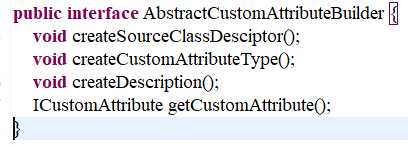
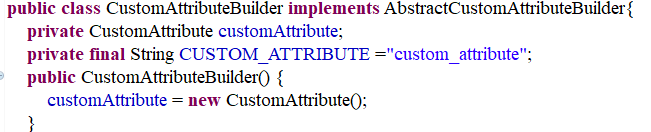
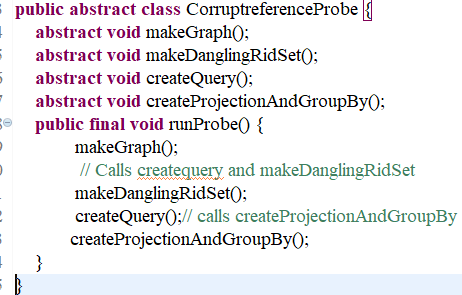
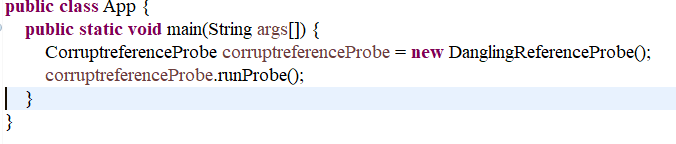
**Builder:** CustomAttribute



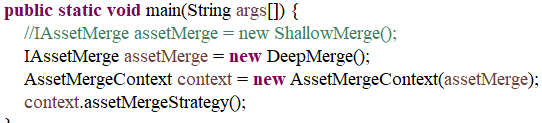


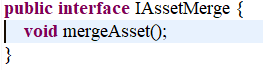
**template:** CorruptProbe

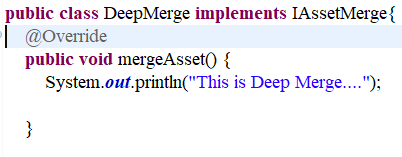


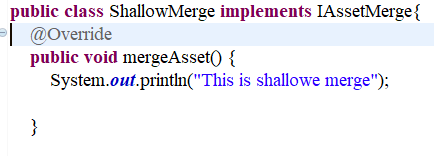


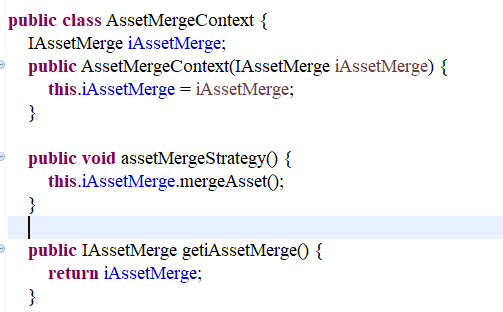
**Stretegy Design Pattern:**



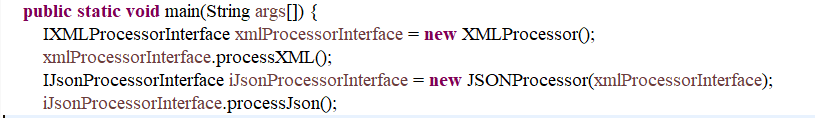


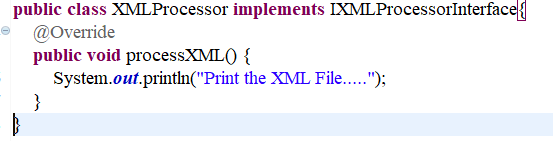


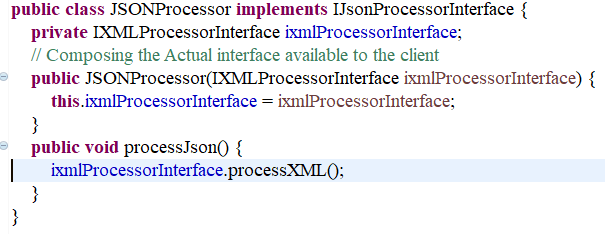




**Adapter Pattern**





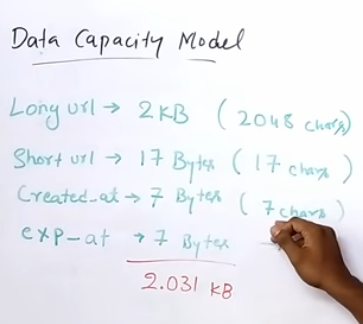


**System Design:**

**1.**

**Tiny URL**

1. Load of requests
2. Character of bytes for short url
3. 1 million user / month
4. Mapping between Actual Url and tiny URL
5. Prequeisite:
6. The size of data to be inserted into Database:
7. Called :DataCapacity Model

