

[Draft] Age Range System Guideline Document

I. Introduction

Restore NuGet packages before building solution.

Database: there were 3 *.db files:

1. Databases\RunningDb\AgeRanger.db : for development
2. Databases\IntegrationTestDb\AgeRanger.db for running integration tests. No need to change ConnectionString
3. Databases\AcceptanceTestDb\AgeRanger.db for running acceptance tests. Remember change connectionString in AgeRange/AgeRanger.WebApp/ConnectionString.config and point it to this database when perform Acceptance test.

II. Development Environment

Visual Studio 2017 Pro.

Back End:

- ASP.Net MVC 5 & Web API 2.
- .Net Framework 4.5.2
- Autofac (DI)
- Log4Net
- EntityFramework 6
- SQLite 3

Front End:

- Angularjs 1.6 && angular-ui
- Twitter Bootstrap 3
- toastrjs (john papa)(popup notifier)

Testing:

- Moq
- Microsoft VS Test Framework
- Selenium

III. Business Requirements

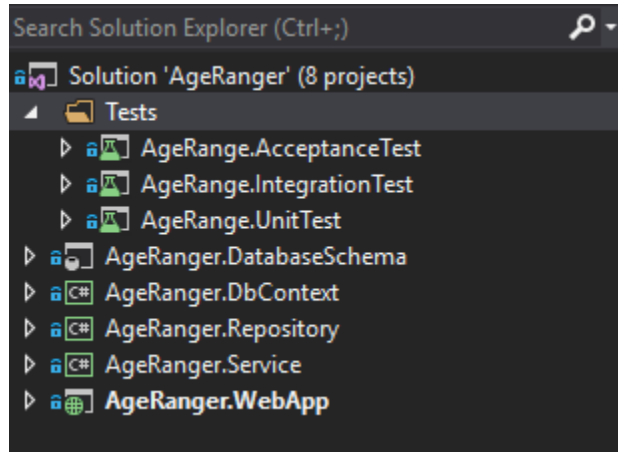
1. [Done] Displays a list of people in the DB with their First and Last names, age and their age group.
2. [Done] Search for a person by his/her first or last name and displays all relevant information for the person - first and last names, age, age group.
3. [Done] Add a new person - every person has the first name, last name, and age;
4. [Done] Edit existing person records and expose a WEB API
5. [Done] Delete an existing person.
6. [Done] All Unit Test are covered.
7. [ToDo] Migrating Data to SQL Server.

IV. System Design Requirements

1. [Done] Single Page Application (Refer to ../AgeRange.WebApp/app/app.js)

2. [Done] Unit testing with all cases covered.
3. [Done] Integration testing.
4. [Done] Acceptance testing.

V. Solution Folders:



- **AgeRanger.WebApp** - Presentation layer with ASP.NET MVC Web Site with a SPA using AngularJS. Perform all UI logic.
- **AgeRanger.Service** - Domain Model and Service. Service is responsible for perform business logic and return Domain Model included all information that Presentation layer need.
- **AgeRanger.Repository** - Generic of Data access layer (as a wrapper). Accessing to Db to retrieve data.
- **AgeRanger.DbContext** - Use EF6 and SQLite to communicate with Db. All entities and Entity-Mappings are defined here.
- **AgeRanger.DataBaseSchema** - Support DBA to migrate data to SQL Server.
- **AgeRange.Tests.UnitTest** : Test all methods with no dependencies
- **AgeRange.Tests.IntegrationTest**: Start test point from service layers to database: Service->Repository->Db
- **AgeRange.Tests.AcceptanceTest**: Automation Test performed by selenium to make sure all feature work well before release time.

VI. Next Improvements

1. Security - Security consideration:
 - JSON Web Token Authentication
 - Cross Site Anti Forgery
 - Role Based Security : only administrators have access right to Add/edit.
2. Consistent Error handling and message between contrpllerpi and client
3. Implement validation at client side
4. Confirmation popup when delete person
5. Adding Unit Test in AngularApp

VII. Overview Architecture Diagram

Dev Environment

Visual Studio 2017 Pro.

