# C# Web Basics – Exam Preparation

# PANDA

**PANDA** (**P**ackage **A**cceptance and **N**ational **D**elivery **A**pplication) is a platform for package deliveries, which is a fast-rising Start-Up, which lacks a web application. You have been employed by the **KFC** (**K**ung-**F**u-**C**hicken) Corporation to implement a web platform for **PANDA**. However, there are specific requirements that must be followed.

## Technological Requirements

• Use the **SIS.WebServer**

• Use the **SIS.MvcFramework**

• Use Entity Framework Core

The Technological Requirements are ABSOLUTE. If you do not follow them, you will NOT be scored for other Requirements.

Now that you know the Technological Requirements, let us see what the Functional Requirements are.

## Database Requirements

The **Database** of the PANDA application needs to support **3 entities**:

### User

* Id - a GUID String**, Primary Key**
* Username - a **string** with **min length** **5** and **max length 20** (**required**)
* Email - a **string** with **min length** **5** and **max length 20** (**required**)
* Password - a string – hashed in the database (**required**)
* Packages – a **Collection** of type Packages
* Receipts – a **Collection** of type Receipts

### Package

* Id - a GUID String**,** **Primary Key**
* Description - a **string** a **string** with **min length** **5** and **max length 20** (**required**)
* Weight - a **floating-point** number
* Shipping Address - a **string**
* Status - can be one of the following values ("Pending", "Delivered")
* Estimated Delivery Date - a DateTime object
* RecipientId - **GUID foreign key (required)**
* Recipient - a User object

### Receipt

* Id - a GUID String**, Primary Key**
* Fee - a **decimal** number
* Issued On - a DateTime object
* RecipientId - **GUID foreign key (required)**
* Recipient - a User object
* PackageId - **GUID foreign key (required)**
* Package – a Package object

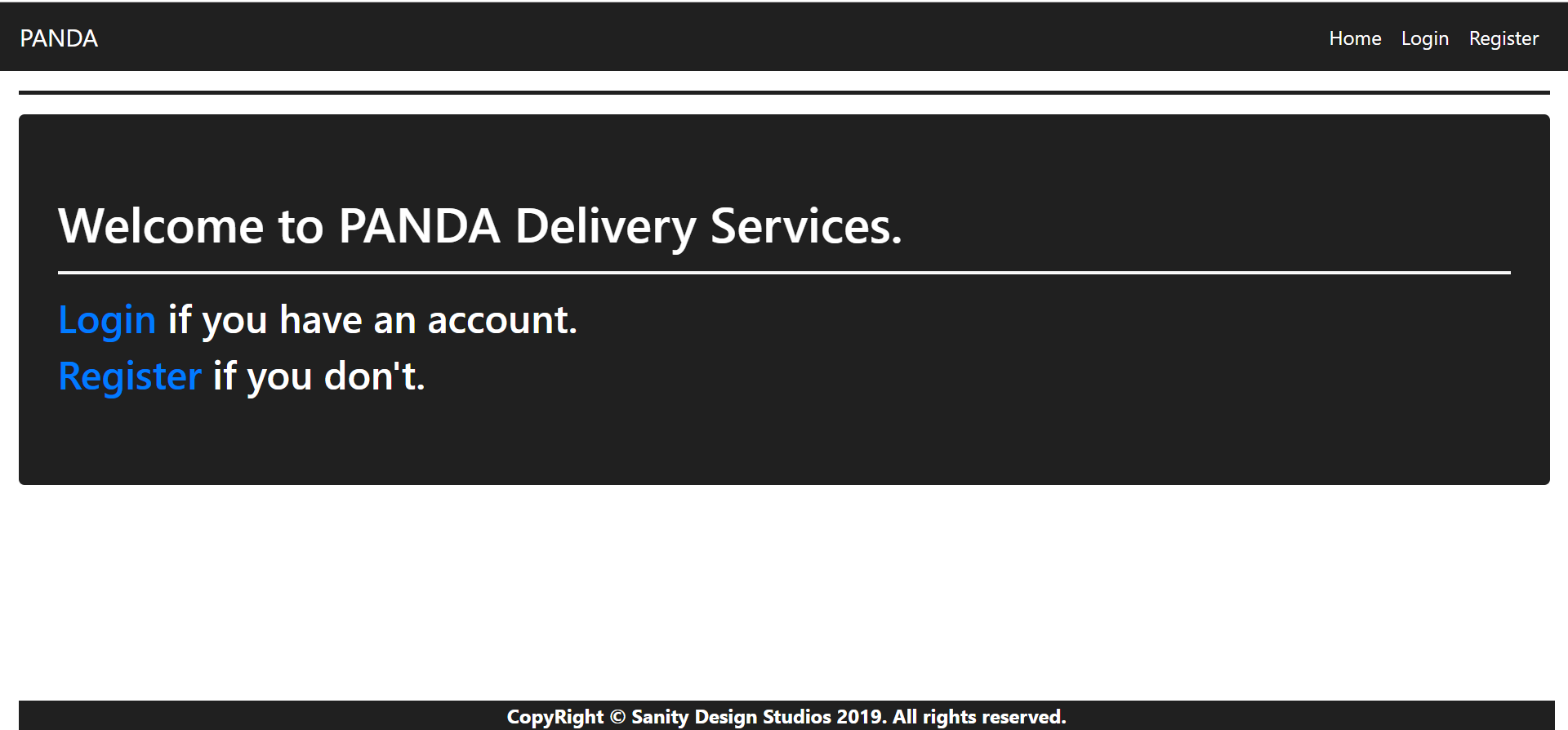
Implement the entities with the **correct datatypes**.

