

Санкт-Петербургский Государственный Университет  
Математико-механический факультет

Программная инженерия

Пыхтин Александр Васильевич

# Распределенная система хранения 3D моделей людей и метаданных

Бакалаврская работа

Научный руководитель:  
профессор Н. А. Терехов

Рецензент:  
технический директор ООО Системы КМ А. Г. Петров

Санкт-Петербург  
2017

St. Petersburg State University  
Faculty of Mathematics and Mechanics

Software engineering

Pykhtin Aleksandr Vasilevich

# Distributed storage system of 3D people models and their metadata

Graduation Thesis

Scientific supervisor:  
professor N. A. Terhov

Reviewer:  
Technical Director of KM Systems A. G. Petrov

St. Petersburg  
2017

# Оглавление

<b>Введение</b>	<b>4</b>
<b>1. Постановка задач</b>	<b>6</b>
<b>2. Обзор предметной области</b>	<b>7</b>
2.1. Существующие решения . . . . .	7
2.2. Используемые технологии . . . . .	7
<b>3. Описание архитектуры</b>	<b>10</b>
3.1. Архитектура сервиса хранения моделей . . . . .	10
3.2. Схема работы сервера . . . . .	11
<b>4. Реализация</b>	<b>12</b>
4.1. База данных . . . . .	12
4.2. Регистрация . . . . .	13
4.3. Аутентификация . . . . .	14
4.4. Получение списка моделей . . . . .	14
4.5. Получение модели . . . . .	15
4.6. Загрузка модели . . . . .	17
4.7. Комментирование . . . . .	18
4.8. Сжатие модели . . . . .	19
<b>5. Мобильное приложение</b>	<b>21</b>
5.1. Схема работы мобильного приложения . . . . .	21
5.2. Отображение моделей в мобильном приложении . . . . .	22
<b>Результаты</b>	<b>25</b>
<b>Список литературы</b>	<b>26</b>

# Введение

Устройства для сканирования частей человеческого тела в настоящее время становятся очень популярны. Множество компаний предлагают услуги по 3D-сканированию, продают устройства для создания моделей. Например 3D-сканеры устанавливают в торговых центрах. Всё это позволяет практически каждому человеку сделать модель своего тела. Подобные 3D-модели могут быть полезны для самых различных целей. Остановимся на двух областях применения.

В медицине и пластической хирургии, у докторов нет права на ошибку, поэтому моделирование является неотъемлемой частью их работы. К тому же в области пластической хирургии очень важно получить положительный обратный отзыв от клиента. Чем раньше будет получен обратная связь от пациента, тем раньше можно будет продолжить дальнейшую работу над планированием его внешнего вида. В этом сильно помогают 3D-модели, которые можно изменить, отправить пациенту на просмотр и получить быстрый отзыв.

Сейчас стало очень популярно следить за своим здоровьем. Все больше людей уделяют внимание своему телу и начинают заниматься фитнесом. Однако часто бывает, что в связи с плотным рабочим графиком, командировками и переездами личные контакты с тренером становятся редким явлением. Это негативно сказывается на результатах тренировок. 3D-сканирование помогает решить эту проблему. Благодаря 3D-моделям любой человек может удалённо контактировать с тренером, показывать свой прогресс и получать дальнейшие инструкции.

Однако, несмотря на возрастающую популярность 3D-моделирования, в настоящее время все ещё существует ряд проблем, решение которых сделает использование 3D моделей более повсеместным:

- Процесс создания подобной модели довольно продолжителен.
- Устройства для сканирования дороги и громоздки, а торговые центры не всегда находятся в шаговой доступности.

- Различные сканеры никак не связаны между собой, из-за чего вся работа по взаимодействию с моделями переносится на пользователя
- Нет простой и удобной системы для их взаимодействия между пациентами и врачами, тренерами и тренирующимися.

Распределённая система хранения, обработки и демонстрации 3D-моделей людей и метаданных позволит легко и удобно работать с 3D-моделями. Так же система позволяет решить часть вышеперечисленных проблем.

# 1. Постановка задач

Целью данной работы является разработка распределённой системы хранения 3D моделей людей и метаданных. Для реализации поставленной цели были выделены следующие задачи:

- Разработать архитектуру распределённой системы хранения.
- Разработать систему хранения и обработки 3D-моделей и API к ней
- Разработать мобильный клиент для демонстрации и работы с 3D-моделями

## 2. Обзор предметной области

### 2.1. Существующие решения

- Naked3d[7] - Система слежения за спортивным прогрессом. Система позволяет осуществлять сканирование своего тела, отслеживать спортивный прогресс, а так же ставить задачи по увеличению объёма мышечной массы. Naked3d обладает довольно узкой специализацией, применяется только для задач фитнеса и не обладает достаточной мощностью для работы с 3D-моделями.
- Sketchfab[11] - Система хранения 3D-моделей Служет для хранения и просмотра созданных пользователями 3D-моделей. Модели открыты для просмотра и комментирования. Sketchfab по своей идее схож с сервисом Instagram, то есть служит для показа моделей публике и получения отзывов о проделаной художником работе.

Как видно из описания, существующие решения обладают довольно узкой специализацией, что не позволяет в полной мере выполнить задачи хранения и обработки 3D-моделей. Так же данные системы не позволяют работать с моделями в медицинских целях и обеспечить функции необходимые для задач моделирования в пластической хирургии. Предложенная в рамках работы система позволит подключать различные модули для обработки 3D-моделей, что позволит выполнить широкий спектр задач. Разработанное приложение позволит получить доступ к просмотру своих моделей в любое время, что ускорит взаимодействие клиентов с докторами или тренерами.

