**Application Version: 1.0**

1. **Introduction**

This application searches a record in the database depending on the given input. If the record doesn’t exist, an appropriate message would be displayed.

It also has facility to seed the database or add individual records into the database. When the user selects Seed Database option, 10 records will be added into the database.

All the fields are compulsory while adding individual record.

1. **Installation Steps**
2. Open the solution file using Visual Studio 2012 or higher versions
3. Navigate to the *App.Config* file present in both the projects viz. *PersonSearch* and *UnitTest* and change the Server Name in the following lines (DataSource). Also, if possible, please use Windows Authentication(marked in grey color) while creating the server:

<defaultConnectionFactory type="System.Data.Entity.Infrastructure.SqlConnectionFactory, EntityFramework">

<parameters>

<parameter value="Data Source=KRSNA\PERSONSERVER; Initial Catalog=PersonDB; Integrated Security=True;" />

</parameters>

</defaultConnectionFactory>

and

<connectionStrings>

<add name="PersonContextContainer"

providerName="System.Data.SqlClient"

connectionString="Data Source=KRSNA\PERSONSERVER; Initial Catalog=PersonDB; Integrated Security=True; MultipleActiveResultSets=True;App=EntityFramework"/>

</connectionStrings>

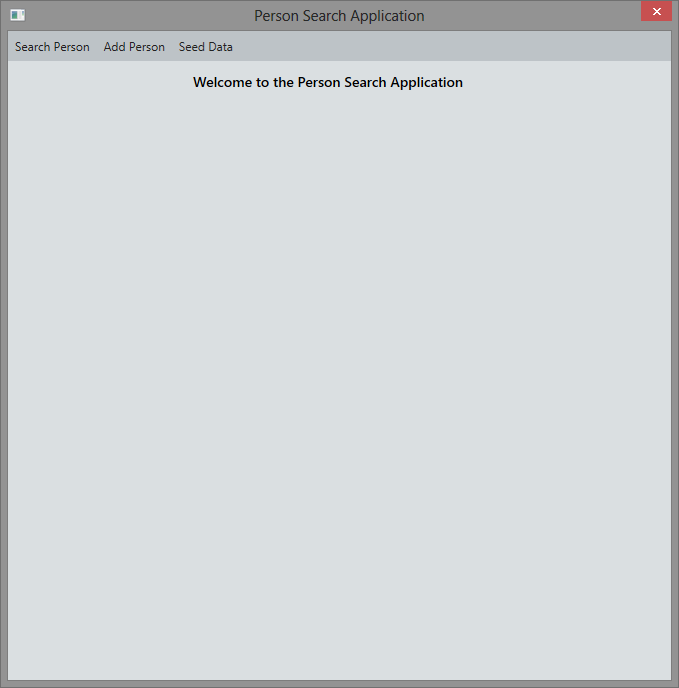
Now the application can be run.