## Page Requirements

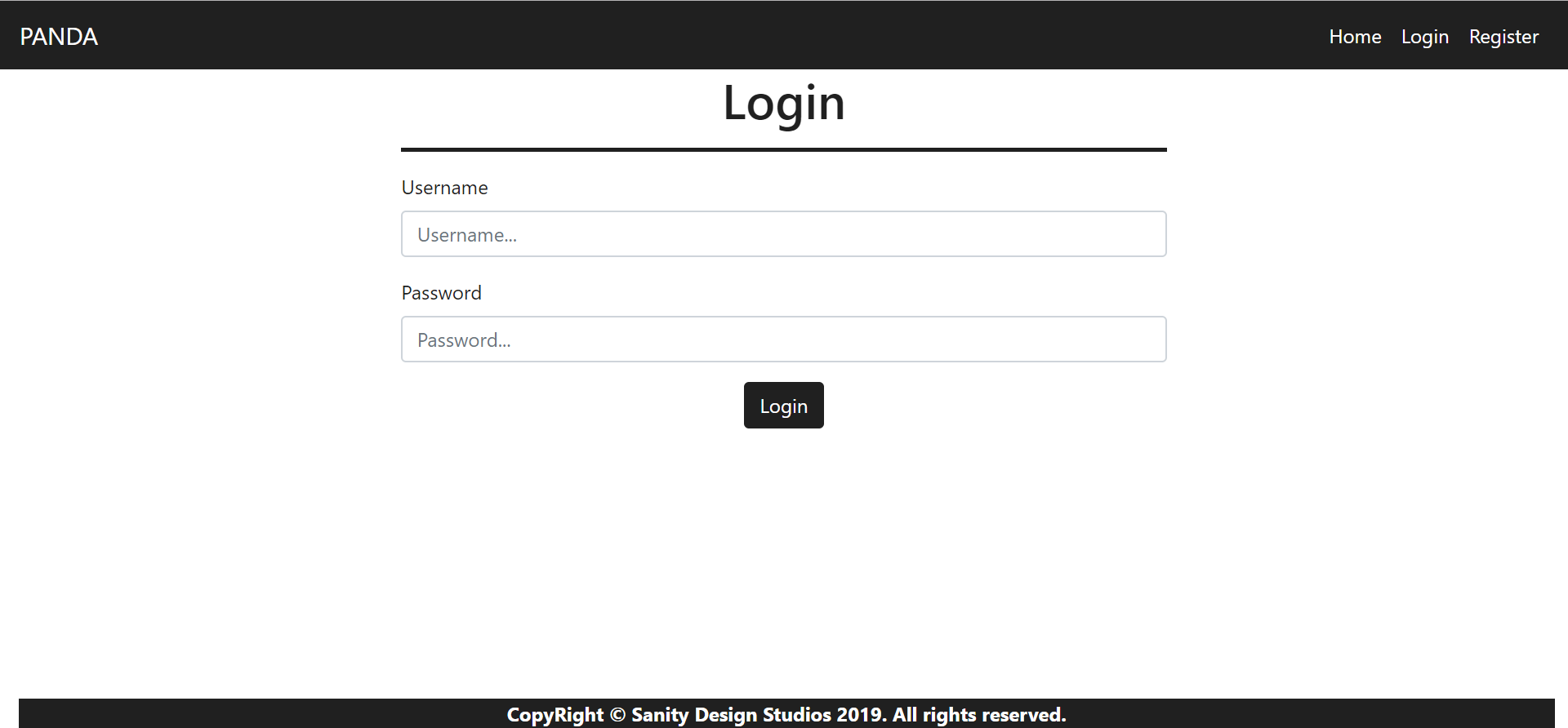
### Guest Pages

These are the **templates** and **functionalities**, accessible