

Designing Youtube or Netflix

Let's design a video sharing service like Youtube, where users will be able to upload/view/search videos. Youtube is one of the most popular video sharing websites in the world. Users of the service can upload, view, share, rate, and report videos as well as add comments on videos.

For the sake of this exercise, we plan to design a simpler version of Youtube with following requirements:

Functional Requirements:

1. Users should be able to upload videos.
2. Users should be able to share and view videos.
3. Users should be able to delete his own videos.
4. Users should be able to perform searches based on video titles, keywords, ratings, publication date, publishing user and so on.
5. Users should be able to specify own preferences based on keywords (Drama, Thriller and so on)
6. Our services should be able to record stats of videos, e.g., likes/dislikes, total number of views, etc.
7. Users should be able to add and view comments on videos.
8. System should be able to provide video recommendations.
9. System should be able to list the most popular videos, channels, subscriptions.

Non-Functional Requirements:

1. The system should be highly reliable, any video uploaded should not be lost.
2. The system should be highly available. Consistency can take a hit (in the interest of availability); if a user doesn't see a video for a while, it should be fine.
3. Users should have a real time experience while watching videos and should not feel any lag.

ER Design:

In the first phase of your project, you will do the requirement analysis using ER-model. All the requirements can be obtained from the functional requirements section of this document. You must decide what kind of storage you want to use (SQL vs NoSQL) for the different entities in your system. After this phase you should generate an ER-diagram with any other supporting documentation, describing the assumptions you made. For the ER-diagram you can use any

graphical editor you want, and you should finally create a PDF file using ER notations from the lectures/lab/book. You can make reasonable assumptions on your design, as long as:

- That you state them clearly in the documentation for this phase.
- They do not contradict the system requirements analysis we provide.

Schema Design:

Your task in this phase will be to translate the ER design that you made in the first step to relational database schema and determine how data will be stored in distributed file system. This includes File format, Compression and Data storage system. The relational database schema will be in a form of a single executable SQL script (*.sql file with SQL statements).

Implementation:

Your tasks in this phase will be to develop either:

- A standalone client application. This type of software development involves creation of desktop apps like Word, Excel, and Photoshop that run in the classic desktop environment and take full advantage of that environment's specific features.
- A service-based solution with a web app. This type of applications can be accessed through the Internet (or through an Intranet) using a web browser with an active internet connection. These applications are programmed using a client-server modeled structure—the user ("client") is provided services through an off-site server that is hosted by a third-party. Examples of commonly used web applications include webmail, online retail sales, online banking, and online auctions.

Because of the complexities of the service-based solutions projects that are implemented this way will receive extra bonus during the grading. Also, teams that have more than 3 members are required to implement their app as a service-based solution.

Documentation:

Product documentation describes the product that is being developed and provides instructions on how to perform various tasks with it. In general, product documentation includes requirements, tech specifications, business logic, and manuals. There are two main types of product documentation:

- System documentation represents documents that describe the system itself and its parts. It includes requirements documents, design decisions, architecture descriptions, program source code, and FAQs.
- User documentation covers manuals that are mainly prepared for end-users of the product and system administrators. User documentation includes tutorials, user guides, troubleshooting manuals, installation, and reference manuals.

Please make sure you take care of the documentation during the project development.