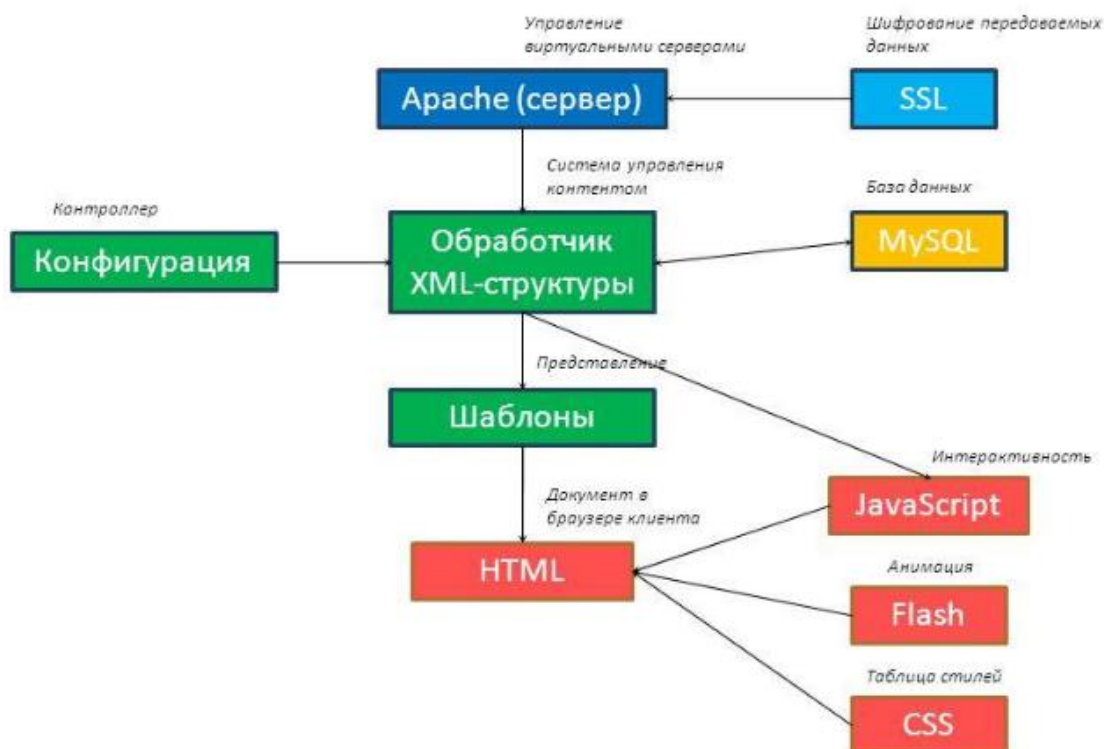


Рисунок 1.11 Архитектура PHP приложения.

2) Переносимое решения. PHP достаточно независим от платформы, он может быть запущен почти на любом сервере и на любой платформе.

В тоже время, имеется широкий выбор хостингом которые с поддержкой PHP, и вам не нужно арендовать полноценный сервер с SSH доступом чтобы запустить проект. Из этого следует, что интеграция PHP проектов и их развертывание несколько проще для небольших компаний или для отдельных лиц который могут запускать и управлять приложениями без знаний консольных команд, системы Linux

3) Спроектирована для WEB. В отличии от Java или Python и других языков программирования общего назначения, PHP был разработан специально для Web. Именно поэтому он содержит всю необходимую функциональность для работы с HTML, серверами, базами данных.

Недостатки PHP

1) Плохое разделение ответственности (SoC). В PHP плохо реализован паттерн MVC, который является рекомендованным в веб-разработке. Очень часто можно встретить объединение HTML и PHP, что выливается в сложно поддерживаемый код, где бизнес логик смешана с представлением.

2) Устаревшая клиент-серверная модель

PHP следует классической клиент-серверной модели где запрос страницы инициирует приложение, соединение с базой данных, их обработкой и рендерингом HTML. Это делает PHP несколько медленнее в сравнении с Node.js приложениями, которые инициируются при запуске, и поэтому Node больше подходит для написания приложений реального времени.

В данном проекте выбор обозначен в сторону Nodejs, как инструмента поддерживающего библиотеки работы с медиаконтентом и платформу AWS.

1.6 Выводы по главе

В результате исследования проведенного в первой главе была определена актуальность предметной области.

Выбран инструментарий реализации проекта.

В разделе разработка планируется создать следующие модули системы: клиентское приложение Web приложение или мобильное приложение, серверное приложение (бизнес логика), AWS Lambda и AWS S3 функции и триггеры для обработки и хранения видео информации.

2 РАЗРАБОТКА СИСТЕМЫ

Разрабатываемая система состоит из двух частей фронтенд (Nodejs) фактически являющимся витриной и основой интерфейса приложения и бекэнд (службы AWS).

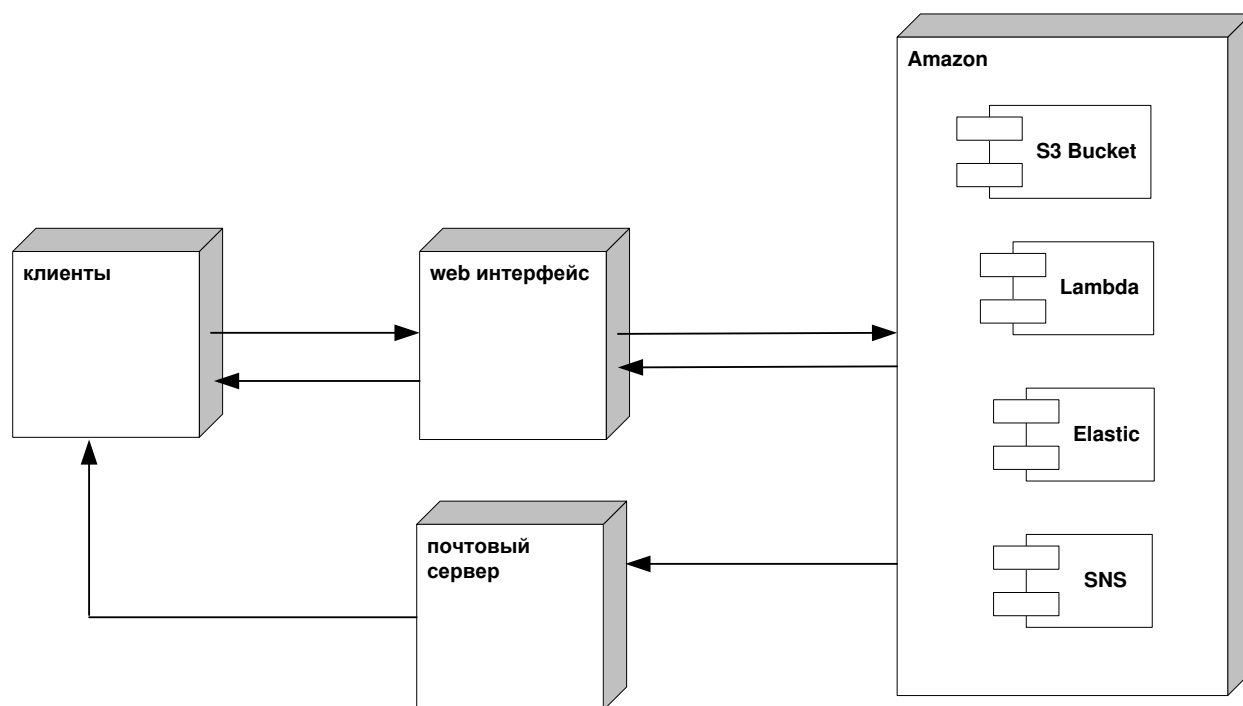


Рисунок 2.1 Архитектура реализуемой системы

Потоки информации в данном приложении проходят стадии анализа, обработки и визуализации.

Представим эти стадии в виде данных и функций, как показано на рисунке 1.5.

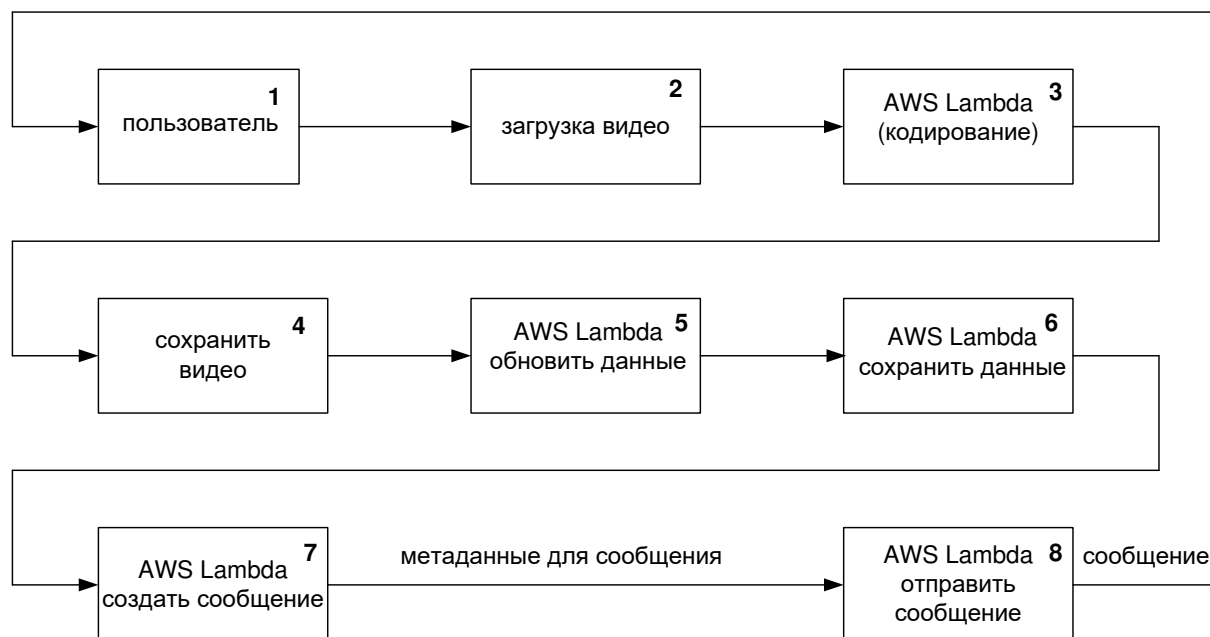


Рисунок 2.2 Поток данных в системе

Система должна реализовывать следующие функции: загрузка и кодирование видео файлов, посылка сообщения на e-mail всем заинтересованным лицам, что в систему добавлено новое видео, просмотр добавленного видео зарегистрированными пользователями, регистрация и вход по логину и паролю пользователям системы.

2.1 Реализация интерфейса регистрации и входа в систему

Сервис Amazon Cognito обеспечивает аутентификацию, авторизацию и управление пользователями для веб-приложений средствами Amazon. Он выполняет те же функции, что и многие другие популярные платформы аутентификации, такие как Auth0, Identity Server и веб-токены JWT, но Cognito сохраняет и синхронизирует данные конечного пользователя в облаке, а не в серверной части приложения.

Amazon Cognito управляет следующими инструментами регистрации и авторизации: пулы пользователей и пулы удостоверений.

Пул пользователей позволяет разработчикам добавлять функции регистрации и входа в веб-приложения. Он поддерживает регистрацию пользователей и вход в систему, а также предоставление идентификационных токенов для зарегистрированных пользователей.

Пул удостоверений позволяет разработчикам создавать уникальные удостоверения для пользователей и проверять их подлинность с помощью поставщиков федеративных удостоверений. Фактически инструмент позволяет создавать временные учетные данные AWS с ограниченными правами для безопасного доступа к другим сервисам AWS, таким как Amazon DynamoDB, Amazon S3 и Amazon API Gateway.

В разрабатываемом приложении регистрация пользователей, аутентификация и восстановление учетных записей будет выполняться пулами пользователей, тогда как пул удостоверений позволяет пользователям использовать различные сервисы AWS. Это позволит пользователю получить доступ к облачным S3-контейнерам, чтобы загрузить файлы.

Вся процедура реализации системы авторизации состоит из двух частей: создавать наш пул пользователей в Cognito и реализовать программный код на стороне клиента для использования сервисов Cognito.

При переходе на сервисы AWS и Cognito появляется страница (рисунок 2.3), которая позволяет выбрать: управлять пользовательскими пулами, либо управлять личными пулами.

Поскольку для реализации проекта собираемся создать пул пользователей, нажмите «Управление пользовательскими пулами» и «Создать новый пул пользователей». Ниже показана страница, на которую вы направляетесь после нажатия на создание пула пользователей (рисунок 2.4).



Рисунок 2.3 Начальная страница Cognito

Здесь можно выбрать один из двух вариантов: использовать настройки по умолчанию для пула пользователей или настроить параметры. Большинство из этих настроек по умолчанию требуется проверить для данного сценария разработки, поэтому выберем пошаговые настройки.

Рисунок 2.4 Ввод уникального имени пула.

					ПГУ 1.090501.17.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		28

Далее можно выбрать то, что нужно от пользователя при входе в систему при регистрации. По умолчанию он выбирает имя пользователя, но также можем указать либо адрес электронной почты, либо номер телефона. Во втором разделе отображаются атрибуты, которые будут необходимы в процессе регистрации.

Car

How do you want your end users to sign in?

You can choose to have users sign in with an email address, phone number, username or preferred username plus their password. [Learn more.](#)

☒ Username - Users can use a username and optionally multiple alternatives to sign up and sign in.

☐ Also allow sign in with verified email address

☐ Also allow sign in with verified phone number

☐ Also allow sign in with preferred username (a username that your users can change)

☐ Email address or phone number - Users can use an email address or phone number as their "username" to sign up and sign in.

☐ Allow email addresses

☐ Allow phone numbers

☐ Allow both email addresses and phone numbers (users can choose one)

Which standard attributes do you want to require?

All of the standard attributes can be used for user profiles, but the attributes you select will be required for sign up. You will not be able to change these requirements after the pool is created. If you select an attribute to be an alias, users will be able to sign-in using that value or their username. [Learn more about attributes.](#)

Required

☐

Attribute

address

☐

birthdate

☒

email

☐

family name

☐

gender

☐

given name

☐

locale

☐

middle name

☐

nickname

☐

phone number

☐

picture

☐

preferred username

☐

profile

☐

zoneinfo

☐

updated at

☐

website

Рисунок 2.5 Выбор параметров авторизации.

Далее мы направляемся на страницу настройки пароля. Во втором разделе данной страницы можно дать пользователям возможность регистрироваться самостоятельно, а не только администраторам.

					ПГУ 1.090501.17.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		29

Cancel

What password strength do you want to require?

Minimum length

8

- ☒ Require numbers
- ☒ Require special character
- ☒ Require uppercase letters
- ☒ Require lowercase letters

Do you want to allow users to sign themselves up?

You can choose to only allow administrators to create users or allow users to sign themselves up. [Learn more.](#)

☐ Only allow administrators to create users
☒ Allow users to sign themselves up

How quickly should temporary passwords set by administrators expire if not used?

You can choose for how long until a temporary password set by an administrator expires if the password is not used. This includes accounts created by administrators.

Days to expire

7

Back

Next step

Рисунок 2.6 Выбор параметров паролей и авторизации.

На следующей странице можно настроить, нужно ли пользователям использовать многофакторную аутентификацию (MFA).

MFA добавляет дополнительный уровень безопасности, где пользователи будут проверяться с использованием нескольких учетных данных, а не только с использованием имени пользователя и пароля. MFA требует других дополнительных учетных данных, таких как ответ на секретный вопрос, отпечаток пальца или распознавание лица.

[Cancel](#)

Do you want to enable Multi-Factor Authentication (MFA)?

Multi-Factor Authentication (MFA) increases security for your end users. If you choose 'optional', individual users can have MFA enabled. You can only choose 'required' when initially creating a user pool, and if you do, all users must use MFA. Phone numbers must be verified if MFA is enabled. You can configure adaptive authentication on the Advanced security tab to require MFA based on risk scoring of user sign in attempts. [Learn more about multi-factor authentication.](#)

Note: separate charges apply for sending text messages.

☒ Off
 ☐ Optional
 ☐ Required

Which attributes do you want to verify?

Verification requires users to retrieve a code from their email or phone to confirm ownership. Verification of a phone or email is necessary to automatically confirm users and enable recovery from forgotten passwords. [Learn more about email and phone verification.](#)

☒ Email
 ☐ Phone number
 ☐ Email or phone number
 ☐ No verification

You must provide a role to allow Amazon Cognito to send SMS messages

Amazon Cognito needs your permission to send SMS messages to your users on your behalf. [Learn more about IAM roles.](#)

New role name

NodeJstutorialUsers-SMS-Role

Create role

Рисунок 2.7 Дополнительные уровни безопасности авторизации.

Далее требуется настроить способ отправки приглашения авторизации по электронной почте пользователям.

[Cancel](#)

Do you want to customize your email address?

You can send emails from an SES verified identity. [Learn more about SES verified identities and domains.](#)

SES Region

US East (Virginia) ▼

FROM email address

Default ▼

You must verify your email address with Amazon SES before you can select it. [Verify an SES identity.](#)

REPLY-TO email address

Do you want to send emails through your Amazon SES Configuration?

Select Yes if you require higher daily email limits otherwise select No. [Learn more about Cognito daily email limits.](#) If you choose Yes, Cognito will send emails through your Amazon SES configuration. [Refer to this documentation for additional steps.](#)

☐ Yes - Use Amazon SES
 *Requires FROM email address
 ☒ No - Use Cognito (Default)

Рисунок 2.8 Настройка отправки приглашения авторизации на e-mail.

Далее настраиваем систему на режим работы, при котором пользователю предоставляется возможность сохранить пользовательские устройства, чтобы пользователь всегда входил в систему или нет.

Do you want to remember your user's devices?

Always User Opt In No

Back Next step

Рисунок 2.9 Настройка опций собрания в системе устройств пользователя.

Далее нам нужно создать программные функции на стороне Amazon, с которыми будет взаимодействовать web приложение Nodejs.

Для этого установите флажок «Включить API входа для аутентификации», чтобы приложение Nodejs обращалось к пулу пользователей Cognito для аутентификации. Клиенты приложений могут быть созданы и после создания пользовательских пулов.

Do you want to customize workflows with triggers?

You can make advanced customizations with AWS Lambda functions. Pick AWS Lambda functions to trigger with different events if you want to customize workflows and the user experience. Visit the [AWS Lambda console](#) to create your functions before selecting them below. [Learn more about triggers.](#)

Pre sign-up

This trigger is invoked when a user submits their information to sign up, allowing you to perform custom validation to accept or deny the sign up request.

Lambda function

none ▾

Pre authentication

This trigger is invoked when a user submits their information to be authenticated, allowing you to perform custom validations to accept or deny the sign in request.

Lambda function

none ▾

Custom message

This trigger is invoked before a verification or MFA message is sent, allowing you to customize the message dynamically. Note that static custom messages can be edited on the Verifications panel.

Lambda function

none ▾

Post authentication

This trigger is invoked after a user is authenticated, allowing you to add custom logic, for example for analytics.

Lambda function

none ▾

Post confirmation

Define Auth Challenge

This trigger is invoked to initiate the custom authentication flow.

Рисунок 2.10 Создание Amazon Lambda функций авторизации.

На этом этапе требуется назначить лямбда-функции событиям, с которыми будет работать Cognito. Например, в событии «Предварительная регистрация» требуется написать лямбда-функцию для форматирования ввода перед отправкой его в Cognito API (исходный код представлен в приложении).

При создании пула пользователей приложения, в общих настройках можно видеть, что для пула пользователей создан идентификатор пула и идентификатор клиента приложения. Оба этих идентификатора будут необходимы при вызове функций в приложении Nodejs.

Pool Id us-east-1_gf5MTchOu

Pool ARN arn:aws:cognito-idp:us-east-1:914556018196:userpool/us-east-1_gf5MTchOu

Рисунок 2.11 Созданные пулы

Рисунок 2.12 ID приложения для программного доступа к AWS

Далее требуется реализовать программный код, позволяющий клиентскому приложению авторизовать пользователя средствами AWS, для этого приложению требуется реализовывать три функции: регистрация, вход в систему, проверка токенов.

Для каждой функции понадобятся отдельные файлы API в приложении и на сервере AWS.

Для этого в каталог Controllers добавим файл с именем AuthController.js. Также в папку с именем Services добавим файл с именем AuthService.js. Установим пакеты Nodejs, в которых body-parser требуется для анализа тела запроса, отправленного на сервер, второй пакет - это Cognito SDK, который будет использоваться для запросов к Cognito API.

```
npm установить body-parser - сохранить
npm установить amazon-cognito-identity-js -save
npm isntall node-fetch --save
```

В файл AuthService.js добавим следующий код:

```
global.fetch = require ('node-fetch'); global.navigator = () =>
null;
const AmazonCognitoIdentity = require ('amazon-cognito-identity-
js');
const poolData = {
  UserPoolId: "{pool_id}",
  ClientId: "{app_client_id}"
};
```

```

const pool_region = "{регион}"; const userPool = new
AmazonCognitoIdentity.CognitoUserPool (poolData); exports.Register
= function (body, callback) {
    var name = body.name;
    var email = body.email;
    var password = body.password;
    var attributeList = [];
    attributeList.push (new
AmazonCognitoIdentity.CognitoUserAttribute ({Name: "email", Value:
email}));
    userPool.signUp (name, password, attributeList,
        if (err)
            callback(err);
            var cognitoUser = result.user;
            callback(null, cognitoUser);
        })
    }

```

В приведенном выше коде первым делом определяется CognitoUserPool, который дает pool ID и app client id приложения. После этого анализируется тело пользовательского запроса на username и password. Так как в коде был запрос e-mail для регистрации, то программно собираемся прикрепить его как CognitoUserAttribute. Затем вызываем метод signUp, который создаст нашего пользователя.