by Guests (**logged out** users).

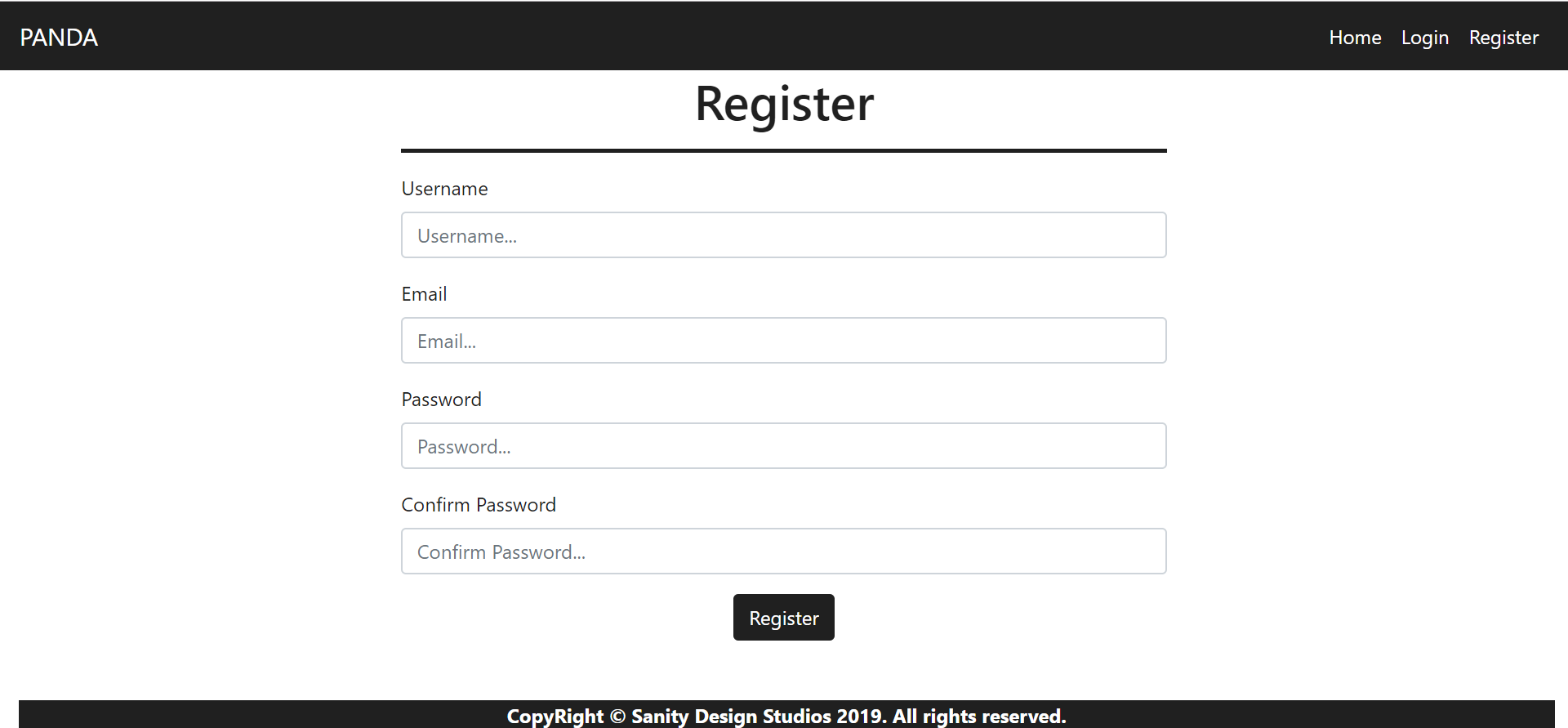
#### Index Page (route = "/Home/Index") (logged out user)