### 2.2. Используемые технологии

- В качестве платформы был выбран [Amazon][1]. Amazon предлагает огромное количество сервисов, прекрасно взаимодействующих между собой. Это позволило абстрагироваться от настройки окружения и заняться разработкой архитектуры.

- В качестве базы данных был выбран [PostgreSQL][8], поскольку он является свободной базой данных с большим сообществом пользователей и обладает достаточными мощностями для дальнейшего расширения сервиса.
- В качестве языка разработки была выбрана [Java][9]. Данный язык обладает обширным сообществом, а так же имеет большое количество библиотек, помогающих в написании разнообразных сервисов. В качестве альтернативы были предложены фреймворки Python: Django и Bottle. Python позволяет быстро разрабатывать прототипы, но не является достаточно мощной платформой для дальнейшего масштабирования.
- Сервер был разработан с использованием контейнера сервлетов [Eclipse Jetty][4]. Jetty позволяет с легкостью поднять HTTP сервер не испытывая проблем с настройкой.
- Для работы с базой данных был использована библиотека [Hibernate][5]. Hibernate решает задачи объектно-реляционного отображения, освобождает разработчика от необходимости писать SQL код, чем ускоряет процесс разработки.
- Компоненты системы были развернуты с использованием [Docker][3] контейнеров. Docker позволяет упаковывать в контейнер приложение вместе с его окружением. Это необходимо для удобства запуска и переноса системы в случае необходимости.
- Для разработки мобильного приложения был выбран фреймворк [Apache Cordova][2]. Apache Cordova позволяет создавать кроссплатформенные мобильные приложения используя JavaScript и CSS.
- Для отрисовки 3D-моделей на мобильном устройстве была использована библиотека для создания и отображения анимированной компьютерной 3D графики [Three.js][10]. Three.js является



кроссбраузерной библиотекой, что позволяет использовать её совместно с Apache Cordova для разработки кроссплатформенного мобильного приложения.

### 3. Описание архитектуры

Для выполнения поставленной цели был разработан сервис, обладающий следующей архитектурой

#### 3.1. Архитектура сервиса хранения моделей

Схема 1: Архитектура сервиса



Система имеет модульную архитектуру. Каждая часть системы находится в отдельном docker контейнере. Подобный подход обеспечивает свободное масштабирование, а так же упрощает перенос и разворачивание в случае необходимости. Компоненты системы обладают собственным REST API для удалённой работы.

Кленты обращаются к основному серверу. Сервер обеспечивает сохранение метаданных в БД, работу клиентов с облаком Amazon S3 для хранения моделей, а так же взаимодействие с сервисом сжатия моделей. 3D-модель представляет собой композицию описания набора точек и текстуры. Это позволяет архивировать данные при пересылке, что снижает размер файлов и экономит трафик. Архивация важна для работы с мобильными устройствами трафик которых ограничен.

## 3.2. Схема работы сервера

Работа с системой происходит согласно следующей схеме

Схема 2: Схема работы сервера



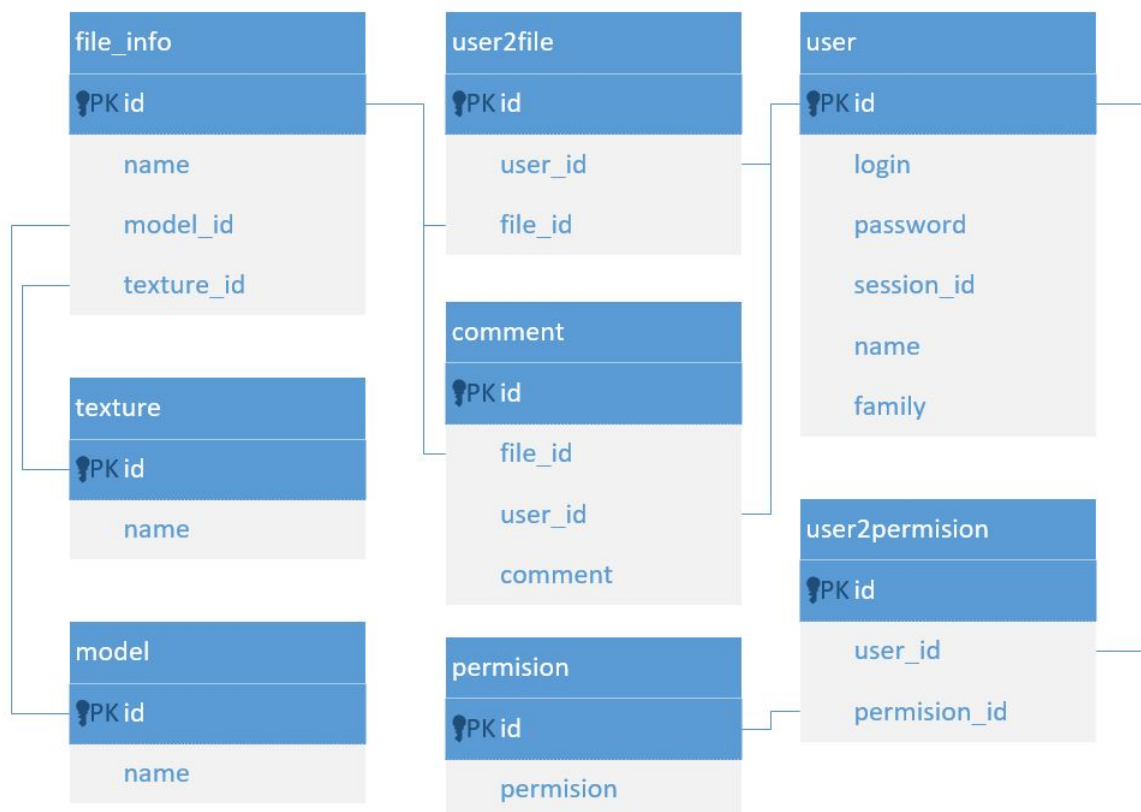
Для доступа к системе пользователю необходимо зарегистрироваться и пройти авторизацию. После авторизации клиент получает доступ к своим моделям, комментариям, а так же получает возможность загружать и выгружать модели. Вся метайнформация о работе с моделями сохраняется в базу данных.