****

**Details to persist in database:**

**LongURL | ShortURL | created time(epoch) | expiry at**

**The above is entry of data per URL Shortener**

Now, to decide the database: DBMS or NOSQL need to understand the number of read/ write operations

The TinyURL have a ***domain.name/uniqueCode***

The Unique Code is generated:

Base 62 / Md5 Hash

MD5 takes string as input and might give outputs with collision

So use Base 62 and Base 10

As Base 62 has values from A\_ Z , a-z and 0-9

Hence, 62 pow 7 will give 3.5 trillion combinations of UniqueIds

**Database:**

RDBMS: ACID but not scalable as per month 30 Million read/write

NO SQL: Highly Available, scalable BUT *Eventual Consistency*

Use base 62 libraray to convert ramdomnumber to base 62

But, multiple APPServer might generate same unique ID so;

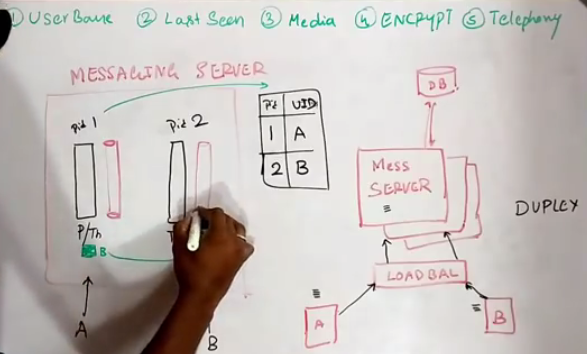
**2.**

**Google AutoSearch:**

Advantages of Cassandra/NOSQL over RDBMS

**3.**

**WhatsApp:**



**Inputs needed:**

1.Scale and User base

2. last Seen time need?

3. Support media messgaes?

4. Encryption needed?

5. Telephony

Load Balancer ------------------ > Messaging Server --------------🡪 DB(to store arbitrary statges)

Connection is **Duplex**(Bi-Directional)i.e. **TCP/UDP/Socket**

Message saved in mobile database(locally) when NOT yet sent to server

On message received server checks for the process that deals with the message, if B is NOT online, it remains saved in DB. Always Client establishes the connection and Never the Server.

**Acknowledgement: heartbeat ???**

**UniqueID used for each message**

A process/Thread is created in the Messaging Server and a corresponding queue associated with the process/Thread

In Database; will have table with Column: **PID** and **UID**

Thread keeps on listinging to the Connection with a particular Client

The Client A PID receiving the message and passes it to the Client B Thread’s queue which is further sent by client B thread to Client B

**If Client B is offline; ??? Than NO process will be running in the messaging server for Client B**

**How Client A knows that Client B is Offline? Ans: When It doesn’t find any process running corresponding to client B**

Last Seen: Based on Heart Beat Saved in table which perhaps was sent by Client

**Case4: Twitter**

**Features:**

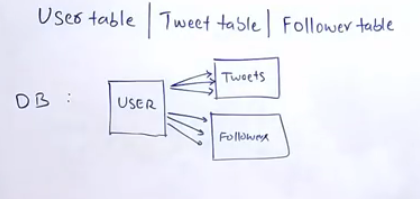
* **Tweet**
* **Timeline : Home timeline / User timeline / Search Timeline**
* **Trends**

Read heavy

**Eventual consistency**: consistency not major factor

Storage cost??

User Table | Tweets Table | Follower Table



Caching layer needed for a read heavy Twitter

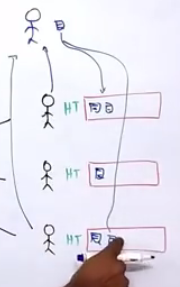
In redis the structure to be as :

User\_ID : USER\_treet IDS[1,2,3………….] as list

Tweet\_ID: User\_id

BottleNeck is multiple DB queries: Alternatively: NoSQL/Nodes

Data need to create the Hometimeline? Using FANOUT to push the tweet made by person to his followers



Get the tweets from the Cache rather than from Database based on User ID

how much latency is acceptable in database query?

