

# Design Part 2

# Object-Oriented Design

- Given a **technical/real-world problem**
- **Simplify** to basic parts
- Sketch out the **classes** and **methods**
- Think like a **designer for a game**

# 4 Steps

- Clarify the question
- Start with main objects
- Try link them up
- Action with examples

# Clarify the Question

- Interviewer is hiding the detail
- Ask like a co-worker
- who, where, when, why, how, what
  - 1. Internal/External/International user
  - 2. FE/BE
  - 3. Rush hour? Flat? Normal distributed?
  - 4. Current solution not good enough
  - 5. What's the specific requirement

# Start with Main Objects

- Consider the main parts
- "Design a Parking Lot"
  - Parking Lot
    - Controller
    - ParkingSpace
      - RegularParkingSpace
      - HandicappedParkingSpace
      - CompactParkingSpace
      - EV, RV, ExpectedMother, Truck
    - Car, Sign, Floor, Parker, PayStation, Valet

# Try Link Them Up

1. Relationship between objects
  2. Member of.
  3. Inherit & Implement
- ParkingLot should have an array of Floors
  - Floor should have an array of ParkingSpace
  - RegularPS, CompactPS, HandicappedPS... inherit from ParkingSpace
  - Sign should be Enum: Full, Empty, Invalid

# Action with Examples

1. Already have the basic outline of your design
  2. Take a example and walk through
  3. Consider the key actions
  4. Update the design if forgot any object
- SpaceController
    - assignSpace
    - getCar
  - Floor
    - getVacantSpace#
    - isFull
  - Space
    - sign
    - type
  - Car
    - Plate#
    - type

# Short URL

# Short URL

In Twitter, Weibo, you have character limit. Also normal URL will contain some invalid ASCII code that you cannot put into the content.

So a lot of company has its own Short URL

Example: <http://t.cn/RGKGR3T>

# Short URL

Design A System that can generate this Short URL

- The user input an URL, the server returns a short one
- Other user click this short URL, it will redirect to the real URL
- If the URL has not been used (no one clicks for 72 hours, configurable), it will be removed

# How to generate a short URL

UUID  
MD5

Are these too long? Could we find a better way?

Think about the requirement  
one URL could be used released after some day

# How to generate a short URL

We do not generate a short URL everytime, we  
just lookup the table

# How to send request

Apache Tomcat

# Design a Database

Short URL -> Long URL, time

Pretty Easy? If we have a high QPS?

# What DB we should use?

Relational Database: Oracle, MySql

Non-Relational Database (NO Sql):  
MongoDB, CouchDB, Couchbase, Redis

# RDBMS vs NoSQL

RDBMS(Relational Database Management System)

- Strong mathematical basis
- Declarative syntax
- A well-known language in Structured Query Language (SQL)
- Real-time Read
- Read/Write Consistency

# RDBMS vs NoSQL

## NoSQL (Not Only SQL)

- It does not have schema
- Read/Write throughput is very high
- Horizontal scalability can be achieved easily
- Will support Bigdata in volumes of Terra Bytes & Peta Bytes
- Faster development life cycles for developers
- Can be hosted in cheaper hardware machines
- In memory caching option is available to increase the performance of queries

**Lots of companies combine  
these two together**

# For this question

We could just use NoSQL

Reason: we do not have complicated query to fetch data, we only need original URL and short URL (timestamp maybe)

# How to remove unused links?

How to update the Store?

How to remove them?

# How to remove unused links?

short -> Long

short -> time

Is it okay?

# How to remove unused links?

If we could stretch the time

short -> Long

short -> time

Daily data purge job

# If the QPS is very high Everything is different

Do we really need to be accurate to seconds to  
purge the data?

Since we have a daily job, it seems not necessary  
Message System  
Apache Kafka

# RDBMS + NOSQL

Relational Database cannot handle high reading QPS  
NOSQL has a lot limit in relation and maintenance

So sometimes we just hybrid them together

# RDBMS + NOSQL

For common get queries, we store them in NOSQL

For specific queries and write operation, we still use RDBMS

How to update NOSQL? Database Listener

# Summary

Database: Relational and NoSql -> consider to use which one

Network: Apache, Tomcat, etc -> Rest API, JSON

Distributed System: Load balancing, Consistent Hash

Security: HTTPS, Encryption/Decryption