

**МИНОБРНАУКИ РОССИИ**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**  
**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Факультет компьютерных наук  
Кафедра технологий обработки и защиты информации

Аналог видеохостинга “YouTube”  
Курсовой проект

09.03.02 Информационные системы и технологии  
Обработка информации и машинное обучение

Обучающийся \_\_\_\_\_ Д.М. Молин, 3 курс, д/о  
Обучающийся \_\_\_\_\_ В.Е. Бондаренко, 3 курс, д/о  
Руководитель \_\_\_\_\_ В.С. Тарасов, ст. преподаватель  
Руководитель \_\_\_\_\_ К.В. Зенин, ассистент

Воронеж 2022

<b>Введение.....</b>	<b>3</b>
<b>1      Постановка задачи .....</b>	<b>5</b>
<b>2.     Анализ предметной области .....</b>	<b>7</b>
<b>2.1.   Анализ деятельности информационной системы видеохостинга</b>	<b>7</b>
<b>2.2.   Обзор существующих решений .....</b>	<b>8</b>
<b>2.2.1   YouTube.....</b>	<b>8</b>
<b>2.2.2   Discord.....</b>	<b>10</b>
<b>3.     DRF .....</b>	<b>11</b>
<b>3.1.1   Web API.....</b>	<b>14</b>
<b>3.1.2   REST .....</b>	<b>15</b>
<b>3.1.3   DRF.....</b>	<b>16</b>
<b>4.     Анализ задачи.....</b>	<b>19</b>
<b>4.1.   Навигация .....</b>	<b>19</b>
<b>5.     Диаграммы .....</b>	<b>21</b>
<b>Заключение .....</b>	<b>25</b>
<b>Список использованных источников.....</b>	<b>26</b>

## Введение

Интернет и видеохостинги — это самые быстро развивающиеся технологии нашего мира.

Сейчас почти происходит в интернете. В 14 февраля 2005 года на свет появился видеохостинг «Youtube».

YouTube — видеохостинг, предоставляющий пользователям услуги хранения, доставки и показа видео. YouTube стал популярнейшим видеохостингом и вторым сайтом в мире по количеству посетителей. Пользователи могут загружать, просматривать, оценивать, комментировать, добавлять в избранное и делиться видеозаписями, не нарушая правила и политику пользования сервисом.

В январе 2012 года ежедневное количество просмотров видео на сайте достигло 4 млрд. На сайте представлены фильмы, музыкальные клипы, трейлеры, новости, образовательные передачи, а также любительские видеозаписи, включая видеоблоги, летсплеи, слайд-шоу, юмористические видеоролики и прочее. Также на сайте есть различные музыкальные чарты, показывающие предпочтения пользователей в зависимости от географического положения.

Пользователи могут комментировать, оценивать чужие комментарии, добавлять аннотации и титры к видео, а также выставять рейтинг просмотренным видео (но автор видео может скрыть количество лайков и дизлайков, если пожелает). Человек, загрузивший видео, также может запретить «встраивание» (embedding) своего видео на другие сайты, блоги и форумы. Также, по выбору, он может преобразовать загруженное видео из 2D в 3D. Первоначально YouTube предлагал просмотр видео только в одном уровне качества с разрешением  $320 \times 240$  пикселей с использованием кодека Sorenson Spark (вариант H.263) с монозвуком MP3.

В 2007 году разработчики YouTube уже предпринимали попытку

сделать более продвинутый видеоредактор с использованием технологии Adobe Flash, он назывался YouTube Remixer, но разработки были приостановлены.

С января 2009 года YouTube предоставляет возможность скачивать некоторые видеоролики напрямую с сайта возможно сохранение без помощи сторонних приложений. Сохранённое видео размещается в кэше браузера (если ролик имеет большой размер, в кэше может оказаться только его часть, которая просматривалась последней, как правило этого не происходит с роликами длительностью менее 15 минут).

В данной курсовой работе предполагается сделать аналог видеохостинга Youtube, но с чуть-чуть большим функционалом. Добавить текстовый чат, чтобы пользователи могли общаться и комментировать просмотренное видео в реальном времени.

## **1 Постановка задачи**

Исходя из современной ситуацией в мире, YouTube может быть вовсе заблокирован на территории РФ. Но пользователям все равно хотелось бы просматривать видео, самим создавать видео. И также преимуществами видеохостинга является неограниченное количество загруженных видео.

Главной задачей является создание сайта, в котором пользователь мог бы создавать аккаунты, загружать видео, просматривать видео других пользователей ставить лайки и писать комментарии, а также смог позвонить другому пользователю в чате и обсудить видео, которое они посмотрели.

Большое внимание следует уделить и дизайну. Проанализировав несколько средств разработки выбрать оптимальное.

Актуальность данной работы обусловлена востребованностью создания видеохостинга который может функционировать независимо от ситуаций в мире. Способ привлечения пользователей посредством видеохостинга отличается относительно низкими затратами и большим количеством целевой аудитории.

Объектом разработки является сайт видеохостинга.

Предметом разработки является разработка сайта видеохостинга.

Целью работы является разработка web-сайта интернет-магазина, предназначенного для продажи товаров и ознакомления потенциальных клиентов с компанией.