How much fast does a Cache makes the operation as compared to DB query?

Also maintain HomeTimeLine in Redis Cache

Do we need to precompile the HT and UT for user who haven’t logged to their twitter account?

No,We can avoid the processing

**#Trends:**

*Streaming????*

Stream processing Framwork as **Apache Storm**

**Or Kakfa**

ZooKeeper??

Use cache against distributed DB / nodes using FANOUT

Redis a cache db??

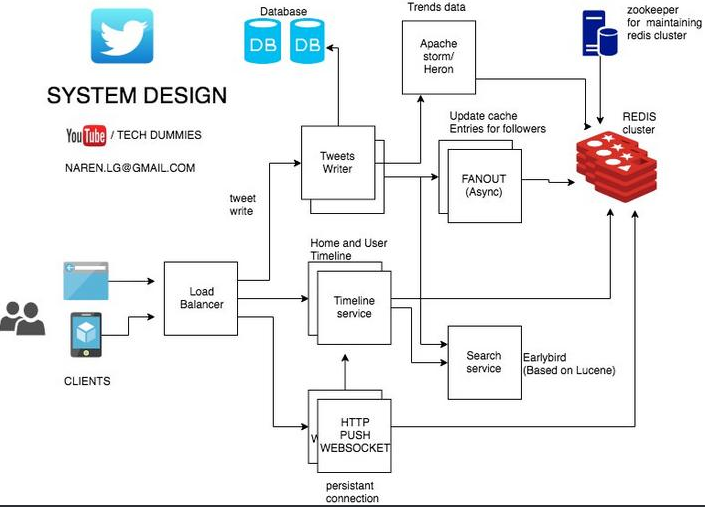
API are exposed form Redis to be consumed further

**In memory database?? How Redis maintains persistence of data when machine crashes**

Trends: volume of data/ **stream processing: storm or kafka**

Validation

Indexinf for seach/ lucene

****

**Netflix system design:**

* OpenConnect
* BackEnd
* Client

***CDN: Content Delivery network***

***Open Connect***

***Elastic Load Balancer***

***Transcoding/Encoding : to format the video as per client***

================================================================================ TO DO===============

https://www.nginx.com/blog/building-microservices-using-an-api-gateway/

Cache:

https://dzone.com/articles/introducing-amp-assimilating-caching-quick-read-fo

ormalization:

https://beginnersbook.com/2015/05/normalization-in-dbms/

1. Distributed database System
2. Cache : in memory and Database/ LRU

Memcache

1. Load Balancing

How is Client mapped in Server

Duplex Connection

1. Proxy Server
2. Scalability
3. Security
4. API Design
5. MicroServices

Distributed datastores

Scale Database:

1. Master-Slave Architecture

Write in master while Read from Slaves

-ve ; Not a consistent Database:CAP

2. To fix above; Database Sharding

-ve: Network call needed across two systems in case need join

1. Distributed database System
2. Cache : in memory and Database/ LRU

Memcache

1. Load Balancing

How is Client mapped in Server

Duplex Connection

1. Proxy Server
2. Scalability
3. Security
4. API Design
5. MicroServices

**GeeksForGeeks**

1. [**https://www.geeksforgeeks.org/snake-ladder-problem-2/**](https://www.geeksforgeeks.org/snake-ladder-problem-2/)
2. [**https://www.geeksforgeeks.org/implementation-of-tic-tac-toe-game/**](https://www.geeksforgeeks.org/implementation-of-tic-tac-toe-game/)

**Currently Working Project**

InfoSphere Information Governance Catalog: Used by Business Analysts for create and manage enterprise vocabulary and information governance practices

Build a common language between business and information technology

designated users can create glossary assets: terms, categories, information governance policies, and information governance rules. In addition, users can define relationships between the glossary assets that they create and other catalog assets

In addition to glossary assets, the catalog can contain metadata about information assets, which are assets other than glossary assets. Examples of information assets are implemented data resources, such as database tables and columns, ETL jobs, profiling processes, routines, and functions

These information assets typically come from other IBM InfoSphere Information Server components and are stored in the metadata repository of InfoSphere Information Server.

