**Design Assignments:**

**Class Design:** Please find the below functional declaration and Datatypes for the given design

**Equipment Class:**

public class Equipment

{

public int Id { get; set; }

public string Type { get; set; }

public Equipment(int id, string type);

public static List<Equipment> GetAllEquipments();

public void SaveToDatabase();

}

**Machine Event class:**

public class MachineEvent

{

public int Id { get; set; }

public DateTime Timestamp { get; set; }

public string EventType { get; set; }

public int EquipmentId { get; set; }

public MachineEvent(int id, DateTime timestamp, string eventType, int equipmentId);

public static MachineEvent GetEventById(int eventId);

public void SaveToDatabase();

}

**Database Design:**

Based on the Data requirements we need below tables

1. **Equipment Table:** The Equipment table stores information about different types of equipment.

**Table 1: Equipment**

Columns:

* EquipmentId (Primary Key)
* Type

1. **Events Table:** The Events table stores machine events with a foreign key reference to the corresponding equipment.

**Table 2: Events**

Columns:

* EventId (Primary Key)
* Timestamp
* EventType
* EquipmentId (Foreign Key referencing Equipment.EquipmentId)

1. **BinInfoTypeA:** This table store bin information specific to equipment typeA.

**Table 3: BinInfoTypeA (For Equipment Type A)**

Columns:

* BinId (Primary Key)
* Color

1. **BinInfoTypeB:** This table store bin information specific to equipment typeB.

**Table 4: BinInfoTypeB (For Equipment Type B)**

Columns:

* BinId (Primary Key)
* Color

1. **Equipment Bin Mapping:** The EquipmentBinMapping table maps equipment to its corresponding bins, allowing flexibility for different equipment types.

**Table 5: EquipmentBinMapping**

Columns:

* EquipmentId (Foreign Key referencing Equipment.EquipmentId)
* BinId (Foreign Key referencing BinInfoTypeA.BinId or BinInfoTypeB.BinId based on equipment type)

This design allows for normalization by separating equipment, events, and bin information into distinct tables. Each table serves a specific purpose and can be independently managed and queried. I believe this design also ensures data integrity and makes it easier to manage and query the database, facilitating reporting and analysis of machine performance.