

Program.cs.....

```
using System;
using StoresScheduleSystemBLL;

namespace StoresScheduleSystemConsoleUi
{
    class Program
    {
        static void Main(string[] args)
        {
            string Continue = "y";
            StoreScheduler StoreScheduler = new StoreScheduler();
            if (Continue.ToLower() == "n")
            {
                System.Environment.Exit(1);
            }
            else
            {
                while (Continue.ToLower() == "y")
                {
                    Console.WriteLine("-----");
                    Console.WriteLine("===== GHMC Stores Scheduler =====");
                    Console.WriteLine("-----");
                    try
                    {
                        FindAndUpdateStore(StoreScheduler);
                    }
                    catch (Exception ex)
                    {
                        Console.WriteLine("Error : " + ex.Message);
                    }
                    finally
                    {
                        Console.WriteLine();
                        Console.Write("Do you want to continue for another store(y/n) :");
                        Continue = Console.ReadLine();
                    }
                }
            }
        }
    }
}
```

```

private static void FindAndUpdateStore(StoreScheduler StoreScheduler)
{

    StoreScheduler storeScheduler = new StoreScheduler();
    Console.WriteLine("Enter store id");
    int id = Convert.ToInt32(Console.ReadLine());

    Store store = storeScheduler.GetStoreById(id);
    if (store.StoreId == id)
    {
        Console.WriteLine("Found a store with following details");
        Console.WriteLine("StoreName : " + store.StoreName);
        Console.WriteLine("OwnerName : " + store.OwnerName);
        Console.WriteLine("MobileNo : " + store.MobileNo);
        Console.WriteLine("StoreAddress : " + store.StoreAddress);
        Console.WriteLine("SellsEssentials : " + store.SellsEssentials);
        Console.WriteLine("OpeningTime : " + store.OpeningTime);
        Console.WriteLine("ClosingTime : " + store.ClosingTime);
        Console.WriteLine("Enter the Timeslot assigned to store (TimeSlotA/TimeSlotB)");
        try
        {
            string timeslot = Console.ReadLine();

            storeScheduler.AssignTimings(store, timeslot);
            storeScheduler.UpdateStoreTimings(store, timeslot);

            Console.WriteLine("Time slot updated for the store {0}", store.StoreName);

        }
        catch (Exception e)
        {
            Console.WriteLine(e.Message);
        }
    }
    else
    {
        Console.WriteLine("No store found for store id : {0}", id);
    }
}
}

```

```
}
```

Store.cs.....

.....

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace StoresScheduleSystemBLL
{
    public class Store
    {
        public int StoreId { get; set; }
        public string StoreName { get; set; }
        public string OwnerName { get; set; }
        public string MobileNo { get; set; }
        public string StoreAddress { get; set; }
        public bool SellsEssentials { get; set; }
        public string OpeningTime { get; set; }
        public string ClosingTime { get; set; }
    }
}
```

StoreScheduler.cs.....

.....

```
using System;
using System.Data;
using StoresScheduleSystemDAL;

namespace StoresScheduleSystemBLL
{
    public class StoreScheduler
    {
        StoresScheduleDAO StoresScheduleDAO = new StoresScheduleDAO();
        public StoreScheduler()
        {
            // Instantiate fields here
        }
        public Store AssignTimings(Store store, string timeSlot)
        {
            TimeSlotA timeSlotA = new TimeSlotA();
        }
    }
}
```

```

TimeSlotB timeSlotB = new TimeSlotB();
if (timeSlot == "timeSlotA" || timeSlot=="TimeSlotA")
{
    timeSlotA.SetIsEssentialItemsStore(Convert.ToBoolean(store.SellsEssentials));
    store.OpeningTime = timeSlotA.GetOpeningTime();
    store.ClosingTime = timeSlotA.GetClosingTime();
}
else if(timeSlot == "timeSlotB" || timeSlot == "TimeSlotB")
{
    timeSlotB.SetIsEssentialItemsStore(Convert.ToBoolean(store.SellsEssentials));
    store.OpeningTime = timeSlotB.GetOpeningTime();
    store.ClosingTime = timeSlotB.GetClosingTime();
}
else
{
    throw new ArgumentException("Error : Invalid time slot");
}

return store;
}

public Store GetStoreById(int storeId)
{
    Store store = new Store();
    DataTable dt = StoresScheduleDAO.FindStore(storeId);
    if(dt!=null && dt.Rows.Count>0)
    {
        store.MobileNo = dt.Rows[0]["MobileNo"].ToString();
        store.StoreName = dt.Rows[0]["StoreName"].ToString();
        store.OwnerName = dt.Rows[0]["OwnerName"].ToString();
        store.StoreId = Convert.ToInt32(dt.Rows[0]["StoreId"]);
        store.StoreAddress = dt.Rows[0]["StoreAddress"].ToString();
        store.SellsEssentials = Convert.ToBoolean(dt.Rows[0]["SellsEssentials"]);
        store.OpeningTime = dt.Rows[0]["OpeningTime"].ToString();
        store.ClosingTime = dt.Rows[0]["ClosingTime"].ToString();