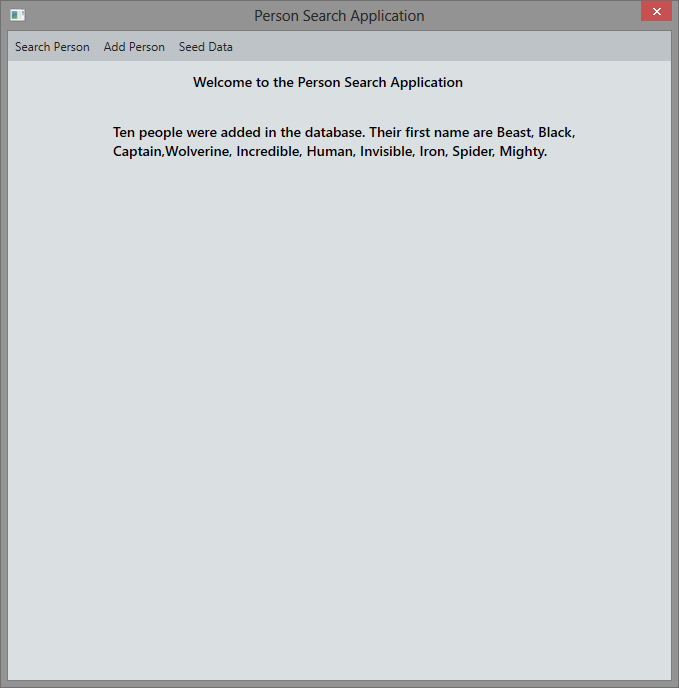
1. **Technical Details**

This application has been developed using Visual Studio 2012 and SQL Server 2008 on a 64 bit machine. It uses Entity Framework Code First Model and MVVM Pattern. It also contains a Unit Test Project which can be run to test the functionalities of the application. Please refer *section 7* for more details on running the Unit Tests.

1. **Seed Data**

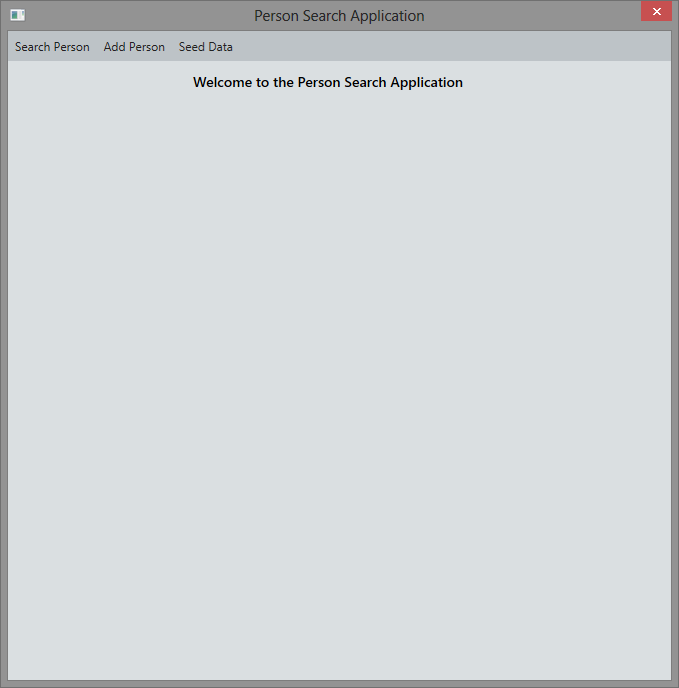
User can seed data by adding 10 records into the database. Click on *Seed Data* Menu button. This will add 10 records in the database and appropriate success message would be displayed. The steps can be displayed as follows:

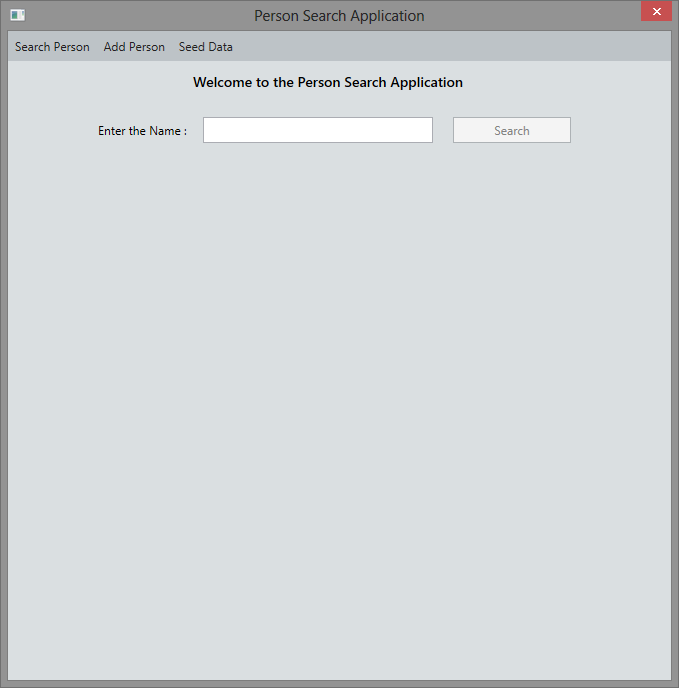




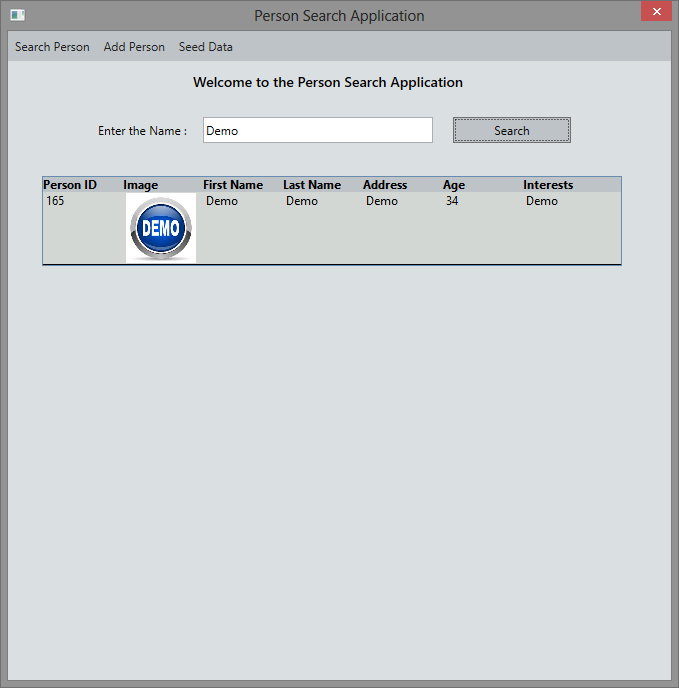
1. **Search Person**

The user can search for a record by clicking on the *Search Person* menu button. A textbox would be displayed for accepting the input. The Search button would be disabled till user enters some data into the textbox.

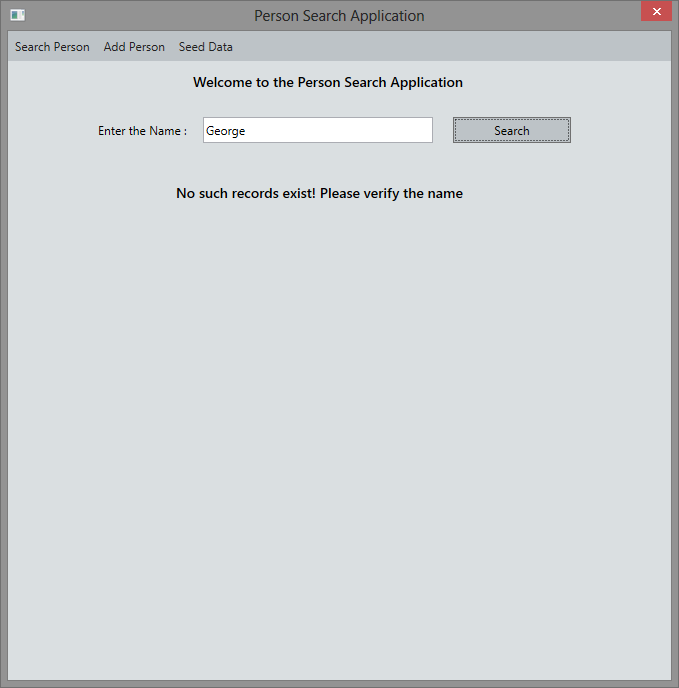




A grid containing corresponding list of People would be displayed.

