Assignment Description:

• An Expense Management System for an organization to track expenses of all employees.

• A distributed systems web-application, where end-user sends a REST request to a proxy-server. The proxy server is connected to three back-end servers and three-cache machines.

Proxy-Server:

• The proxy server uses rendezvous hashing algorithm to choose a back-end server for a given request. Once a server is selected, the request is forwarded to that back-end server. The proxy server then waits for a response and forwards it to end-user.

• Similarly, the proxy server also selects a cache machine through a separate rendezvous hashing algorithm for a given request to store the response in that cache machine. Once a similar request is received in future, it can then be fetched from the cache machines.

• The proxy server keeps track of all running servers with the help of a key-value Redis Database.

Back-end server:

• Each back-end server is connected to individual databases to implement database sharding. Database sharding helps to optimize performance by faster query operations.

• To enhance availability, each server also stores a copy of the data in the next server. So, if a server goes down, data can be fetched from the next server.

Cache Machine:

• The cache machines help to reduce the load on back-end servers and respond to requests quickly.

• When a request is sent by a client, the proxy-server first hashes the request (In this example done on the basis of unique ID) and each cache machine, and chooses the server with maximum weight.

• Once cache machine is chosen, proxy server checks if a cached copy of response is available for that request.

* If yes, that copy is sent back to the end user.
* Else, the proxy server forwards the request to one of the back-end servers and waits for a response. Once the response is received it is stored in that cache machine and then the response is sent back to the ens-user.

Technologies:

* Server-side: Python Flask
* Database: SQLAlchemy, Redis



Fig: Architecture Diagram