EXPERIMENT NO. 1

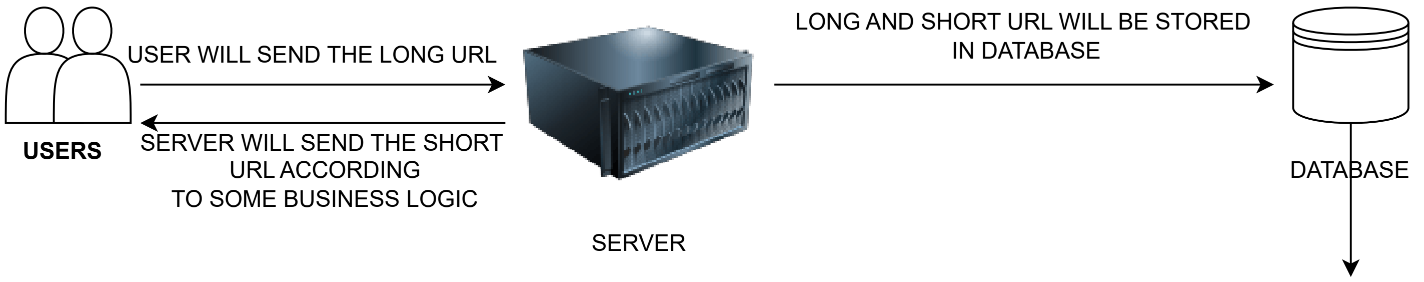
DESIGNING A URL SHORTENER :

FUNCTIONAL REQUIREMENTS :
1. GETTING THE LONG URL
2. REDIRECTING TO LONG URL
3. ENSURE UNIQUENESS
4. HANDLE URL VALIDATION

NON-FUNCTIONAL REQUIREMENTS:
1. PERFORMANCE
2. AVAILABILITY
3. SCALABILITY
4. SECURITY

HIGH LEVEL DESIGN :
SO ACCORDING TO THE REQUIREMENT WE CAN UNDERSTAND THAT THERE WILL BE A USER WHO WILL REQUEST TO THE API OR SERVER BY SENDING THE LONG URL AND REQUEST FOR SHORT URL

