



.NET Core Azure Sprint-1 **Cargo Manager System**

Document Revision History

| Date | Revision No. | Author | Summary of Changes |
|------------|--------------|--------------|--------------------|
| 13-07-2017 | 1.0 | Pankaj Patil | Case Study Created |
| | | | |
| | | | |
| | | | |
| | | | |



Table of Contents

| | |
|---|----------|
| Introduction | 4 |
| Setup Checklist | 4 |
| Instructions | 4 |
| Problem Statement | 5 |
| Objective | 5 |
| Project structure | 9 |
| Implementation | 10 |
| Summary of the functionality to be built: | 10 |

INTRODUCTION

This document outlines a project for the .NET Line of Technology (LOT). The project is to develop Cargo Manager System. This document contains the requirements, work flow of the system and gives guidelines on how to build the functionality gradually in each of the course modules of the .NET LOT.

SETUP CHECKLIST

Minimum System Requirements

- Intel Pentium 4 and Windows 2010
- Memory 4 GB
- Internet Explorer 11.0 or higher / Chrome
- SQL Server 2014 or 2016 client and access to SQL Server 2014 or 2016 server
- Visual Studio 2019
- Visual Studio Code
- Git

INSTRUCTIONS

- The code modules in the mini project should follow all the coding standards.

PROBLEM STATEMENT

OBJECTIVE

Need to develop Cargo Manager System.

Abstract of the project:

Cargo Manager is a comprehensive cargo management module, designed for addressing the areas of General cargo, Bulk cargo operations through Aviation. All the aspects of cargo like documentation handling, movement, and storage are addressed in this module.

Cargo Manager system will have following features:

- General Cargo Booking (Manifestation of cargo)
- Storing of Cargo (actual cargo weight, volume etc. are recorded)
- Generating the gate pass for bringing the cargo into the warehouse
- Marking the truck exit
- Delivery order for rejected or damaged cargo
- Generating gate pass for redelivery of cargo
- Housekeeping movements for maintenance i.e., moving the cargo from one place to another
- Extensive report and querying facilities

Users of the System will be: Cargo Manager Administrator

Role of Cargo Manager Administrator:

- Login to System
- Registration on System
- Search for Address
- Calculating Cargo Price
- Book a Cargo
- View Status of Cargo
- Cancel the Cargo

Phase 1: Create Client Application in Angular and Services using ASP.NET Core Web API

MODULE LIST and MODULE DETAILS

Modules:

- Customer Class :- This class will contain methods which will allow to manage all Customers
- Employee class :- This class will contain methods to manage Employee
- CargoOrder Class :- This class will Contain methods which will allow to manage all Order Placed by Customer
- CargoOrderDetails Class :- :- This class will Contain methods which will allow to manage all Order Placed by Customer
- City Class :- This class will contain methods to manage City details

- **Product Class :** This class will contain methods to manage Product
 - i. **Login / Sign In:** Login screen would display asking Cargo Manager user to enter 'User Id' & 'Password'. If the supplied user credentials are valid the HomePage would be displayed, else appropriate Login error message would be displayed
 - ii. **Home Page:** On successful user authentication (validation of userid/password provided by the user in login screen) the homepage would be displayed. The Homepage would contain below sections/contents:

Header section: The header section would be common across all the pages and would mainly have –

 - Sign In Link – On click it would take to Login page
 - Search: Search for information

Main content section: It would display generic Welcome message giving overview of the site. Should have menu for Administrator Employee, Customer
- a. **For Administrator :**
 - i. Register a new Employee on Cargo Manager System
 - ii. Delete an existing Employee on Cargo Manager System
 - iii. Update an existing Employee on Cargo Manager System
 - iv. Search for Employee and Customer.
 - v. Add new Cargo Type.

- vi. Update Cargo Type.
- vii. Update Places and Prices.
- b. For Employee :
 - i. Update self-Record on Cargo Manager System
 - ii. Search for Customer.
 - iii. Search for Cargo fare
 - iv. Search for Order
 - v. Update Cargo Type.
 - vi. Update Places
 - vii. Update Customer
- c. For Customer
 - i. User will get option to add, update its details
 - ii. View the details of all Order Placed

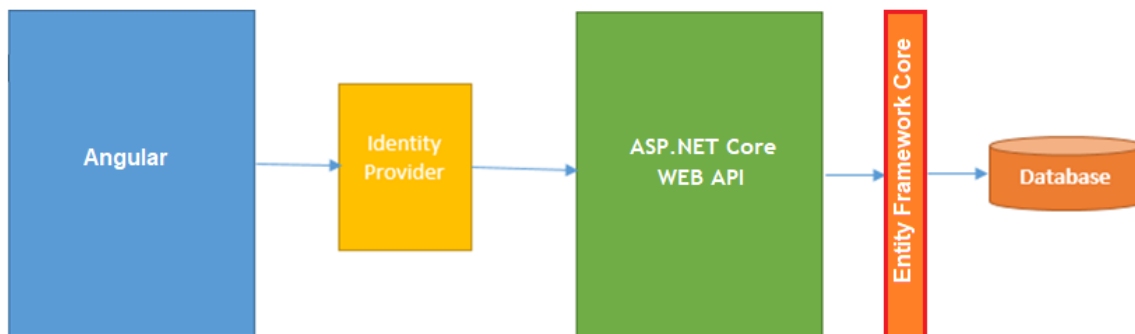
PROJECT STRUCTURE

Description: Create below sample reference project structure, which will help to reuse most of module for Web application. You are allowed to bring your own project structure and project to achieve the requirement.

In this sprint you have to use

- Angular UI as the presentation layer
- ASP.NET Core WEB API to create services
- Entity Framework Core to interact with the database.

The project should have minimum target framework – .Net Core 3.1 for Core WEB API Services and Angular 10 for UI Design



Design guidelines

- All the exceptions/errors to be captured and user friendly message to be displayed on the Common Error page.

IMPLEMENTATION

SUMMARY OF THE FUNCTIONALITY TO BE BUILT:

The participants need to develop the Cargo Manager System by building the functionality incrementally in each of the course modules of .NET LOT.

| Sr. No | Course | Duration | Functionality to be built |
|--------|--|----------|---|
| | | (in PDs) | |
| 1 | Angular ASP.NET Core Web API Entity Framework Core SQL Server | 5 | Developing Presentation components (Angular), Business components (ASP.NET Core WEB API) and Data access components (Entity Framework Core) |