

Q.1

a)

Currently sonic live company is located in UK only, and customers will deal with the company without any online platform to request equipment. If the company expanded worldwide, the data generation will be drastically increase and RDBMS will not be able to retrieve data quickly and up to date as per customer's requests. With the current RDBMS, it will be not able to support the current application for projected user volume growth. Further, it will require scaling the application where RDBMS cannot provide on demand scalability except scaling-up. Scaling-up is very expensive which means whenever business has grown enough to meet maximum server capacity, it is required to replace and upgrade server machines to fulfill more reads and writes.

SonicLive is also expect to access by mobile and web-based applications, so the company wants to engage the mobile and web customers in real time and will be required to add new features and demands after deployment. For that, company will require very agile system, which is able to process unstructured data easily, and system engagement need to be extremely dynamic. Therefore, it is recommend going for a NoSQL database rather than RDBMS. NoSQL support for dynamic requirement, where RDBMS slow to such changes since its rigid data structure. NoSQL can handle agile development. As NoSQL database can be proposed mongo dB and Cassandra.

Mongo is a document database and Cassandra is a column family database. MongoDB **support for ad hoc queries**, therefore user can search the equipment and other supports by using field, range query and this will support for regular expression searches. MongoDB support to **index** any field in document. MongoDB is a master slave replication. A replica set consists of two or more mongo DB instances. Two or more mongoDB instances consists in replica set. One replica is act as a master replica and others as slave replica (secondary) at any time. Master replica will interacts with clients and perform all the read and write operations. The slave

replicas will maintain a copy of the master using built in replication. Whenever master replica failed, a slave replica will become master slave automatically. So this will ensure the **high availability** of the database to the users.

MongoDB has the ability to run over multiple servers, therefore it can store data duplicating to keep system up and running whenever hardware failure encounter.

When handling large data sets, MongoDB uses **sharding** to support deployment and high throughput operations. MongoDB support horizontally scaling by using sharding over different servers. Even one or more shards are unavailable, a sharded cluster can do continues partial read and writes that help to achieve high availability. Sharding require careful planning, execution, and maintenance for gain better performances.

Load balancing is another key feature of mongoDB. It use sharing to scale horizontally by dividing data across multiple instances. By running over multiple servers MonogDB used to balancing the load and duplicating data to ensure up and running, the system whenever encounter a hardware failure.

When considering the disadvantages of MongoDB , it is not support to transactions. Though it is support for handle lots of data, relationship between them is weak. If the SonicLive willing to acquire online payment method then it is not suitable mongoDB for that. However, mongoDB version 4 is supported for ACID transactions. MonogDB is alos not supported to join queries.

b.

CAP theorem indicates, Consistency, Availability and Partition tolerance. Consistency mean all the servers in a system will have same data even different users access to different servers can read same data. Availability mean system will always responds to a given request. Partition tolerance mean system granted to operate as whole even if individual server goes down. In a distributed system, one can only have two out of C, A, or P and one of them will be not supported. MongoDB is a known as CP system where it sacrifices availability for the sake of consistency and partition tolerance.

As per the proposed architecture in part a, Sonic Live users will access the system for get the information about equipment and to check availability of required product. Therefore, it is important to make sure the high consistency and partition tolerance. Because system should able to provide up to date information to its users.

Replica set in mongoDB has one primary node, if primary goes down system become unavailable until new primary is selected. So system will stop accepting writes until it make sure it can safely complete those writes. This is called eventually consistency.

MonogDB has a master slave architecture, since it achieve high consistency. All the writes operations are done using master node and then slave nodes do the readings from master node.