

Log In

Join

Back To Module Home

Design Problems

0% completed

RESHADED Approach for System Design

Design YouTube

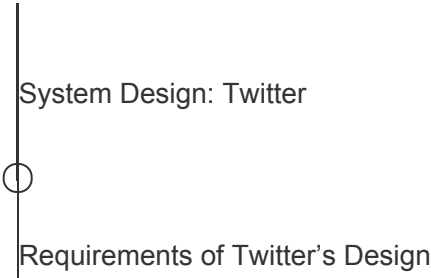
Design Quora


Design Google Maps

Design a Proximity Service / Yelp

Design Uber

Design Twitter





- High-level Design of Twitter
- Detailed Design of Twitter
- Client-side Load Balancer for Twitter
- Quiz on Twitter's Design

Design Newsfeed System

Design Instagram

Design a URL Shortening Service / TinyURL

Design a Web Crawler

Design WhatsApp

Design Typeahead Suggestion

Design a Collaborative Document Editing Service / Google Docs

Conclusion

Mark Module as Completed

System Design: Twitter

Get an overview of Twitter and a brief description of what we'll learn in this chapter.

We'll cover the following

- Twitter
- How will we design Twitter?

Twitter#

Twitter is a free microblogging social network where registered users post messages called “Tweets”. Users can also like, reply to, and retweet public tweets. Twitter has about 397 million users as of 2021 that is proliferating with time. One of the main reasons for its popularity is the vast sharing of the breaking news on the platform. Another important reason is that Twitter allows us to engage with and learn from people belonging to different communities and cultures.

A user performs various operation on Twitter

The illustration below shows the country-wise userbase of Twitter as of January 2022 (source: Statista), where the US is taking the lead. Such statistics are important when we

design our infrastructure because it allows us to allocate more capacity to the users of a specific region, and traffic served from near specific users.

Country wise userbase on Twitter

How will we design Twitter?#

We'll divide Twitter's design into four sections:

1. **Requirements:** This lesson describes the functional and non-functional requirements of Twitter. We'll also estimate multiple aspects of Twitter, such as storage, bandwidth, and computational resources.
2. **Design:** In this lesson, we'll discuss the high-level design of Twitter in this lesson. We also briefly explain the API design and identify the significant components of the Twitter architecture. Moreover, we will discuss how to manage the Top-k problem, such as Tweets liked or viewed by millions of users on Twitter.
3. **Client-side load balancers:** This lesson discusses how Twitter performs load balancing for its microservices system to manage billions of requests between various services' instances. Furthermore, we also see why Twitter uses a customized load-balancing technique instead of other commonly used approaches.
4. **Quiz:** Finally, we'll reinforce major concepts of Twitter design with a quiz.

Let's begin with defining Twitter's requirements.

Back

[Quiz on Uber's Design](#)

Next

[Requirements of Twitter’s Design](#)

[Mark as Completed](#)

[Report an Issue](#)