IBM Stewardship Center sends email notification to specific recipients as soon as a governance event such as creation, modification, import, or workflow change occurs to an asset

 Display a graphical view of asset relationships

 Track the history of changes to terms

Enhance existing metadata with descriptions and associations:

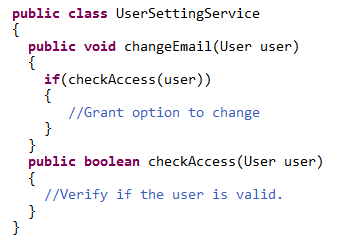
* Associate related assets, stewards, labels, and custom attributes to assets

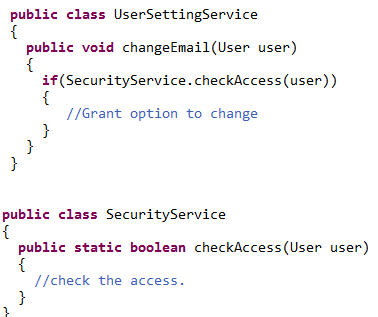
Common metadata assets are stored in the metadata repository and shared between tools in the IBM® InfoSphere® Information Server suite

include logical and physical data model assets, implemented data resources, and business intelligence (BI) assets.

**SOLID**

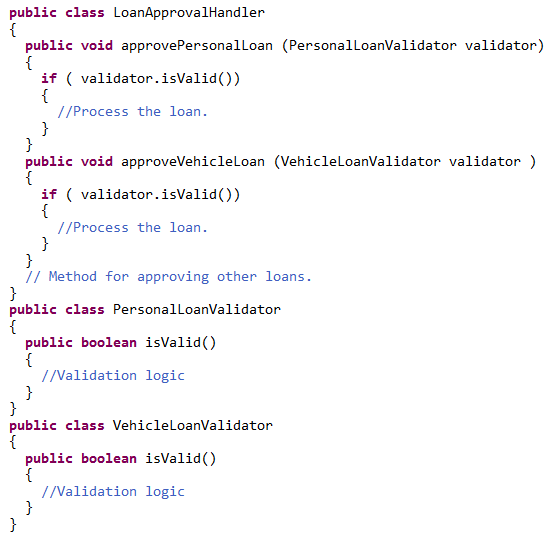
**S: Single Responsibility Principle**



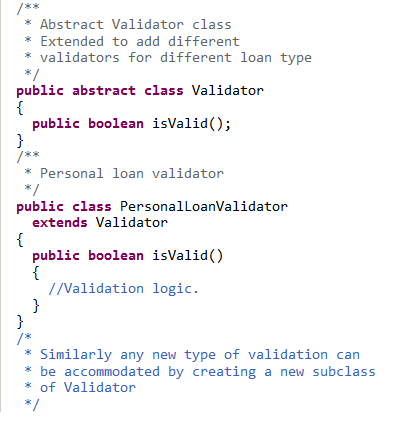


1. Open-Closed Principle:

**Wrong:**



**Correct**



1. Database serialization
2. Design DBMS for Hotel booking
3. Reverse linledlist
4. Dependency Injection

In progress:

State design pattern???????????? WorkFlowID

**Object Oriented Design:**

**Case1: ParkingLot:**

<https://www.educative.io/courses/grokking-the-object-oriented-design-interview/gxM3gRxmr8Z>

[Design a Library Management SystemPreview](https://www.educative.io/courses/grokking-the-object-oriented-design-interview/RMlM3NgjAyR)

[Design a Parking LotPreview](https://www.educative.io/courses/grokking-the-object-oriented-design-interview/gxM3gRxmr8Z)

[Design Amazon - Online Shopping System](https://www.educative.io/courses/grokking-the-object-oriented-design-interview/Bn8PMllro6Q/preview)

[Design Stack Overflow](https://www.educative.io/courses/grokking-the-object-oriented-design-interview/m2YWoEq06AR/preview)

[Design a Movie Ticket Booking System](https://www.educative.io/courses/grokking-the-object-oriented-design-interview/gxkvNgMqDk3/preview)