## 4. Реализация

### 4.1. База данных

Схема базы данных выглядит следующим образом

Схема 3: Схема базы данных



Вся информация о модели содержится в таблице `file_info`. Поскольку 3D-модель состоит из 2х файлов(файла описания точек и файла текстуры), `file_info` агрегирует в себе информацию об этих файлах. `file_info` связана с таблицей пользователей `user` отношением многие ко многим через таблицу `user2file`. Таким образом у одного пользователя может быть множество моделей, а у одной модели несколько владельцев. Владельцы могут обладать различными уровнями доступа к моделям. Уровни доступа перечислены в таблице `permission`. В настоящий момент существует 4 уровня доступа к модели. Доступ к просмотру, доступ к загрузке и выгрузке, доступ к обновлению существующей модели, доступ к комментированию. Права связаны с пользователями отношением многие ко многим через таблицу `user2permission`. Комментарии к моделям хранятся в таблице `comment` и связаны с моделью и пользо-

вателем отношением многие к одному.

## 4.2. Регистрация

Регистрация проходит по следующей схеме

Диаграмма 1: Регистрация



Пользователь шлёт запрос

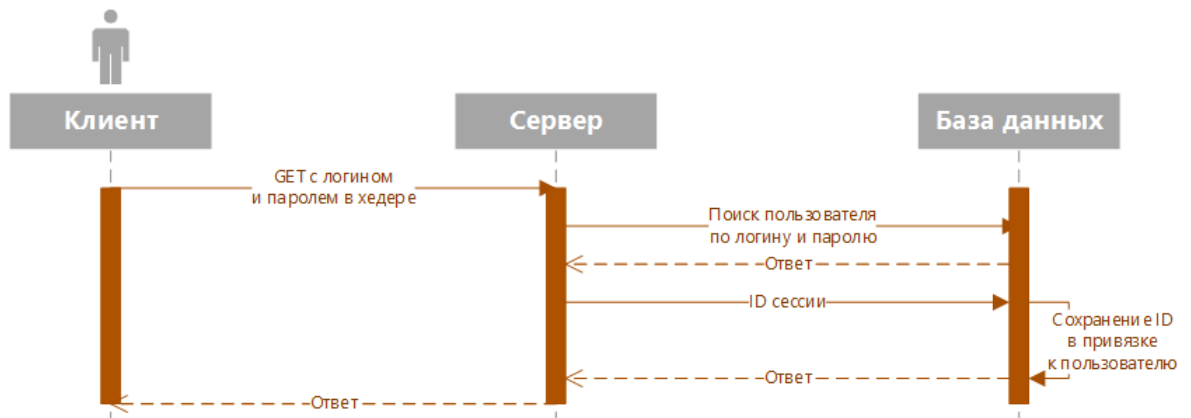
```
POST 52.3.236.245:8080/register
{
  "login": "login",
  "password": "password"
}
```

по адресу

После этого данные о пользователе вносятся в БД и он может пройти аутентификацию

### 4.3. Аутентификация

Диаграмма 2: Аутентификация



Для аутентификации пользователь посылает запрос

```
GET 52.3.236.245:8080/auth
```

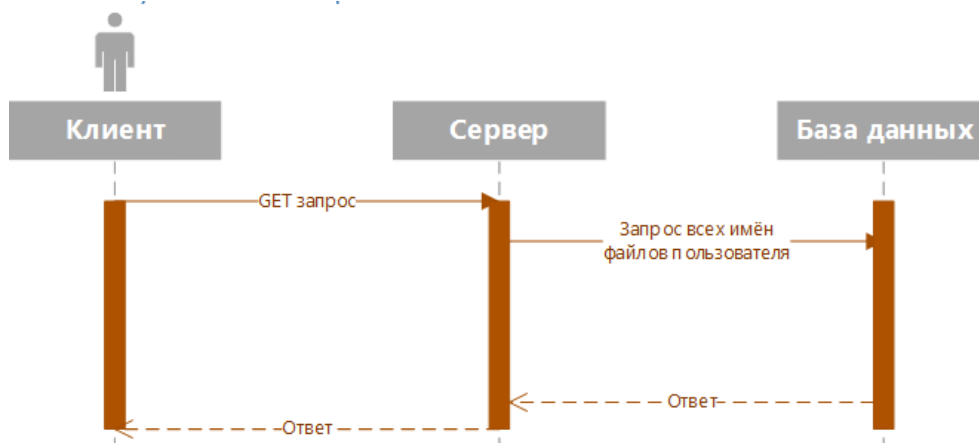
```
Authentication: login:password
```

после чего сервер сохраняет ID сессии в БД с привязкой к пользователю. Дальнее общение происходит с указанием ID в хэдере Cookie.