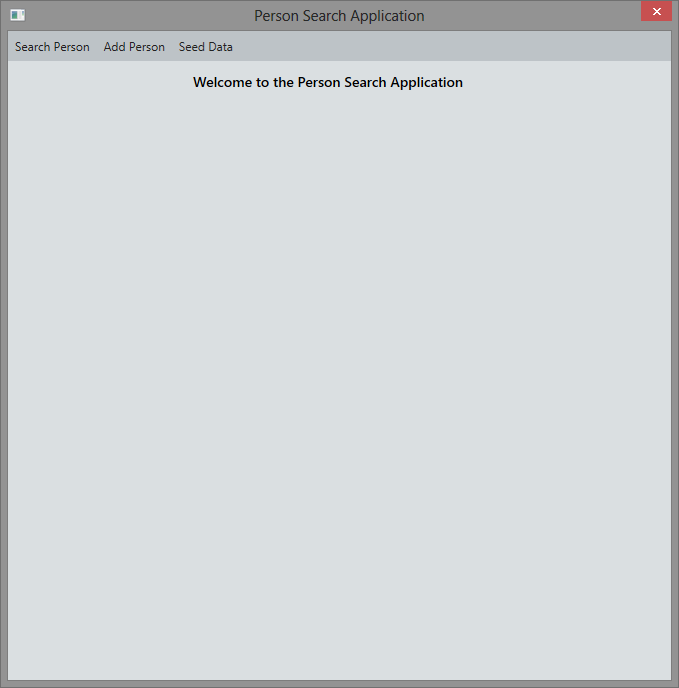


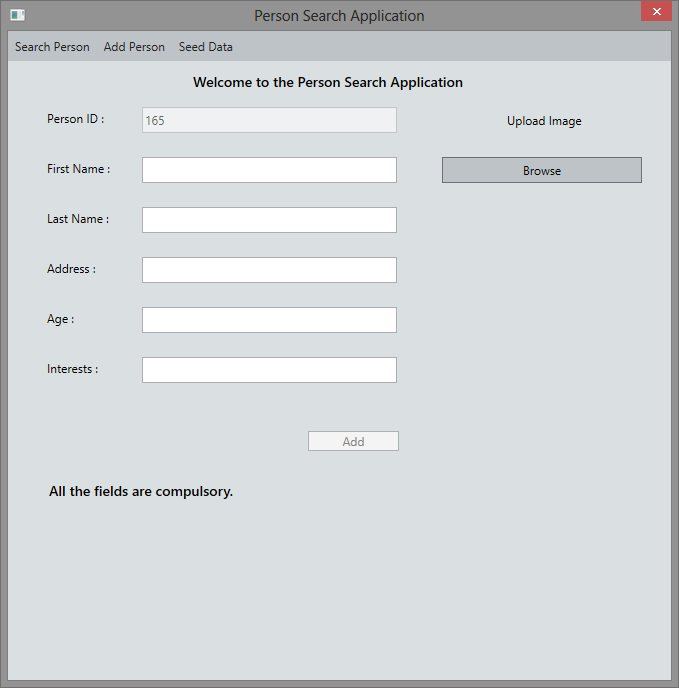
If no records exist, then appropriate message would be displayed