[**Design an ATM**](https://www.educative.io/courses/grokking-the-object-oriented-design-interview/m22LWKgQ4Wr/preview)

[**http://www.newthinktank.com/2012/12/object-oriented-design-8/**](http://www.newthinktank.com/2012/12/object-oriented-design-8/)

[Design an Airline Management System](https://www.educative.io/courses/grokking-the-object-oriented-design-interview/RMjqP6Vw98R/preview)

[Design Blackjack and a Deck of Cards](https://www.educative.io/courses/grokking-the-object-oriented-design-interview/YQ5gm2APRnp/preview)

[Design a Hotel Management System](https://www.educative.io/courses/grokking-the-object-oriented-design-interview/39Ek39vZBy9/preview)

<https://dba.stackexchange.com/questions/125162/database-design-of-room-booking-system-in-a-hotel>

[Design a Restaurant Management system](https://www.educative.io/courses/grokking-the-object-oriented-design-interview/xV8p1GA6K0r/preview)

[Design Chess](https://www.educative.io/courses/grokking-the-object-oriented-design-interview/JP7BXYkj3DK/preview)

[Design an Online Stock Brokerage System](https://www.educative.io/courses/grokking-the-object-oriented-design-interview/RM8ol3m2o3w/preview)

[Design a Car Rental System](https://www.educative.io/courses/grokking-the-object-oriented-design-interview/B8QoxDl6YON/preview)

[Design LinkedIn](https://www.educative.io/courses/grokking-the-object-oriented-design-interview/gxMOMDVKlBk/preview)

[Design Cricinfo](https://www.educative.io/courses/grokking-the-object-oriented-design-interview/YQ7lDLlNl5A/preview)

[Design Facebook - a social network](https://www.educative.io/courses/grokking-the-object-oriented-design-interview/7n94JNyPOMw/preview)

**TO DO:**

1. **CALLABLE JDBC to use Storedprocedure**
2. **Batch update**
3. **Which amd when to use jdbc driver**
4. **Thin vs thick driver**
5. **Execute(when query not known at beginning; at runtime: boolean) vs executeQuery: ResutSet vs executeUpdate: int vs executeBatch**

**(Mapper for resultSet)**

1. **CLOB and BLOB datatype save in DB**
2. **4 execute method:**
3. **Date in java: java.util (BOTH DATE AND TIME)and java.sql(for DB operations only)(ONLYSDATE)**

**To insert java.sql.Date in DB should use preparedstatement**

1. **Connection timeout value in DB ConnectionPool.**

**Hibernate and session tutorial:**

[**https://www.youtube.com/watch?v=TXAW6ndnSeY**](https://www.youtube.com/watch?v=TXAW6ndnSeY)

[**https://stackoverflow.com/questions/28486850/what-is-the-difference-between-a-session-and-a-connection-in-hibernate**](https://stackoverflow.com/questions/28486850/what-is-the-difference-between-a-session-and-a-connection-in-hibernate)

[**https://howtodoinjava.com/hibernate/hibernate-3-introduction-and-writing-hello-world-application/**](https://howtodoinjava.com/hibernate/hibernate-3-introduction-and-writing-hello-world-application/)

**TO DO:**

**HIBERNATE DONE TILL THE BASIC MULTIPLE SESSION; NOW PROCEEDING WITH datastructure implementation with UI**

**Implementtaion od DS:**

[**https://www.youtube.com/watch?v=ZNH0MuQ51m4**](https://www.youtube.com/watch?v=ZNH0MuQ51m4)

**Save image/ in database???DB2**

**Save image directly in database or save urlt o image on sysem?**

**Implement in memory cache for images**

**Why and when to use wrapper classes??**

**JSON Service for REST API**

[**https://crunchify.com/create-very-simple-jersey-rest-service-and-send-json-data-from-java-client/**](https://crunchify.com/create-very-simple-jersey-rest-service-and-send-json-data-from-java-client/)

**to find list of sequence:**

**select \* from syscat.sequences;**

**>>>>>>>>>>>>>>>>>>>>**

**Create web app using maven:**

mvn archetype:generate -DgroupId=com.sample.library -DartifactId=java-project -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

[**https://stackoverflow.com/questions/577659/real-world-examples-of-tree-structures**](https://stackoverflow.com/questions/577659/real-world-examples-of-tree-structures)