В соответствии с поставленной целью в работе определены следующие задачи:

- изучить технологии по разработки сайтов;
- определить цели, идеи, потребности видеохостинга;
- разработать техническую концепцию сайта (структуру);

- разработать программный продукт;
- Результатом работы является сайт видеохостинга.

## **2. Анализ предметной области**

### **2.1. Анализ деятельности информационной системы видеохостинга**

В настоящее время в глобальной сети интернет существует бесчисленное множество сайтов. Все сайты очень разноплановые и отличаются друг от друга по большому количеству параметров.

По типам предоставляемых сервисов Web-сайты можно разделить на коммерческие и некоммерческие. К коммерческим сайтам относят те сайты, которые непосредственно связаны с ведением бизнеса. Среди них можно выделить: продвигающие «офф-лайн»-бизнес (т.е. бизнес, который существует вне Интернета) и ориентированные на онлайн-коммерцию (те виды бизнеса, которые без Интернета невозможны, например интернет-торговля). Основной аудиторией коммерческого сайта являются действительные и потенциальные клиенты. По своим функциям и свойствам сайты бывают:

- информационными сайтами;
- визитками;
- электронными магазинами;
- корпоративными представительствами, — системами управления предприятием;
- видеохостингами;
- порталами.

В соответствии с целью данной курсовой работой рассмотрим видеохостинги подробнее.

Видеохостинг – веб-сервис, позволяющий загружать и просматривать видео в браузере, например, через специальный проигрыватель. Содержит

строку поиска видео, кнопки загрузки видео, строку с уже готовыми видео. Большое количество сайтов видеохостинга тематически не ограничивают своё наполнение. Однако, некоторые видеохостинги занимают специализированные секторы, предлагая тематические порталы. Особое место занимают сервисы публикации научного, научно-популярного и учебного видеоконтента. В то время как на некоторых сайтах проводится жёсткий контроль закачанных видеофайлов, многие видеохостинги испытывают проблемы, связанные с тем, что пользователи закачивают видеоклипы, не являясь их правообладателями. Так, против YouTube время от времени возбуждаются судебные разбирательства, в которых производители музыкальных видеоклипов, фильмов или телесериалов требуют от Google (владельца сервиса) денежной компенсации.

## 2.2. Обзор существующих решений

### 2.2.1 YouTube

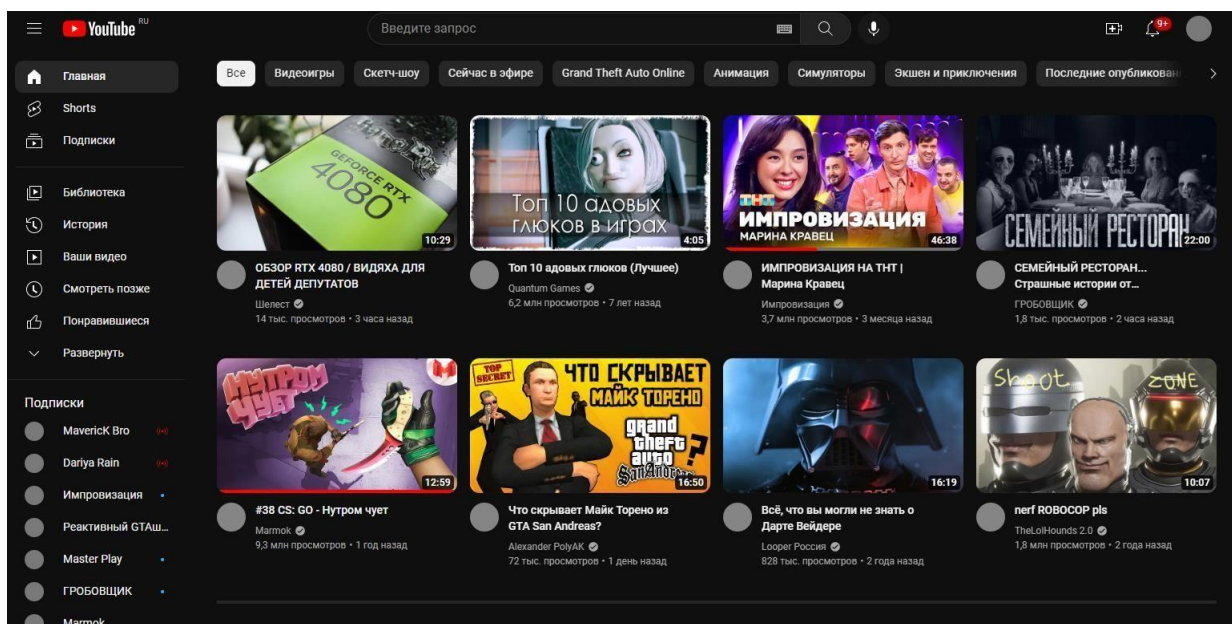


Рисунок 1 - главная страница YouTube.