После прохождения аутентификации пользователь получает доступ к различным функциям системы

### 4.4. Получение списка моделей

Диаграмма 3: Получение списка моделей



Клиент может получить список всех своих загруженных моделей, отправив запрос.

GET 52.3.236.245:8080/download

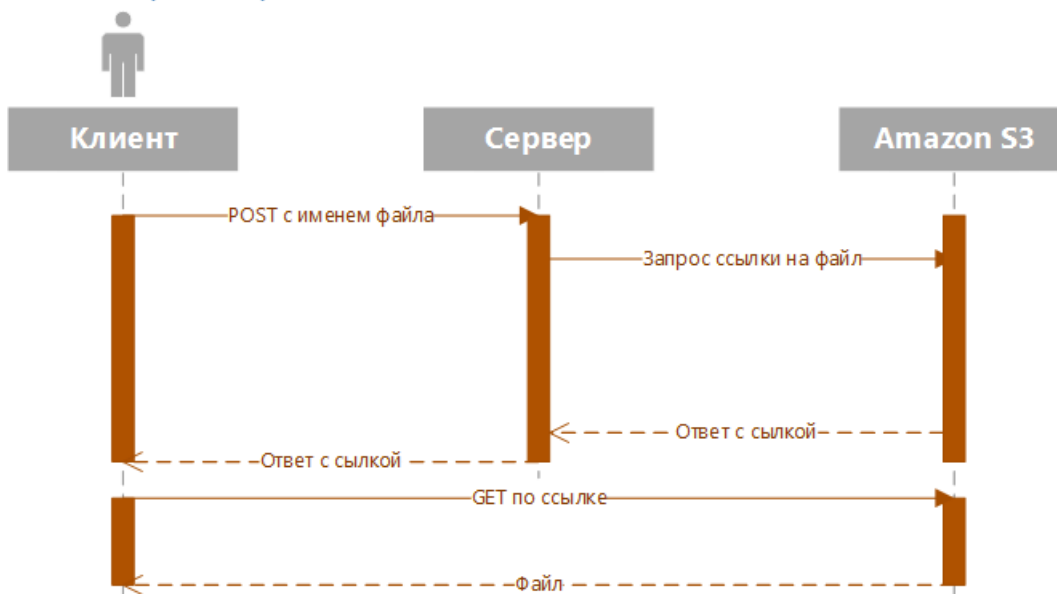
В ответ вернется JSON вида

```
{
  "files":
  [
    "filename1",
    "filename2",
    ...
    "filenameN"
  ]
}
```

Используя список моделей можно выгрузить превью, а так же сами модели.

## 4.5. Получение модели

Диаграмма 4: Получение модели из облака



Получение моделей происходит в 2 шага. Сначала шлется запрос вида

```
POST 52.3.236.245:8080/download
```

```
{
  "files":
  [
    "filename1",
    "filename2",
    ...
    "filenameN"
  ]
}
```

в ответ придёт JSON вида

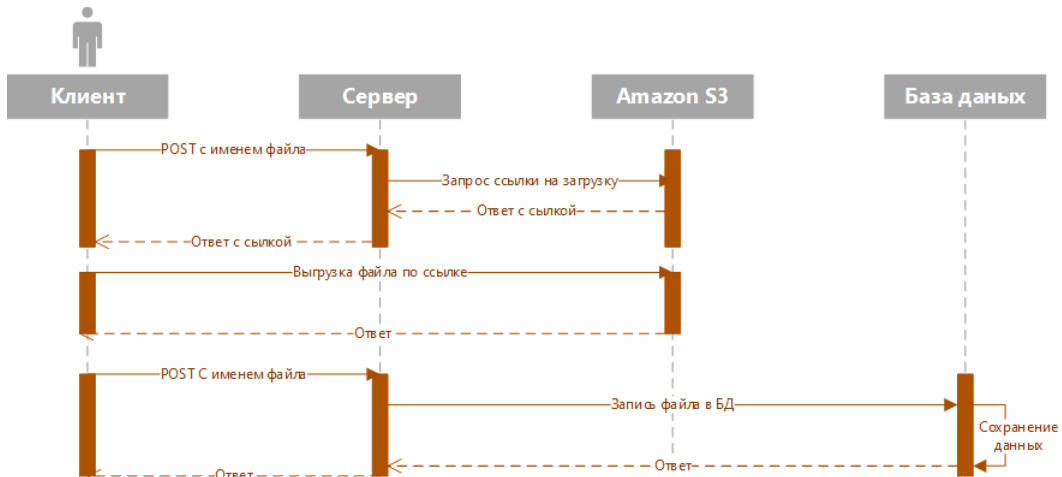
```
{
  "refs":
  [
    "ref1",
    "ref2",
    ...
    "ref3"
  ]
}
```

Далее необходимо выполнить GET запросы по возвращенным ссылкам. В ответ будут приходить файлы моделей.



