

Servicing

View Models

```
public class CustomerView
{
    public int CustomerID { get; set; }
    public string Name {get; set;}
    public string Address {get; set;}
    public string Phone {get; set;}
}
```

csharp



```
public class CustomerVehicleView
{
    public string VehicleIdentificationNumber { get; set; }
    public string MakeModel {get; set;}
}
```

csharp



```
public class CategoryView
{
    public int CategoryID { get; set; }
    public string Description {get; set;}
}
```

csharp



No model class, individual parameters

```
public int GetCoupon(string coupons)
```

csharp



```
public class PartView
{
    public int PartID { get; set; }
    public string Description { get; set; }
}
```

csharp



```
public class JobView
{
    public int EmployeeId { get; set; }
    public decimal ShopRate { get; set; }
    public string VehicleIdentificationNumber { get; set; }
    public decimal TaxAmount { get; set; }
    public decimal SubTotal { get; set; }
}
```

```
public class JobDetailView
{
    public string Description { get; set; }
    public decimal JobHour { get; set; }
    public string Comment { get; set; }
    public int CouponId { get; set; }
    public int DiscountPercent { get; set; }
}
```

Business Processing Requirements

Query Service Methods


List of customer

```
public List<CustomerView> Service_GetCustomer(string lastName)
{
}
```

- customer name submitted
 - no
 - throw NullArgumentException (no name provided)
- customers found
 - no
 - throw NullArgumentException (no customer found)

List of customer vehicle


```
public List<CustomerVehicleView> GetCustomerVehicle(int customerID)
{
}
```

csharp 

- customer has a vehicle
 - no
 - throw NullArgumentException (no vehicle provided)

List of standard services


```
public List<string> Service_GetStandardService()
{
}
```

csharp 

- filter by Servicing

Get coupon amount

```
public int GetCoupon(string coupons)
{
}
```

csharp 

- check that coupon is valid
 - no
 - return zero (0)
 - yes
 - return discount amount

Command Methods

Register Job

```
public void RegisterJob(JobView job, List<JobDetailView> jobDetails)
{

}
```

- job details count is greater than zero (0)
 - no
 - throw NullArgumentException (no job details)
- foreach jobDetail in jobDetails
 - Service is not empty
 - errorList.Add(new Exception(\$"Service description is needed"));
 - Quantity is greater than zero (0)
 - errorList.Add(new Exception(\$"Hours for job detail () must have a positive value"));
- check for errors
 - Yes
 - throw the list of all collected exceptions
 - no
 - save all work to the database
 - create record in Job
 - set JobDateIn to today
 - update all fields
 - create records in JobDetails
 - update all fields
- **Disable Register Job Button**

Form Methods

Add Standard Service

```
public void AddStandardService(string standardService)
{

}
```

- Add data shown in the standard service drop-down into the job details
 - Add coupon discount value to job detail
 - clear coupon???
-

Add Service

csharp 

```
public void AddService(string service)
{

}
```

- is the field empty
 - yes
 - throw NullArgumentException (Service value cannot be empty)
 - no
 - Add service into the job details
 - clear coupon???
-

RemoveService

csharp 

```
public void Remove(string service)
{

}
```

Reset Service

csharp 

```
public void Reset()
{

}
```

- clear service text box

Clear

csharp 

```
public void Clear()  
{  
  
}
```

- clear all values