### Достоинства:

- + Возможность публиковать видео очень большого размера;
- + огромное разнообразие видеоконтента – всевозможные темы, форматы, жанры;
- + общедоступность и бесплатность;
- + возможность загружать и просматривать видеоролики во многих видеоформатах;
- + интеграция с телевидением, СМИ, социальными сетями и т. д.;

### Недостатки:

- повсеместная реклама, которая вставлена практически везде. Чтобы выложить собственный ролик, нужно устанавливать какую-либо рекламу.
- принудительная ограниченность контента. В отличие от классического телевидения, много важной информации просто не допускается администрацией для показа по различным соображениям – политическим, религиозным, стратегическим, меркантильным и т. д.;
- плохая служба поддержки. В основном работает автоматика, которая ограничена программным алгоритмом. Это вызывает частые блокировки аккаунтов, каналов или видеороликов из-за ошибочно воспринятых причин.

## 2.2.2 Discord

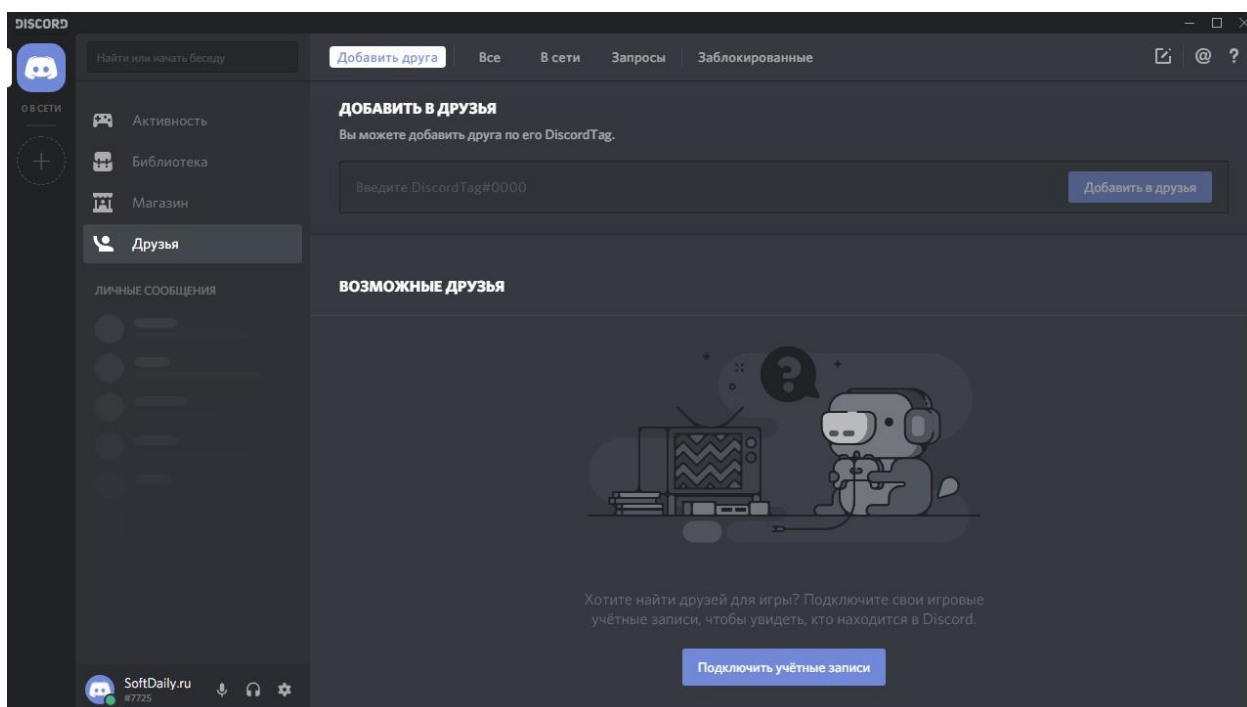


Рисунок 2 - главная страница Discord

Достоинства:

- + Бесплатный доступ;
- + Возможность организации аудио и видео конференций;
- + Высокое качество звука и видео;
- + Понятный интерфейс;
- + Боты. В сервисе есть большой каталог официальных ботов с заданными свойствами и функциями.

Недостатки:

- Отсутствие поддержки программы на ранних версиях Windows;
- Невозможность публикации файлов более 8 Мб без оформления платной подписки;

- Отсутствие функции записи видео;
- Низкая популярность в корпоративной среде.

### **3 Реализация**

Текущее приложение состоит из двух модулей: Back-end и Front-end.

#### **3.1 Back-end**

Для разработки back-end приложения существует множество языков программирования, такие как: Java, Python, C++, C#, Go и т.д. Наше приложение использует фреймворк Django, написанный на языке Python. Django - это свободный фреймворк для разработки быстрых и безопасных веб-приложений и сайтов на языке Python. Использует шаблон проектирования MVC.

Базовая структура фреймворка включает следующие компоненты:

- URL-маршрутизаторы, перенаправляющие HTTP-запрос от браузера любого иного веб-клиента в представления;
- представление, которое обрабатывает запрос, обращается к модели и сообщает ей, какие именно данные из БД нужно задействовать, чтобы удовлетворить запрос;
- модель (менеджер базы данных, ORM), «вытаскивающую» нужную информацию из БД и передающую ее представлению;
- HTML-шаблоны, которые используются представлением для демонстрации пользователю полученных от модели данных.