Добавим функциональность входа в файл AuthService.js и логику контроллера в AuthController.js.

```

exports.Login = function (body, callback) {
    var userName = body.name;
    var password = body.password;
    var authenticationDetails = new
AmazonCognitoIdentity.AuthenticationDetails({
        Username: userName,

```

```

        Password: password
    });    var userData = {
        Username: userName,
        Pool: userPool
    }    var cognitoUser = new
AmazonCognitoIdentity.CognitoUser(userData);
cognitoUser.authenticateUser(authenticationDetails, {
    onSuccess: function (result) {
        var accesstoken = result.getAccessToken().getJwtToken();
        callback(null, accesstoken);
    },
    onFailure: (function (err) {
        callback(err);    })
})
};

```

Для проверки токена понадобится несколько пакетов для web приложения на базе Nodejs. Установите следующие пакеты и импортируйте их в AuthService.js:

```

npm установить запрос - сохранить
npm установить jwk-to-pem - сохранить
npm установить jsonwebtoken - сохранить

```

```

exports.Validate = function(token, callback){
    request({
        url : `https://cognito
idp.${pool_region}.amazonaws.com/${poolData.UserPoolId}/.well-
known/jwks.json`,
        json : true
    }, function(error, response, body){
        if (!error && response.statusCode === 200) {
            pems = {};
            var keys = body['keys'];
            for(var i = 0; i < keys.length; i++) {
                var key_id = keys[i].kid;
                var modulus = keys[i].n;
                var exponent = keys[i].e;
                var key_type = keys[i].kty;
            }
        }
    });
}

```

```

        var jwk = { kty: key_type, n: modulus, e:
exponent};
        var pem = jwkToPem(jwk);
        pems[key_id] = pem;
    }
    var decodedJwt = jwt.decode(token, {complete:
true});
    if (!decodedJwt) {
        console.log("Not a valid JWT token");
        callback(new Error('Not a valid JWT token'));
    }
    var kid = decodedJwt.header.kid;
    var pem = pems[kid];
    if (!pem) {
        console.log('Invalid token');
        callback(new Error('Invalid token'));
    }
    jwt.verify(token, pem,
function(err, payload) {
    if(err) {
        console.log("Invalid Token.");
        callback(new Error('Invalid token'));
    } else {
        console.log("Valid Token.");
        callback(null, "Valid token");
    }
    });
} else {
    console.log("Error! Unable to download JWKs");
    callback(error);
}
});
}

```

В конечном итоге добавим маршруты контроллера в файл с именем route.js и добавим маршруты в точку входа пользовательского приложения файл app.js.

И протестируем результат, запустив в браузере ресурс по адресу localhost: 3000/ (рисунок 2.13)

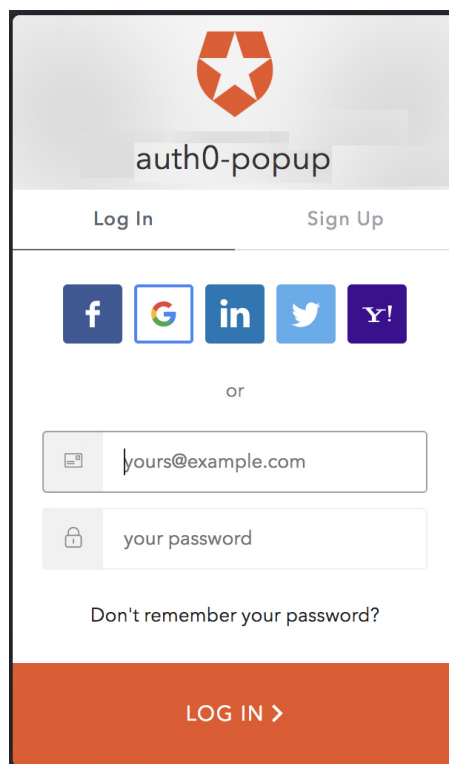


Рисунок 2.13. Страница авторизации приложения

2.2 Реализация функций видеоплеера

Функции видео плеера включают в себя сохранение видео в облаке, получения списка видео файлов на страницу ресурса, воспроизведение видео и отправка сообщения на e-mail заинтересованных лиц сообщения, если пользователь загрузил видео.

Начнем реализацию данных функций с функции кодирования видео файлов попадающих в систему под общий стандарт mp4, т.к. он проще всего отображается при реализации интерфейса библиотекой videojs и инструментарием AWS Lambda, Elastic Transcoder, S3 с уведомлением AWS SNS.

Amazon Elastic Transcoder это транскодирование медиа в облаке. Он разработан как способ для конвертировать мультимедийные файлы из их исходного формата в версии, которые будут воспроизводиться на ПК. Три

основных компонента AWS Elastic Transcoder: конвейер, задания, шаблоны. Все это позволяет в различных режимах захватить видео из хранилища, обработать его и сохранить в хранилище в нужном формате.

Создадим сегмент S3 с именем **transcode-vdo-to-audio** и две папки в этом сегменте.

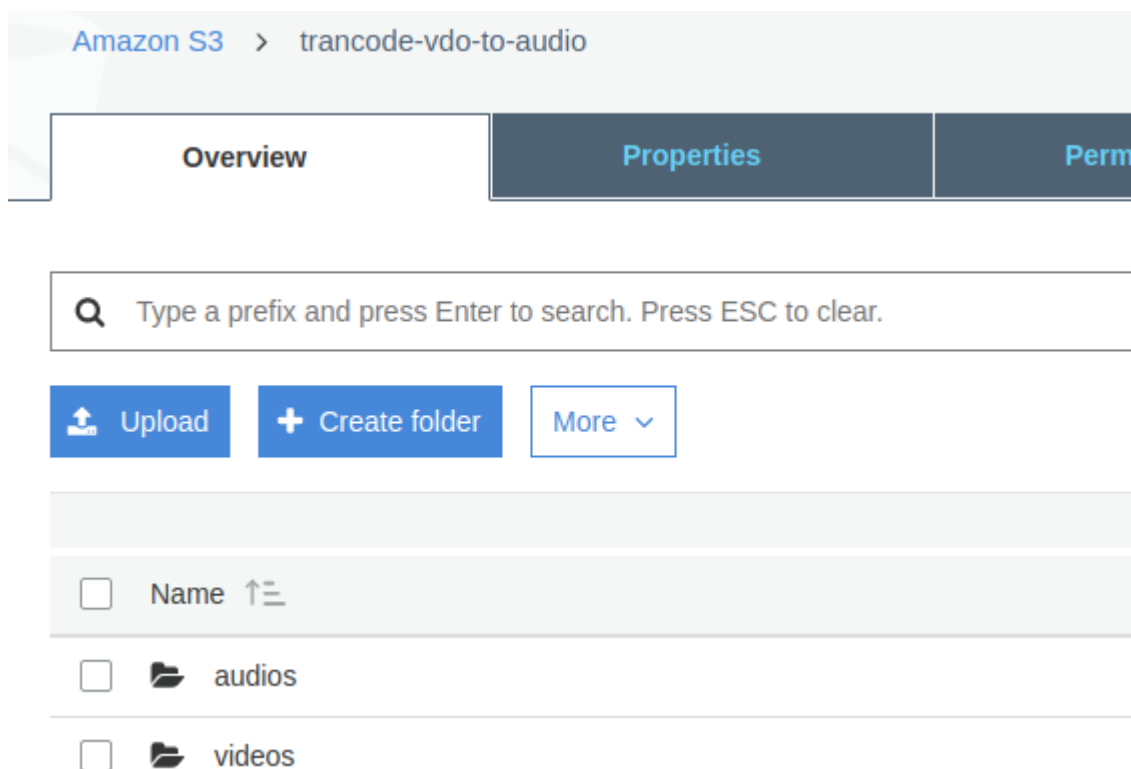


Рисунок 2.14 Настройка хранилища файлов

Затем перейдите к AWS SNS и создайте две темы с именем error и complete для получения уведомлений, если транскодер выдаст ошибку или завершит конвейер процесса транскодирования.

Topics

Publish to topicCreate new topicActions ▾

Filter

☐	Name	ARN
☐	complete	arn:aws:sns:ap-south-1:852015740547:complete
☐	error	arn:aws:sns:ap-south-1:852015740547:error

Рисунок 2.15 Настройка уведомлений

В поле Endpoint укажите адрес в домене Gmail. Создайте еще одну подписку для отправки сообщений об ошибке темы ошибки. После чего на адрес электронной почты должно прийти электронное письмо для подтверждения подписки.

Следующим шагом создайте новый pipeline на панели управления Elastic Transcoder, для обработки видео с разрабатываемого сайта.

Edit Transcoding Pipeline

A pipeline is a queue for your transcoding jobs. You can have more than one pipeline per AWS account. You can use multiple pipelines to organize y for high-priority jobs.

Pipeline Name ⓘ

Input Bucket ⓘ

IAM Role ⓘ
Elastic Transcoder previously created a default IAM role for this AWS account. [View the policy.](#)

Configuration for Amazon S3 Bucket for Transcoded Files and Playlists

Bucket ⓘ

Storage Class ⓘ

[+ Add Permission](#)

Configuration for Amazon S3 Bucket for Thumbnails

Bucket ⓘ

Storage Class ⓘ

Рисунок 2.16 Настройка кодирования видео

▼ Notifications (Optional)

On Progressing Event ☒ No Notifications ⓘ
☐ Use an existing SNS topic
☐ Create a New SNS Topic

On Warning Event ☒ No Notifications ⓘ
☐ Use an existing SNS topic
☐ Create a New SNS Topic

On Completion Event ☐ No Notifications ⓘ
☒ Use an existing SNS topic
☐ Create a New SNS Topic

Select a Topic ⓘ

On Error Event ☐ No Notifications ⓘ
☒ Use an existing SNS topic
☐ Create a New SNS Topic

Select a Topic ⓘ

▼ Encryption (Optional)

AWS KMS Key ARN ☒ Default ⓘ
☐ Custom

Рисунок 2.17 Настройка кодирования видео (продолжение)

					ПГУ 1.090501.17.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		41

После создания конвейера система генерирует уникальный код `pipe_id`. Скопируйте этот код для клиентской части приложения `pipe_id`. Данный код является ключом для загрузки приложением видео файлов с систему через web интерфейс.

Далее нужно создать роль IAM для лямбды, чтобы AWS функция могла обращаться к s3, clouwatch, transcoder и sns. Перейдите в IAM и создайте роль с именем `lamda_s3_role`, чтобы предоставить ему необходимый доступ. Теперь перейдите на панель управления сервисом Lambda и нажмите кнопку «Создать функцию».

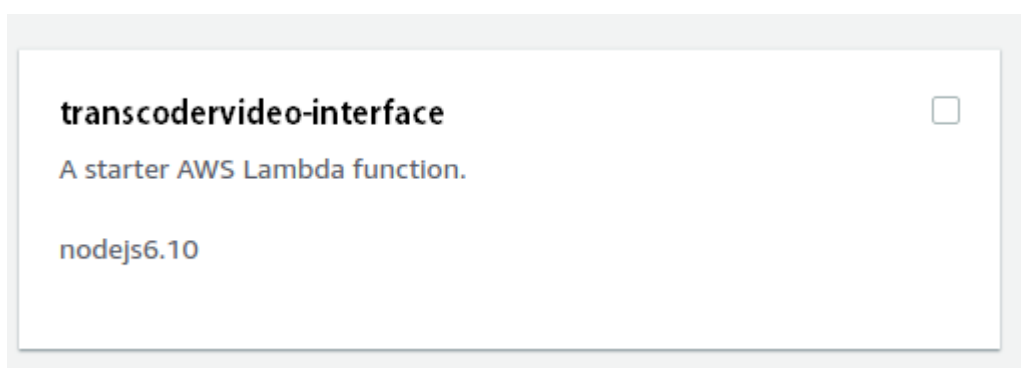


Рисунок 2.18 Выбор языка для AWS Lambda

Далее создаем саму AWS Lambda для реализации приема и кодирования видео из хранилища.

В появившемся окне ввода тела функции требуется ввести код ее исполнения:

```
'use strict';var AWS = require('aws-sdk'),
    transcoder = new AWS.ElasticTranscoder({
    });exports.handler = (event, context, callback) => {
    let fileName = event.Records[0].s3.object.key;
    var srcKey =
    decodeURIComponent(event.Records[0].s3.object.key.replace(/\+/g, "
    "));
    var newKey = fileName.split('.')[0];
```

```

    console.log('New video has been uploaded:',
fileName);transcoder.createJob({
    PipelineId: process.env.PIPELINE_ID,
    Input: {
        Key: srcKey,
        FrameRate: 'auto',
        Resolution: 'auto',
        AspectRatio: 'auto',
        Interlaced: 'auto',

```

Basic information [Info](#)

Name*

myVideoTranscoderLambdaFunction

Role*

Defines the permissions of your function. Note that new roles may not be available for a few minute about Lambda execution roles.

Choose an existing role ▼

Existing role*

You may use an existing role with this function. Note that the role must be assumable by Lambda an Logs permissions.

lambda_s3_role ▼

Рисунок 2.19 Реализация AWS Lambda

```

    Container: 'auto'
},
Output: {
    Key: getOutputName(fileName),
    ThumbnailPattern: '',
    PresetId: '1351620000001-300040',
    Rotate: 'auto'

```

```

    }
  }, function(err, data){
    if(err){
      console.log('Something went wrong:',err)
    }else{
      console.log('Converting is done');
    }
    callback(err, data);
  });
};function getOutputName(srcKey){
  let baseName = srcKey.replace('videos/', '');
  let withoutExtension = removeExtension(baseName);
  return 'audios/' + withoutExtension + '.mp3';
}function removeExtension(srcKey){
  let lastDotPosition = srcKey.lastIndexOf(".");
  if (lastDotPosition === -1) return srcKey;
  else return srcKey.substr(0, lastDotPosition);
}

```

Далее в переменные среды AWS добавляем триггер s3, чтобы при загрузке в хранилище видеофайла срабатывало выполнение лямбда-функции.

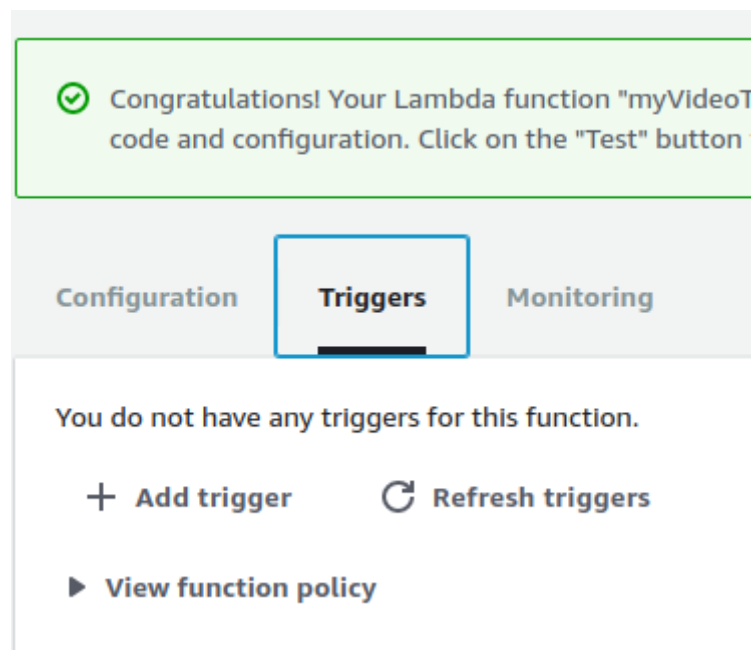


Рисунок 2.20 Создание триггера

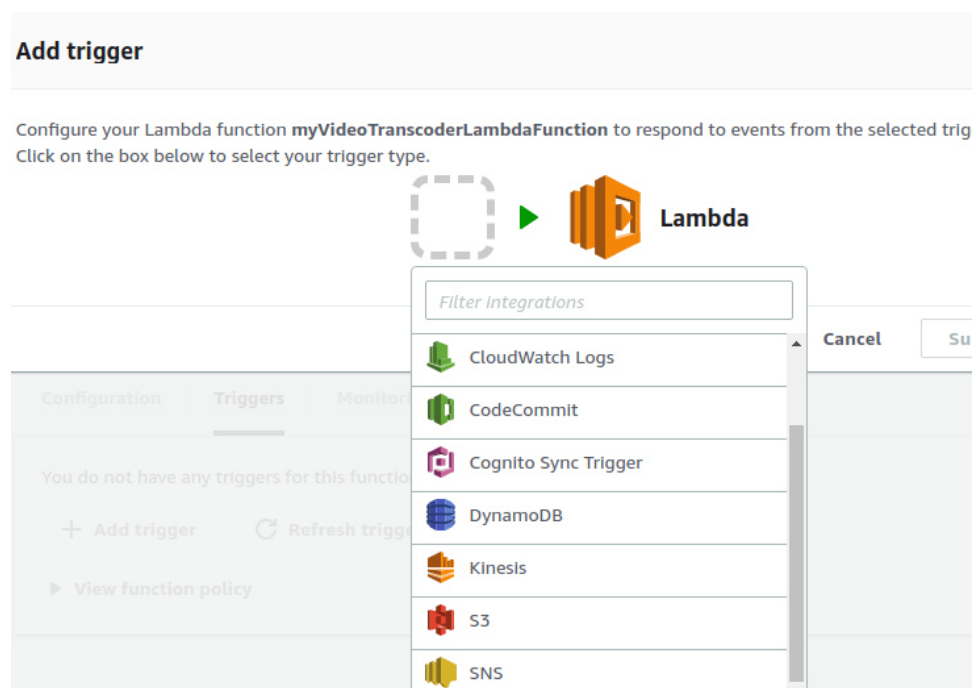


Рисунок 2.21 Настройка триггера на событие

Click on the box below to select your trigger type.



Bucket
Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.

trancode-vdo-to-audio

Event type
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.

Object Created (All)

Prefix
Enter an optional prefix to limit the notifications to objects with keys that start with matching characters.

videos/

Suffix
Enter an optional suffix to limit the notifications to objects with keys that end with matching characters.

e.g. .jpg

Lambda will add the necessary permissions for Amazon S3 to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

Enable trigger
Enable the trigger now, or create it in a disabled state for testing (recommended).

☒

Cancel Submit

Рисунок 2.22 Настройка связи хранилища и Lambda функции

Для тестирования системы загружаем файл в хранилище Amazon.

Pipelines	Create New Job	Enter New Search Para
Jobs	Cancel Copy	
Presets	Filter:	View
	ID	Pipeline ID
	Submit Time	First Input Key
	# of Inputs	# of Outputs
	Status	
	1510503061642-ys3fz6	1510502344860-xumx5j
	2017-11-12T22:11:01+0600	videos/video1.3gp
	1	1
	Complete	

Рисунок 2.23 Загрузка файла в систему

После чего проверим отчеты системы и почтовый ящик.

CloudWatch > Log Groups > /aws/lambda/myVideoTranscoderLambdaFunction >
 [\$LATEST]172c95da280d47dfab4de4363bf3a114

Expand all ☒ Row ☐ Text   

Filter events

all30s5m1h6h1d1wcustom

Time (UTC +00:00)	Message
No older events found at the moment. Retry .	
▶ 16:11:00	START RequestId: 12be2cc4-c7c4-11e7-8040-59ccb2fcd59a Version: \$LATEST
▶ 16:11:00	T16:11:00.801Z 12be2cc4-c7c4-11e7-8040-59ccb2fcd59a New video has been uplo
▶ 16:11:01	T16:11:01.805Z 12be2cc4-c7c4-11e7-8040-59ccb2fcd59a Converting is done
▶ 16:11:01	END RequestId: 12be2cc4-c7c4-11e7-8040-59ccb2fcd59a
▶ 16:11:01	REPORT RequestId: 12be2cc4-c7c4-11e7-8040-59ccb2fcd59a Duration: 1063.31 ms Billed Du
No newer events found at the moment. Retry .	

Рисунок 2.24 Отчет о загрузке и кодировании файла в систему

Теперь осталось только реализовать просмотр данных файлов и загрузку их через web интерфейс (приложение) и данная часть системы полностью готова к использованию.

2.3 Реализация интерфейса видеоплеера

В данном разделе опишем способы реализации воспроизведения видео на странице разрабатываемого ресурса. Добавление и управление элементами управления объектов video средствами java script и html.

В библиотеке Videojs существует объект video, который сам по себе уже обеспечивает необходимый набор контролов для управления проигрыванием. Чтобы была видна панель управления воспроизведением, достаточно указать атрибут controls.

```
<video src="trailer_480p.mp4" width="480" height="270" poster="poster.gif"
controls />
```

Однако с стандартными элементами управления есть проблема, которая заключается в том, что выглядят они нестандартно. Другими словами, в каждом браузере они выглядят по-своему.

Для работы с медиаданными в библиотеке есть DOM интерфейс HTMLVideoElement, наследующий в свою очередь интерфейс HTMLMediaElement.

Это общий интерфейс для обоих медиа-элементов (аудио и видео), описывающий доступ к базовым возможностям работы с медиа-контентом: контроль источника контента, управление воспроизведением, изменение уровня звука и обработка ошибок.

Основные свойства и методы, которые нам понадобятся:

Состояние сети и готовность к работе: *src* ссылка (url) на воспроизводимый контент, *buffered* буферизованные куски видео.

Воспроизведение и контролы: *currentTime* текущий момент проигрывания, *duration* длительность медиа-контента), *paused* находится ли воспроизведение на паузе, *ended* закончилось ли проигрывание, *muted* включение/выключение звука, *volume* уровень звука [0, 1], *play()* начать проигрывание, *pause()* поставить на паузу.

События: *oncanplay* можно начать проигрывание, *ontimeupdate* изменена позиция проигрывания, *onplay* запущено проигрыв, *onpause* нажата пауза, *onended* воспроизведение закончилось.

Видео отличается от аудио несколькими дополнительными свойствами: *width* и *height* ширина и высота контейнера для проигрывания видео; *videoWidth* и *videoHeight* внутреннее значение ширины и высоты видео, если размеры не известны, равны 0; *poster* ссылка на картинку, которую можно показывать, пока видео недоступно (обычно это один из первых непустых кадров).

Разница между *width/height* и *videoWidth/videoHeight* в том, что последние это собственные характеристики видео, в частности, с учетом соотношения сторон и других характеристик, в то время как контейнер для видео может быть любых размеров (больше, меньше, с другой пропорцией).

Создание видео-плеера начинается с задачи запускать видео на проигрывание и останавливать воспроизведение. Для этого понадобятся методы `play()` и `pause()` и несколько свойств, описывающих текущее состояние видео-потока (также будет использоваться библиотека jQuery).

```
<div>
  <video id="myvideo" width="480" height="270" poster="poster.gif" >
    <source src="trailer_480p.mp4" type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"' />
    <source src="trailer_480p.webm" type='video/webm; codecs="vorbis, vp8"' />
  </video>
</div>
<div id="controls">
  <span id="playpause" class="paused" >Play</span>
</div>
```

После того как каркас видео плеера готов, необходимо реализовать функцию js, чтобы нажатие на кнопку play переключало ее состояние и соответственно запускало видео-ролик или ставило его на паузу.

```
$(document).ready(function(){
    var controls = {
        video: $("#myvideo"), playpause: $("#playpause")
    };

    var video = controls.video[0];

    controls.playpause.click(function(){
        if (video.paused) {
            video.play();
            $(this).text("Pause");
        }
        else {
            video.pause();
            $(this).text("Play");
        }
        $(this).toggleClass("paused"); });
});
```

Для реализации функции Play и кликабельное видео, чтобы проигрывание и пауза переключались по нажатию на само видео добавим следующий код:

```
video.addEventListener("ended", function() {  
    video.pause();  
    controls.playpause.text("Play");  
    controls.playpause.toggleClass("paused");  
});  
  
controls.video.click(function() {  
    controls.togglePlayback();  
});
```

В результате в браузере появится следующий промежуточный результат:



Рисунок 2.25 Настройка элементов управления.