1. SHORT URL GENERATION

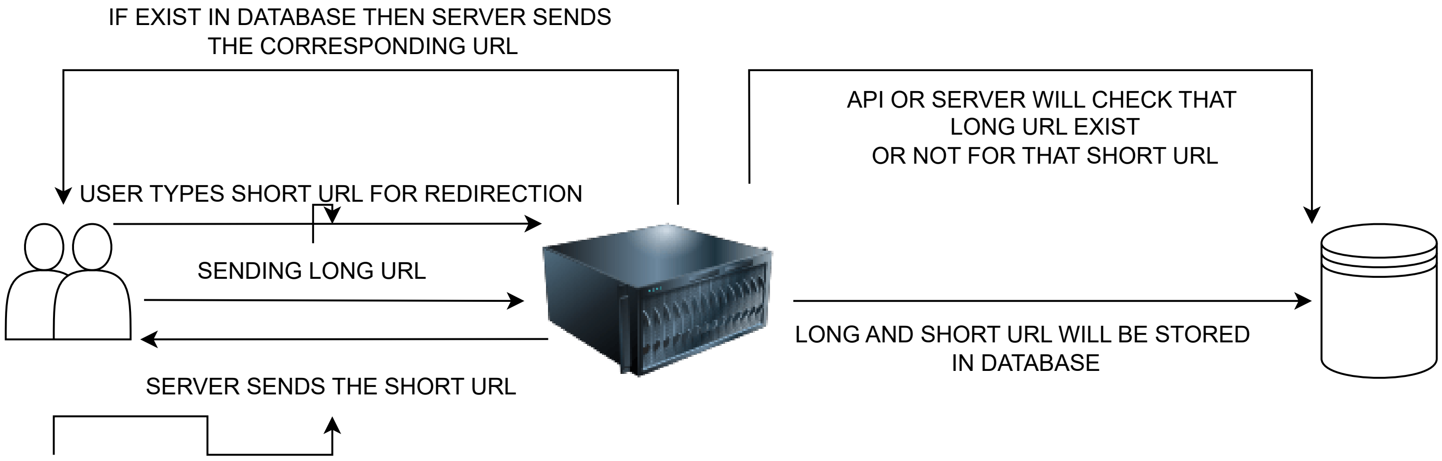


USER WILL SEND THE LONG URL

SERVER WILL SEND THE SHORT URL ACCORDING TO SOME BUSINESS LOGIC

LONG AND SHORT URL WILL BE STORED IN DATABASE

2. REDIRECTION:: WHEN USER ENTER THE SHORT URL..



IF EXIST IN DATABASE THEN SERVER SENDS THE CORRESPONDING URL

API OR SERVER WILL CHECK THAT LONG URL EXIST OR NOT FOR THAT SHORT URL

USER TYPES SHORT URL FOR REDIRECTION

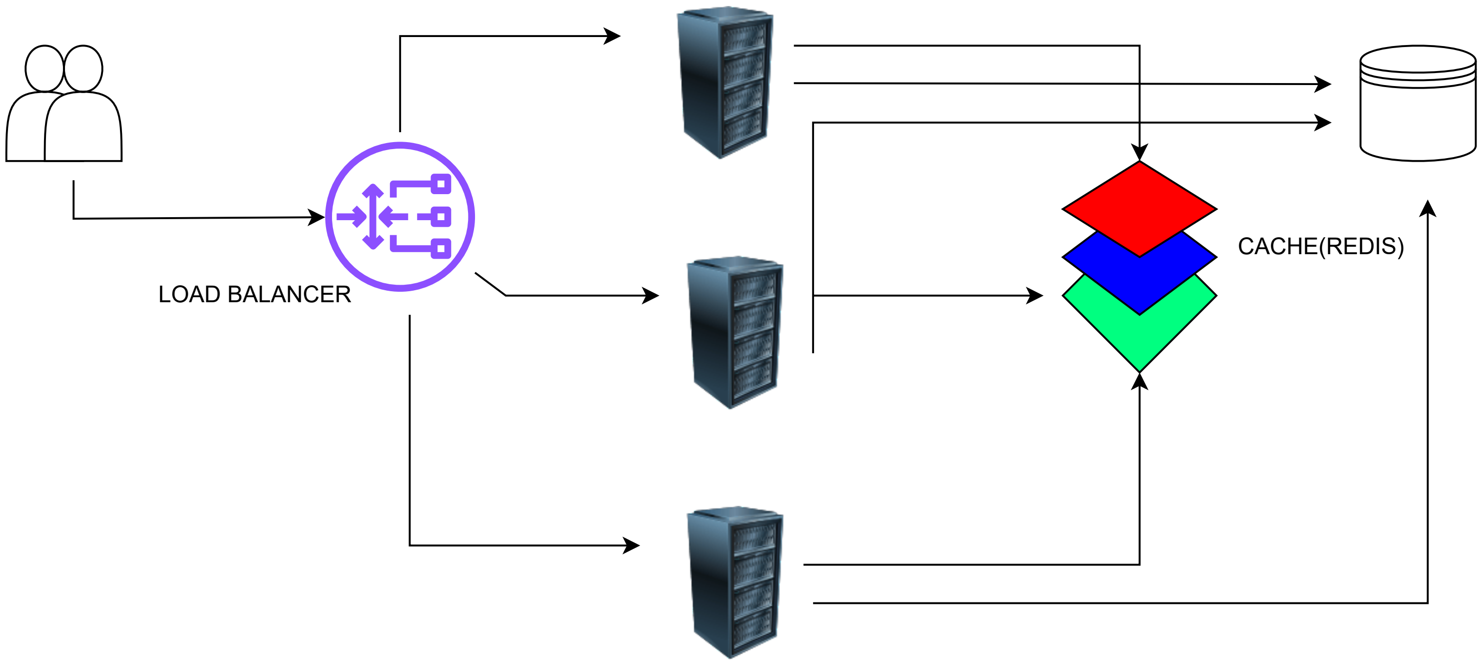
SENDING LONG URL

LONG AND SHORT URL WILL BE STORED IN DATABASE

SERVER SENDS THE SHORT URL

APPROACH 1 :

AS THERE IS ONE SERVER ONLY SO WHAT IF THIS SERVER CRASHES THEN OUR WHOLE SYSTEM WILL SHUT DOWN SO WE NEED TO ADD MORE SERVERS SO THAT OUR SYSTEM CAN WORK PROPERLY .



PROBLEM WE FACED :

Text

1. WE USED MORE THAN ONE SERVER SO THAT OUR SYSTEM CONTINUES EVEN IF OUR ONE SERVER CRASHES SO WE LOAD BALANCER SO THAT WE CAN DISTRIBUTE THE USERS OR THE LOAD

2. WE USED CACHE MEMORY BECAUSE EVERY TIME WE REDIRECT THE SHORT URL TO GET THE LONG URL THENEVERY TIME SAME QUERY EXECUTED THATS WHY HERE IN THE CACHE MEMORY LATEST OR VIRAL URL WILL BE THERE SO THAT LATENCY CAN BE DECREASED