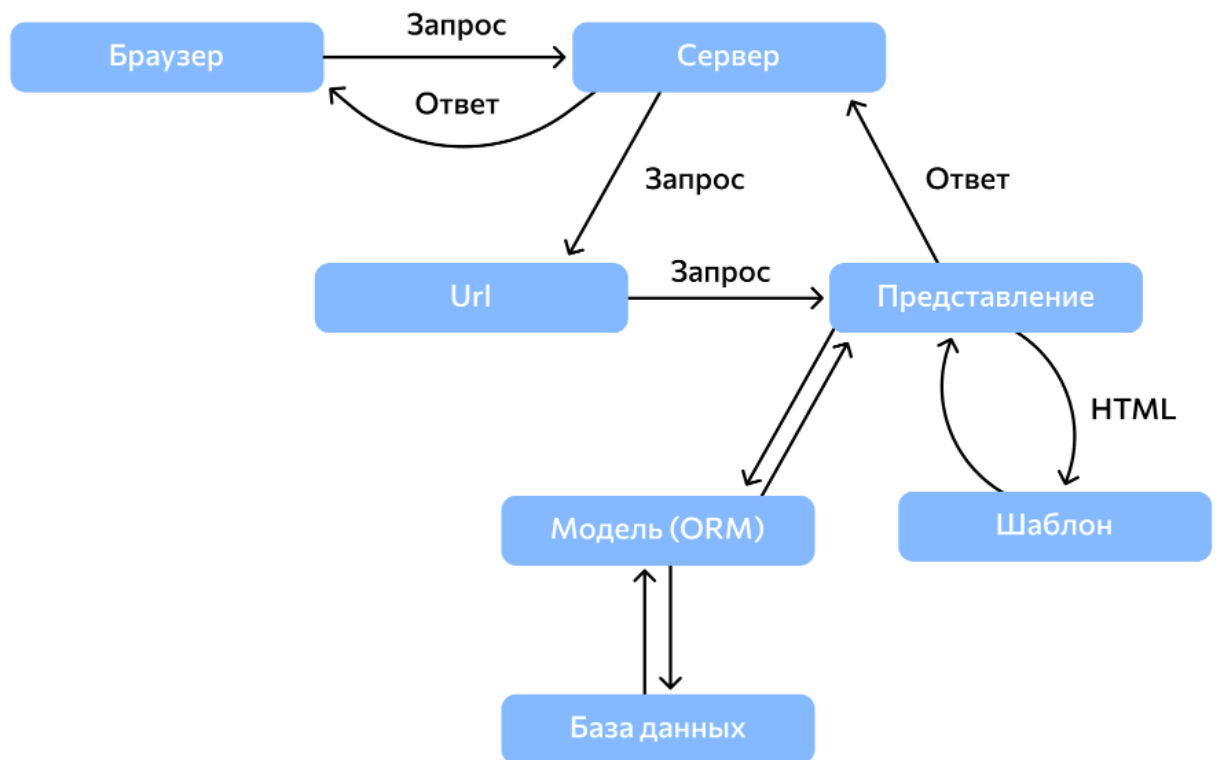


Рисунок 3 - Схема работы Django

#### Преимущества Django:

- Фреймворк Django спроектирован по принципу «Все включено». Разработчик может с его помощью создать веб-приложение без сторонних компонентов. Это полезно для начинающих программистов, так как им не приходится отвлекаться на поиск дополнительных инструментов для решения типовых задач;

- Django подходит для разработки веб-сайтов и приложений любого типа: систем управления контентом, новостных или информационных ресурсов, видеохостингов, социальных сетей и т.д. Он может работать с любыми типами файлов, различными базами данных, клиентскими средами. Если в «коробочной» версии фреймворка нет компонента для реализации конкретного продукта, его можно получить из сторонних источников;

- Django поддерживается и развивается Django Software Foundation, а также сообществом сторонних разработчиков. Фреймворк обновляется и

совершенствуется, проверяется на ошибки. Созданные приложения стабильные и содержат меньше багов по сравнению с разработанными на других фреймворках, CRM или написанными с нуля;

- Архитектура Django строится на независимости составляющих частей. Любой компонент можно заменить или модифицировать, не затрагивая другие. Возможности Django позволяют как разрабатывать приложения, так и расширять их при увеличении трафика и нагрузки;

- Django позволяет повторно использовать код, группировать связанные функции в отдельные модули. Это облегчает, сокращает и упрощает структуру приложения. Если у проекта сменился разработчик, он сможет быстро разобраться в архитектуре ПО и обеспечить качественную поддержку;

- У Django есть встроенные инструменты защиты от распространенных хакерских атак. Также он позволяет эффективно распределять доступ к данным среди пользователей разного уровня. Это повышает безопасность продукта и стабильность его работы;

- Язык программирования Django — Python, адаптированный ко всем распространенным платформам. Разработанные на фреймворке приложения одинаково хорошо работают на Windows, Mac OS X и Linux-based операционных системах, а также их версиях. Django поддерживают веб-хостинги с подходящей инфраструктурой и документацией;

- Django — открытое ПО, поэтому любой желающий может использовать его для создания приложения или сайта, в том числе для коммерческого использования.