То что место воспроизведения видео пустое, говорит о том что в коде отсутствует ссылка на видео.

Перейдем к отображению прогресса проигрывания. Для начала необходимо добавить несколько элементов управления, которые будут использоваться для отображения текущего состояния и управления текущей позицией:

```
<span id="progress">  
<span id="total">
```

```

<span id="buffered"><span id="current"></span></span> </span>
</span>
<span id="time">
<span id="currenttime">00:00</span> /
<span id="duration">00:00</span>
</span>

```

Во вторых нужно понять, какова длительность ролика, для этого у video-элемента есть свойство duration. Отследить это значение можно, например, в момент готовности ролика к проигрыванию по событию oncanplay:

```

video.addEventListener("canplay", function() {

controls.hasHours = (video.duration / 3600) >= 1.0;

controls.duration.text(formatTime(video.duration, controls.hasHours));

controls.currentTime.text(formatTime(0),controls.hasHours);}, false);

```

Также в коде используется специальная функция formatTime для перевода секунд в формат HH:mm:ss или mm:ss:

```

function formatTime(time, hours) {
    if (hours) {
        var h = Math.floor(time / 3600);
        time = time - h * 3600;
        var m = Math.floor(time / 60);
        var s = Math.floor(time % 60);
        return h.lead0(2) + ":" + m.lead0(2) + ":" + s.lead0(2);
    } else {
        var m = Math.floor(time / 60);
        var s = Math.floor(time % 60);
        return m.lead0(2) + ":" + s.lead0(2);
    }
}

Number.prototype.lead0 = function(n) {

```

```

    var nz = "" + this;
    while (nz.length < n) {
        nz = "0" + nz;
    }
    return nz;
};

```

Для отображения процесса проигрывания понадобится событие `ontimeupdate`, срабатывающее при изменении текущего момента:

```

video.addEventListener("timeupdate", function() {
    controls.currentTime.text(formatTime(video.currentTime,
controls.hasHours));
    var progress = Math.floor(video.currentTime) / Math.floor(video.duration);
    controls.progress[0].style.width = Math.floor(progress *
controls.total.width()) + "px"; }, false);

```

Свойство `currentTime` выдает в секундах текущее время. Его же можно использовать, чтобы изменить время проигрывания используя данную конструкцию:

```

controls.total.click(function(e) {
    var x = (e.pageX - this.offsetLeft)/$(this).width();
    video.currentTime = x * video.duration;
});

```

Для буферизации видео можно использовать событие `onprogress`, срабатывающее при загрузке новых порций видео:

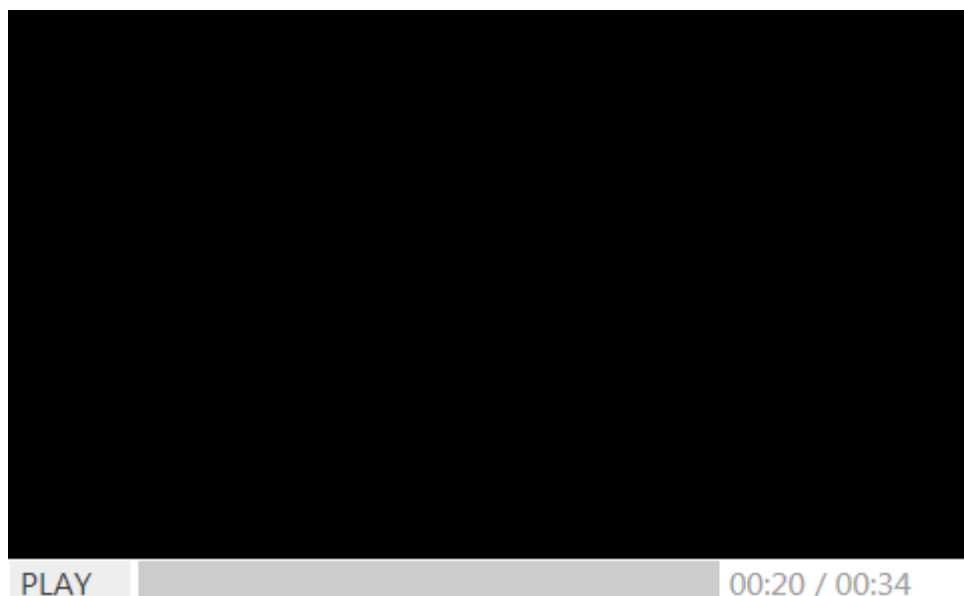


Рисунок 2.26 Настройка элементов управления.

Видео пока не загружено, но при выставлении длины видео, как константы, можно кликами мышки увидеть изменение тайминга.

Добавим к видео-плееру возможность включать и выключать звук. Для этого добавим элемент с динамиком (SVG-иконка):

```
<span id="volume"> <svg id="dynamic" version="1.0" id="Layer_1"
xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"
x="0px" y="0px" width="16px" height="16px" viewBox="0 0 95.465 95.465"> <g >
<polygon points="39.323,20.517 22.705,37.134 0,37.134 0,62.865 22.705,62.865
39.323,79.486 "/> <path d="M52.287,77.218c14.751-15.316,14.751-39.116,0-54.436c-
2.909-3.02-7.493,1.577-4.59,4.59
c12.285,12.757,12.285,32.498,0,45.254C44.794,75.645,49.378,80.241,52.287,77.218L5
2.287,77.218z"/> <path d="M62.619,89.682c21.551-22.103,21.551-57.258,0-79.36c-
2.927-3.001-7.515,1.592-4.592,4.59
c19.08,19.57,19.08,50.608,0,70.179C55.104,88.089,59.692,92.683,62.619,89.682L62.6
19,89.682z"/> <path d="M75.48,99.025c26.646-27.192,26.646-70.855,0-98.051c-2.936-
2.996-7.524,1.601-4.592,4.59
c24.174,24.674,24.174,64.2,0,88.871C67.956,97.428,72.545,102.021,75.48,99.025L75.
48,99.025z"/> </g> </svg> </span>
```

С соответствующими стилями для включенного и выключенного состояний:

```
#dynamic { fill:#333; padding:0 5px; }
#dynamic.off { fill:#ccc; }
```

Для переключения состояния динамика нам понадобится свойство mute:

```
controls.dynamic.click(function() {
    var classes = this.getAttribute("class");
    if (new RegExp('\\boff\\b').test(classes)) {
        classes = classes.replace(" off", "");
    } else {
        classes = classes + " off";
    } this.setAttribute("class", classes);
    video.muted = !video.muted;
});
```

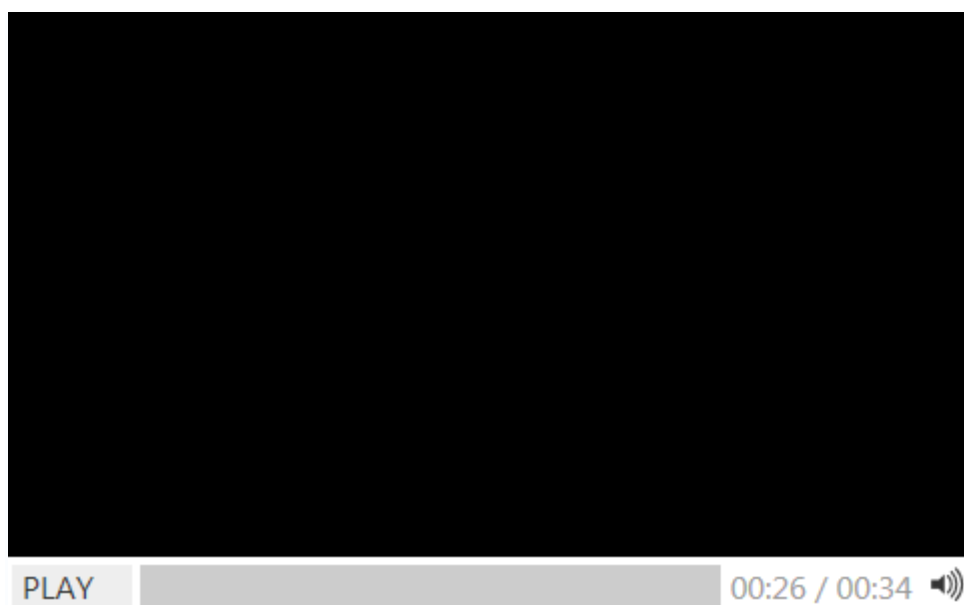


Рисунок 2.27 Настройка элементов управления.

2.4 Описание работы интерфейса системы

Работу системы с точки зрения пользователя можно показать временной диаграммой.

					ПГУ 1.090501.17.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		54

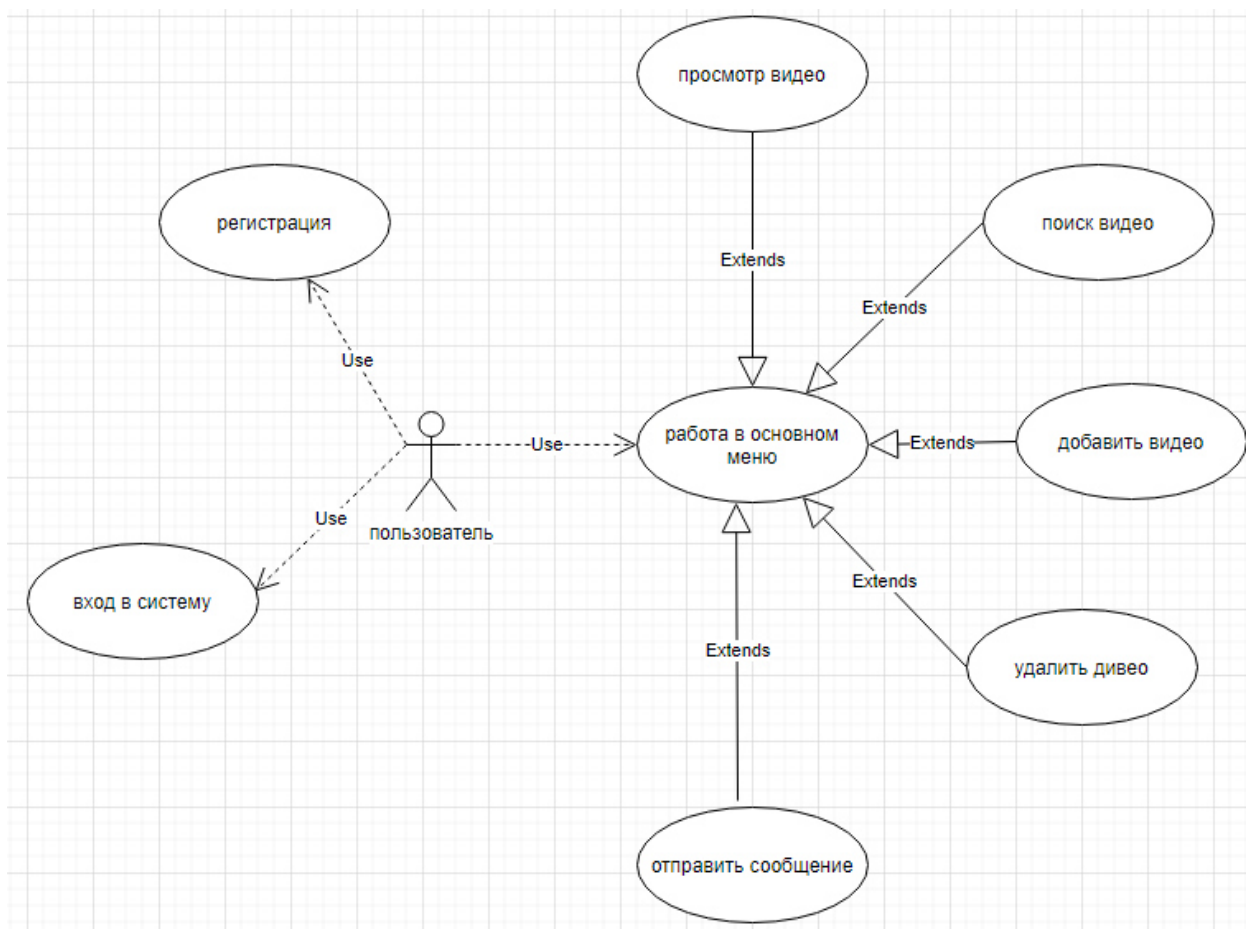


Рисунок 2.28 Функционал интерфейса системы.

Как видно из uml диаграммы, пользователь, прежде чем приступить к работе, должен выполнить одну из двух задач: войти в систему под своим логином и паролем. В данном случае это или e-mail пользователя и пароль, сохраненный в сервисах Amazon или логин и пароль социальных сетей пользователя, которые доступны системе благодаря сервису Cognito.

Если пользователь регистрируется в системе через web интерфейс, то ему на e-mail придет сообщение о его регистрации в системе.

Далее пользователь получает доступ к основному меню приложения и может реализовать следующие действия: просмотреть видео архив, увидеть новые видео в системе, просмотреть видео, добавить комментарий, удалить

и/или добавить видео, также каждому пользователю доступен список пользователей, с которыми он связан.

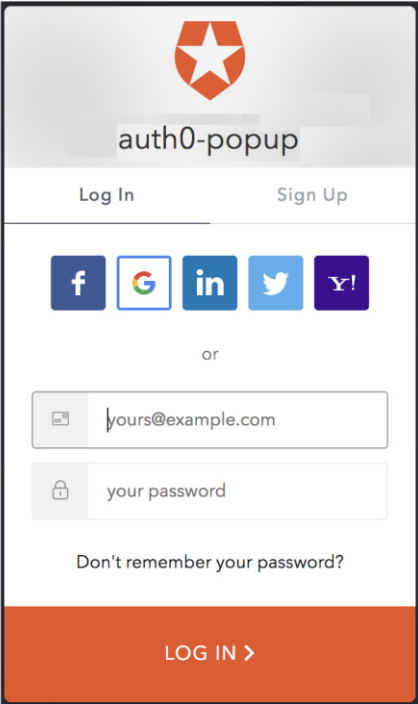


Рисунок 2.29 Вход в систему

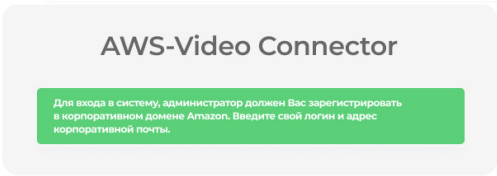
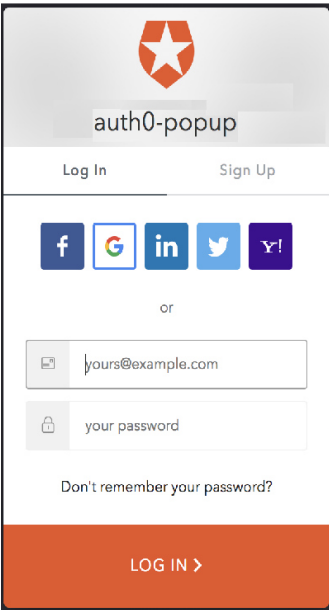


Рисунок 2.30 Ошибка входа в систему

AWS-Video Connector

НОВЫЕВИДЕОПОЛЬЗОВАТЕЛИЗАГРУЗИТЬНАЙТИ

Новые видео

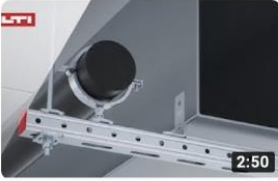


4:00

Монтаж нутрянки БС-117
г. Саратов



ОТПРАВИТЬ



2:50

Монтаж микро БС-97
г. Пенза



ОТПРАВИТЬ

Рисунок 2.31 Главная страница системы

AWS-Video Connector

НОВЫЕВИДЕОПОЛЬЗОВАТЕЛИЗАГРУЗИТЬНАЙТИ

Монтаж нутрянки БС-117 г. Саратов

Проект типового каркаса из прокатных двутавров

Уклон кровли 10%



PLAY00:34 / 4:00

Качество: **Высокое**

ОТПРАВИТЬ

Комментарии

Петров Иван Денисович

Федоров Иван Петрович, Требуется новый коммутатор второго уровня и пара SFP модулей. Закажите на складе. Иван заберет из офиса.

Рисунок 2.32 Просмотр конкретного видео

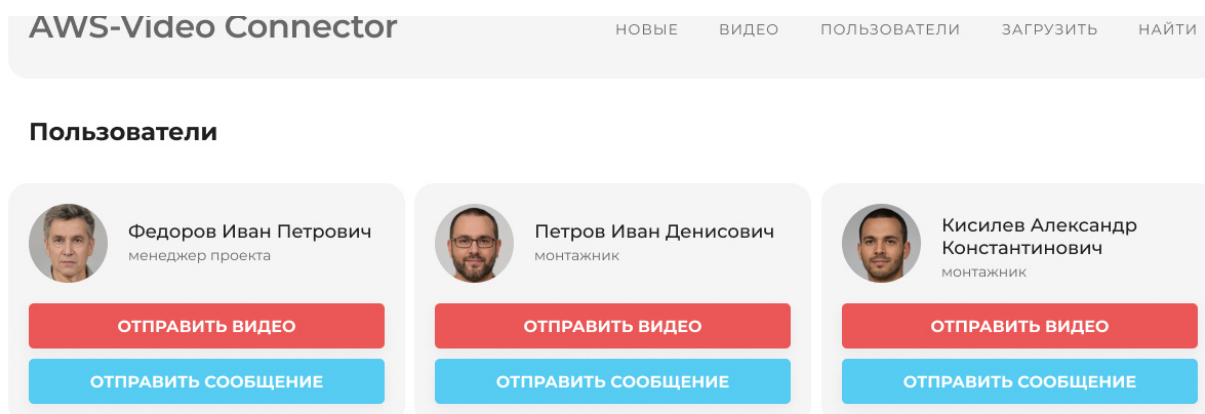


Рисунок 2.33 Список пользователей зарегистрированных в системе

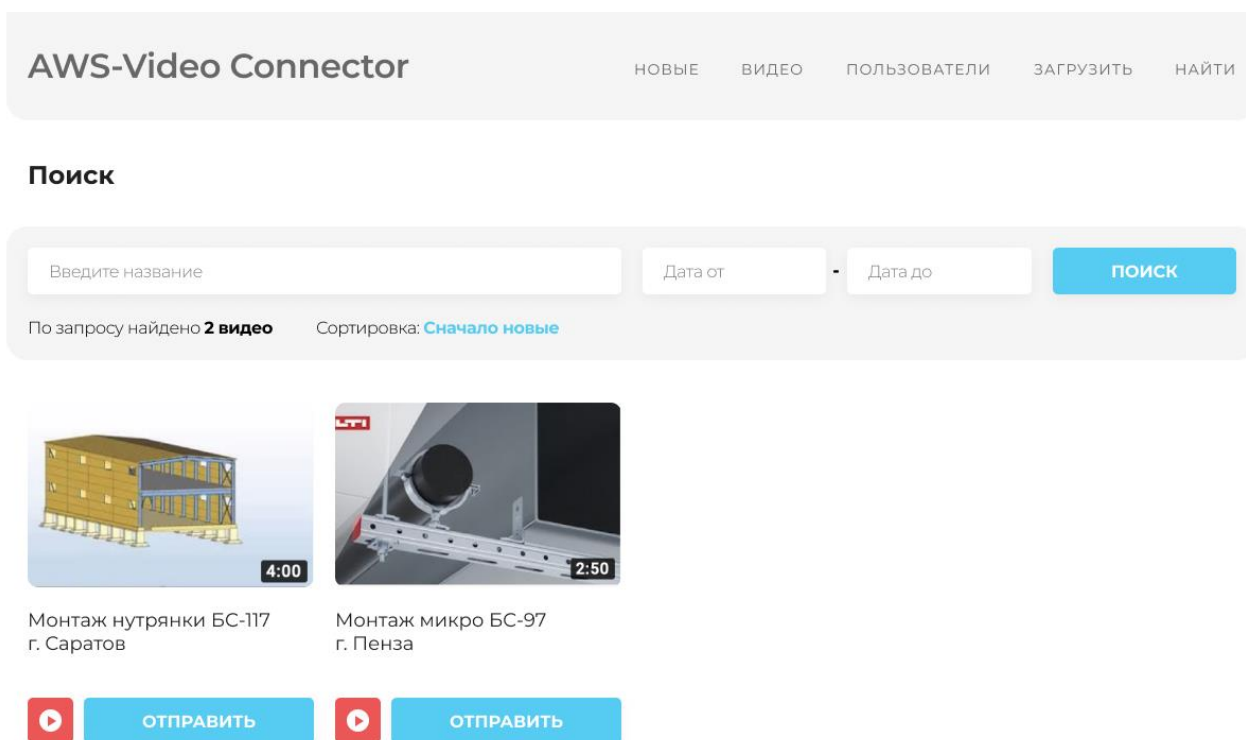


Рисунок 2.34 поиск видео в системе

Отправка видео

Сообщение получателю

Введите текст сообщения...

Кому:

+ Добавить получателя

Видео:

+ Добавить видео

ОТПРАВИТЬ

Рисунок 2.35 Шаблон отправки сообщения

Отправка видео

Сообщение получателю

Введите текст сообщения...

Кому:

+ Добавить получателя

Кисилев Александр
Константинович
монтажникФедоров Иван Петрович
менеджер проектаПетров Иван Денисович
монтажник

Рисунок 2.36 Добавление пользователей к сообщению

Отправка видео

Сообщение получателю

Введите текст сообщения...

Кому:



Кисилев Александр
Константинович
монтажник



+ Добавить получателя

Видео:

+ Добавить видео

ОТПРАВИТЬ

Рисунок 2.37 Удаление пользователей прикрепленных к сообщению

Отправка видео

Сообщение получателю

Введите текст сообщения...

Кому:

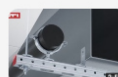
+ Добавить получателя

Видео:

+ Добавить видео



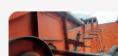
Монтаж нутрянки
БС-117 г. Саратов



Монтаж микро БС-97
г. Пенза



Ремонт БС-56 г. Пенза



Ремонт БС-56 г. Пенза



Рисунок 2.38 Добавление видео к сообщению

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.17.001 ПЗ

Лист

60

Отправка видео

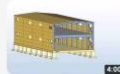
Сообщение получателю

Введите текст сообщения...

