بسم الله الرحمن الرحيم

**Mrsh Pay**

**Online Payement Solution**

**Database System Main Concepts:**

The database system will be built with regarding to the following key principles:

1. Scalability
2. No single point of faliure
3. Well Structured
4. Secured

**Scalability:**

refers to the capability of a system to handle a growing amount of work, or its potential to perform more total work in the same elapsed time when processing power is expanded to accommodate growth. A system is said to be scalable if it can increase its workload and throughput when additional resources are added.

There are two broad categories for scaling database systems: **vertical scaling** and **horizontal scaling.**

**Vertical scaling:**

Also known as scaling up, is the process of adding resources, such as memory or more powerful CPUs to an existing server. Removing memory or changing to a less powerful CPU is known as scaling down.

**Horizontal scaling:**

Sometimes referred to as scaling out, is the process of adding more hardware to a system. This typically means adding nodes (new servers) to an existing system. Doing the opposite, that is removing hardware, is known as scaling in.

**No single point of failure:**

A single point of failure (SPOF) is a system component which, upon failure, renders an entire system unavailable or unreliable. When you design a highly available deployment, you identify potential SPOFs and investigate how these SPOFs can be mitigated.

SPOFs can be divided into three categories:

* Hardware failures, for example, server crashes, network failures, power failures, or disk drive crashes
* Software failures, for example, Directory Server or Directory Proxy Server crashes
* Database corruption

## Mitigating SPOFs:

You can ensure that failure of a single component does not cause an entire directory service to fail by using **redundancy**. Redundancy involves providing redundant software components, hardware components, or both. Examples of this strategy include deploying multiple, replicated instances of Directory Server on separate hosts, or using redundant arrays of independent disks (RAID) for storage of Directory Server databases. Redundancy with replicated Directory Servers is the most efficient way to achieve high availability.

You can also use **clustering** to provide a highly available service. Clustering involves providing pre-packaged high availability hardware and software. An example of this strategy is deploying Sun Cluster hardware and software.

**Will Structured – Secured [ Will be discussed later]**

**Database Schema:**

The database schema is consisted with the following tables:

**Country Table:**

which contains the following fields:

1. Id
2. Name
3. Name\_Abbreviation

**State Table:**

which contains the following fields:

1. Id
2. Name
3. County\_Id (Foreign Key)

**City Table:**

which contain the following fields:

1. Id
2. Name
3. Sate\_Id (Foreign Key)

**Currency Table:**

which contain the following fields:

1. Id
2. Name
3. Currency Symbol
4. Country\_Id (Foreign Key)

**User Table:**

which contains the following fields:

1. Id
2. Full\_Name
3. Username
4. Email
5. Password
6. Gender
7. Phone\_Number
8. City\_Id (Foreign Key)
9. Street\_Address
10. Type

**Bank Table:**

which contains the following fields:

1. Id
2. Name
3. City\_Id (Foreign Key)

**Bank Account Table:**

which contains the following fields:

1. Id
2. Bank\_Id (Foreign Key)
3. Number
4. Issue\_Date
5. Issue\_Time
6. Balance

**Wallet Table:**

which contains the following fields:

1. Id
2. User\_Id (Foreign Key)
3. Bank\_Account\_Id (Foreign Key)
4. Code
5. Issue\_Date
6. Issue\_Time
7. Balance
8. Id\_Number
9. qr\_code

**Card Table:**

which contains the following fields:

1. Id
2. Name
3. User\_Id (Foreign Key)
4. Code
5. Issue\_Date
6. Issue\_Time
7. Balance

**Service Table:**

which contains the following fields:

1. Id
2. Name
3. Service\_Type
4. Account\_id

**Wallet To Wallet Transaction Table:**

which contains the following fields:

1. Id
2. Sender\_Wallet (Foreign Key)
3. Receiver\_Wallet (Foreign Key)
4. Service\_Fee
5. Amount\_Transferred
6. Date
7. Time

**Card To Card Transaction Table:**

which contains the following fields:

1. Id
2. Sender\_Card (Foreign Key)
3. Receiver\_Card(Foreign Key)
4. Service\_Fee
5. Amount\_Transferred
6. Date
7. Time

**Card To Wallet Transaction Table:**

which contains the following fields:

1. Id
2. Sender\_Card (Foreign Key)
3. Receiver\_Wallet (Foreign Key)
4. Service\_Fee
5. Amount\_Transferred
6. Date
7. Time

**Wallet To Card Transaction Table:**

which contains the following fields:

1. Id
2. Sender\_Wallet (Foreign Key)
3. Receiver\_Card (Foreign Key)
4. Service\_Fee
5. Amount\_Transferred
6. Date
7. Time

**Wallet To Service Transaction Table:**

which contains the following fields:

1. Id
2. Sender\_Wallet (Foreign Key)
3. Service\_Id (Foreign Key)
4. Invoice\_Number
5. Amount\_Transferred
6. Service\_Fee
7. Date
8. Time

**Card To Service Transaction Table:**

which contains the following fields:

1. Id
2. Sender\_Card (Foreign Key)
3. Service\_Id (Foreign Key)
4. Invoice\_Number
5. Amount\_Transferred
6. Date
7. Time

**\* Ticket Table:**

which contains the following fields:

1. Id
2. Name
3. Qr\_Code
4. Quantity
5. Price
6. User\_Id (Foreign Key)
7. Issue\_Date
8. Issue\_Time
9. Description
10. Valid\_To\_Date
11. Valid\_To\_Time

**\* Wallet To Ticket Transaction Table:**

which contains the following fields:

1. Id
2. Wallet\_Id (Foreign Key)
3. Ticket\_Id (Foreign Key)
4. Transferred\_Amount
5. Service\_Fee
6. Quantity
7. Date
8. Time

**\* Card To Ticket Transaction Table:**

which contains the following fields:

1. Id
2. Card\_Id (Foreign Key)
3. Ticket\_Id (Foreign Key)
4. Transferred\_Amount
5. Service\_Fee
6. Quantity
7. Date
8. Time

**Notification Table:**

which contain the following columns:

1. Id
2. Title
3. Content
4. Date
5. Time

**\* Tables marked with this sign are not yet implemented in designs**