**PROPOSED SYSTEM**

In this subsection, we describe this research’s objective in an illustrative diagram to explain the problem in Load Balancing and the role of the proposed LB algorithm, as seen in figure 3 below. This proposed model’s main goal is to provide efficient resource allocation in a cloud environment whereby it avoids unbalanced workload in Cloud Computing applications. This model resolves issues related to workload migration and task rejection in the cloud. The proposed framework consists of two layers:

• Top Layer: deals with requests from multiple different clients (application’s users) of both mobile and desktop. Clients can access the Internet using different devices to send requests to the cloud. In this layer, the model uses the Cloudlet Scheduler Time Shared algorithm to submit tasks in a random order (Arrival Time) and schedule them to Virtual Machines by considering two main parameters: Deadline and Completion Time. In Cloud Computing, Data Center (DC) can be described as big storage for cloud servers and data. DC receives requests and sends them to the active load balancer. In this layer of the model, the proposed algorithm is implemented as a Load Balancer, which acts as the primary balancer in the cloud environment to perform migration in the case of violation, which has not been addressed in the previous literature up to the author’s knowledge.

• Bottom Layer: deals with allocation of user requests to Virtual Machines (VMs). As the figure illustrates, we have a primary batch of VMs; VM2’s status is set to high priority since it violates the SLA requirement, which means its Completion Time is higher than the Deadline. Thus, the proposed LBA should apply a migration technique to transfer the workload to another available Virtual Machine by reconfiguring the MIPS of both VMs before and after allocating the resources to them. The allocation table is then updated whenever a Virtual Machine becomes violated or not, along with the number of requests it’s been allocated. There is a case where there is no SLA violation. Suppose the Time to Complete (TTC) is less than SLA (Deadline) given for tasks to run on VMs. Then, no SLA violation occurs.

Overall, the proposed framework supports dynamic scheduling and load balancing to fully utilize the CPU and fully the cloud resources.