Недостатки Django:

- «Джанго» развивается как единый и самодостаточный комплекс инструментов разработчика. Это позволяет реализовывать даже крупные проекты, не обращаясь к сторонним приложениям и сервисам, но и тормозит

развитие самого фреймворка. Разработчикам приходится тратить силы на все инструменты, входящие в его ядро;

- У Django есть свой менеджер базы данных, которая позволяет работать с различными типами БД. Однако у нее отсутствуют некоторые полезные функции, которые есть в других ORM и активно используются разработчиками. Основной недостаток — отсутствие интеграции с SQLAlchemy, являющейся основным инструментом работы с базами данных на языке Python;

- «Джанго» не позволяет отдельным процессам работать с несколькими запросами одновременно. Несмотря на то что разработчики пытаются решить проблему, им приходится использовать различные подходы.

### **3.1.1 Web API**

Сегодня сеть интернет построена по принципу Клиент-Серверного взаимодействия. Клиент посылает запрос — Сервер ему отвечает. В случае, когда между собой общаются два Сервера, мы условно называем Клиентом того, который отправил запрос и ожидает ответ, а Сервером будет тот, кто принимает запрос и отвечает не него. Взаимодействие браузеров и веб-сайтов (первые выступают в роли Клиента, а вторые в роли Сервера) традиционно делалось при помощи технологии html-рендеринга, именно так изначально это делал Django. Чтобы получить данные с веб-сайта, браузер отправляет запрос GET к Серверу. Сервер формирует ответ в виде html-страницы и передает ее браузеру. Так Сервер передает данные браузеру, но как браузер может передать данные Серверу? В этой самой html-странице Сервер заложил все необходимые веб-формы, заполнив которые, пользователь мог бы передать свои данные обратно на сервер. Когда вы ввели свои данные в форму на сайте, браузер отправляет Серверу запрос POST, в котором содержатся ваши данные, а Сервер обрабатывает их и записывает в базу

данных.

Все это отлично работало, но уже в середине нулевых такой подход перестал удовлетворять возрастающим требованиям в веб-разработке. Появлялись мобильные приложения, различные гаджеты с доступом в интернет, и для них уже не подходил стандартный способ html-рендеринга на сервере, ведь теперь каждому клиенту нужно было отрисовать данные по-своему. Постоянно увеличивалось взаимодействие серверов друг с другом, и html-формат уже не подходил. Для всех этих задач есть другой способ обмена данными — Web API. Смысл этого способа в том, что Сервер передает Клиенту не html-страницу, а непосредственно данные, никак не влияя на то, как эти данные будут в итоге представлены. Наиболее популярными форматами для передачи данных становятся XML и JSON. Таким образом Сервер полностью избавляется от задачи отрисовки данных. Какое-то время длился переходный период, когда разработчикам веб-приложений на Сервере приходилось поддерживать оба способа одновременно: html рендерился на Сервере для браузеров, а Web API использовался для мобильных приложений и интеграции с другими серверами. Понятно, что разработчикам приложений на Сервере приходилось делать двойную работу. Но в начале десятых ситуация стала меняться в пользу Web API. Этому способствовало молниеносное развитие инструментов на языке JavaScript, а также появление различных веб-фреймворков, одним из которых и является предмет данной статьи.

### **3.1.2 REST**

В 2000 году Рой Филдинг написал докторскую диссертацию, где изложил концепцию REST. Там были рекомендации о том, как спроектировать Сервер, чтобы ему было удобно общаться с Клиентами. Выделю два главных принципа создания приложений в стиле REST:

Сервер не должен ничего знать о текущем состоянии Клиента. В

запросе от Клиента должна быть вся необходимая информация для обработки этого запроса Сервером.

Каждый ресурс на Сервере должен иметь определенный Id, а также уникальный URL, по которому осуществляется доступ к этому ресурсу.

На данный момент мы можем найти фреймворк для создания приложений в стиле REST практически для каждого языка программирования, используемого в веб-разработке. Логика построения Web API на Сервере в этих фреймворках реализована одинаково.

Действия для управления данными привязаны к определенным HTTP-методам. Существует несколько стандартных действий для работы с данными — это *Create, Read, Update, Delete*. Часто их обобщают как CRUD.

Для создания объекта используется http-метод POST;

- Для чтения — http-метод GET;
- Для изменения — http-метод PUT;
- Для удаления — http-метод DELETE;

### **3.1.3 DRF**

Django Rest Framework— это библиотека, которая работает со стандартными моделями Django для создания гибкого и мощного API для проекта.

API DRF состоит из 3-х слоев: сериализатора, вида и маршрутизатора.

Сериализатор: преобразует информацию, хранящуюся в базе данных и определенную с помощью моделей Django, в формат, который легко и эффективно передается через API.

Представление: определяет функции (чтение, создание, обновление,



удаление), которые будут доступны через API.

Маршрутизатор: определяет URL-адреса, которые будут предоставлять доступ к каждому виду.



### 3.2 Front-end

Для разработки front-end части приложения использованы следующие инструменты: html, css, js. Так же использовался шаблонизатор Django для рендеринга, сбор в единое целое, html документа на сервере.

HTML — это язык разметки, который состоит из различных команд – "тегов". Всего существует более ста тегов, но чаще всего приходится взаимодействовать примерно с третью. Про остальные теги необходимо помнить, чтобы представлять все возможности HTML и пользоваться ими в нужный для того, чтобы оформить элемент страницы.