Кому:

+ Добавить получателя

Видео:

Монтаж нутрянки
БС-117 г. Саратов

+ Добавить видео

ОТПРАВИТЬ

Рисунок 2.39 Удаление видео прикрепленных к сообщению

Сообщение получателю

Введите текст сообщения...

Кому:

Кисилев Александр
Константинович
МОНТАЖНИК

+ Добавить получателя

Видео:

Монтаж нутрянки
БС-117 г. Саратов

+ Добавить видео

ОТПРАВИТЬ

Рисунок 2.40 Вариант отправки сообщения

Загрузка видео

ВЫБРАТЬ ФАЙЛ ДЛЯ ЗАГРУЗКИ

Рисунок 2.41 Загрузить видео

Загрузка видео

ВЫБРАТЬ ДРУГОЙ ФАЙЛ

Монтаж нутрянки БС-117 г. Саратов



Выбрать качество ▼

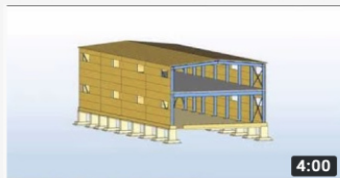
ЗАГРУЗИТЬ

Рисунок 2.42 Выбрать другой файл

Загрузка видео

ВЫБРАТЬ ДРУГОЙ ФАЙЛ

Монтаж нутрянки БС-117 г. Саратов



Выбрать качество ▼

Высокое

Низкое

ЗАГРУЗИТЬ

Рисунок 2.42 Выбрать качество загружаемого файла

Загрузка видео

Видео загружено!

Видео [Монтаж нутрянки БС-117 г. Саратов](#) было успешно загружено

ЗАГРУЗИТЬ ЕЩЁ

или вернуться на [главную страницу](#)

Рисунок 2.42 Выбранный файл загружен

3 ТЕХНОЛОГИЯ ХРАНЕНИЯ МЕДИАКОНТЕНТА В БЕССЕРВЕРНЫХ ПРИЛОЖЕНИЯХ

В данной главе рассмотрим приемы работы с хранилищем данных, которое фактически является СУБД для данного приложения.

Amazon S3 позволяет хранить контент и получать к нему доступ. Amazon S3 хранит данные в виде объектов в корзинах (bucket).

Каждый объект представляет собой файл и, опционально - метаданные, которые этот объект описывают (например Content Type и/или JSON).

Чтобы сохранить данные в S3 необходимо загрузить данные в корзину. После того, как данные загружены необходимо установить ограничения доступа к нему. Корзина - это контейнер, который хранит объекты. Можно иметь неограниченное количество корзин, и для каждой установить доступ (список пользователей с правом создавать, удалять и просматривать список объектов в корзине), просматривать логи доступа к корзине и объектам в ней. Далее рассмотрим процесс создания контейнеров и загрузки файлов более подробно. В данном разделе будут рассмотрены два способа работы с контейнерами: графический web интерфейс и командный режим.

3.1 Создание Bucket

Для создания корзины через веб-интерфейс необходимо перейти в <https://console.aws.amazon.com/s3/> и нажать кнопку Create bucket:



Рисунок 3.1 Начало работы с контейнером

Указываем имя (которое должно быть уникальным среди всех S3 корзин всех пользователей) и регион:

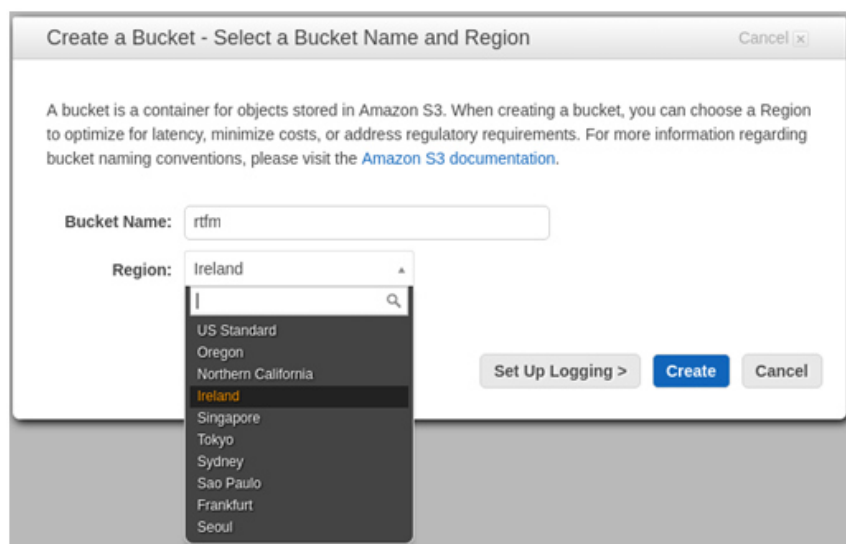


Рисунок 3.2 Начало работы с контейнером

После создания корзины её нельзя переименовать, а её имя будет включено в код приложения, для реализации доступа web интерфейса к контенту.

3.2 Добавление объектов

Объектом для AWS S3 может быть любой файл. После его добавления разработчик и/или пользователь может добавить метаданные и установить права доступа к этому объекту.

Для добавления через веб-интерфейс требуется нажать кнопку Upload:

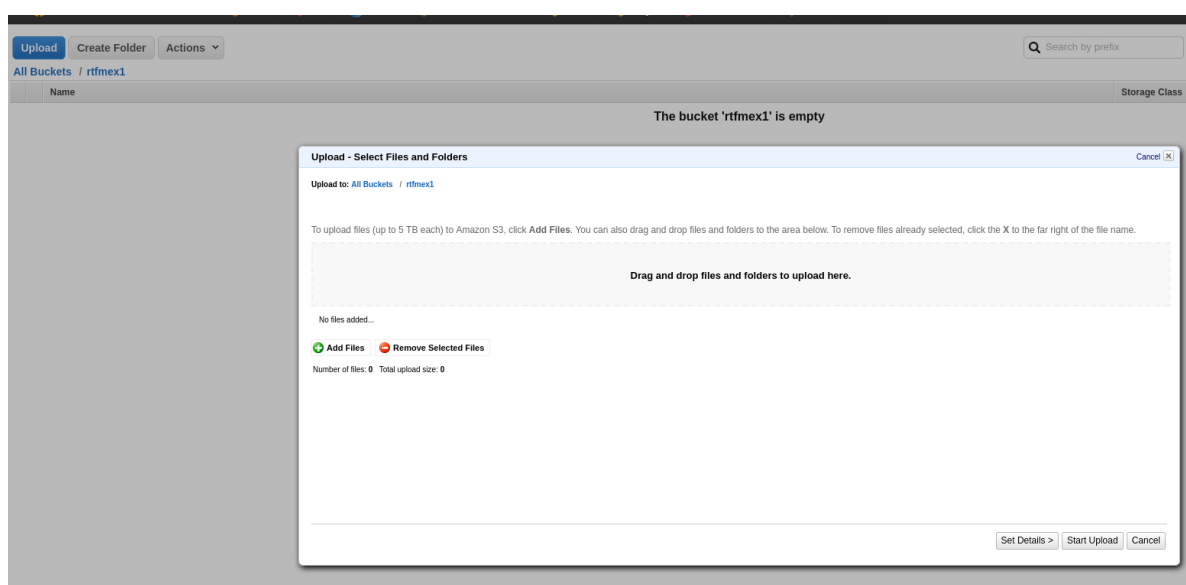


Рисунок 3.3 Шаги загрузки файла в облачное пространство AWS S3

Далее выбирается файл для загрузки, и нажимается кнопка Start upload:

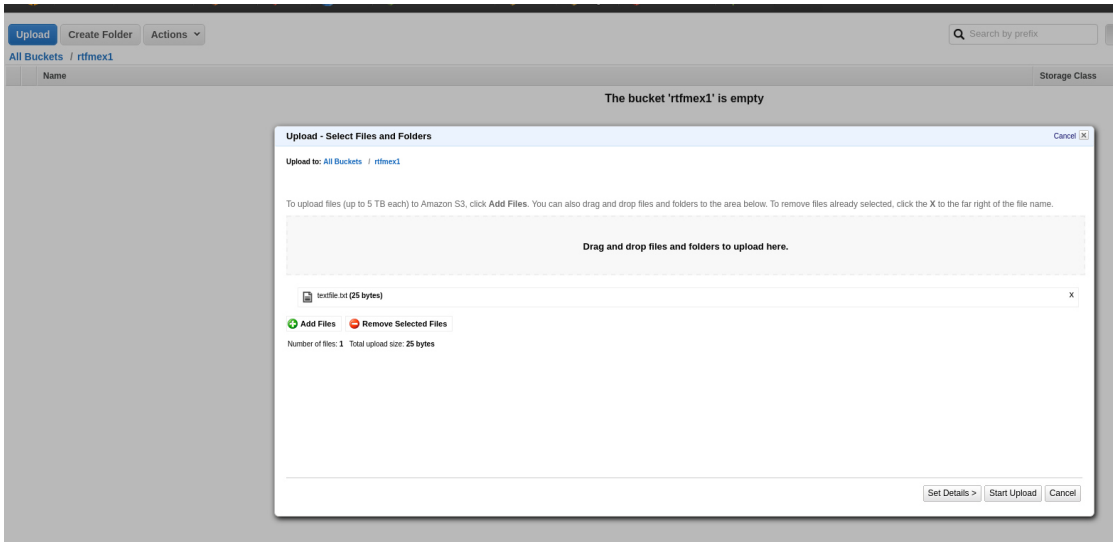


Рисунок 3.4 Шаги загрузки файла в облачное пространство AWS S3

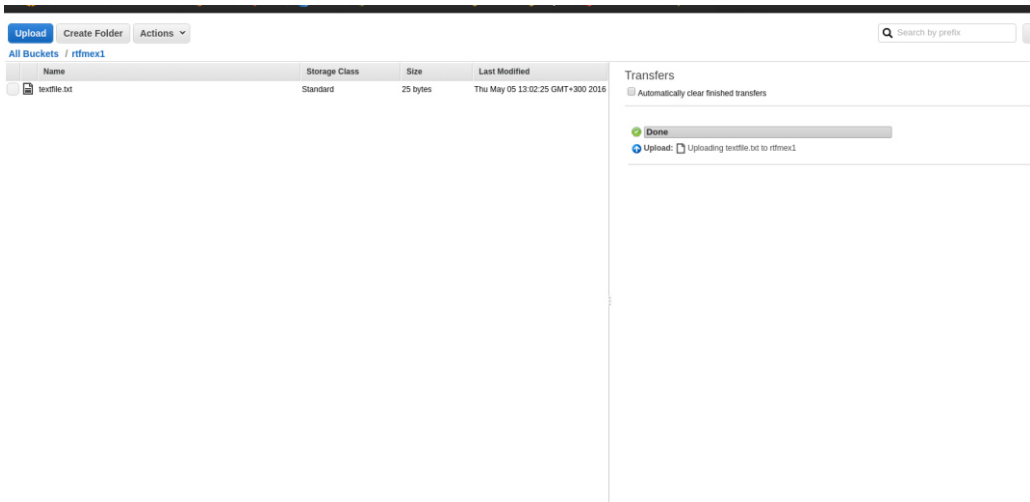


Рисунок 3.5 Шаги загрузки файла в облачное пространство AWS S3

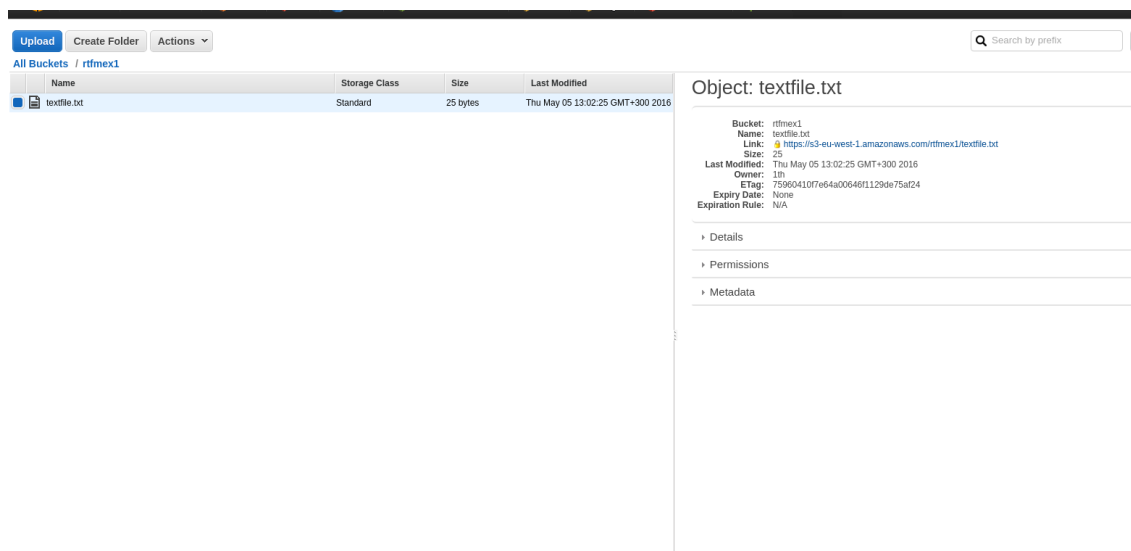


Рисунок 3.6 Просмотр списка файлов в контейнере

Просмотр загруженных файлов с помощью консоли CLI дает тот же результат.

3.3 Права доступа

По умолчанию AWS S3 корзины и объекты в них являются приватными и недоступны для чтения:

\$ curl https://s3-eu-west-1.amazonaws.com/rtfmex1/textfile.txt

<?xml version="1.0" encoding="UTF-8"?>

Результатом запроса будет вывод об ошибке:

```
<Error><Code>AccessDenied</Code><Message>Access
Denied</Message>
```

Все права доступа к корзинам и объектам в них записываются в ACL (Access Control List), которые можно просмотреть с помощью AWS CLI s3api. Просмотреть ACL корзины:

\$ aws s3api get-bucket-acl --bucket rtfmex2

```
$ aws s3api get-bucket-acl --bucket rtfmex2
{
  "Owner": {
    "DisplayName": "username",
```

```

      "ID":
"a8e627b97012bf714c8308e668fe092628e6e69006499397b6cb8f337322a7b
1"
    },
    "Grants": [
      {
        "Grantee": {
          "Type": "CanonicalUser",
          "DisplayName": "username",
          "ID":
"a8e627b97012bf714c8308e668fe092628e6e69006499397b6cb8f337322a7b
1"
        },
        "Permission": "FULL_CONTROL"
      }
    ]
  }
}

```

Просмотр ACL объекта:

```

$ aws s3api get-object-acl --bucket rtfmex2 --key textfile.txt
{
  "Owner": {
    "DisplayName": "username",
    "ID":
"a8e627b97012bf714c8308e668fe092628e6e69006499397b6cb8f337322a7b
1"
  },
  "Grants": [
    {
      "Grantee": {
        "Type": "CanonicalUser",
        "DisplayName": "username",
        "ID":
"a8e627b97012bf714c8308e668fe092628e6e69006499397b6cb8f337322a7b
1"
      },
      "Permission": "FULL_CONTROL"
    }
  ]
}

```

Права доступа могут быть установлены для другого аккаунта AWS или встроенных групп. Владелец по умолчанию имеет полный доступ. Права доступа к корзине и объектам в ней абсолютно независимы: объект в корзине не наследует права доступа этой корзины. Например, если создаётся корзина и даются права записи для другого пользователя — создатель не может

получить доступ к созданным им данным, пока владелец корзины явно не откроет доступ на чтение объектов в созданной корзине. Чтобы установить права доступа выбирается объект или корзина, правой кнопкой и выбирается кнопка Properties:

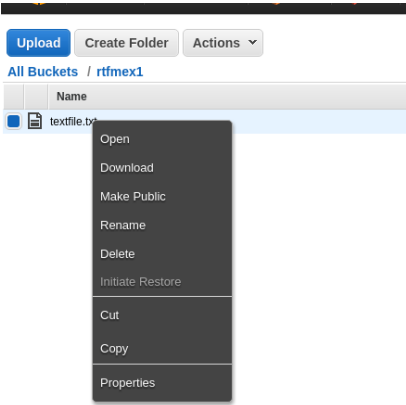


Рисунок 3.7 Права доступа

Затем конкретному файлу добавляем дополнительные права Add more permissions:

Из выпадающего списка в разделе Grantee выбираем следующие опции:
Everyone: открыть анонимный доступ всем,
Authenticated Users: доступ для любого авторизованного пользователя AWS или AIM роли,
Log Delivery: открывает доступ, когда корзина используется в роли хранилища логов сервера,
Me: только root аккаунт, но не AIM пользователь.

В системе имеется возможность указать права доступа, используя canonical user ID или почтовый адрес в поле Grantee.

Найти canonical user ID можно в панели управления > Security credentials > Account Identifiers.

Для AWS S3 имеется три группы правил доступа:
Open/Download: чтение/загрузка объекта залогиненному пользователю

View Permissions: просмотр прав доступа, связанных с этим объектом

Edit Permissions: изменение прав доступа, связанных с этим объектом

Открываем общий (Everyone) доступ на чтение файла:

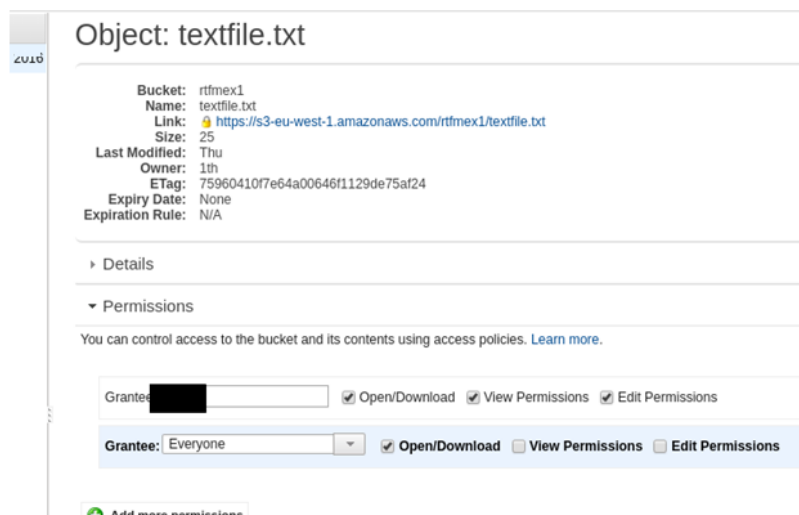


Рисунок 3.8 Права доступа к файлу

Проверка доступа по URL осуществляется в браузере командой:

```
$ curl https://s3-eu-west-1.amazonaws.com/rtfmex2/textfile.txt
```

Результатом такой команды будет ответ сервера: This is textfile from S3.

После настройки прав доступа к контейнеру и файлам в нем, с контейнером можно работать в режиме web доступа.

В результате данной главы были рассмотрены способы работы с хранилищем Amazon и способ проверки доступа пользователя к файлам хранилища из браузера.

4 ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ЭФФЕКТИВНОСТИ ИНВЕСТИЦИЙ ВО ВНЕДРЕНИЕ СИСТЕМЫ УПРАВЛЕНИЯ МЕДИАДАНЫМИ

4.1 Введение

Целью технико-экономического обоснования разработки является количественное и качественное доказательство экономической целесообразности усовершенствования программы, а также определение организационно-экономических условий ее эффективного функционирования.

Эффективность программного модуля определяется его качеством и эффективностью процесса разработки и сопровождения. Качество программного изделия определяется тремя составляющими:

- с точки зрения специалиста-пользователя данного программного продукта;
- с позиции использования ресурсов и их оценки;
- по выполнению требований на программное изделие.

Программное изделие должно быть разработано так, чтобы оно выполняло свои функции без лишних затрат ресурсов (оперативной памяти ПК, машинного времени, пропускной способности каналов передачи данных и др. – на стадии функционирования; времени разработки и денежных ресурсов – на стадии использования программного изделия).

Содержание технико-экономического обоснования разработки программы заключается в следующем:

- расчет технико-экономических показателей и выбор базы сравнения; определение трудоемкости и стоимости программного обеспечения (ПО);

- расчет цены ПО;
- расчет капитальных и эксплуатационных затрат на разработку;
- определение показателей финансово-экономической эффективности.

Приведенные в данном разделе дипломной работы результирующие таблицы, перечни данных, позволяют сопоставить результаты разработки и затраты на нее, чтобы сделать вывод об эффективности проекта. Исходные данные для расчета экономических показателей приведены в таблице 4.1.

Таблица 4.1 Исходные данные для расчета экономических показателей

Обозначение	Наименование показателя	Единицы измерения	Значение показателя		
$C_{ЭВМ}$	Стоимость ЭВМ	тыс. руб.	20		
D_M	Среднее количество дней в месяце	дни	22		
ρ_n	Норматив рентабельности	—	0,24		
ω_d	Коэффициент, учитывающий дополнительную заработную плату разработчика программы	—	0,20		
ω_c	Коэффициент, учитывающий начисления органам социального страхования	—	0,26		
ω_n	Коэффициент, учитывающий накладные расходы организации	—	1,35		
q_I	Количество I-задач, решаемых потребителем	зад. год	60		
$t_{M.B.I}$	Время решения I-ой задачи разработанной программой	маш. час	1		
n_p	Количество организаций, которые приобретут данную программу	шт.	1г	2г	3г
			10	17	25
$Z_{ЭЛ}$	Тариф за 1 кВт/час	руб.	3,44		
T_c	Срок службы разработанной программы	год	3		
$НДС$	Налог на добавленную стоимость	%	20		
T_p	Количество рабочих дней в году	дн	264		
N_{CM}	Количество смен работы ЭВМ	—	1		

Продолжение таблицы 4.1

Обозначение	Наименование показателя	Единицы измерения	Значение показателя
$t_{\text{см}}$	Продолжительность смены	ч	8
α	Простои ЭВМ	%	5
P	Мощность, потребляемая ЭВМ	кВт	0,4
$N_{\text{ср}}$	Среднее количество ремонтов в год	—	2
$S_{\text{д}}$	Стоимость деталей, заменяемых при ремонте	руб.	1300

4.2 Расчет затрат на разработку программного обеспечения

Суммарные затраты на разработку программы рассчитываются по следующей формуле:

$$S_{\text{РП}} = S_{\text{ЗП}} + S_{\text{НАК}}, \quad (4.1)$$

где:

$S_{\text{ЗП}}$ – затраты по заработной плате инженера-программиста;

$S_{\text{НАК}}$ – накладные расходы.