    }
    else
    {
        return null;
    }

    return store;
}

```

```

    }

    public bool UpdateStoreTimings(Store store, string timeSlot)
    {
        bool IsUpdated = false;
        // Donot change method signature
        // Implement code here

        TimeSlotA timeSlotA = new TimeSlotA();
        TimeSlotB timeSlotB = new TimeSlotB();
        if (timeSlot == "timeSlotA" || timeSlot=="TimeSlotA")
        {
            timeSlotA.SetIsEssentialItemsStore(Convert.ToBoolean(store.SellsEssentials));
            store.OpeningTime = timeSlotA.GetOpeningTime();
            store.ClosingTime = timeSlotA.GetClosingTime();
        }
        else if(timeSlot == "timeSlotB" || timeSlot == "TimeSlotB")
        {
            timeSlotB.SetIsEssentialItemsStore(Convert.ToBoolean(store.SellsEssentials));
            store.OpeningTime = timeSlotB.GetOpeningTime();
            store.ClosingTime = timeSlotB.GetClosingTime();
        }
        else
        {
            throw new Exception("Error : Invalid time slot");
        }
        StoresScheduleDAO storesScheduleDAO = new StoresScheduleDAO();

        int a = storesScheduleDAO.UpdateStore(store.StoreId, store.OpeningTime,
store.ClosingTime);
        if (a > 0)
            IsUpdated = true;

        return IsUpdated;
    }
}

```

```

StoresScheduleDAO.cs.....
.....
using System;
using System.Collections.Generic;
using System.Data;

```

```

using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Configuration;

namespace StoresScheduleSystemDAL
{
    public class StoresScheduleDAO
    {
        public SqlConnection connection;
        public SqlCommand command;
        public SqlDataAdapter adapter;
        public string connstring =
ConfigurationManager.ConnectionStrings["StoresConnection"].ConnectionString;

        public StoresScheduleDAO()
        {
            // Instantiate fields here
        }

        public DataTable FindStore(int storeId)
        {
            DataTable table = new DataTable();

            connection = new SqlConnection(connstring);
            string query = "select * from Stores where storeId= " + storeId;

            connection.Open();

            adapter = new SqlDataAdapter(query, connection);

            adapter.Fill(table);

            connection.Close();

            return table;
        }

        public int UpdateStore(int storeId, string openingTime, string closingTime)
        {

```

```

        int RowsAffected;

        connection = new SqlConnection(connstring);
        string query = "update dbo.Stores set OpeningTime= " + openingTime + "
,ClosingTime= " + closingTime + " where StoreId= " + storeId;
        connection.Open();
        command = new SqlCommand(query, connection);
        RowsAffected = command.ExecuteNonQuery();
        connection.Close();

        return RowsAffected;
    }
}

```

TimeSlot.cs.....

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace StoresScheduleSystemBLL
{
    public class TimeSlot
    {
        protected bool SellsEssentialItems;

        public void SetIsEssentialItemsStore(bool isEssentialItemsStore)
        {
            SellsEssentialItems = isEssentialItemsStore;
        }

        public virtual string GetOpeningTime()
        {
            return "08:00 AM";
        }

        public virtual string GetClosingTime()

```

```

        {
            return "08:00 PM";
        }
    }
}
TimeSlotA.cs.....
.....

```

```

using System;

```

```

namespace StoresScheduleSystemBLL

```

```

{
    public class TimeSlotA:TimeSlot
    {

        public override string GetOpeningTime()
        {
            // Donot change method signature
            // Implement code here

            if (SellsEssentialItems)
            {
                return "08:00 AM";
            }
            else
            {
                return "10:00 AM";
            }

        }

        public override string GetClosingTime()
        {

            if (SellsEssentialItems)
            {

                return "02:00 PM";
            }
            else
            {
                return "02:00 PM";
            }
        }
    }
}

```



```

    }

    }
}

TimeSlotB.cs.....
.....

using System;

namespace StoresScheduleSystemBLL
{
    public class TimeSlotB:TimeSlot
    {

        public override string GetOpeningTime()
        {

            if (SellsEssentialItems)
            {
                return "02:00 PM";
            }
            else
            {
                return "04:00 PM";
            }

        }

        public override string GetClosingTime()
        {

            if (SellsEssentialItems)
            {
                return "08:00 PM";
            }
            else
            {
                return "08:00 PM";
            }

        }

    }
}

```

}  
}