CSS — это язык описания стилей, который определяет, как будет наглядно отображаться HTML-документ. CSS работает с шрифтами на странице, изображениями, высотой и шириной объектов, цветом, полями, а также с позиционированием элементов на странице.

JavaScript - прототипно-ориентированный сценарный язык программирования. Обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое


применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам. Основные архитектурные черты:

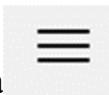
- динамическая типизация;
- слабая типизация;
- автоматическое управление памятью;
- прототипное программирование;
- функции как объекты первого класса.

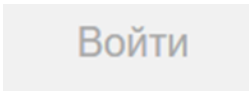
## 4. Анализ задачи

### 4.1. Навигация

Основная задача видеохостинга – загрузка и просмотр видео, общение в чате, но, чтобы это стало доступно нужно пройти регистрацию или авторизоваться.

Кнопка  – Открывает главную страницу, на которой показаны загруженные видео


Кнопка  – открывает шторку, в которой можно перейти на главную страницу

Кнопка  - открывает страницу авторизации/регистрации

Кнопка Чат – открывает чат

При нажатии на видео – открывается плеер, где можно посмотреть видео, почитать комментарии

При авторизации:

- Кнопка -открывает ваш профиль;
- Кнопка Мои видео–показывает ваши видео;
- Кнопка Подписки-показывает ваши подписки на другие каналы;
- Кнопка Понравилось–показывает видео, которые понравились пользователю;
- Кнопка Не понравилось–показывает видео, которые не понравились пользователю;
- Кнопка Создать видео–позволяет загрузить и дать название вашему новому видео;

- Кнопка  -позволяет выйти со своего аккаунта.