Затраты по заработной плате инженера-программиста рассчитываются по формуле:

$$S_{\text{ЗП}} = \text{ОЗП} \cdot (1 + \omega_{\text{с}}) \cdot (1 + \omega_{\text{д}}) \cdot t_{\text{pi}}, \quad (4.2)$$

где:

ОЗП – основная заработная плата инженера-программиста за месяц (10000 руб.);

t_{pi} – время, необходимое для разработки программы программистом i -го разряда (чел.-мес.);

ω_d – коэффициент, учитывающий дополнительную заработную плату разработчика программы, в долях к сумме основной заработной платы;

ω_c – коэффициент, учитывающий начисления органам социального страхования на заработную плату разработчика программы, в долях к сумме основной заработной плате разработчика.

Программа разрабатывалась 30 дней, если учесть, что в одном месяце 22 рабочих дня, то:

$$t_{p12} = \frac{30}{22} = 1,36 \text{ (чел. –мес.)} \quad (4.3)$$

Таким образом, затраты по заработной плате инженера-программиста составят:

$$S_{зп} = 10000 \cdot (1+0,26) \cdot (1+0,20) \cdot 1,36 = 20693,76 \text{ (руб.)}$$

Накладные затраты рассчитываются с учетом ω_n – коэффициента, определяющего уровень накладных расходов организации по формуле:

$$S_{нак} = ОЗП \cdot \omega_n \cdot t_{p12} , \quad (4.4)$$

$$S_{нак} = 10000 \cdot 1,35 \cdot 1,36 = 18360 \text{ (руб.)}$$

Таким образом, суммарные затраты на разработку программного обеспечения составляют:

$$S_{рп} = 20693,76 + 18360 = 39053,76 \text{ (руб.)}$$

4.3 Расчет цены разработанной программы

Оптовая цена разработанной программы определяется по следующей формуле:

$$Z_{п} = S_{рп} + П , \quad (4.5)$$

где:

$Z_{п}$ – оптовая цена (цена разработчика) (руб.);

$S_{рп}$ – суммарные затраты на разработку программы (руб.);

Π – прибыль, рассчитанная по формуле:

$$\Pi = \rho_n \cdot S_{\text{РП}}, \quad (4.6)$$

где:

ρ_n – норматив рентабельности, учитывающий прибыль организации, разрабатывающей данную программу в долях ко всем затратам данной организации на разработку программы.

Итак,

$$Z_{\Pi} = S_{\text{РП}} \cdot (1 + \rho_n), \quad (4.7)$$

$$Z_{\Pi} = 39053,76 \cdot (1 + 0,24) = 48426,7 \text{ (руб.)}$$

Розничная цена программы рассчитывается с учетом налога на добавленную стоимость ($\text{НДС} = 20\%$) по формуле:

$$Z_{\text{Пр}} = Z_{\Pi} + \text{НДС} = Z_{\Pi} + \frac{Z_{\Pi} \cdot 20}{100} = Z_{\Pi} \cdot (1 + 0,20), \quad (4.8)$$

$$Z_{\text{Пр}} = 48426,7 \cdot (1 + 0,20) = 58112,04 \text{ (руб.)}$$

Выручка от продаж при условии n_{Π} – количество организаций, желающих приобрести программу, составит:

$$B = Z_{\text{Пр}} \cdot n_{\Pi}, \quad (4.9)$$

$$B_1 = 58112,04 \cdot 10 = 581120,4 \text{ (руб.)}$$

$$B_2 = 58112,04 \cdot 17 = 987904,68 \text{ (руб.)}$$

$$B_3 = 58112,04 \cdot 25 = 1452801,0 \text{ (руб.)}$$

4.4 Расчет капитальных вложений

Капиталовложения, связанные с работой ЭВМ рассчитываются по формуле:

$$K_{\text{ЭВМ}} = C_{\text{ЭВМ}} + S_3, \quad (4.10)$$

где:

$C_{\text{ЭВМ}}$ – стоимость ЭВМ (руб.);

S_3 – стоимость запасных частей (руб.);

Итак, произведем расчет коэффициентов входящих в формулу расчета величины капиталовложений:

$$S_3 = 0,15 \cdot C_{\text{ЭВМ}}; \quad (4.11)$$

$$S_3 = 0,15 \cdot 20000 = 3000 \text{ (руб.)};$$

Капиталовложения в ЭВМ составляют:

$$K_{\text{ЭВМ}} = 20000 + 3000 = 23000 \text{ (руб.)}$$

4.5 Расчет эксплуатационных расходов

Эксплуатационные расходы на ЭВМ рассчитываются по формуле:

$$E = (T_{\text{М.В.}} \cdot e_{\text{ч}}) + \frac{Z_{\text{П}}}{T_{\text{С}}}, \quad (4.12)$$

где:

$T_{\text{М.В.}}$ – машинное время для решения задач с помощью разработанной программы, (маш.час/год);

$e_{\text{ч}}$ – эксплуатационные расходы, приходящиеся на 1 час работы ЭВМ;

$Z_{\text{П}}$ – цена, по которой продается программа (руб.);

$T_{\text{С}}$ – срок службы программы (г).

Полезный фонд времени работы ЭВМ рассчитывается по формуле:

$$T_{\text{ПОЛ}} = T_{\text{ОБЩ}} \cdot t_{\text{СМ}} \cdot N_{\text{СМ}} \cdot \left(1 - \frac{\alpha}{100}\right), \quad (4.13)$$

где:

$T_{\text{ОБЩ}}$ – общий фонд времени работы ЭВМ (дни); $T_{\text{ОБЩ}} = T_{\text{Р}}$;

$N_{\text{СМ}}$ – количество смен работы ЭВМ;

$t_{\text{СМ}}$ – время одного рабочего дня (час);

α – простои ЭВМ (в % от общего фонда времени работы ЭВМ).

Полезный фонд времени работы ЭВМ получим:

$$T_{\text{ПОЛ}} = 264 \cdot 8 \cdot 1 \cdot \left(1 - \frac{5}{100}\right) = 2006,4 \text{ (Маш. час /год)}.$$

Машинное время для решения задач с помощью данной программы рассчитывается по формуле:

$$T_{\text{М.В.}} = q_{\text{I}} \cdot t_{\text{М.В. I}}, \quad (4.14)$$

где:

q_{I} – количество I-задач, решаемых потребителем в год(шт.);

$t_{\text{М.В. I}}$ – время решения I-ой задачи, разработанной программой (Маш. час).

$$T_{\text{М.В.}} = 60 \cdot 1 = 60 \text{ (Маш. час / год)}.$$

Эксплуатационные расходы, приходящиеся на 1 час работы ЭВМ, оцениваются по формуле:

$$e_{\text{ч}} = \frac{(A_{\text{О}} + S_{\text{ЗП}} + S_{\text{ЭЛ}} + R_{\text{РМ}})}{T_{\text{ПОЛ}}}, \quad (4.15)$$

где:

A_O – амортизационные отчисления (руб.);

$S_{ЗП}$ – затраты по заработной плате инженера в год (руб./год);

$S_{ЭЛ}$ – стоимость потребляемой энергии (руб.);

$R_{РМ}$ – затраты на ремонт ЭВМ (руб.);

$T_{ПОЛ}$ – полезный годовой фонд работы ЭВМ, (маш.час/год).

Амортизационные отчисления рассчитываются с учетом нормы амортизации ($a_n = 12,5 \%$);

$$A_O = a_n \cdot K_{ЭВМ} = 0,125 \cdot 23000 = 2875,0 \text{ (руб.)}. \quad (4.16)$$

Затраты по заработной плате инженера за год рассчитывается по формуле:

$$S_{ЗП} = (1 + \omega_c) \cdot (1 + \omega_d) \cdot ОЗП \cdot 12, \quad (4.17)$$

где:

ω_c – коэффициент, учитывающий начисления органам социального страхования на заработную плату разработчика программы, в долях к сумме основной заработной плате разработчика.

ω_d – коэффициент, учитывающий дополнительную заработную плату разработчика программы, в долях к сумме основной заработной платы;

$ОЗП$ – основная заработная плата инженера за месяц 8-го разряда (10000 руб.);

Рассчитываем годовые затраты по заработной плате и социальным отчислениям для инженера:

$$S_{ЗП} = (1 + 0,268) \cdot (1 + 0,20) \cdot 10000 \cdot 12 = 182592 \text{ (руб./ год)}$$

Стоимость потребляемой энергии оценивается по формуле:

$$S_{ЭЛ} = P \cdot T_{ПОЛ} \cdot Z_{ЭЛ}, \quad (4.18)$$

где:

P – мощность, потребляемая ЭВМ (кВт);

$T_{\text{пол}}$ – полезный годовой фонд работы ЭВМ (маш.час/год);

$Z_{\text{эл}}$ – тариф за 1 кВт/час (руб. /кВт).

Итак, произведем расчет стоимости потребляемой энергии:

$$S_{\text{эл}} = 0,4 \cdot 2006,4 \cdot 3,44 = 2760,81 \text{ (руб.)}.$$

Затраты на ремонт ЭВМ вычисляются по формуле:

$$R_{\text{рм}} = N_{\text{ср}} \cdot S_{\text{д}}, \quad (4.19)$$

где:

$N_{\text{ср}}$ – среднее количество ремонтов в год;

$S_{\text{д}}$ – стоимость деталей заменяемых при одном ремонте, в среднем.

$$R_{\text{рм}} = 2 \cdot 1300 = 2600 \text{ (руб.)}.$$

Произведем вычисление эксплуатационных расходов, приходящихся на 1 час работы ЭВМ:

$$e_{\text{ч}} = \frac{(2875 + 182592 + 2760,81 + 2600)}{2006,4} = 95,12 \text{ (руб./час)}.$$

Далее вычислим эксплуатационные расходы на ЭВМ:

$$E = (60 \cdot 95,12) + \frac{48426,7}{3} = 21849,43 \text{ (руб.)}.$$

4.6 Определение расходов на интернет трафик

Для облегчения работы программиста и повышения его производительности в проекте предусмотрено использование глобальной сети интернет.

В настоящее время на пензенском рынке интернет очень много провайдеров предлагающие свои услуги. Цены предлагаемые основными провайдерами сведем в таблицу 4.2, для анализа и экономически целесообразного выбора поставщика услуги по интернет трафику.

Таблица 4.2 Таблица интернет трафиков и провайдеров

Название провайдера	Предлагаемый тариф	Стоимость тарифа (руб.)	Скорость доступа (Мбит/с)
Ростелеком	Пакет-Для впечатлений	699	100
ЗАО «Золотая Линия»	VIP	500	100
Дом.РУ	Скорость света	550	100

Т.к. выше перечисленные провайдеры предлагают без лимитные интернет трафики с одинаковой скоростью доступа в сеть интернет, то экономически целесообразно выбрать трафик с самой низкой стоимостью. Выбираем ЗАО "Золотая линия", тариф VIP, с абонентской платой в месяц 500 рублей.

4.7 Заключение

Обобщенные технико-экономические показатели разработки программы сведены в таблицу 4.3.

Таблица 4.3 Итоговая таблица

Показатель	Значение
Капитальные вложения (руб.)	23000
Эксплуатационные расходы (руб.)	21849,43

Продолжение таблицы 4.3

Показатель	Значение
Затраты на разработку ПО (руб.)	39053,76
Затраты на интернет трафик (руб.)	500
Затраты на заработную плату программисту (руб.)	20693,76

5. БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ В ПОМЕЩЕНИЯХ ВЫЧИСЛИТЕЛЬНЫХ ЦЕНТРОВ.

Одним из направлений научно-технического прогресса, является развитие компьютерной техники и ее широкое применение на всех стадиях научно-исследовательской, проектно-конструкторской работ.

При этом надо отметить, что работы связанные с применением компьютеров, требуют:

1. обеспечение электро-безопасности
2. обеспечение противопожарной безопасности
3. обеспечение санитарно-бытовых условий
4. освещение

5.1 Обеспечение электро-безопасности

В основе защитного действия заземления лежит явление стекания тока в землю при непосредственном соприкосновении проводника с током и земли, что используется не только для создания коллективных средств защиты человека от поражения электрическим током, но и для обеспечения нормальной работы электроустановок и другого оборудования, где необходимо поддерживать нулевой потенциал.

Заземлением называется намеренное электрическое соединение нормально не находящихся под напряжением металлических частей электроустановки непосредственно с землей или ее эквивалентом.

Существуют три вида заземления: рабочее, физическое и защитное.

Рабочим заземлением называется заземление какой-либо точки токоведущих частей электроустановки, необходимое для обеспечения её работы.

Физическое заземление применяется в научно-исследовательских лабораториях для заземления корпусов, экранов, цепей физических приборов и установок.

Защитным заземлением называется заземление частей электроустановок с целью обеспечения электробезопасности.

Заземлителем называется проводник или совокупность электрически соединённых между собой проводников, находящихся в непосредственном соприкосновении с землёй.

Естественными заземлителями называют находящиеся в соприкосновении с землёй электропроводящие части коммуникаций и сооружений производственного или другого назначения.

Заземляющим проводником называется проводник, соединяющий заземляемые части установки с заземлителем.

Заземляющим устройством называется совокупность заземлителя и заземляющих проводников.

Замыканием на корпус называется замыкание, возникшее в электрических машинах, аппаратах, приборах, сетях на конструктивные части электроустановок, нормально не находящихся под напряжением.

Напряжением прикосновения называется напряжение между двумя точками цепи тока замыкания на землю (на корпус) при одновременном прикосновении к ним человека.

Напряжением шага называется напряжение между двумя точками земли, обусловленное растеканием тока замыкания на землю, при одновременном касании их ногами человека.

5.1.1 Классификация помещений по степени опасности поражения людей электрическим током

В соответствии с Правилами устройства электроустановок (ПУЭ) помещения по степени опасности поражения людей электрическим током подразделяются на три класса:

1. Помещения без повышенной опасности
2. Помещения с повышенной опасностью
- 3.. Помещения особо опасные

В зависимости от класса опасности помещения применяются различные средства защиты от поражения человеком электрическим током.

Например, в ПУЭ определены допустимые безопасные значения напряжения прикосновения:

- 12 В - для помещений особо опасных
- 42 В - для помещений с повышенной опасностью
- 65 В - для помещений без повышенной опасности.

5.1.2 Типы заземляющих устройств

Назначение защитного заземления - устранение опасности поражения людей электрическим током при замыкании на корпус.

Принцип действия защитного заземления - снижение до безопасных значений напряжения прикосновения, обусловленного замыканием фазы на корпус. Это достигается уменьшением потенциала заземленного оборудования, а также уменьшением напряжения прикосновения за счёт увеличения потенциала основания, на котором стоит человек, до потенциала, близкого по значению к потенциалу заземлённого оборудования.

Область применения защитного заземления - трёхфазные трёхпроводные сети напряжением до 1000 В с изолированной нейтралью, а выше 1000В с любым режимом нейтрали.

Различают два типа заземляющих устройств:

- выносные (или сосредоточенные)
- контурные (или распределительные).

Выносное заземляющее устройство характеризуется тем, что заземлители вынесены за пределы площадки, на которой размещено заземляемое оборудование, или сосредоточены на некоторой части этой площадки. Защитное действие выносного заземляющего устройства заключается в снижении потенциала на заземлённом оборудовании из-за перегрузки источника тока и перекоса фаз. Данный тип заземляющего устройства применяется лишь в сетях малой мощности. Преимуществом такого типа заземляющего устройства является возможность выбора места размещения электродов с наименьшим сопротивлением грунта (сырое, глинистое, в низинах и т.д.).

Контурное заземляющее устройство характеризуется тем, что его заземлители размещают по контуру (периметру) площадки, на которой находится заземляемое оборудование, или распределяют по всей площадке по возможности равномерно.

Безопасность при контурном заземлителе обеспечивается увеличением потенциала на защищаемой территории до потенциала фазы. В результате этого напряжение прикосновения уменьшается до безопасных значений.

5.1.3 Требования к величине сопротивления заземляющих устройств

Согласно ПУЭ в электроустановках напряжением до 1000 В с изолированной нейтралью сопротивление заземляющего устройства должно быть не более 4 Ом.

При суммарной мощности источников тока до 100 кВт·А, заземляющие устройства могут иметь сопротивление не более 10 Ом.

5.1.4 Оборудование, подлежащее защитному заземлению

Защитному заземлению в соответствии с ГОСТ 12.1.030-81 подлежат металлические части электроустановок (с изолированной нейтралью),

					ПГУ 1.090501.17.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		85

доступные для прикосновения человека и не имеющие других видов защиты, обеспечивающих электробезопасность.

Защитное заземление следует выполнять:

- при номинальном напряжении 300 В и выше переменного тока и 440В и выше постоянного тока - во всех случаях;
- при номинальном напряжении от 42 В до 300 В переменного тока и от 110 В до 440 В постоянного тока при работах в условиях с повышенной опасностью и особо опасных;
- во взрывоопасных помещениях заземление выполняется независимо от значения напряжения.

Проверка заземляющего устройства осуществляется по окончании монтажа, а затем периодически в процессе эксплуатации устройства не реже одного раза в год в период наименьшей проводимости почвы: летом - при наибольшем просыхании, а зимой - при наибольшем промерзании.

5.1.5 Пример электроустановки, подключенной к сети с изолированной нейтралью.

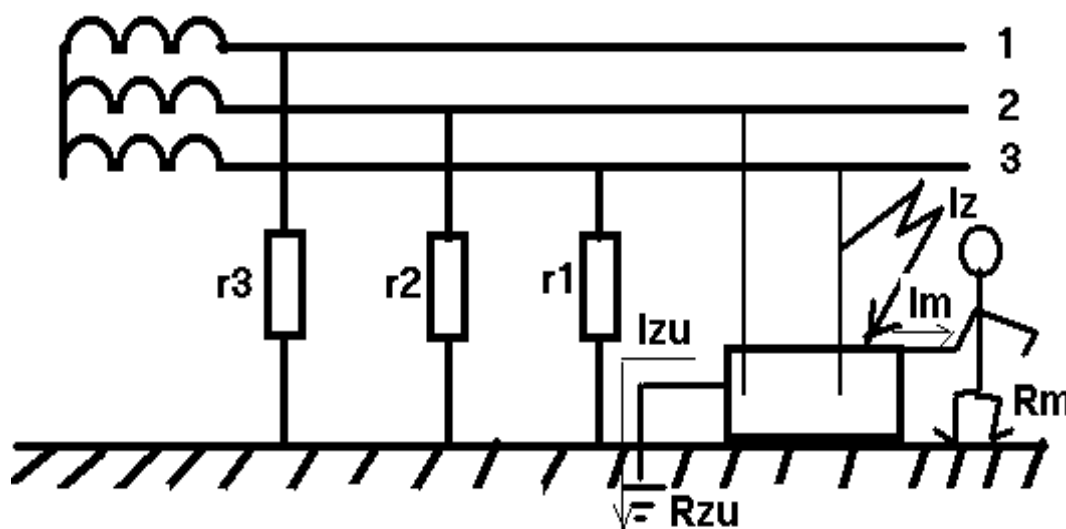


Рисунок 5.1. Схема защитного заземления.

На рисунке 5.1 показана электроустановка, подключенная к сети с изолированной нейтралью, в которой произошло замыкание одной из фаз на корпус. Корпус электроустановки, которого касается человек, соединен с землей через заземляющее устройство с сопротивлением растеканию тока R_{zu} . Предположим, что $R_{zu} \ll R_m$, $I_z \cdot R_{zu} = I_m \cdot R_m$ тогда, как $R_{zu} \parallel R_m$, $I_{zu} \gg I_m$ и $I_z = I_{zu} + I_m \cong I_{zu}$.

Но сила тока замыкания I_z в сети с изолированной нейтралью равна $I_z \cong 3 \cdot U / r_{iz}$ тогда из соотношения можно найти силу тока, протекающего через тело человека:

$$I_m = 3 \cdot U / r_{iz} \cdot \left(\frac{R_m}{R_{zu}} \right) \quad (5.1)$$

Отсюда можно сделать вывод, что принцип действия защитного заземления заключается в уменьшении силы тока, протекающего через тело человека.

5.2 Расчет заземляющего устройства

Расчет заземляющего устройства при проектировании защитного заземления сводится к определению числа вертикальных электродов и определению сопротивления заземляющего устройства, состоящего из рассчитанного числа вертикальных электродов, соединенных металлической полосой и заземляющего проводника.

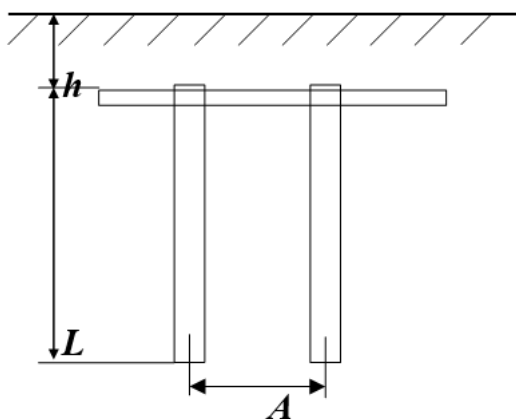


Рисунок 5.2 Определение числа вертикальных электродов

Число одиночных необъединенных вертикальных электродов, одинаковых по размерам и симметрично расположенных в однородном грунте, рассчитывают методом последовательного приближения по формуле (5.2):

$$n = R_0 \cdot \eta_c / R_{zd} \cdot \eta_T \quad (5.2)$$

где n - число вертикальных электродов;

R_0 - измеренное сопротивление одиночного вертикального электрода;

R_{zd} - предельно допустимая величина сопротивления заземляющего устройства;

η_c - коэффициент сезонности (таблица 5.1);

η_T - коэффициент экранирования (взаимного влияния) вертикальных электродов. (рис.5.3).