#### Login Page (route = "/Users/Login") (logged out user)



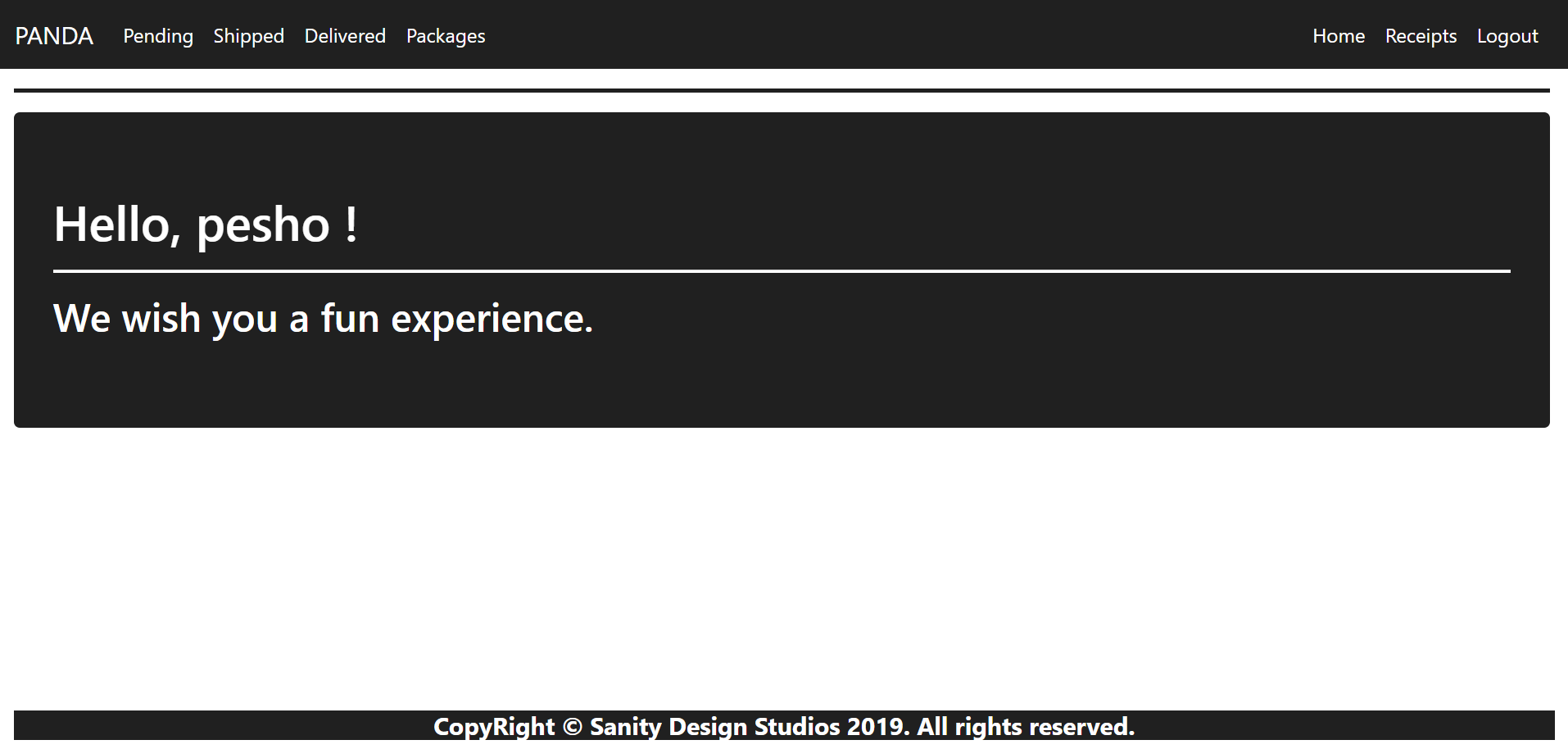
#### Register Page (route = "/Users/Register") (logged out user)