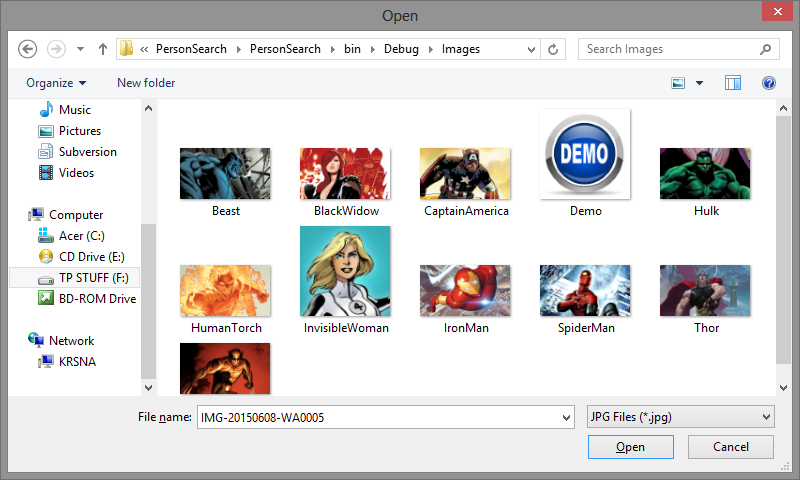


1. **Add Person**

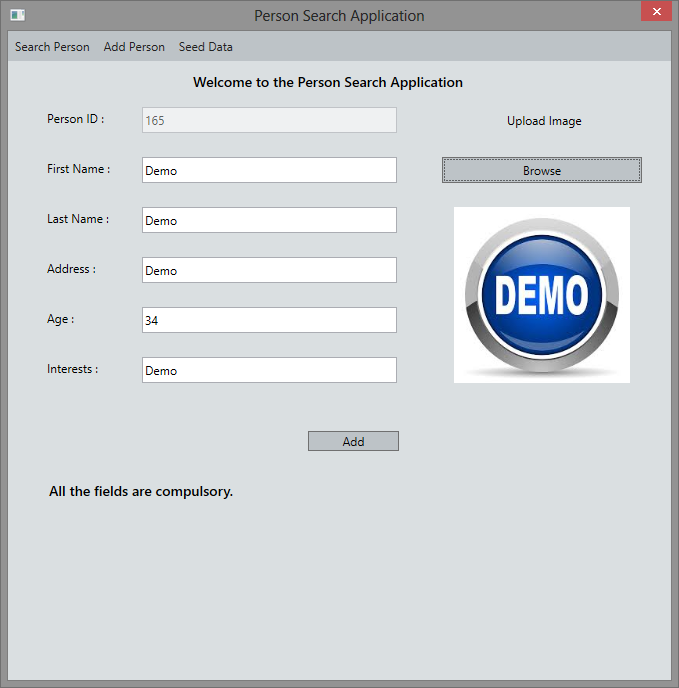
User can also add individual record into the database. Click on *Add Person* menu button. A set of controls would be displayed on the screen. All the fields are mandatory for adding the Person details and Add button would be disabled till the user inputs all the data. *Person ID* field is read-only field and this would be auto-populated. There is a provision to add profile picture of the person. The file browser would only shows images having *jpg*, *jpeg*, *png* and *gif* format.