Back End:

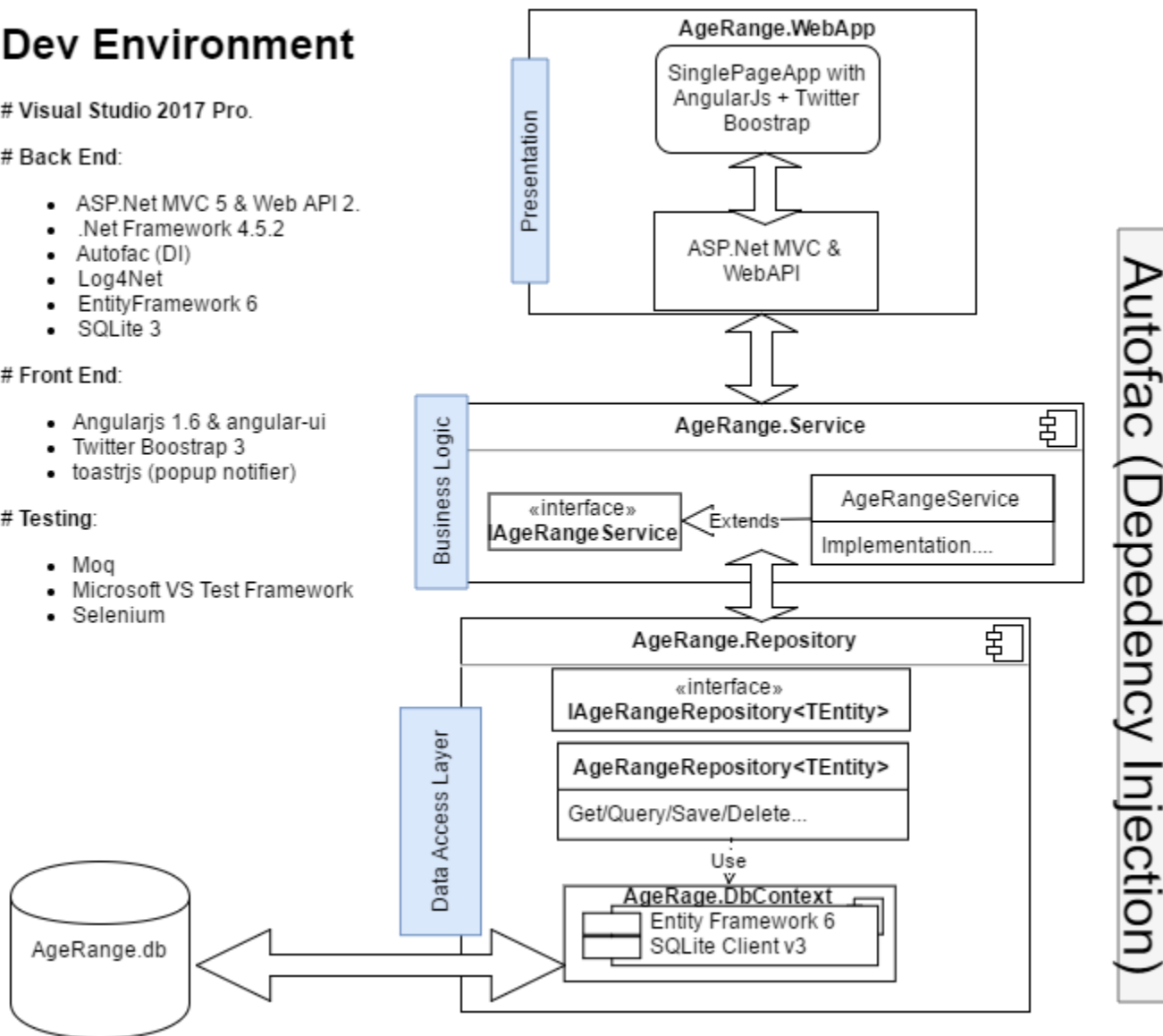
- ASP.Net MVC 5 & Web API 2.
- .Net Framework 4.5.2
- Autofac (DI)
- Log4Net
- EntityFramework 6
- SQLite 3

Front End:

- Angularjs 1.6 & angular-ui
- Twitter Bootstrap 3
- toastrjs (popup notifier)