## 4.6. Загрузка модели

Диаграмма 5: Загрузка модели в облако



Загрузка файла происходит в 3 шага. Сначала шлетя запрос вида  
POST 52.3.236.245:8080/upload

```
{
  "files":
  [
    "filename1",
    "filename2",
    ...
    "filenameN"
  ]
}
```

в ответ придёт JSON вида

```
{
  "refs":
  [
    "ref1",
    "ref2",
    ...
    "ref3"
  ]
}
```

Для загрузки моделей в облако необходимо выполнить PUT запросы по возвращенным ссылкам с файлом в теле запроса. После успешной загрузки моделей необходимо передать серверу информацию об успешно загруженных моделях. Для этого отправляется запрос

```
POST 52.3.236.245:8080/confirm
{
  "files":
  [
    "filename1",
    "filename2",
    ...
    "filenameN"
  ]
}
```

Сервер проверяет, что модели были успешно загружены, и сохраняет в базу данных информацию о загруженных моделях

## 4.7. Комментирование

Диаграмма 6: Комментирование

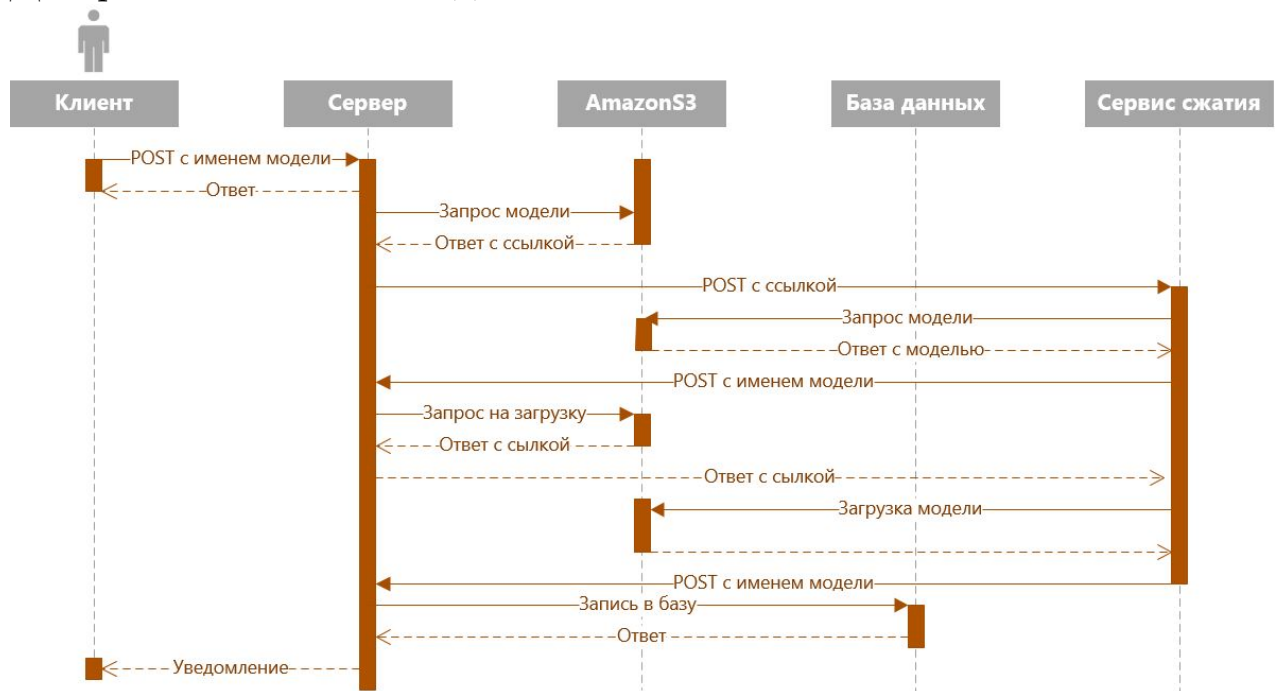


Для комментирования модели необходимо опрарвить запрос

```
POST 52.3.236.245:8080/confirm
{
  "model":
  {
    "user": "login",
    "file": "filename"
  },
  "comment": "comment text"
}
```

## 4.8. Сжатие модели

Диаграмма 7: Сжатие модели



Для сжатия модели используется сторонний сервис, который обеспечивает сжатие с потерями по алгоритму HASC[6]. Сжатие происходит в несколько этапов. Сначала клиент отправляет запрос на сжатие загруженной модели

```
POST 52.3.236.245:8080/compress
{
  "files":
  [
    "filename1",
    "filename2",
    ...
    "filenameN"
  ]
}
```

после этого сервер получает ссылку на модель и отправляет её сервису сжатия

```
POST 52.3.236.245:8085/compress
{
  "files":
  [
    "ref1",
    "ref2",
    ...
    "ref3"
  ]
}
```

Сервис выкачивает модель, сжимает её и загружает в облако(см. Выгрузка модели) Сервер отправляет клиенту информацию об успешном совершении операции.