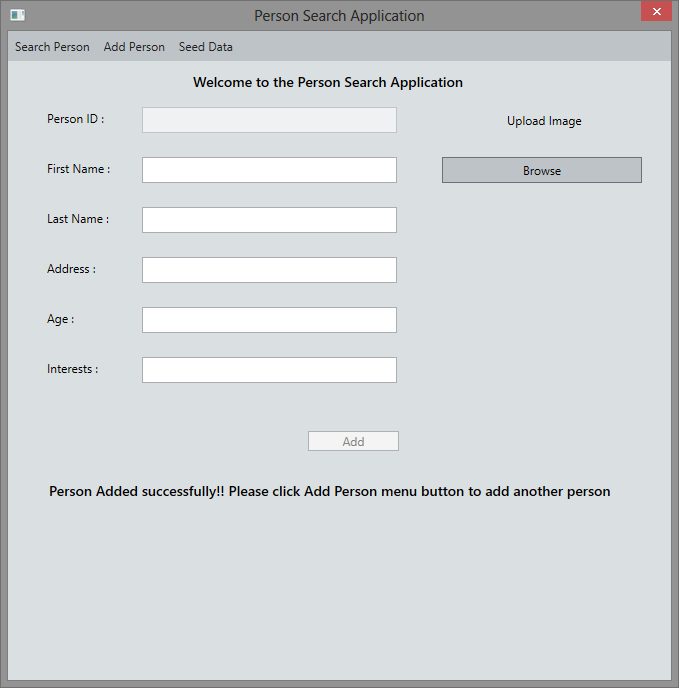




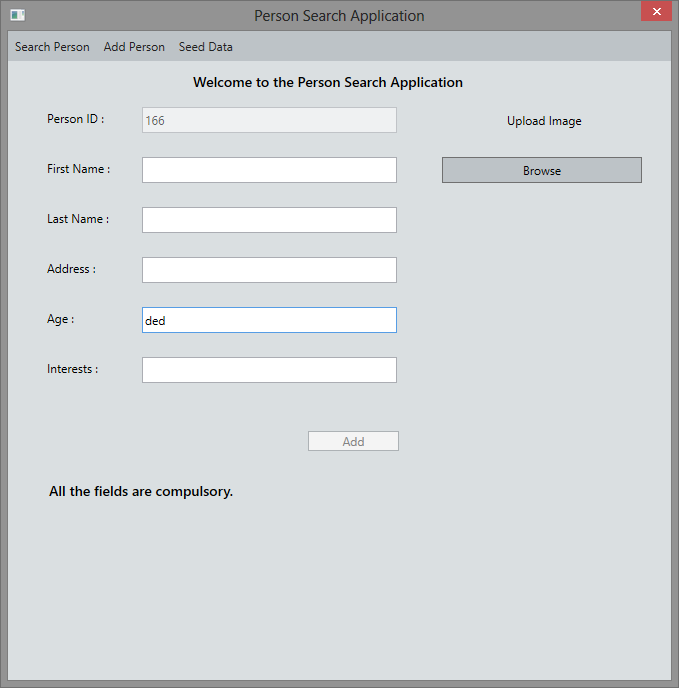


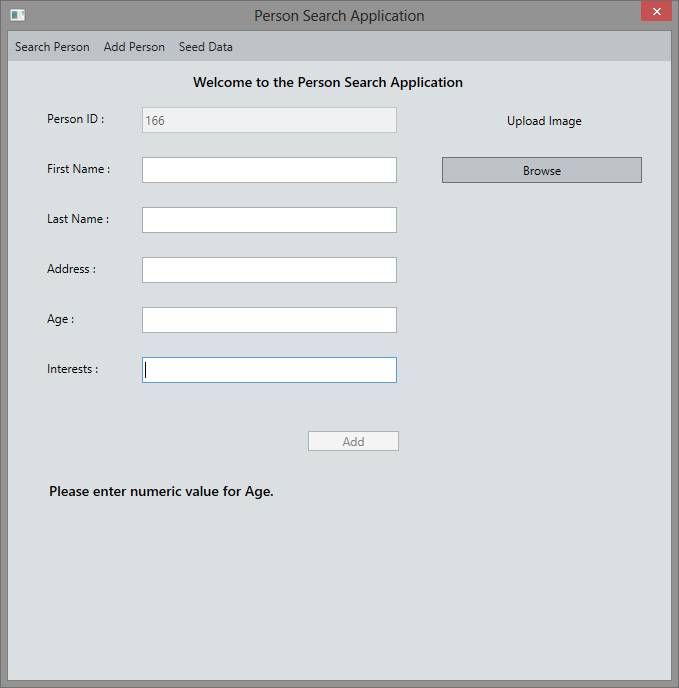
On successful entry of the Person, appropriate message would be displayed.





The *Age* field has to have Numeric value. If a non-numeric value is entered, on exiting the textbox, appropriate message would be displayed and the textbox would be cleared.





1. **Unit Tests**

The Unit Test project has been developed using Visual Studio 2012. This project tests the functionality for searching the non-existent record in the database, adding a record in the database and then searching an existing record. It first searches for the record ***ABCUNIQUEXYZ123*** in the database. Please make sure that the database does not contain the record having first name or last name as ***ABCUNIQUEXYZ123.*** The test application then adds a record having first name as ***ABCUNIQUEXYZ123*** and then searches the database for this newly inserted record. It checks for appropriate messages to test the status of the operation.

1. **Error handling**

Error handling has been implemented for all the modules. All the errors would be handled in the code and appropriate message would be displayed to the user.