Testing:

- Moq
- Microsoft VS Test Framework
- Selenium



Feature traceability Table

Features	Angular App (UI)	PersonApiController (Server)	Service	Repository
Add Person	/ app /controllers/add PersonModalContr oller.js	[HttpPost] public HttpResponseMessage Post(PersonModel personModel)	SavePerson(PersonModel)	ForceSaveOrUp dateImmediate ly(PersonEnti ty)
Edit Person	/ app /controllers/edi tPersonModalCont roller.js	// POST: api/person/update/{filter} public HttpResponseMessage Update(long id, PersonModel personModel)		
Get Person	/ app /controllers/edi tPersonModalCont roller.js	// GET: api/Person/5 public HttpResponseMessage Get(long id)	GetPersonById (id)	Get(id)
Delete Person	/ app /controllers/per sonController.js	// DELTE: api/person/delete/{id} public HttpResponseMessage Delete(long id)	DeletePersonById(id)	DeleteById(id)
Filter Person	/ app /controllers/per son.js	// GET: api/Person/{filter} public HttpResponseMessage Filter(string name)	FindPeople(name)	List(Expressi on @query) GetAll()
Get AgeGroup	N/A	N/A	GetAgeGroupsByAgeRange (long minAge, long maxAge)	Query(Express ion @where)
Migration Db	N/A	N/A	N/A	N/A

Testing Results:

Note: There are several missing test cases should be added later

Solution Explorer

Search Solution Explorer (Ctrl+;)

Solution 'AgeRanger' (8 projects)

- Tests
 - AgeRange.AcceptanceTest
 - AgeRange.IntegrationTest
 - AgeRange.UnitTest
- AgeRanger.DatabaseSchema
- AgeRanger.DbContext
- AgeRanger.Repository
- AgeRanger.Service
- AgeRanger.WebApp

Test Explorer

Run All | Run... | Playlist : All Tests

AgeRange.AcceptanceTest.BrowerTests (8)

- ✗ AddNewPersonWithValidData 16 sec
- ✗ AllPeopleAreShownWhenPageLoaded 10 sec
- ⚠ AddNewPersonWithInvalidValidData
- ⚠ EditPersonWithInvalidValidData
- ✓ AddNewPersonPopupLoadingCorrectly 21 sec
- ✓ EditPersonPopupLoadingCorrectly 20 sec
- ✓ FilteringPeopleByFirstOrLastName 14 sec
- ✓ IsSearchTextBoxAvailable 12 sec

AgeRange.IntegrationTest.ServiceTest (5)

AgeRange.UnitTest.Service (6)

AgeRange.UnitTest.WebApp.ApiControllers (11)

AgeRange.UnitTest.WebApp.Controllers (1)

Summary

Last Test Run Failed (Total Run Time 0:01:46.3822793)

- ✗ 2 Tests Failed
- ⚠ 2 Tests Skipped
- ✓ 27 Tests Passed

Screenshots:

Search People

Add New Person

#	First Name	Last Name	Age	Age Group	Edit	Delete
10	Test1 1234	Test1L	100	Very old	✍	✖
9	Kitty	Nguyen	203	Vampire	✍	✖
8	Sam	Vo	110	Very old	✍	✖
7	Newbei	New	1000	Vampire	✍	✖
6	Crazy	Kity	5000	Kauri tree	✍	✖
5	Tinh	Tinh Heo	345	Vampire	✍	✖
4	Tim	Heact	120	Crazy ancient	✍	✖
3	Fitch	Heavy	123	Crazy ancient	✍	✖
2	Mad	Todd	20	Teenager	✍	✖
1	Jindo	nho	10	Child	✍	✖

Data Migration

Total: 10

Add New Person

FirstName

LastName

Age

Add

Cancel

Heact	120	Crazy ancient
Heavy	123	Crazy ancient
Todd	20	Teenager
nho	10	Child

Edit Person (#: 7)

Age Group: Vampire

FirstName

LastName

Age

Update

Cancel

Heact	120	Crazy ancient
Heavy	123	Crazy ancient
Todd	20	Teenager