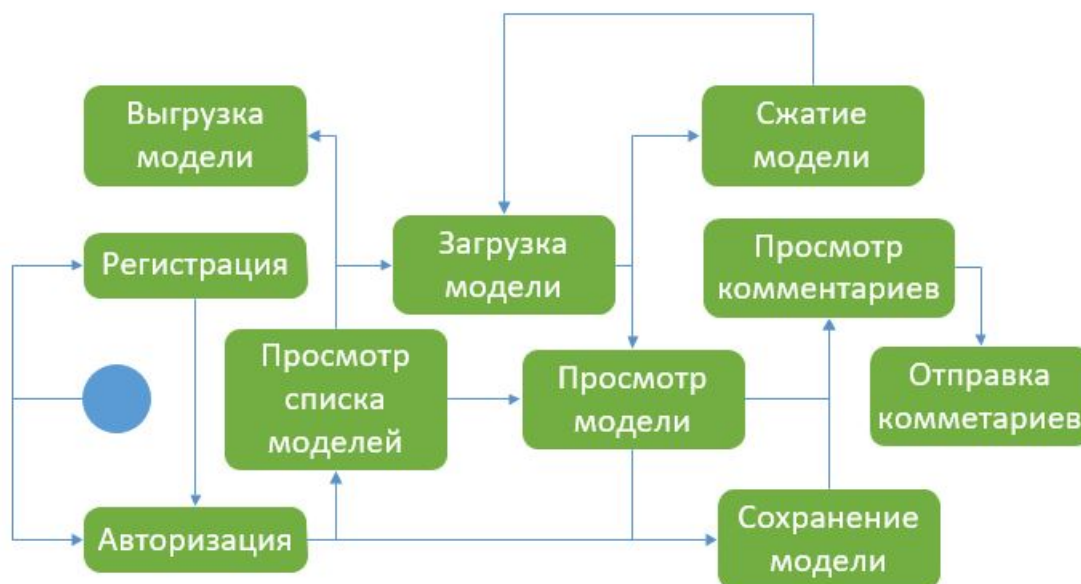
## 5. Мобильное приложение

В ходе работы был разработан прототип мобильного приложения обладающего следующим функционалом

- Регистрация
- Авторизация
- Просмотр моделей
- Загрузка моделей
- Выгрузка моделей
- Комментирование
- Сжатие моделей

### 5.1. Схема работы мобильного приложения

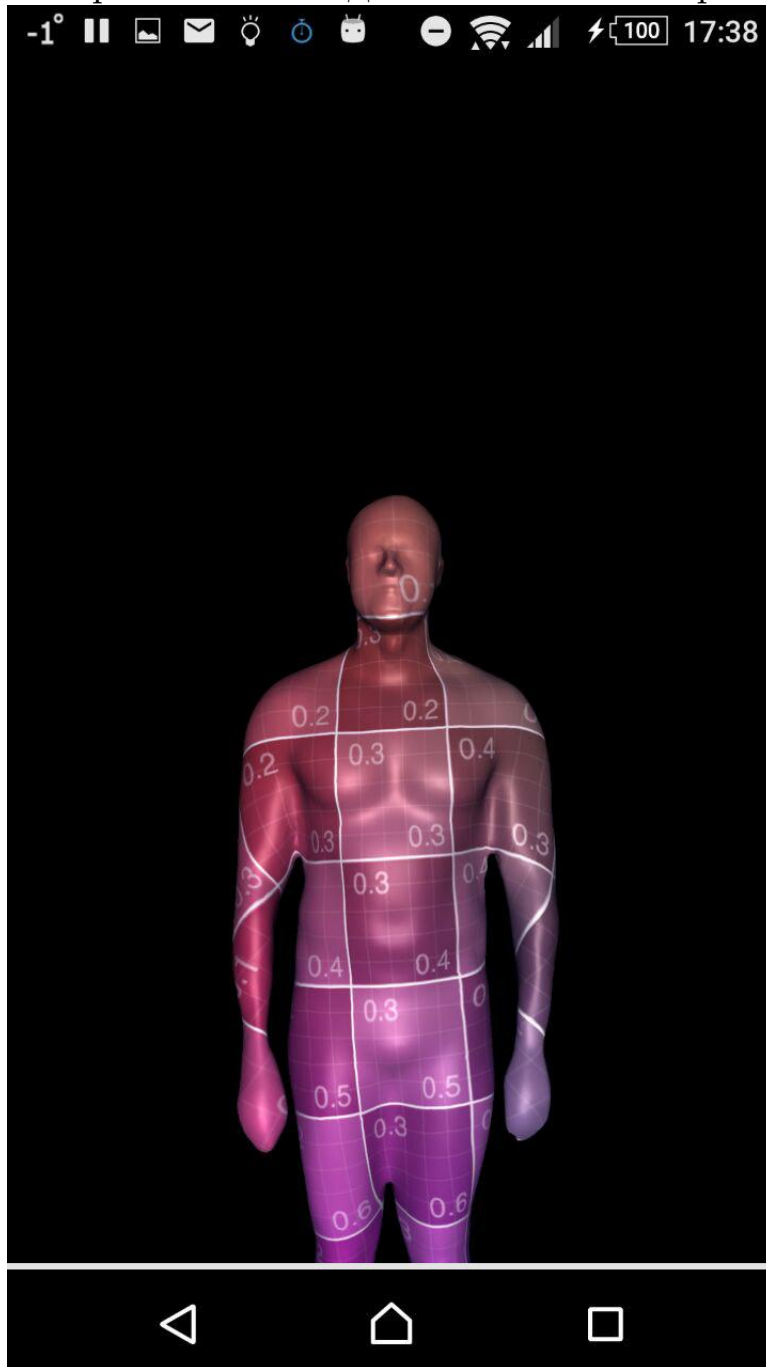
Схема 4: Схема работы мобильного приложения