## 5. Диаграммы

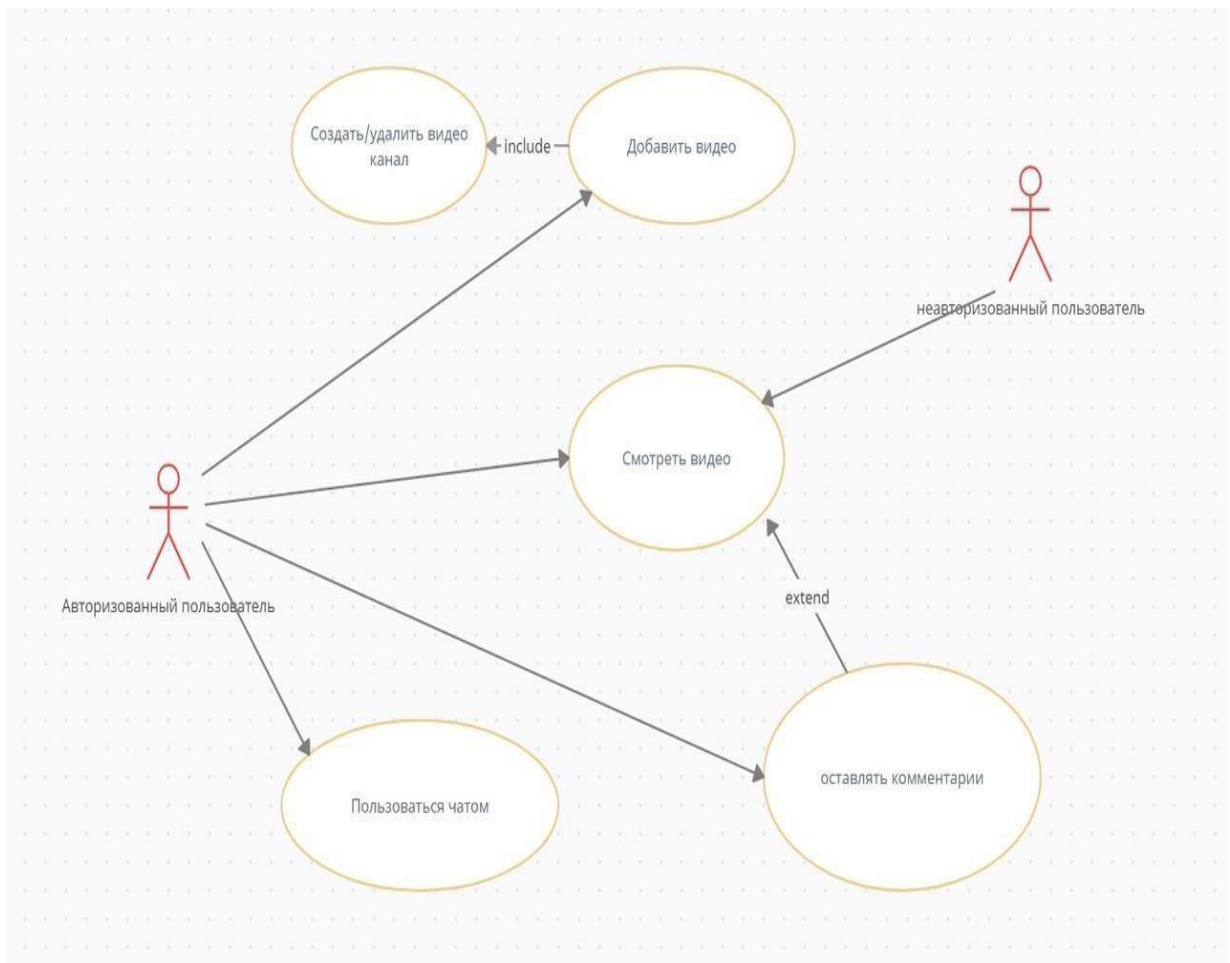


Рисунок 4 - Диаграмма Use Case

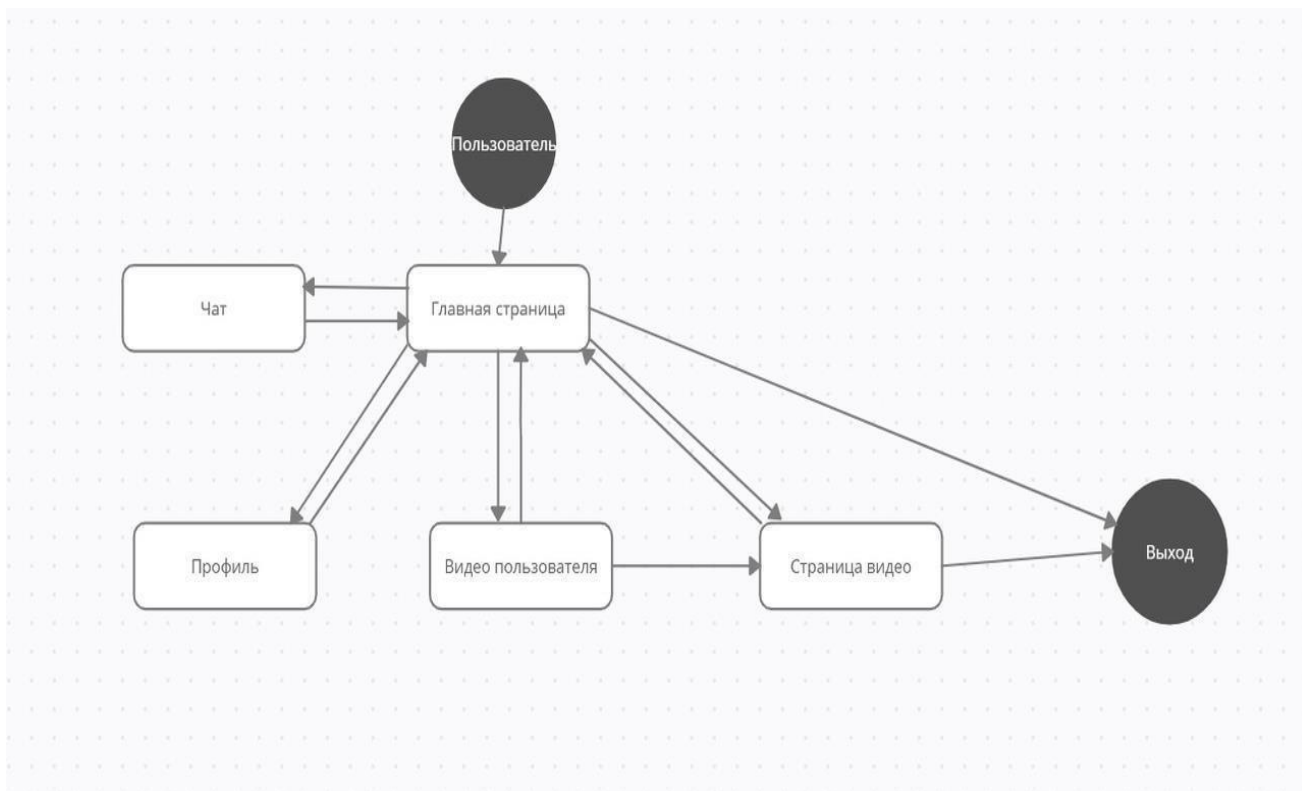


Рисунок 5 - Диаграмма состояния

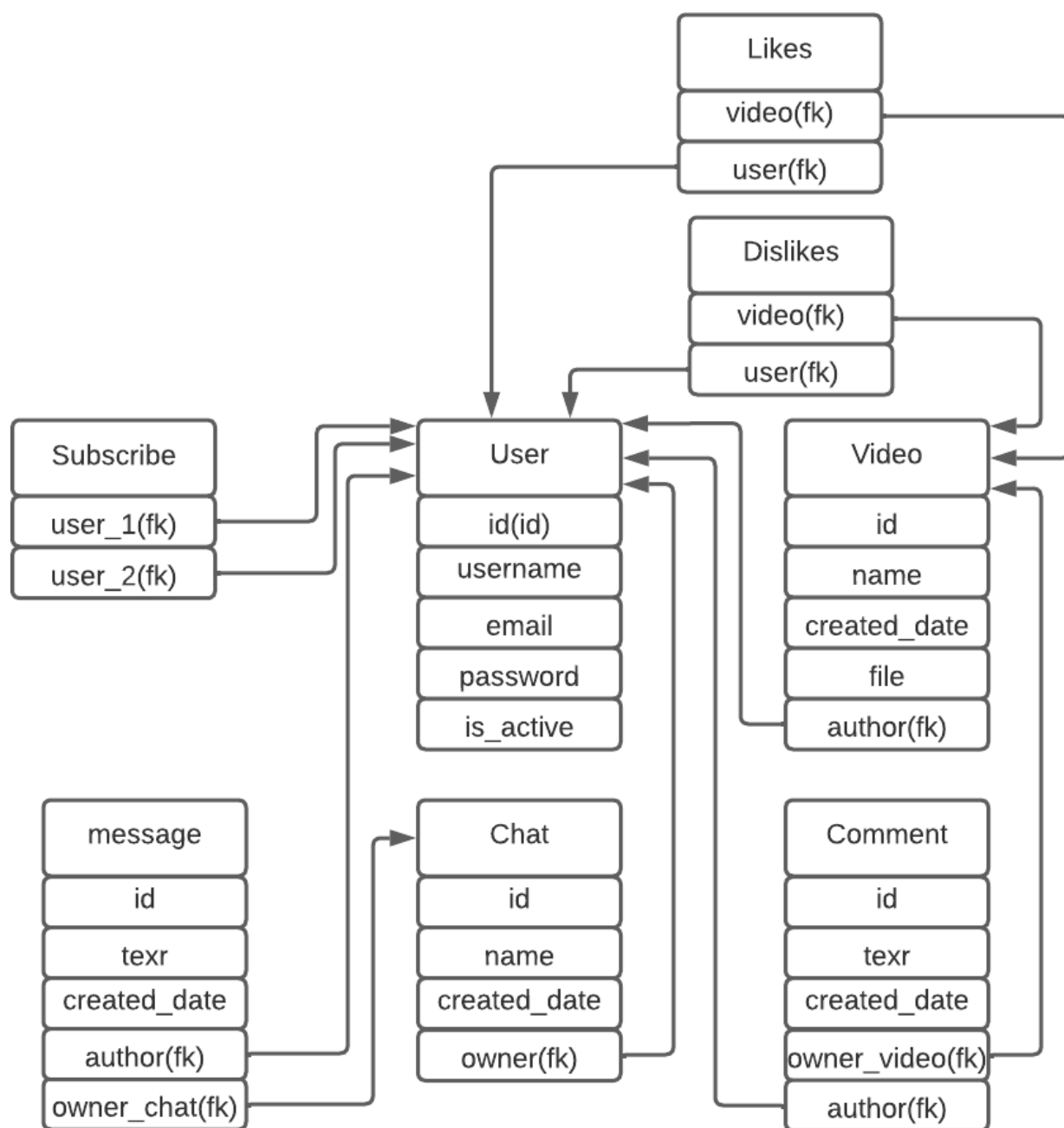


Рисунок 6 - ER-Диаграмма

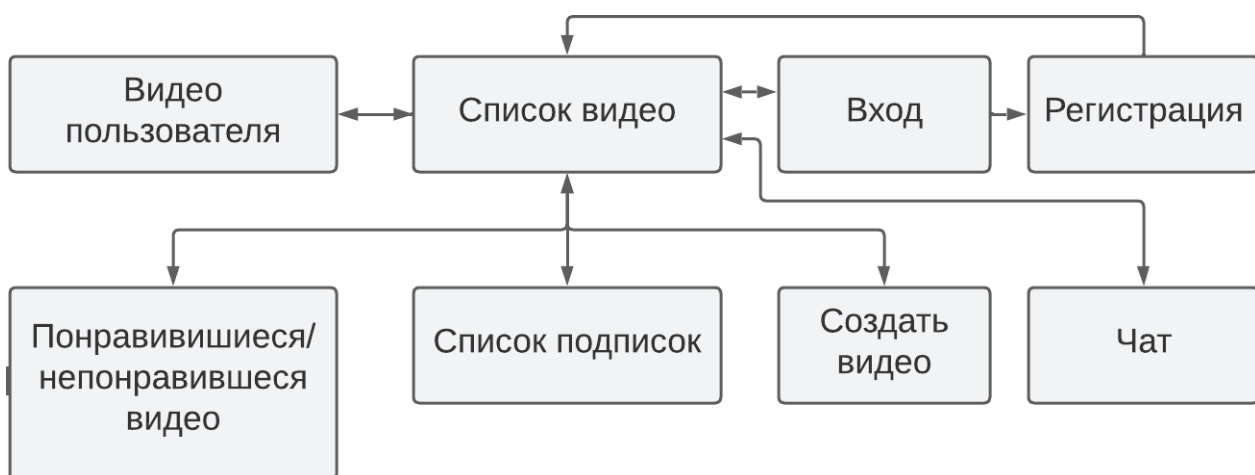


Рисунок 7 - Структура сайта



## **Заключение**

В ходе выполнения данной курсовой работы был разработан аналогвидеохостинга YouTube – AnalogYoutube.

Цель выполнена, сайт был успешно разработан. Он нормальнофункционирует и готов к использованию. На сайте очень просто ориентироваться.

Сайт выполняет функции, которые на него возлагались и предоставляетнужную информацию пользователю

## **Список использованных источников**

1. Django. Разработка веб-приложений на Python — Джефф Форсье, ПолБиссекс, Уэсли Дж. Чан
2. Алексеев А.П., Введение в Web-дизайн: учебное пособие. - М.: СОЛОН-ПРЕСС, 2008
3. Буч Г., Рамбо Д., Джекобсон А., Язык UML для пользователя: Пер. с англ. - М.: ДМК, 2000