Таблица 5.1-Приближенные значения коэффициентов сезонности - η_c

Тип заземлителей	Глубина заложения заземлителя h, м	Коэффициент (по количеству влаги)			
			Сыро	Влажно	Сухо
Труба, угловая сталь, металлические стержни длиной 2- 3 м (вертикальные заземлители)	Вертикальный конец заземлителя расположен на глубине 0,8 м от поверхности грунта	c1	2,0	1,5	1,4
Полоса, круглая проволока и т.п. протяженные (горизонтальные) заземлители	h = 0,5 м	2	6,5	5,0	4,5
	h = 0,8 м		3,0	2,0	1,6

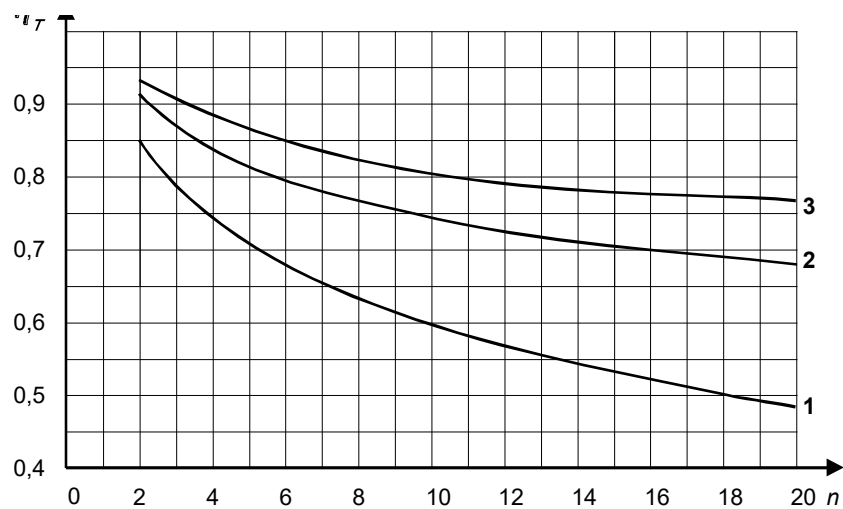


Рис.1 Значение коэффициента использования вертикальных заземлителей η_T , расположенных в ряд.
1 - $A/L = 1$; 2 - $A/L = 2$; 3 - $A/L = 3$;

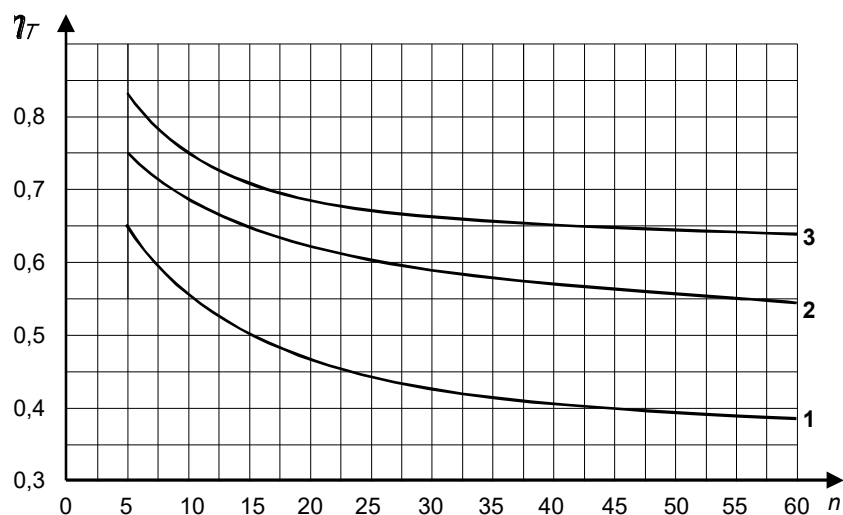


Рис.2 Значение коэффициента использования вертикальных заземлителей η_T , расположенных по контуру.
1 - $A/L = 1$; 2 - $A/L = 2$; 3 - $A/L = 3$;

Рисунок 5.3. Значения коэффициентов использования вертикальных заземлителей

Метод последовательного приближения состоит в том, что на первом шаге принимают $\eta_T = 1$ и определяют n_1 . По n_1 находят соответствующее ему значение η_{T2} (рис. 5.3) и по формуле (5.2) определяют новое значение n_2 . По n_2 находят соответствующее значение η_{T3} и по формуле (5.2) новое значение n_3 . Так продолжают расчет до тех пор, пока разница между двумя

последующими числами электродов будет меньше единицы (т.е. $n_i - n_{i-1} \leq 1$).
Полученное число электродов.

5.3 Расчет сопротивления заземляющего устройства

Сопротивление не объединенных в один контур вертикальных заземлителей рассчитывают по формуле:

$$R_{\Sigma} = \frac{R_o \cdot \eta_c}{n \cdot \eta_{\text{ТФ}}}, \quad (5.3)$$

где R_o - сопротивление одиночного вертикального заземлителя, Ом;

n - число вертикальных электродов (заземлителей);

η_c - коэффициент сезонности;

$\eta_{\text{ТФ}}$ - фактический коэффициент использования вертикальных заземлителей.

Длина соединительной полосы определяется из выражения:

$l_{\Pi} = 1,05 \cdot A(n-1)$ - при расположении вертикальных электродов по контуру или

$l_{\Pi} = 1,05 \cdot A \cdot n$ - при расположении вертикальных электродов в ряд,

где A - расстояние между электродами, м;

n - число вертикальных электродов.

Сопротивление растеканию тока полосы, считая ее единичным горизонтальным заземлителем, рассчитывают по формуле:

$$R_n = \frac{\rho}{2\pi \cdot l_{\Pi} \cdot \eta_{\Pi}} \ln \frac{2l_{\Pi}^2}{b \cdot h} \quad (5.4)$$

где ρ - удельное сопротивление грунта, Ом·м (табл.5.2 Приложения);

l_{Π} - длина полосы, м;

Таблица 5.2 Удельные сопротивления грунтов

Вид грунта	ρ , Ом·м
Песок	$7 \cdot 10^2$
Супесок	$3 \cdot 10^2$
Суглинок	$1 \cdot 10^2$
Глина	$0,4 \cdot 10^2$
Чернозем	$2 \cdot 10^2$

η_{II} - коэффициент использования полосы (рисунок 5.4);

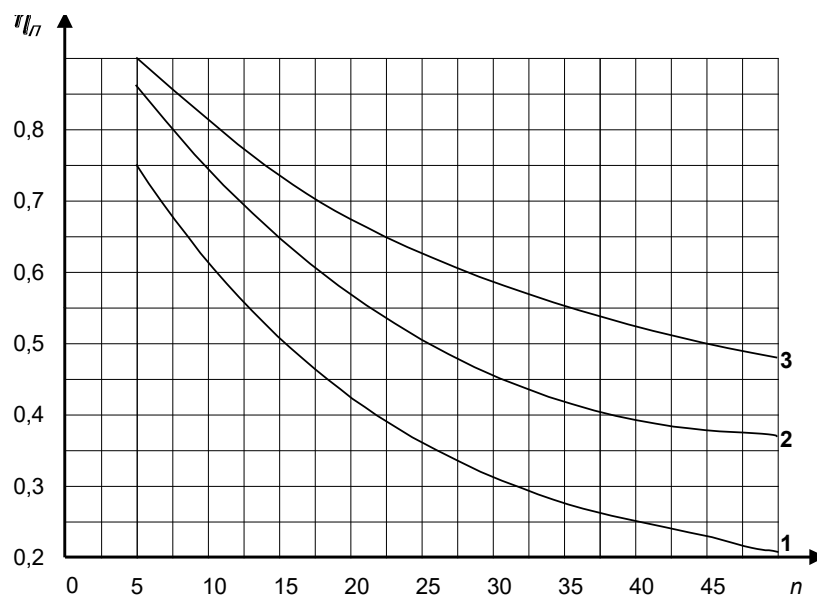


Рис.3 Значения коэффициента использования соединительной полосы $\eta_{л}$ при расположении вертикальных заземлителей в ряд.
1 - $A/L = 1$; 2 - $A/L = 2$; 3 - $A/L = 3$;

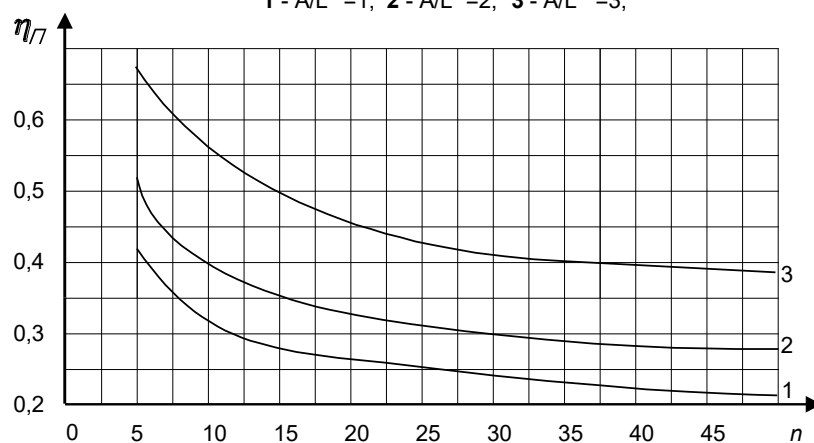


Рис.4 Значения коэффициента использования соединительной полосы $\eta_{л}$ при расположении вертикальных заземлителей по контуру.
1 - $A/L = 1$; 2 - $A/L = 2$; 3 - $A/L = 3$;

Рисунок 5.4 Значения коэффициента использования соединительной полосы.

b - ширина полосы, м;

h - глубина заложения полосы.

Сопротивление заземляющего устройства определяют по формуле:

$$R_{з.у.} = \frac{R_{\Sigma} \cdot R_{л}}{R_{\Sigma} + R_{л}} + R_{з.п.}, \quad (5.5)$$

где R_{Σ} - суммарное сопротивление всех вертикальных заземлителей;

$R_{л}$ - сопротивление соединительной полосы;

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.17.001 ПЗ

Лист
93

$R_{з.л.}$ - сопротивление заземляющего проводника.

Полученное сопротивление заземляющего устройства не должно превышать допустимого значения $R_{з.у.} < R_{з.д.}$.

5.4 Обеспечение противопожарной безопасности

5.4.1 План эвакуации

Вынужденная эвакуация при пожаре протекает в условиях нарастающего

действия опасных факторов пожара. Поэтому безопасность людей находится в прямой зависимости от времени пребывания их в здании при пожаре.

Наиболее распространенным путем эвакуации являются проходы, коридоры, фойе и лестницы.

Выходы считаются эвакуационными, если они ведут из помещений: первого этажа наружу непосредственно или через тамбур, вестибюль, лестничную клетку; любого этажа, кроме первого, в коридор, ведущий на лестничную клетку.

Требования к эвакуационным путям:

1. В полу на эвакуационных путях не допускаются перепады высот более 45 см. В местах перепада высот следует предусматривать лестницы с числом ступеней не менее трех.
2. Ширина маршрута лестницы не менее ширины двери эвакуационного выхода на лестничную клетку.

Схема эвакуации из помещения представлена на рисунке 5.6:



Рисунок 5.6. Схема эвакуации из помещения.

5.4.2 Пожарные краны.

В зданиях ВЦ, пожарные краны устанавливают в коридорах, на площадках лестничных клеток, то есть в доступных и заметных местах. Пожарные краны располагают в нишах на высоте 1,35 м., где также находятся пожарный ствол с напорным рукавом из тканевого материала длиной 10-20 см. Напор воды должен обеспечивать радиус действия компактной части струи воды, достаточный для достижения наиболее удаленной и возвышенной части здания, но не менее 6м.

5.4.3 Огнетушители.

5.4.3.1 ОГNETУШИТЕЛЬ ВОДНЫЙ

НАЗНАЧЕНИЕ

Предназначен для использования как первичного средства пожаротушения пожаров класса А (твердых горючих материалов, горение которых сопровождается тлением, В (горючие жидкости и твердые плавящиеся материалы, С (горючие газы).

Огнетушитель не применяется для тушения пожаров электроустановок под напряжением, щелочных и щелочно-земельных металлов. Огнетушитель должен эксплуатироваться при температуре от -50 до +50°C Срок службы огнетушителя 10 л.



Рисунок 5.7 огнетушитель водный мелкодисперсный ОВМ-10

Таблица 5.3 - Характеристики водного мелко дисперсионного огнетушителя ОВМ 10

Показатели	
Модель огнетушителя переносного водного с мелкодисперсной распылённой струёй ОВМ-10	
Код продукции	48 5430 5135*

Нормативный документ

ТУ 4854-001-05789995-94

Код предприятия-
изготовителя

05789995

ТАКТИКО-ТЕХНИЧЕСКИЕ И ЭКСПЛУАТАЦИОННЫЕ ХАРАКТЕРИСТИКИ

Способ огнетушащего вещества	вытеснения баллоном сжатого газа	с	Рабочее давление, МПа	в корпусе	1,2 (не более)
				в пусковом баллоне	15 (не более)
Вместимость корпуса, л	10+ 0,3		Продолжительность подачи огнетушащего вещества, с		
				40 (не менее)	
Огнетушащее вещество	Водный раствор антифриза		Длина шланга ,мм		
				500	
Масса огнетушащего вещества, кг	13,5 +0,3		Длина струи огнетушащего вещества, м		
				4,0	
Масса антифриза, кг	6,0+ 0,15		Диапазон температур, ° С		
				- 25...+50	

Огнетушащая способность		3А, 89В	Полная масса, кг	20,0 (не более)
Пусковой баллон	вместимость, см ³	400	Габаритные размеры, мм	225x175x640
	масса СО ₂ ,г	240 +24	Средний срок службы, лет	10

ПРИВЕДЕНИЕ ОГNETУШИТЕЛЯ В ДЕЙСТВИЕ

Поднести огнетушитель к очагу пожара с учетом безопасного от теплового воздействия расстояния

Выдернуть чеку.

Взять в руки шланг с насадкой

Нажать на желтую кнопку до упора и отпустить.

Направить струю мелко распыленного огнетушащего средства на очаг горения

При тушении загорания нефтепродуктов следует отрывать мелко распыленным факелом воды пламя от очага горения

Тушение производить только с наветренной стороны

ПРАВИЛА ЭКСПЛУАТАЦИИ

Огнетушитель может храниться при температуре воздуха от -50 до +50 °С

Работает при температурах от -25 до +50 °С.

Огнетушитель подлежит ежегодной периодической проверке на специализированной станции перезарядки.

5.4.3.2 ОГNETУШИТЕЛЬ ВОЗДУШНО-ПЕННЫЙ

НАЗНАЧЕНИЕ

Огнетушитель ОВП-10.01 предназначен для тушения очагов пожара класса А (горение твердых материалов органического происхождения, горение которых сопровождается тлением - дерево, бумага, ветошь и т.д.) и В (горение жидкостей или твердых тел, превращающихся в жидкости - нефтепродукты, масла, краски и т.п.).

Огнетушитель не может быть применен для тушения веществ, горение которых происходит без доступа воздуха (хлопок, пироксилин и т.п.), горящих металлов (щелочных - натрий и т.п. и легких - магний и т.п.). Запрещается использовать для тушения электроустановок, находящихся под напряжением.

Огнетушитель применяют при температуре окружающего воздуха от +3 до +50 С.



Рисунок 5.8 огнетушитель воздушно-пенный ОВП-10

Таблица 5.4 - Характеристики огнетушителя воздушно-пенного ОВП-10.

Показатели	Модель огнетушителя переносного воздушно-пенного	
	ОВП (Н,С)-10 (б)	ВП-10
Код продукции	48 5432 1120*	48 5432 1119
Нормативный документ	ТУ 10-4854-017-02767964-96	ТУ 4854-009-03936239-97
Код предприятия-изготовителя	02767964	03936239
ТАКТИКО-ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ	И	ЭКСПЛУАТАЦИОННЫЕ
Способ вытеснения огнетушащего вещества	с баллоном сжатого газа	
Вместимость корпуса огнетушителя, л	10+0,5	10+0,5
Объём заряда огнетушащего вещества, л	8,5+0,4	8,1 (6,1-вода, 2,0-заряд)
Огнетушащая способность	2А, 55В	2А, 55В
Рабочее давление, МПа	нет данных	1,2+0,15
Кратность пены	5 и 50	50

Продолжительность подачи огнетушащего вещества, с	40	40
Длина струи огнетушащего вещества, м	3,0	3,0 (не менее)
Диапазон рабочих температур, °С	+5...+50	
Полная масса, кг	нет данных	12,5 (не более)
Габаритные размеры, мм	нет данных	690x150
Средний срок службы, лет	10	10

ПРИВЕДЕНИЕ ОГNETУШИТЕЛЯ В ДЕЙСТВИЕ

Поднести огнетушитель к очагу пожара с учетом безопасного от теплового воздействия расстояния.

Выдернуть чеку.

Взять в руки шланг с насадком.

Нажать на желтую кнопку до упора и отпустить.

Направить струю пены на очаг горения.

При тушении загорании нефтепродуктов следует покрывать пеной всю поверхность, начиная с ближнего края.

Тушение производить только с наветренной стороны.

ПРАВИЛА ЭКСПЛУАТАЦИИ

Огнетушитель соответствует климатическому исполнению УХЛ, категории размещения 4 при рабочем значении температуры воздуха при

					ПГУ 1.090501.17.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		101

эксплуатации от +3 до +50 С, эксплуатации в атмосфере типов 1 или 2, группы условий эксплуатации 1 по ГОСТ 15150-69.

Огнетушитель подлежит ежегодной периодической проверке на специализированной станции перезарядки.

5.4.3.3 Примечание

В свою очередь существуют углекислые и порошковые огнетушители, которые могут применяться для тушения пожаров в электроустановках под напряжением.

5.5 Обеспечение санитарно-бытовых условий.

Для эксплуатации компьютеров следует предусматривать следующие помещения:

1. машинный зал (основное помещение)
2. помещение для хранения документации
3. помещение для хранения инструмента
4. помещение для курения
5. туалет

К помещениям машинного зала предъявляются особые требования. Высота зала над технологическим полом до подвесного потолка должна быть 3-3.5 м. Расстояние между подвесными и основным потолками при этом должно быть 0.5-0.8 м. Высоту подпольного пространства принимают равной 0.2-0.6 м (при прокладке в нем воздуховодов – не менее 0.3 м, а при наличии только кабелей – не менее 0.15-0.2 м.). Конструкции, расположенные в этом пространстве, не должны иметь острых ребер и заусенцев для исключения повреждений кабелей. Габариты дверей машинного зала ЭВМ принимаются не менее 1,8×1,1 м из расчета возможности транспортирования оборудования. Машинный зал должен иметь как естественное, так и искусственное освещение.

Помещение для хранения документации– высота 3,3 м.; площадь не менее 16 квадратных метров. На ней необходимо разместить нужное

количество тележек и столиков, а также стеллажи и шкафы. Последние изготавливают из дерева, пропитанного огнезащитным составом. Пол, потолок, стены – из негорючих материалов. Двери хранилища должны быть металлическими или деревянными, обитыми листовым железом по войлоку, смоченному раствором глины, или асбесту.

Те же требования предъявляются к помещению для хранения инструмента.

Помещение для курения и туалет могут быть совмещенными помещениями. При этом должен быть соответствующий объем помещения и вентиляция.

5.6 Освещение

К системам производственного освещения предъявляются следующие основные требования:

1. Соответствие уровня освещенности рабочих мест характеру выполняемой зрительной работы.
2. Достаточно равномерное распределение яркости на рабочих поверхностях и в окружающем пространстве.
3. Отсутствие резких теней, прямой и отраженной блескости.
4. Постоянство освещенности во времени.
5. Оптимальная направленность излучаемого осветительными приборами светового потока.
6. Долговечность, экономичность, электро- и пожаробезопасность, эстетичность, удобство и простота эксплуатации.

В зависимости от природы источника световой энергии различают естественное, искусственное и совмещенное.

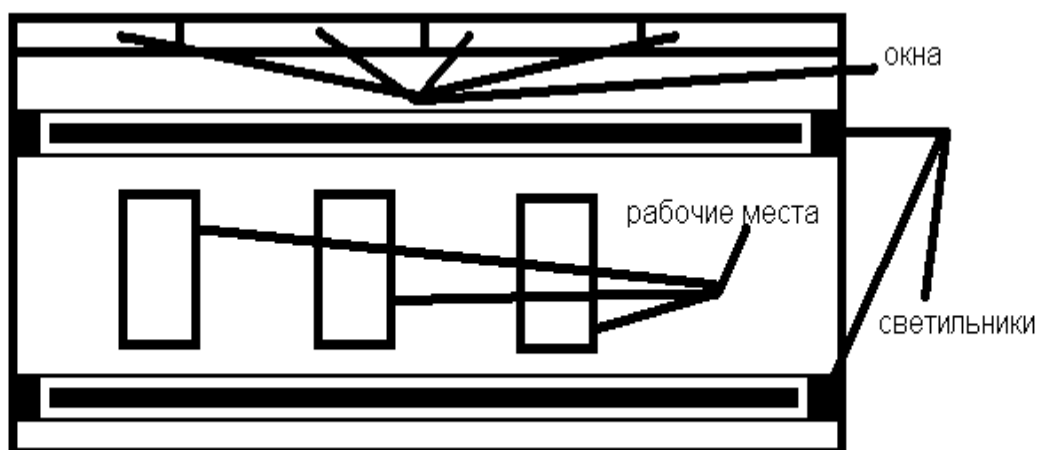


Рисунок 5.9. Расположение светильников общего освещения относительно рабочих мест операторов

5.6.1 Расчет.

Машинный зал ВЦ с размерами $A = 20\text{ м}$, $B = 9\text{ м}$ и высотой $H = 3\text{ м}$ предусмотрены потолочные светильники типа УСП35 с двумя люминесцентными лампами типа ЛБ-40. Коэффициент отражения светового потока от потолка, стен и пола соответственно $\delta_n = 70\%$, $\delta_c = 50\%$, $\delta_{\text{пола}} = 10\%$. Затенения рабочих мест нет. Определить необходимое число светильников при общем равномерном освещении.

Для машинных залов уровень рабочей поверхности над полом составляет $0,8\text{ м}$. Тогда $h = H - 0,8 = 2,2\text{ м}$. У светильников УСП35 наивыгоднейшее отношение $\xi = 1,4$. Отсюда расстояние между рядами светильников $L = \xi \cdot h = 1,4 \cdot 2,2 \approx 3\text{ м}$. Располагаем светильники вдоль длинной стороны помещения. Расстояние между стенами и крайними рядами светильников принимаем равным $l \approx (0,3 \dots 0,5)L$. При ширине машинного зала $B = 9\text{ м}$, число рядов светильников принимаем равным $n \cong \frac{B}{L} = 3$. По таблице «Нормы освещенности для помещений ВЦ», для машинного зала

установлена норма освещенности $E_n = 400 \text{лк}$. С учетом заданных δn , δc , $\delta n_{\text{ола}}$ при $l = \frac{180}{2,2 \cdot (20+9)} = 2,82$ из справочника по данному значению получаем $\eta = 0,45$.

