Test assessment

Intro

Develop a web-application for managing **parcels** and letters in a post office. Parcels and letters are packed into different types of **bags**. One bag can contain either parcels or letters, never mixed. Every parcel has separate details, like unique number, destination county, etc. (see below). Letters don't have any details and don't exist as a separate entity in the system. Several bags, if they are delivered by the same aircraft, are grouped into a **shipment**. One shipment may contain bags of both types (with letters and parcels).

Data is sent to the server via REST API in 4 steps. First request creates a shipment. Second request creates bags in this shipment. Third request creates parcels in specified bags. Last request finalizes shipment, so nothing can be added after finalizing (neither new bags into the shipment nor new parcels into any bag inside)

Home page of this application shows list of all created shipments. If user clicks on a shipment – list of bags is shown (show bag number, type, total amount of parcels or letters inside and total price).

Requirements

Please design data structure and develop required functionality. Use .NET Core 2.1 (or higher). Frontend technology is up to you, but Angular 6 (or higher) is a plus. Authentication is not needed. Data should be stored in a MSSQL (Express) database. Please use Entity Framework to work with database. We expect production-ready solution with error handling and validations. Unit tests would be a big plus.

Please provide for assessment

- Link to git repository (if you used it) or archive with full solution
- Script that creates database and all necessary tables
- Request body (JSON) that was used for testing/debugging

Data description

Tables below describe business specific data for mentioned entities. Everything should be captured by the system and displayed in UI with specified names, format and match to validation rules. Choosing names in the system and data types is a part of the assessment. All fields are mandatory. You can add as many extra fields of any type as you need.

Parcel

Field	Type	Comment
Parcel number	string	Format "LLNNNNNNLL", where L – letter, N – digit Must be unique within entire database
Recipient name	string	Max length 100 characters
Destination country	string	2-letters code, e.g. "EE", "LV", "FI"
Weight	numeric	Max 3 decimals allowed after comma
Price	numeric	Max 2 decimals allowed after comma

Bag with parcels

Field	Type	Comment
Bag number	string	Max length 15 characters, no special symbols allowed Must be unique within entire database
List of parcels	collection	Can't be empty by the moment of finalizing shipment

Bag with letters

Field	Type	Comment
Bag number	string	Max length 15 characters, no special symbols allowed Must be unique within entire database
Count of letters	numeric	Can't be zero
Weight	numeric	Max 3 decimals allowed after comma
Price	numeric	Max 2 decimals allowed after comma

Shipment

Field	Type	Comment
Shipment number	string	Format "XXX-XXXXXX", where X – letter or digit Must be unique within entire database
Airport	enumeration	Possible values are "TLL", "RIX", "HEL"
Flight number	string	Format "LLNNNN", where L – letter, N – digit
Flight date	date and time	Can't be in past by the moment of finalizing shipment
List of bags	collection	Can't be empty by the moment of finalizing shipment