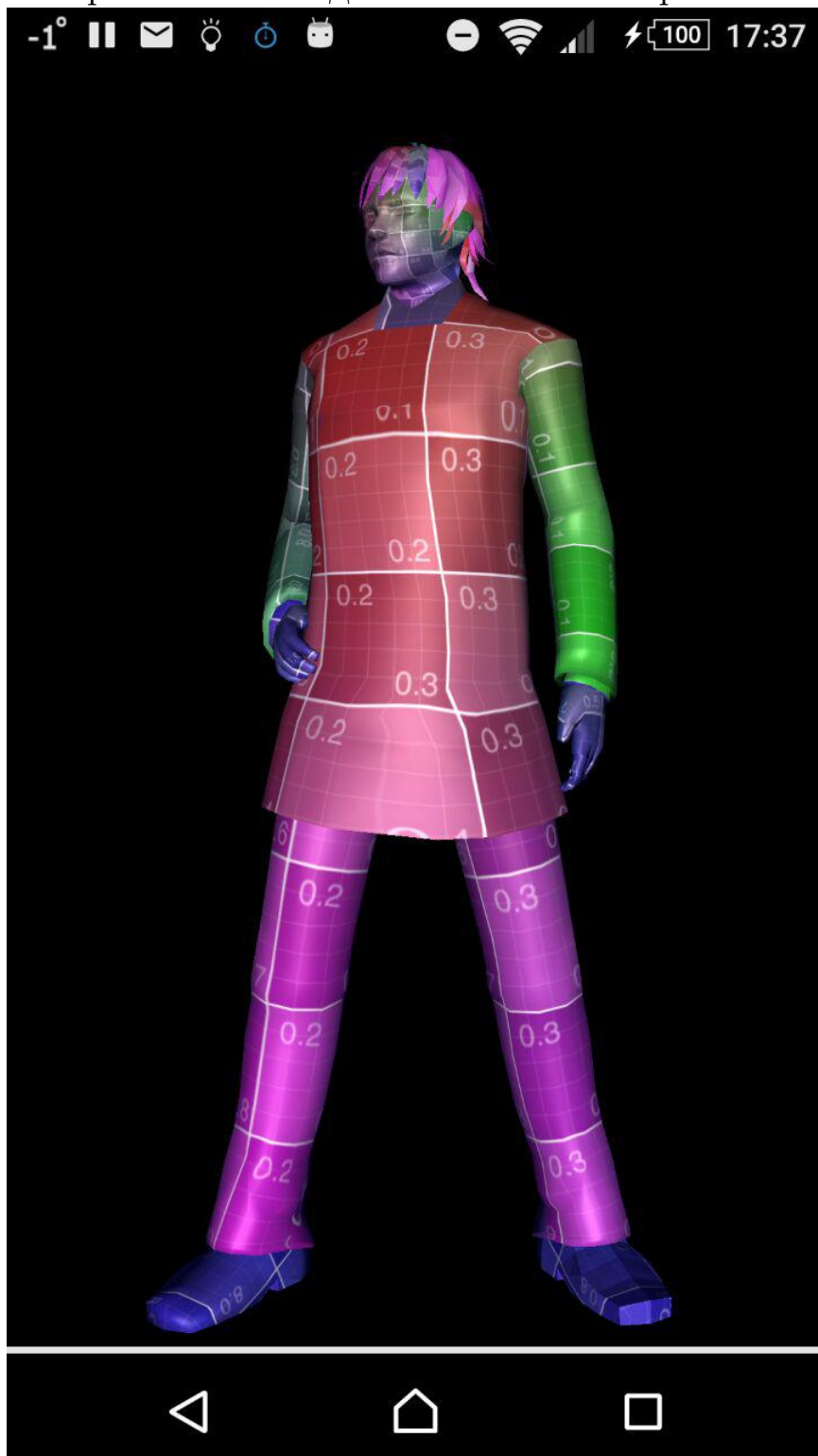
Мобильное приложение позволяет клиенту просматривать свои модели, осавлять комментарии, а так же загружать и выгружать модели. В случае отсутствия сети, приложение позволяет просматривать модели загруженные в память телефона.

