# Delivery Booking System

Customer anyone to Choose Delivery Executive wants to pick up items from their verified address to any place in the same city though a web application.

Users in this document refer to the Delivery Booking sites. The system provided the below capabilities,

1. Customer information
2. Delivery Executive information
3. Choose an Delivery Executive for Pick Up Parcel
4. Delivery Booking Status
5. Booking Details

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

## Detailed Requirements

### Customer Setup

Each Customer will have a unique user name and password. There is have screen to manage the customers from Admin of the System, the data will be managed in the Customer table directly in the database. The database table will have the below data for the entire Customers.

Customer

* 1. Id
  2. Name
  3. User Name
  4. Password
  5. Age
  6. Phone
  7. Address
  8. City
  9. Is Verified

### Delivery Executive Setup

Each Executive will have a unique user name and password. There is have screen to manage the Executive from Admin of the System, the data will be managed in the Delivery Executive table directly in the database. The database table will have the below data for the entire Executives.

Delivery Executive

1. Id
2. Name
3. User Name
4. Password
5. Age
6. Phone
7. Address
8. City
9. Is Verified

### Screen 1: Register

Customer or Executive will use this screen to add the information to the system. Screen will have the below fields,

1. User Id (auto generated number by the system, not editable by the users)
2. Name (No special characters allowed)
3. User Name (No special characters allowed)
4. Password (Should Have min Length of 8)
5. Age
6. Phone
7. Address
8. City
9. User Type(To Select Customer (or) Delivery Executive

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

### Screen 2: Login

This screen will prompt the Customer or Executive to enter their below fields,

* user name
* password
* User Type (Customer or Executive).

All fields are mandatory and to be validated. The user name and password will be validated at the server side against the database table – if the details match and also check Admin can verified their account or not, and then 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” or “Admin cannot verified your Account”.

### Screen 3: Customer Home Page

This screen will be shown only when the Admin Can verifies the Customer Account and also Customer has successfully logged in. This will have the navigation bar which will have the below options,

|  |  |  |
| --- | --- | --- |
| **S.No** | **Menu Name** | **Comments** |
| 1 | Home | This will redirect to the “Home” screen |
| 2 | Booking | This will redirect to the “Add Booking” screen |
| 3 | Booking Status | This will redirect to the “Booking Status” screen |
| 4 | Booking Details | This will redirect to the “Booking Details” |
| 5 | Logout | This will log off the user and redirect to the “Login” screen |

### Screen 4: Booking

Customer will use this screen to setup the Delivery-Executive Bookings. Screen will have the below fields,

1. Date and Time of Pickup
2. Weight of Package
3. Address(on Page Load Filled from DB)
4. City(on Page Load Filled from DB)
5. Select Delivery Executive(on Load Get Executive from db in Same City)

When the users submit the data, validate and save to the database. Each Booking will have a unique id in the database.

### Screen 5: Booking Status

After Add Booking Redirect to Booking Status Page to List the Booking Details with Status Column for Updating of the Executive Response like (Customer Requesting, Executive Accept / Executive Reject).And Also Have another Column for showing Price of the Delivery Booking.

### Screen 6: Booking Details

To Show the List of Booking Details only with The Status of Delivery Packaged.

### Screen 7: Executive Home Page

This screen will be shown only when the Admin Can verifies the Executive Account and also Executive has successfully logged in. This will have the navigation bar which will have the below options,

|  |  |  |
| --- | --- | --- |
| **S.No** | **Menu Name** | **Comments** |
| 1 | Home | This will redirect to the “Home” screen |
| 2 | Bookings | This will redirect to the “Booking Request” screen |
| 3 | Booking Status | This will redirect to the “Booking Status” screen |
| 4 | Booking Details | This will redirect to the “Booking Details” |
| 5 | Logout | This will log off the user and redirect to the “Login” screen |

### Screen 8: Booking Requests

Executive will use this screen to Response the Customers Delivery Bookings. Screen will have the below fields in the List of Customer Requests,

1. Customer Name
2. Address
3. City
4. Date and Time of Pickup
5. Weight of Package
6. Status(Select Accept/Reject)
7. Price(Editable)

When the Executive submit the data, validate and Update to the database.

### Screen 9: Booking Status

After Updating Booking Requests Redirect to Booking Status Page to List the Booking Details With Status Column for Updating of the Customer Response for Price Like (Executive Accept / Executive Reject, Price Accept /Reject).

### Screen 10: Booking Details

To Show the List of Booking Details only with The Status of Delivery Packaged. It Should Update on After the Date and Time Of Pickup.

### 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 users should be shown a message as such. E.g., if the “Add Booking” is successful, it will display “Booking successfully added to the system”. It would be helpful to show the names/ids in these messages as well to make it usable.
* 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

Your 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 Entity details are as follows.

1. Delivery Executive
   * 1. Id
     2. Name
     3. Age
     4. Phone
     5. Address
     6. City
     7. Is Verified
2. Customer
   * 1. Id
     2. Name
     3. Age(optional)
     4. Address
     5. City
     6. Is Verified
3. Bookings
   * 1. OrderID
     2. CustomerId
     3. ExecutiveId
     4. Order Date
     5. Day&TimeOfPickup
     6. Status(CustomerReq,ExecutiveAccept /ExecutiveReject,ExecutivePickuped)
     7. WeightOfPackage
     8. Price

The above-mentioned entities are the basic. Please add more entities or attributes to fulfill all the functional requirements.

Requirements

Customer and executives must register themselves.

Admin can activate their profiles by verifying them.

Admin login can be static

Customer once logged in can request for deliver

Delivery executives can log in to see if any request is available and choose to do it.

*Guideline on technologies to be used*

*Front End – MVC – View -> HTML + CSS + BootStrap*

*Business Logic – Web API / Service in MVC*

*Database – SQL Server*

*Connectivity – EF Core (Database/Model) first approch*