# Contact Information

## Structure

* Database - SQL Server
* Rest Services – Web API
* Front end – MVC

We have following layers in application.

1. MVC
2. ReadModel
3. BusinessService
4. DataService

## Layer Description

### Data Service

Contains DbContext which used code first to generate database

Data Service itself has 2 layers in it. One part is DbContext (Data representation) and another is Generic Repository. We have hide our DBContext from external interactions and so that we have created this generic repository. One more advantage to use repository as generic, is we can reduce code writing as we can share common functionality.

### Business Service

Business Service reads data from repository and sends to MVC layer. And vice versa.

We have used Rest sharp for reading data from DataService.

### MVC

Used for User interface and get data from Business Service and sends to UI and vice versa.

## Technologies Used

1. Entity Framework,
2. Code First,
3. Attribute based routing,
4. Swagger,
5. Dependency Injection,
6. Generic Repository,
7. Newtonsoft.Json,
8. RestSharp RestClient,
9. Data Annotation,
10. Web API

## General Description

We have used RestSharp RestClient for Communication from MVC to BusinessService and BusinessService to DataService. Common code for rest calls is kept on one common location so that we don’t need to write same code again and again.

We haven't created any test project instead of that we have used swagger for some testing.

we just have to add "/Swagger" after url of BusinessService and DataService.

e.g. if url is <http://localhost:49825> then swagger url is <http://localhost:49825/swagger>

## How to install or Debug

### Debug

if you want to chekc code or debug code

1. Open solution in visual studio
2. Change DataService Appsetting Connection string for right database instance
3. Open Package manager console
4. Select DataService from dropdown and insert following command

“update-database -verbose”

1. While -verbose is optional part. it gives information what SQL is fired on database to create required data
2. Right click on solution and select set startup project and select MVC, BusinessService and DataService as startup projects.
3. We have business service and Data service url's in MVC and BusinessService web.config respectively. If you want to change that then please change port in respective project as well.
4. Now run application.

### Deploy

1. Deploy MVC, Business Service, and Data Service on three different port
2. If Port has different values than what we have in web.configs then modify web.configs
3. Change Data Service Appsetting Connection string to point to the right database
4. Launch application

## Application Flow

On landing page we can see customer list, which has following functionality.

### Customer

1. Create New - Create new customer
2. Edit - For Editing Customer information
3. Details - Where you can see details of Customer and his Contact information (Contacts)
4. Delete - For Deleting Cutsomer (when you delete customer it will delete his contacts as well)

### Contacts

We can see contacts on Customer details page. Where we can perform following operation.

1. Create New - Create new contact
2. Edit - For Editing contact information
3. Details - Where you can see details of COntact
4. Delete - For Deleting Contact

### Contact Type

AS I have running out of time I haven't created user interface for create/update/Delete this.

(If you want to add more contact types then please add it in DataService's Migration -> Configuration.cs’s seed method. And use above update command with same Data Service selection in dropdown.

**Note**: I haven’t used Ajax calls so all api request are synchronous. We can use ajax calls as well to run application asynchronously.