## 5.2. Отображение моделей в мобильном приложении

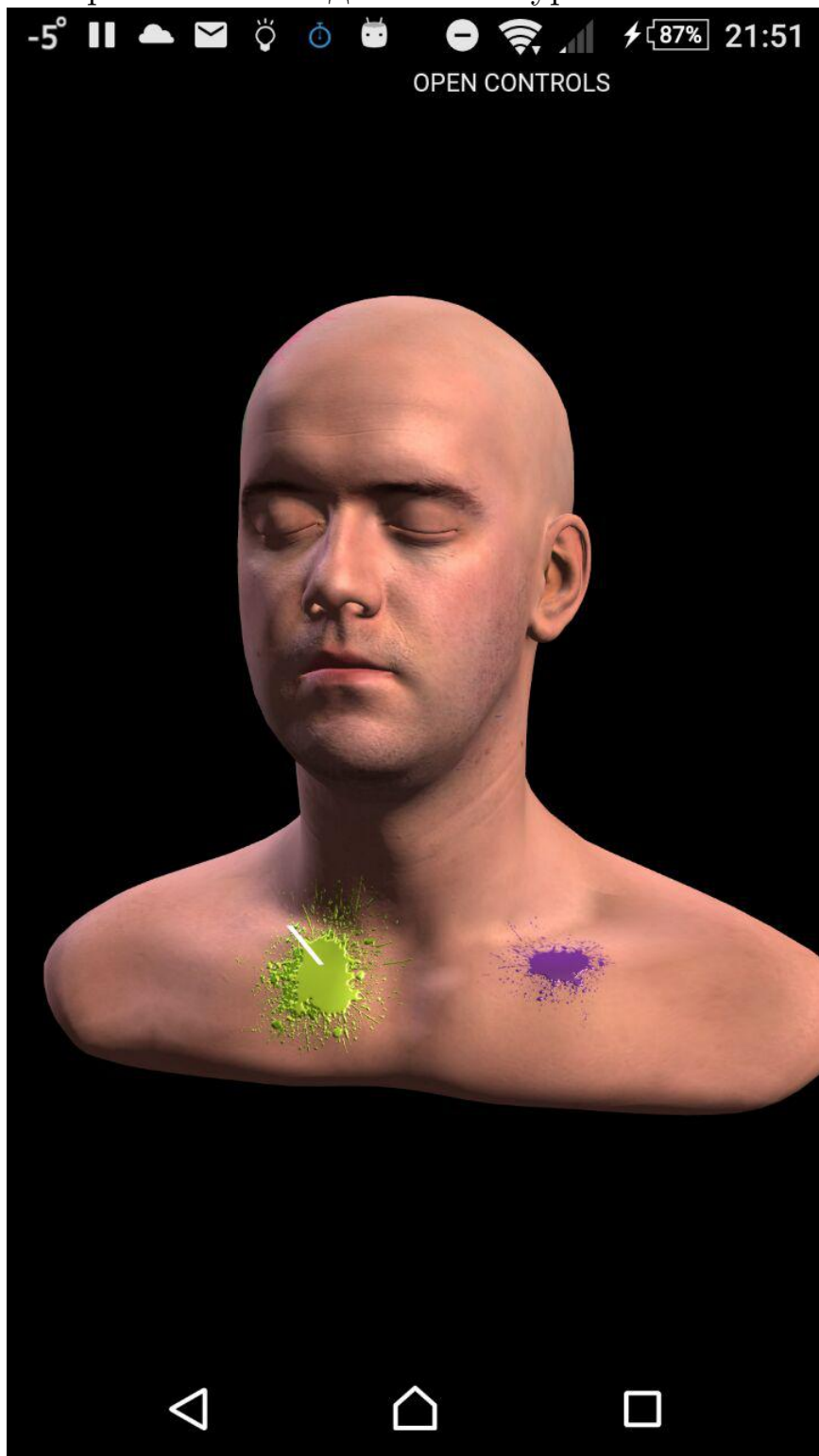
Изображение 1: Модель в мобильном приложении



Изображение 2: Модель в мобильном приложении



Изображение 3: Модель с текстурой в мобильном приложении





# Результаты

В ходе работы была разработана распределённая система хранения 3D моделей людей и метаданных, В том числе были выполнены поставленные задачи.

- Разработана архитектура распределённой системы хранения.

Разработана архитектура, обеспечивающая удалённый доступ к 3D-моделям, позволяющая масштабировать текущее решение, а так же добавлять новые сервисы.

- Разработать систему хранения и обработки 3D-моделей и API к ней.

Разработана система для хранения и обработки 3D-моделей, поддерживающая регистрацию, авторизацию, загрузку и выгрузку моделей, комментирование, а так же сжатие по алгоритму NASD с использованием стороннего сервиса, а так же API для доступа к системе.

- Разработать мобильный клиент для демонстрации и работы с 3D-моделями.

Разработан мобильный клиент, позволяющий работать с созданным сервисом.

## Список литературы

- [1] Amazon. Amazon AWS site // Amazon AWS. — 2017. — URL: <https://aws.amazon.com> (online; accessed: 18.05.2017).
- [2] Cordova. Cordova // Cordova. — 2017. — URL: <https://cordova.apache.org/> (online; accessed: 18.05.2017).
- [3] Docker. Docker // Docker. — 2017. — URL: <https://www.docker.com/> (online; accessed: 18.05.2017).
- [4] Eclipse. Jetty official site // Eclipse Jetty. — 2017. — URL: <http://www.eclipse.org/jetty/> (online; accessed: 18.05.2017).
- [5] Hibernate. Hibernate ORM // hibernate. — 2017. — URL: <http://hibernate.org/> (online; accessed: 18.05.2017).
- [6] MAMOU Khaled. HACD. — 2017. — URL: <http://kmamou.blogspot.ru/2011/10/hacd-hierarchical-approximate-convex.html> (online; accessed: 18.05.2017).
- [7] Naked.fit. — 2017. — URL: <https://naked.fit/> (online; accessed: 18.05.2017).
- [8] PostgreSQL. PostgreSQL official site // PostgreSQL. — 2017. — URL: <https://www.postgresql.org/> (online; accessed: 18.05.2017).
- [9] SE Java. Oracle Java SE // Java SE. — 2017. — URL: <http://www.oracle.com/technetwork/java/javase/overview/index.html> (online; accessed: 18.05.2017).
- [10] Threejs. Threejs // Threejs. — 2017. — URL: <https://threejs.org/> (online; accessed: 18.05.2017).
- [11] sketchfab. — 2017. — URL: <https://sketchfab.com/> (online; accessed: 18.05.2017).