Номинальный световой поток лампы ЛБ-40 $\Phi_l = 3120 \text{лм}$, тогда световой поток, излучаемый светильником, составит $\Phi_{\text{св}} = 2 \cdot \Phi_l = 2 \cdot 3120 = 6240 \text{лм}$. Число светильников в ряду $N = \frac{400 \cdot 1,5 \cdot 180 \cdot 1,15}{3 \cdot 6240 \cdot 0,45} \approx 15 \text{шт}$. При длине одного светильника тина УСП35 с лампами ЛБ-40 $l_{\text{св}} = 1,27 \text{м}$ их общая длина составит $N \cdot l_{\text{св}} = 1,27 \cdot 15 = 19,05 \text{м}$, следовательно светильники размещены в непрерывный сплошной ряд.

ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломной работы было разработано web приложение, обеспечивающее обработку видео архива в рамках корпоративной экосистемы. Данный проект это приложение на основе бессерверного подхода, созданное на основе краудсорсинговых данных пользователей. Приложение интегрируется со сторонними службами для обеспечения безопасности работы с данными. Приложение построено с использованием технологий, которые являются производительными, гибкими и масштабируемыми.

					ПГУ 1.090501.17.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		106

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Сухов, Кирилл Node.js. Путеводитель по технологии / Кирилл Сухов. - Москва: ДМК Пресс, 2015. - 416 с.
2. Хэррон, Дэвид Node.js Разработка серверных веб-приложений на JavaScript / Дэвид Хэррон. - Москва: ДМК Пресс, 2016. - 195 с.
3. Хэррон, Дэвид Node.js Разработка серверных веб-приложений на JavaScript / Дэвид Хэррон. - СПб: Питер, 2016. - 144 с.
4. Мартишин С. А., Храпченко М. В. Базы данных. Практическое применение СУБД SQL- и NoSQL-типа для применения проектирования информационных систем. М :Москва, Бином, 2018. - 301 с.
5. Фаулер М., Прамодкумар Дж. Садаладж NoSQL: новая методология разработки нереляционных баз данных, :Москва, Бином, 2018. - 261 с.

Приложение А

(обязательное)

СИСТЕМА УПРАВЛЕНИЯ МЕДИАДАННЫМИ

КОД ПРИЛОЖЕНИЯ

					ПГУ 1.090501.17.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		108

Файл Index.html

```
<!doctype html>
<!--[if lt IE 7]>          <html class="no-js lt-ie9 lt-ie8 lt-ie7"
lang=""> <![endif]-->
<!--[if IE 7]>             <html class="no-js lt-ie9 lt-ie8"
lang=""> <![endif]-->
<!--[if IE 8]>             <html class="no-js lt-ie9" lang="">
<![endif]-->
<!--[if gt IE 8]><!-->
<html class="no-js" lang="">
<!--<![endif]-->

<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
  <title></title>
  <meta name="description" content="">
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  <link rel="apple-touch-icon" href="apple-touch-icon.png">

  <link rel="stylesheet" href="css/bootstrap.min.css">
  <style>
    body {
      padding-top: 50px;
      padding-bottom: 20px;
    }
  </style>
  <link rel="stylesheet" href="css/bootstrap-theme.min.css">
  <link rel="stylesheet" href="css/main.css">

  <script src="js/vendor/modernizr-2.8.3-respond-
1.4.2.min.js"></script>
```

```

</head>

<body>
  <!--[if lt IE 8]>
    <p class="browserupgrade">Error!!!!.</p>
  <![endif]-->

  <nav class="navbar navbar-inverse navbar-fixed-top"
role="navigation">
    <div class="container">
      <div class="navbar-header">
        <button type="button" class="navbar-toggle collapsed"
data-toggle="collapse" data-target="#navbar" aria-
expanded="false" aria-controls="navbar">
          <span class="sr-only">Toggle navigation</span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        <a class="navbar-brand" href="#">Project name</a>
      </div>
      <div id="navbar" class="navbar-collapse collapse">
        <div class="navbar-form navbar-right">
          <button id="user-profile" class="btn btn-default">
            <img id="profilepicture" />&nbsp;<span
id="profilename"></span>
          </button>

          <button id="auth0-login" class="btn btn-success">Sign
in</button>
          <button id="auth0-logout" class="btn btn-success">Sign
Out</button>
        </div>
      </div>
    <!--/.navbar-collapse -->

```

```

        </div>
</nav>

<!-->
<div class="jumbotron">
    <div class="container">
        <h1>Hello, world!</h1>
        <p><a      class="btn      btn-primary      btn-lg"      href="#"
role="button">Learn more &raquo;</a></p>
    </div>
</div>

<div class="container">
    <!-- Example row of columns -->
    <div class="row">
        <div class="col-md-4">
            <h2>Heading</h2>
            <p><a      class="btn      btn-default"      href="#"
role="button">Видео &raquo;</a></p>
        </div>
        <div class="col-md-4">
            <h2>Heading</h2>
            <p><a      class="btn      btn-default"      href="#"
role="button">Пользователи&raquo;</a></p>
        </div>
        <div class="col-md-4">
            <h2>Heading</h2>
            <p><a      class="btn      btn-default"      href="#"
role="button">View details &raquo;</a></p>
        </div>
    </div>

    <hr>

```

```

    <footer>
      <p>&copy; Company 2016</p>
    </footer>
  </div>
<!-- /container -->

<div id="user-profile-modal" class="modal fade" tabindex="-1"
role="dialog">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-
dismiss="modal" aria-label="Close"><span data-
hidden="true">&times;</span></button>
        <h4 class="modal-title">Поиск</h4>
      </div>
      <div class="modal-body">
        <b>Raw JSON from Auth0:</b>
        <pre id="user-profile-raw-json"></pre>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-primary" data-
dismiss="modal">Close</button>
      </div>
    </div>
  <!-- /.modal-content -->
</div>
<!-- /.modal-dialog -->
</div>
<!-- /.modal -->

```

```

<script
src="//ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js
"></script>

<script>
    window.jQuery          ||          document.write('<script
src="js/vendor/jquery-1.11.2.min.js"></script>')
</script>

<script src="js/vendor/bootstrap.min.js"></script>
<script src="https://cdn.auth0.com/js/lock-9.min.js"></script>

<script src="js/user-controller.js"></script>
<script src="js/config.js"></script>
<script src="js/main.js"></script>

<!-- Google Analytics: change UA-XXXXX-X to be your site's ID.
-->
<script>
    (function(b, o, i, l, e, r) {
        b.GoogleAnalyticsObject = l;
        b[l] || (b[l] =
            function() {
                (b[l].q = b[l].q || []).push(arguments)
            });
        b[l].l = +new Date;
        e = o.createElement(i);
        r = o.getElementsByTagName(i)[0];
        e.src = '//www.google-analytics.com/analytics.js';
        r.parentNode.insertBefore(e, r)
    }(window, document, 'script', 'ga'));
    ga('create', 'UA-XXXXX-X', 'auto');
    ga('send', 'pageview');
</script>
</body>

```

</html>

Файл CustomerAuthorizer.js

```
'use strict';

var jwt = require('jsonwebtoken');

var generatePolicy = function(principalId, effect, resource) {

    var authResponse = {};

    authResponse.principalId = principalId;

    if (effect && resource) {

        var policyDocument = {};

        policyDocument.Version = '2012-10-17'; // default
version

        policyDocument.Statement = [];

        var statementOne = {};

        statementOne.Action = 'execute-api:Invoke'; // default
action

        statementOne.Effect = effect;

        statementOne.Resource = resource;

        policyDocument.Statement[0] = statementOne;

        authResponse.policyDocument = policyDocument;

    }

}
```

					ПГУ 1.090501.17.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		114

```

        return authResponse;
    }

    exports.handler = function(event, context, callback){

        if (!event.authorizationToken) {

            callback('Could not find authToken');

            return;

        }

        var token = event.authorizationToken.split(' ')[1];

        var secretBuffer = new Buffer(process.env.AUTH0_SECRET);

        jwt.verify(token, secretBuffer, function(err, decoded){

            if(err){

                console.log('Failed jwt verification: ', err, 'auth: ',
event.authorizationToken);

                callback('Authorization Failed');

            } else {

                callback(null,          generatePolicy('user',          'allow',
event.methodArn));

            }

        })

    };

```

Файл GetVideoList.js

```
'use strict';

var AWS = require('aws-sdk');

var async = require('async');

var s3 = new AWS.S3();

function createErrorResponse(code, message, encoding) {

    var response = {

        'statusCode': code,

        'headers' : {'Access-Control-Allow-Origin' : '*'},

        'body' : JSON.stringify({'code': code, 'message' : message,
        'encoding' : encoding})

    }

    return response;
}

function createSuccessResponse(result) {

    var response = {

        'statusCode': 200,
```

					ПГУ 1.090501.17.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		116

```

        'headers' : {'Access-Control-Allow-Origin' : '*'},

        'body' : JSON.stringify(result)

    }

    return response;

}

function createBucketParams(next) {

    var params = {

        Bucket: process.env.BUCKET

    };

    next(null, params);

}

function getVideosFromBucket(params, next) {

    s3.listObjects(params, function(err, data){

        if (err) {

            next(err);

        } else {

            next(null, data);

        }

    });

```

```

}

function createList(encoding, data, next) {

    var files = [];

    for (var i = 0; i < data.Contents.length; i++) {

        var file = data.Contents[i];

        if (encoding) {

            var type = file.Key.substr(file.Key.lastIndexOf('-') + 1);

            if (type !== encoding + '.mp4') {

                continue;

            }

        } else {

            if (file.Key.slice(-4) !== '.mp4') {

                continue;

            }

        }

        files.push({

            'filename': file.Key,

            'eTag': file.ETag.replace(/"/g, ""),

            'size': file.Size

        });
    }
}

```

```

    }

    var result = {

        domain: process.env.BASE_URL,

        bucket: process.env.BUCKET,

        files: files

    }

    next(null, result)

}

exports.handler = function(event, context, callback){

    var encoding = null;

    if (event.queryStringParameters.encoding) {

        encoding = decodeURIComponent(event.queryStringParameters.encoding);

    }

    async.waterfall([createBucketParams, getVideosFromBucket,
    async.apply(createList, encoding)],

        function (err, result) {

            if (err) {

```

					ПГУ 1.090501.17.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		119

```

        callback(null, createErrorResponse(500, err, encoding));
    } else {
        if (result.files.length > 0) {
            callback(null, createSuccessResponse(result));
        } else {
            callback(null, createErrorResponse(404, 'No files were
found', encoding));
        }
    }
});
};

```

Файл Email.js

```

'use strict';

var AWS = require('aws-sdk');

var async = require('async');

var SES = new AWS.SES();

function createMessage(toList, fromEmail, subject, message,
next) {
    var params = ({

```

```

    Source: fromEmail,

    Destination: { ToAddresses: toList },

    Message: {

        Subject: {

            Data: subject

        },

        Body: {

            Text: {

                Data: message

            }

        }

    }

});

next(null, params);

}

function dispatch(params, next) {

    SES.sendEmail(params, function(err, data){

        if (err) {

            next(err);

        } else {

            next(null, data);

        }

    });

}

```

```

    }

    })

}

function send(toList, fromEmail, subject, message) {

    async.waterfall([createMessage.bind(this, toList, fromEmail,
subject, message), dispatch],

    function (err, result) {

        if (err) {

            console.log('Error sending email', err);

        } else {

            console.log('Email Sent Successfully', result);

        }

    });

};

module.exports = {

    send: send

};

```

Файл TestConnection.js

```
var chai = require('chai');
```

					ПГУ 1.090501.17.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		122

```

var sinon = require('sinon');

var rewire = require('rewire');

var expect = chai.expect;

var assert = chai.assert;


var sampleData = {

  Contents: [

    {

      Key: 'file1.mp4',

      bucket: 'my-bucket'

    },

    {

      Key: 'file2.mp4',

      bucket: 'my-bucket'

    }

  ]

}


describe('LambdaFunction', function(){

  var listObjectsStub, callbackSpy, module;


  describe('#execute', function() {

    before(function(done) {

```

					ПГУ 1.090501.17.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		123

```

listObjectsStub = sinon.stub().yields(null, sampleData);

callbackSpy = sinon.spy();

var callback = function(error, result) {

    callbackSpy.apply(null, arguments);

    done();

}

module = getModule(listObjectsStub);

module.handler(null, null, callback);

}))

it('should run our function once', function(){

    expect(callbackSpy).has.been.calledOnce;

})

it('should have correct results', function(){

    var result = {

        "baseUrl": "https://s3.amazonaws.com",

        "bucket": "serverless-video-transcoded",

        "urls": [

            {

                "Key": sampleData.Contents[0].Key,

```

```

        "bucket": "my-bucket"

    },

    {

        "Key": sampleData.Contents[1].Key,

        "bucket": "my-bucket"

    }

]

}

    assert.deepEqual(callbackSpy.args, [[null, result]]);

    })

    })

    })

function getModule(listObjects) {

    var rewired = rewire('../index.js');

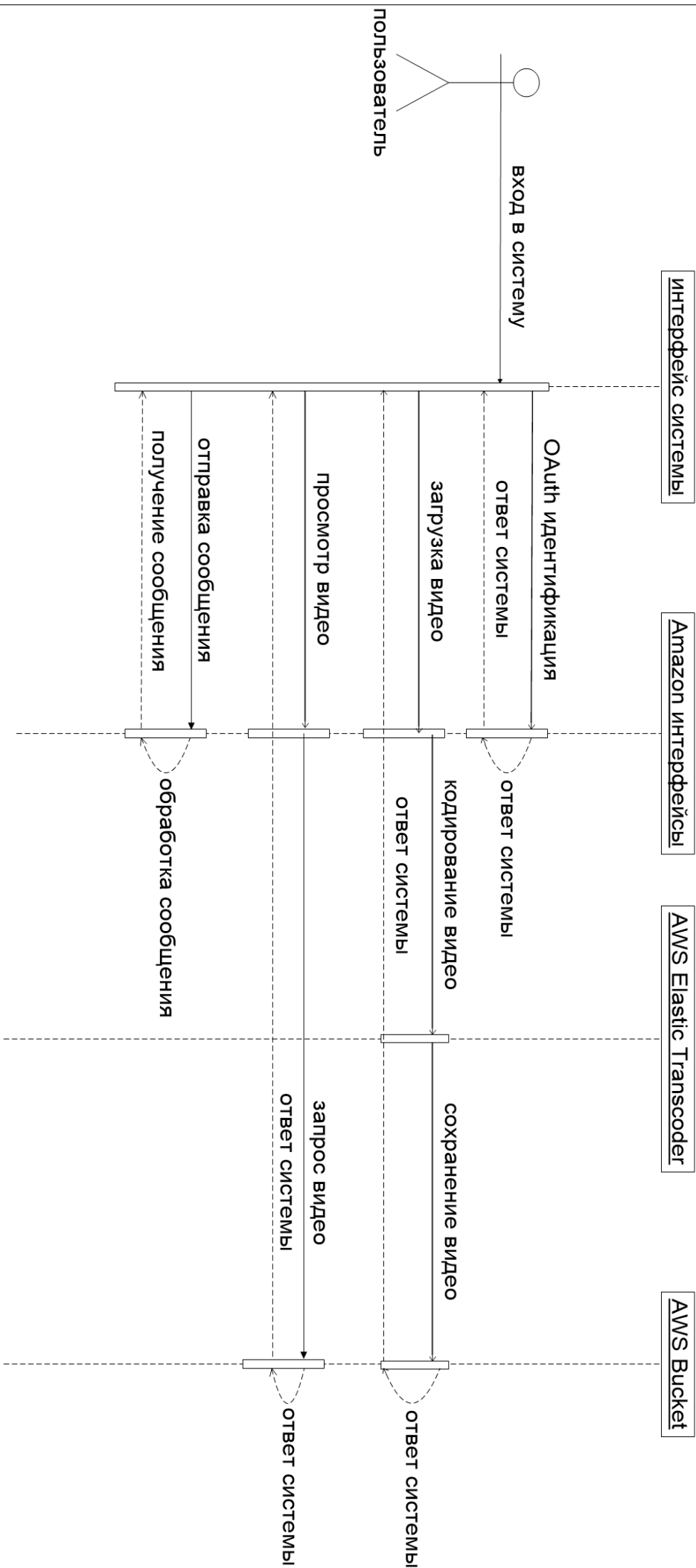
    rewired.__set__({

        's3': { listObjects: listObjects }

    });

    return rewired;

```



					ПГУ 1.090501.17.002 Д01				
Изм	Лист	№ докум.	Подпись	Дата	Система управления медиаданными Диагностика последовательности				
Разраб.		Киселев А.К							
Провер.		Федюкин Р.Н							
Реценз		Васильков А.В.			ФВТ, Каф. ВТ Гр. 15ВО2				
Н. контр.		Кучин А.В.							
Утверд.		Митрохин М.А.							

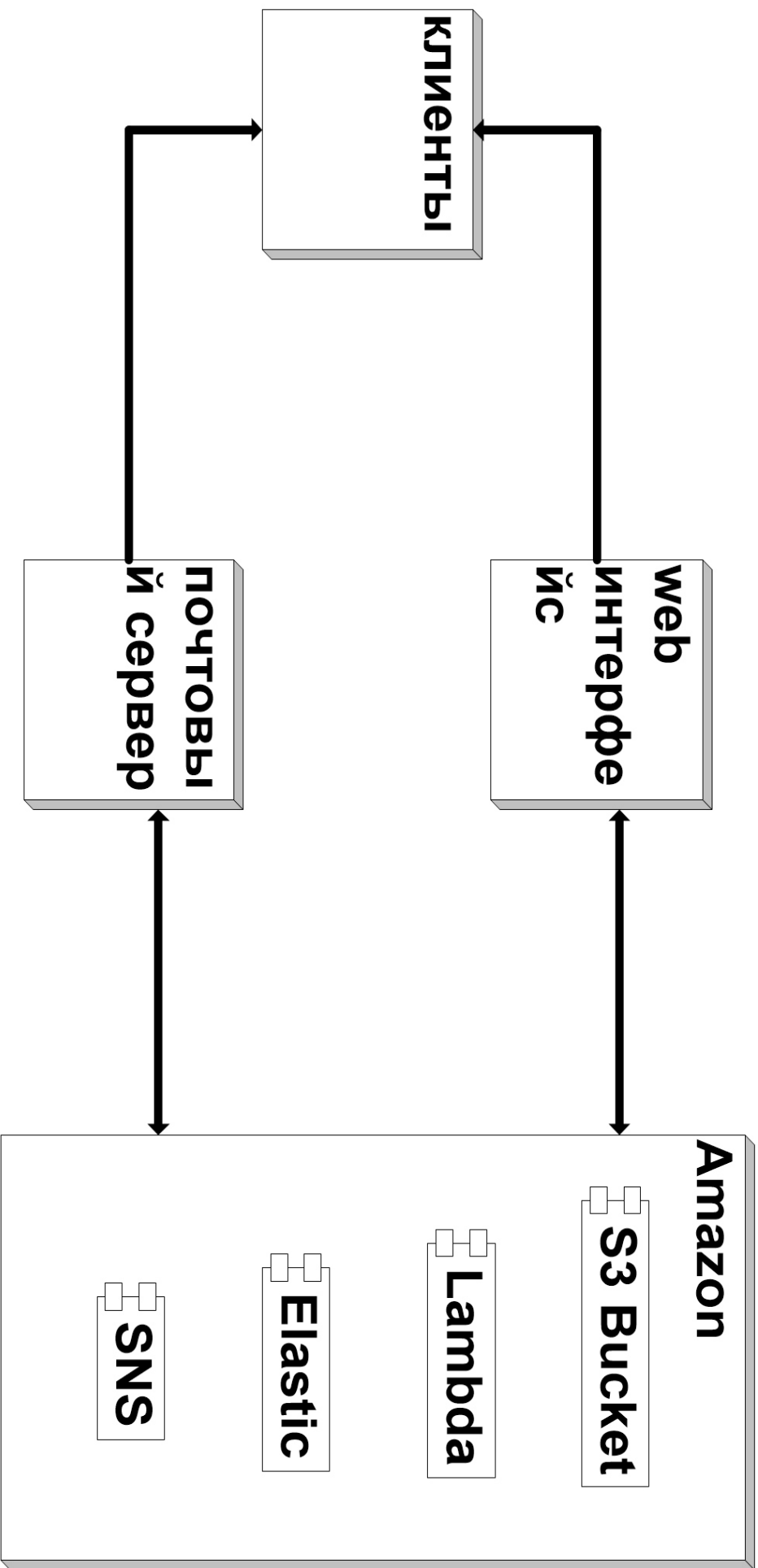
Приложение С

(обязательное)

СИСТЕМА УПРАВЛЕНИЯ МЕДИАДАННЫМИ

СХЕМА СТРУКТУРНАЯ

					ПГУ 1.090501.17.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		127



							ПГУ 1.090501.17.004 Д02			
							Система управления медиаданными Схема структурированная			
Изм	Лист	№ докум.	Подпись	Дата			Лист		Масса	Масштаб
Разраб		Киселев А.К								
Провер.		Федюнин Р.Н.								
Реценз		Васильков А.В.								
Н. контр.		Кучин А.В.								
Утверд.		Митрохин М.А.								
							ФВТ, Каф. ВТ Гр. 15В02			

Приложение D

(обязательное)

СИСТЕМА УПРАВЛЕНИЯ МЕДИАДАННЫМИ

ДИАГРАММА ИСПОЛЬЗОВАНИЯ

					ПГУ 1.090501.17.001 ПЗ	Лист
						129
Изм.	Лист	№ докум.	Подпись	Дата		

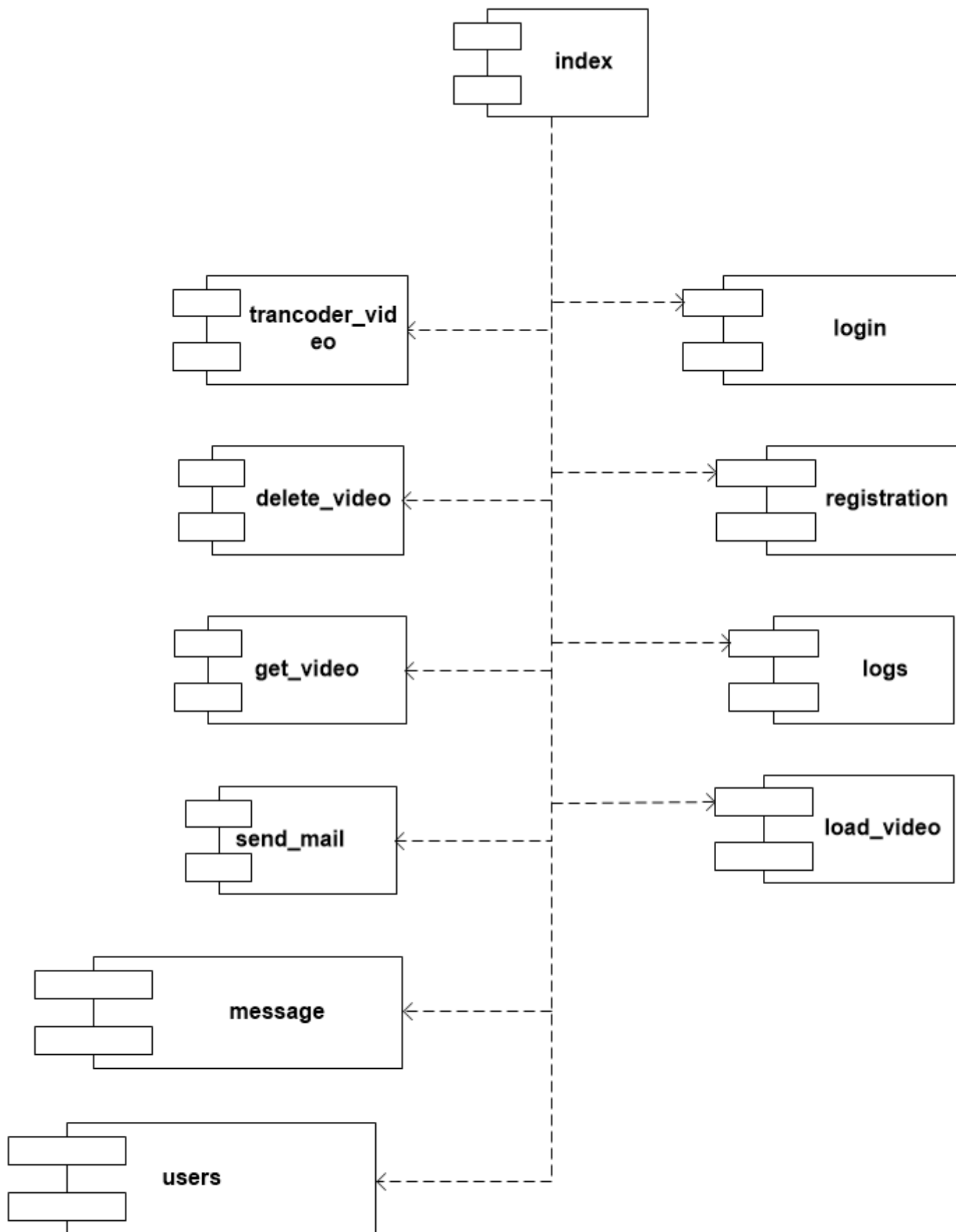
Приложение Е

(обязательное)

СИСТЕМА УПРАВЛЕНИЯ МЕДИАДАННЫМИ

ДИАГРАММА КОМПОНЕНТОВ

					ПГУ 1.090501.17.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		131



Изм.	Лист	№ Докум.	Подпись	Дата	
Разраб.		Киселев А.К.			
Провер.		Федюнин Р.Н.			
Реценз		Васильков А.В.			
Н. контр.		Кучин А.В.			
УТВ.		Митрохин М.А.			

