# Delivery Booking System

A company decides to create a web application which can be used by anyone who wants to pick up items from their verified address to any place in the same city.

The system provided the below capabilities,

1. Customer Registration/ Login
2. Executive Registration/ Login
3. Fix a booking to deliver
4. Cancel booking

The system will be secured that only logged in users can perform any of the above functions.

## Detailed Requirements

### User Setup

Every Executive must register themselves in this web application and can login in to view is there any request for delivery. They can register by giving following details.

* + 1. Id
    2. Name
    3. Age
    4. Phone
    5. Password

The Customer who wants the items to be delivered from this company should also register here and when required for delivery they can login and request for delivery. They can register by giving the following details.

* + 1. Id
    2. Name
    3. Age(optional)
    4. Address
    5. Password

### Screen 1: Login/Register

This screen will prompt the user to enter their username and password. Both fields are mandatory and to be validated. The page will also have a small company logo. If new user register or else login using username and password.

The username and password will be validated at the server side against the database table – if the details match, the users will be redirected to the Home page and if not, they will get the same login page with the error message “You’ve entered an incorrect user name or password”.

### Screen 2: Booking Page for Customers

This screen will be shown only when the Customer has successfully logged in. This will have the company logo (the one used in the login screen) and also a navigation bar which will have the below options,

|  |  |  |
| --- | --- | --- |
| **S. No** | **Menu Name** | **Comments** |
| 1 | Add Booking | This will redirect to the “Add Booking” screen |
| 2 | View Booking | This will redirect to the “View Booking” screen |
| 3 | Cancel Booking | This will redirect to the “Cancel Booking Page” |
| 5 | Logout | This will log off the user and redirect to the “Login” screen |

### Screen 3: Executive’s Page for Delivery

This screen will be shown only when the Executive has successfully logged in. This will have the company logo (the one used in the login screen) and also a navigation bar which will have the below options,

|  |  |  |
| --- | --- | --- |
| **S. No** | **Menu Name** | **Comments** |
| 1 | View Request for delivery | This will redirect to the “View Request” screen |
| 2 | Logout | This will log off the user and redirect to the “Login” screen |

### Screen 4: Add Booking

Customers will use this screen to add the Booking information to the system. Screen will have the below fields,

1. Order Id (auto generated number by the system, not editable by the users)
2. Customer Id (No special characters allowed)
3. Executive Id (No special characters allowed)
4. Date (Delivery Date)
5. Time of Pickup (Time)
6. Weight of package
7. Price

When the users submit the data, validate and save to the database.

### Screen 5: View Booking

Customers will use this screen to View the Booking information to the system. Screen will have the below fields,

1. Order Id (auto generated number by the system, not editable by the users)
2. Customer Id (No special characters allowed)
3. Executive Id (No special characters allowed)
4. Date (Delivery Date)
5. Time of Pickup (Time)
6. Weight of package
7. Price

If they don’t want this booking, they can cancel by clicking ‘Cancel Booking’ so that Your booking will be cancelled.

### General Requirements

* All screens except “Login” and “Home” will have a button “Go to Home” to take the user back to the Home screen
* Whenever an operation is successful, the customers should be shown a message as such. E.g., if the “Add Booking” is successful, it will display “Booking successfully added to the system”.
* On failures, the users should be shown appropriate friendly error message (not exceptions).

### Technical Requirements

* Tables should be designed appropriately (proper data types, table/column names, constraints), use a SQL Server database
* The tables should be created only using SQL Files and these should be available in the repo
* Front end – ASP.NET Core MVC, HTML5, CSS, Bootstrap, JavaScript/jQuery
* Business Logic – Web API/Service in MVC
* Data Access – EF Core (DB/Model first approach)
* All names used in the code (classes, variables, properties etc.,) should be meaningful and follow consistent naming approach (Camel Case, Pascal Case etc.,)
* Classes/methods/properties should have proper accessibility modifiers
* Use code formatting to ensure the code is readable, write comments where required so that the reviewers can understand your code

### IMPORTANT

* Create a new repo “ProjectA” in your account and use that for this exercise
* Push the code to the remote repository often or at least before end of each day.
* Repo should have a folder “DB Script” and it will have all the SQL Files (tables, stored procedure etc.,)
* Repo should have a folder “src” and it will have your ASP.NET Core Projects