

Solution has 5 projects explained below:

Client: (User interface WPF)

- Dependency
 - Infrastructure
 - Automapper
- Content
 - Search for employee using first name.
 - Enter search text and click Search by name button.
 - Retrieve Employee by Id
 - Enter employee Id and click Retrieve by Id button.
 - Details stack
 - Display for Id, Name, Email, Status and Gender
 - Add Employee
 - Add employee based on entered details (After validation).
 - Success message in case of success.
 - Failure message in case of failed.
 - Refresh the view to get the new entry displayed.
 - Update Employee
 - Update employee based on entered details (After validation).
 - Success message in case of success.
 - Failure message in case of failed.
 - Refresh the view to get the new entry displayed.
 - Delete Employee
 - Delete the selected employee (either selected from the view or retrieved by Id).
 - Validation against empty selection.
 - View Employees
 - Display employees (Name, Gender and Status) by page number
 - Page number controller
 - Export to CSV file
 - Export the current page to a csv file in the same location.
- Can be enhanced
 - A better view for different flows instead of one single page.
 - Using more appropriate visual toolbox match fields instead of blank test.
- Screenshots for reference

MainWindow

Advaya Embranthirimale	active
Chaitan Panicker male	active
Hiranmay Joshi 4 female	inactive
Bhudeva Bhattathirimale	active
Mahesh Arora male	inactive
Advaya Embranthirimale	active
Gudakesha Chattopadhyaymale	inactive
Dr. Chakrika Khannafemale	inactive
Diksha Namboothirifemale	inactive

Prev 1 Next

Export current page

Search For Employee by First Name

Search By Name

Retrieve by ID

Retrieve by Id

Id

Name

Email

Gender

Status

Add Employee

Update Employee

Delete Employee

Id

Name

Email

Gender

Status

Add Employee

Update Employee

Delete Employee

Invalid input please check below errors

email is not a valid email

xx is not a valid gender

xx is not a valid status

OK



Are you sure you want to delete this employee

Yes

No

Id	<input type="text" value="401138"/>
Name	<input type="text" value="Advaya Embranthiri"/>
Email	<input type="text" value="embranthiri_advaya@fadel-halvorson.info"/>
Gender	<input type="text" value="male"/>
Status	<input type="text" value="active"/>
<input type="button" value="Add Employee"/>	
<input type="button" value="Update Employee"/>	
<input type="button" value="Delete Employee"/>	

Domain: (contains domain models)

- Dependency
- Content
 - Employee.cs

Application: (contains the Repository contract)

- Dependency
 - Domain
- Content
 - IEmployeeService
 - ItokenProvider

Infrastructure (Repositories with business logic to implement CRUD)

- Dependency
 - Domain
 - Application

- Content
 - Employee Repository
 - Implementation for all services
 - Token Provider
 - Logic related to extract Token (To be extended later)
- Notes to Add
 - Additional validation layer can be added here.
 - Log mechanism for all request and exception from third parties
 - Http Client repository to handle all requests

Testing

- Dependency
 - Xnit
 - Moq
- Content
 - Positive and Negative scenarios to cover all business logic operations on Employee repository.
- Screenshots

▲ ✓ Tests (11 tests)	Success
▲ ✓ Tests (11 tests)	Success
▲ ✓ EmployeeRepositoryTests (11 tests)	Success
✓ AddEmployee_ReturnsError_WhenAddingWrongEmail	Success
✓ AddEmployee_ReturnsError_WhenAddingWrongGender	Success
✓ AddEmployee_ReturnsError_WhenAddingWrongStatus	Success
✓ AddEmployee_ReturnsError_WhenAPIReturnsBadRequest	Success
✓ AddEmployee_ReturnsPositiveInt	Success
✓ DeleteEmployee_ReturnsFalse_WhenAPIReturnsNotFound	Success
✓ DeleteEmployee_ReturnsTrue_WhenAPIReturnsSuccess	Success
✓ GetEmployee_ReturnsEmployee_WhenAPIReturnsSuccess	Success
✓ GetEmployee_ReturnsNull_WhenAPIReturnsNotFound	Success
✓ GetEmployees_ReturnsEmployees_WhenAPIReturnsSuccess	Success
✓ UpdateEmployee_ReturnsTrue	Success