ПГУ1.090501.17.006 Д04

Система управления
медиаданными
Диаграмма компонентов

Лист.	Масса	Масш.
Лист 1	Листов 1	
ФВТ, каф. ВТ группа 15В02		

Приложение F
(обязательное)
ПРЕЗЕНТАЦИЯ

					ПГУ 1.090501.17.001 ПЗ	Лист
						133
Изм.	Лист	№ докум.	Подпись	Дата		

ТЕМА РАБОТЫ:

Система управления медиаданными

Автор работы: Студент группы 15В02 Киселев А. К.
Научный руководитель: к.т.н., доцент Федюнин Р. Н.

ЦЕЛИ ЗАДАЧИ И ОБЪЕКТЫ РАБОТЫ

- ▶ Цель работы - разработать масштабируемую, отказоустойчивую систему загрузки, воспроизведения и руководства видео контентом на базе Amazon.
- ▶ Объектом разработки является программное приложение, обеспечивающее пользовательский доступ к каталогу видео и аудио контенту, администраторам доступ к настройкам системы, а программистам к инструментам видеопотока на базе облачного сервиса.
- ▶ В результате проведенной работы должен быть разработан проект, удовлетворяющий поставленной задаче, а именно, создан программный комплекс – мобильное приложение клиент, web-сервис бизнес логики и облачное хранилище, и разработаны соответствующие алгоритмы, реализующие возможность хранить, обрабатывать и доставлять до потребителя информацию в виде видео потока.
- ▶ Данное приложение рекомендуется использовать для корпоративного вещания. Система может быть внедрена в сегменте пресс службы и рекламы корпоративного сегмента.

Изм.	Лист	№ докум.	Подпись	Дата

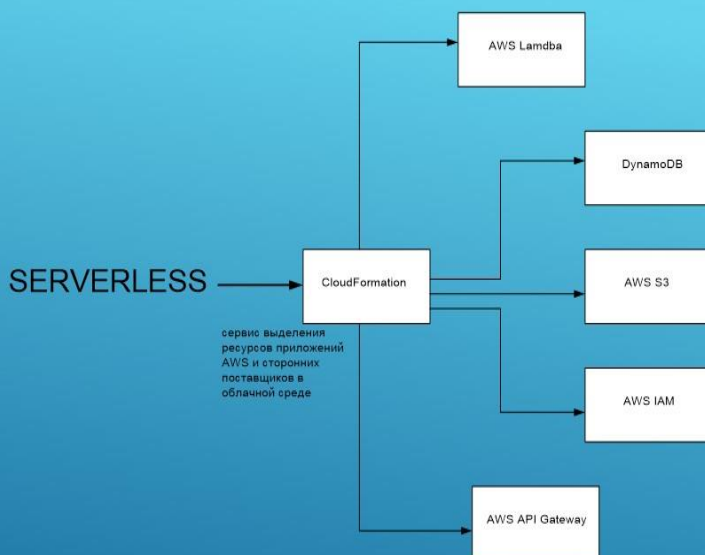
ПГУ 1.090501.17.001 ПЗ

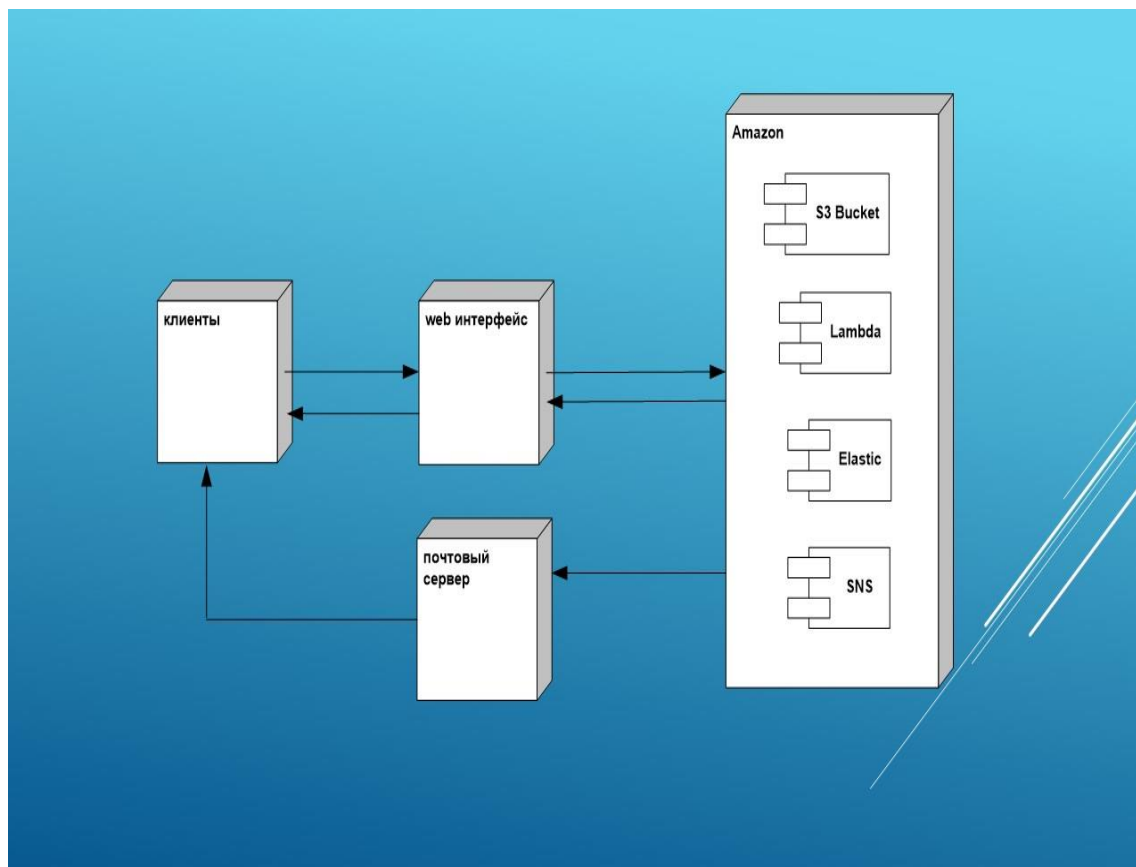
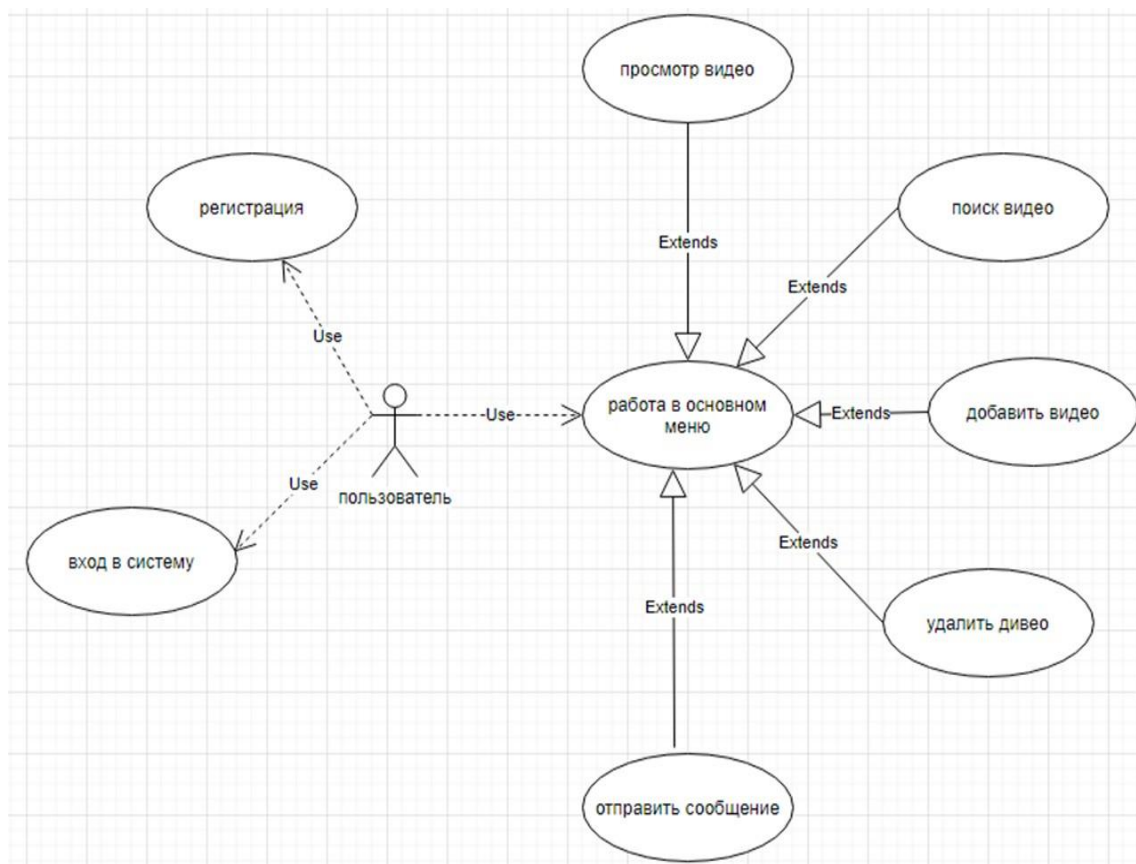
Лист

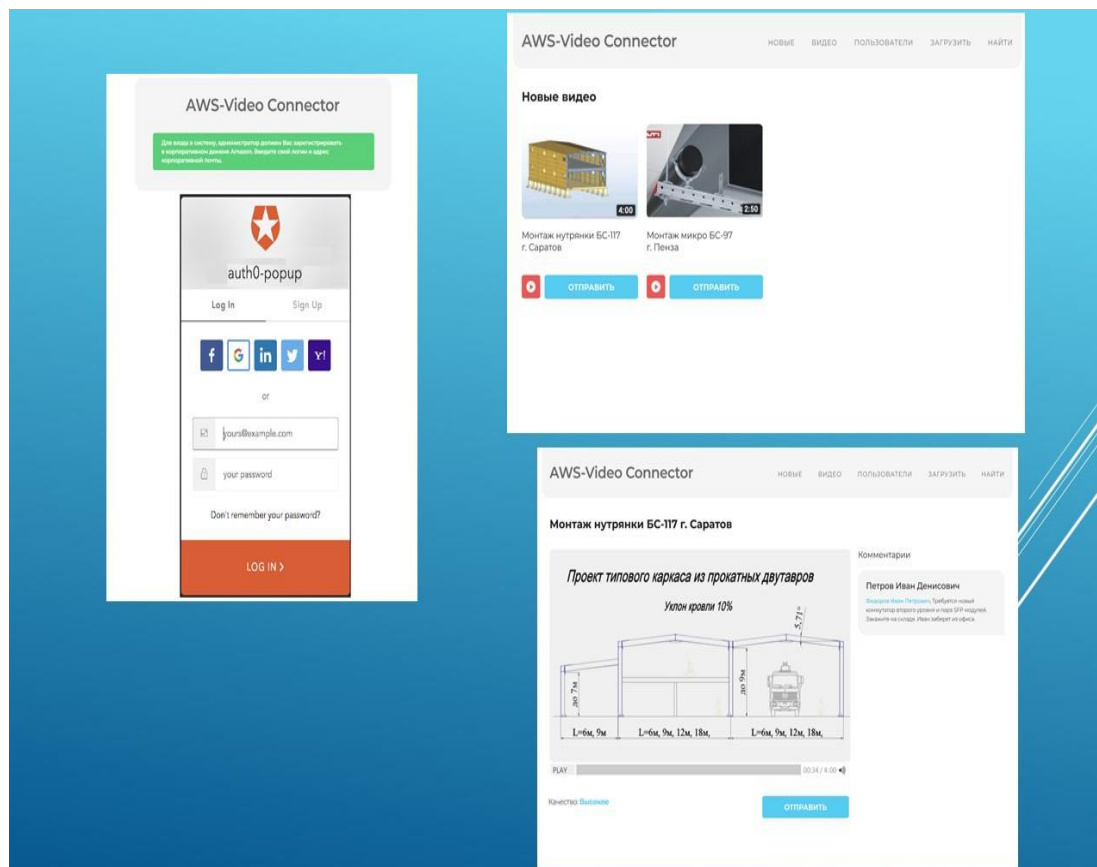
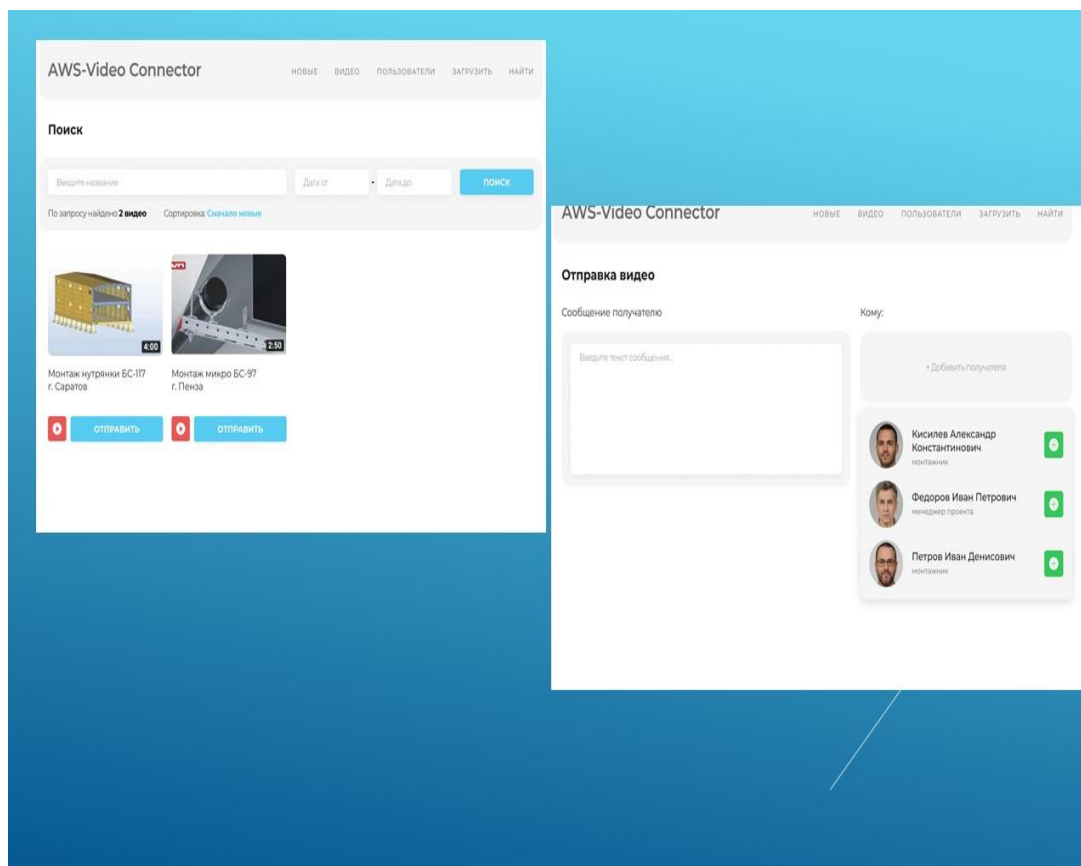
134

ЧТО ТАКОЕ ПОТОКОВОЕ ВЕЩАНИЕ НА БЕССЕРВЕРНОЙ АРХИТЕКТУРЕ

- ▶ В настоящий момент существует множество интернет площадок Youtube, Facebook, с мощной поддержкой систем хранения информации, гибким API для разработки приложений и современным интерфейсом для хранения, обработки и передачи видеоинформации.
- ▶ Основным недостатком вышеперечисленных решений является то, что все они узко специализированы и до конца не могут обеспечить и гарантировать гибкости в обработке информации.
- ▶ Эта причина вынуждает корпоративных потребителей выбирать комплексные решения.
- ▶ В частности одним из таких решений является использование связки облачное решение Amazon, собственный стриминговый сервер и мобильное клиентское приложение.
- ▶ Необходимость данного решения вызвана тем, что порой требуется информирование сотрудников крупных компаний в виде видео инструкции. Для чего основные каналы передачи корпоративной информации e-mail, SMS рассылка, Slack или чат-боты не подходят.





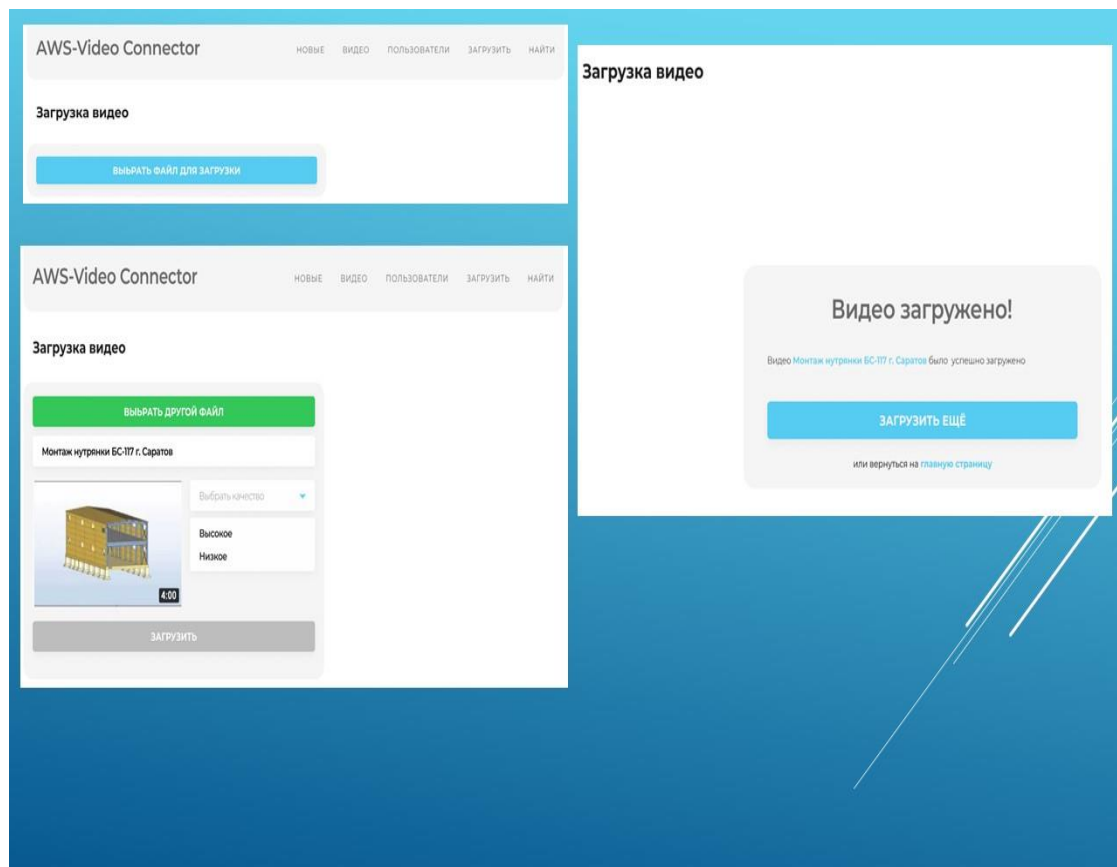


Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.17.001 ПЗ

Лист

137



ЗАКЛЮЧЕНИЕ

В результате выполнения дипломного проекта был осуществлён анализ предметной области на команде разработчиков, изучены особенности её деятельности, определены основные требования к разрабатываемой автоматизированной системе, назначение и функции, было проведено её инфологическое и логическое проектирование, продумана архитектура системы, осуществлена её разработка, отладка и тестирование, подготовлена документация к системе.

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.17.001 ПЗ

Лист

138

СПАСИБО ЗА ВНИМАНИЕ

					ПГУ 1.090501.17.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		139