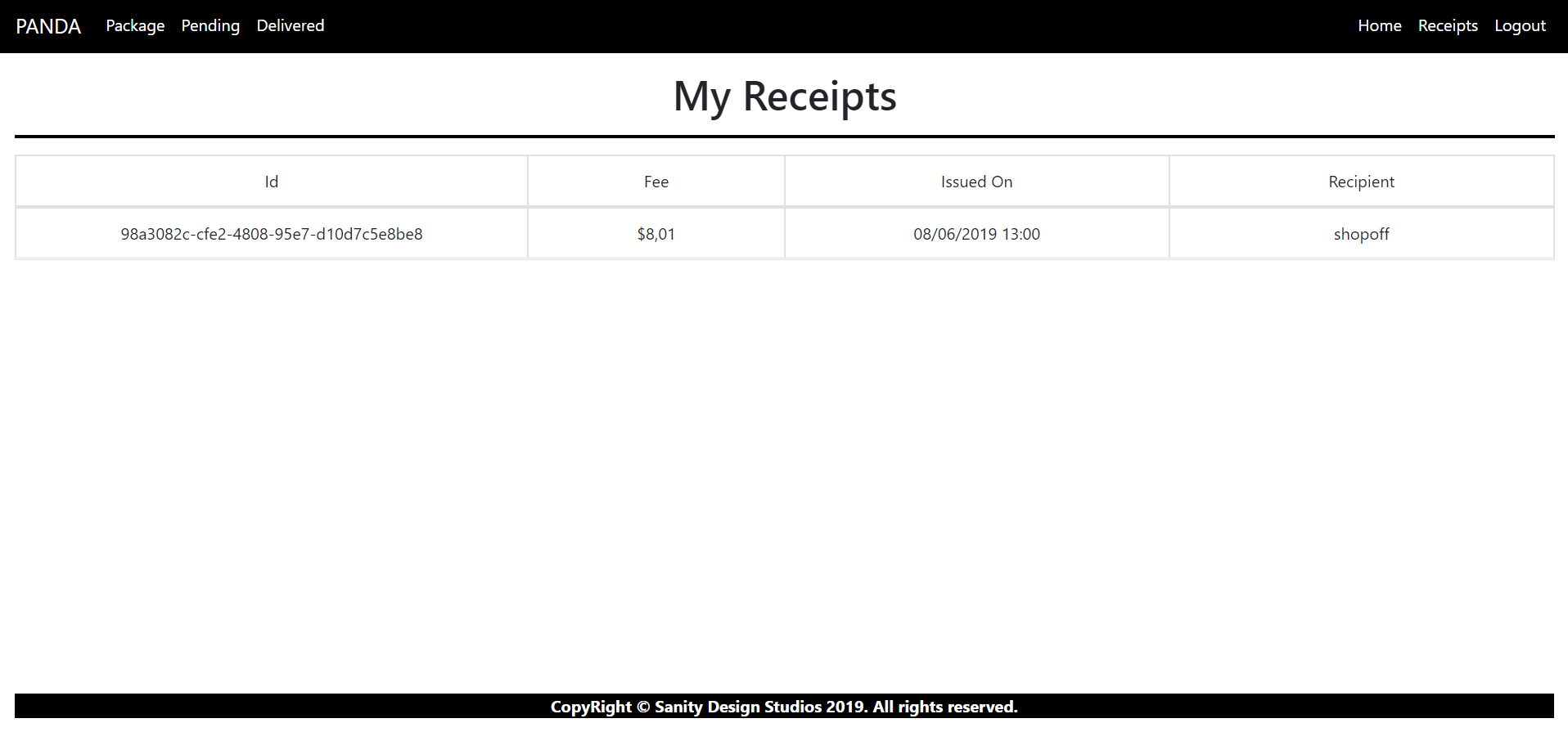
### User Pages

These are the **templates** and **functionalities**, accessible by Users (**logged in** users).

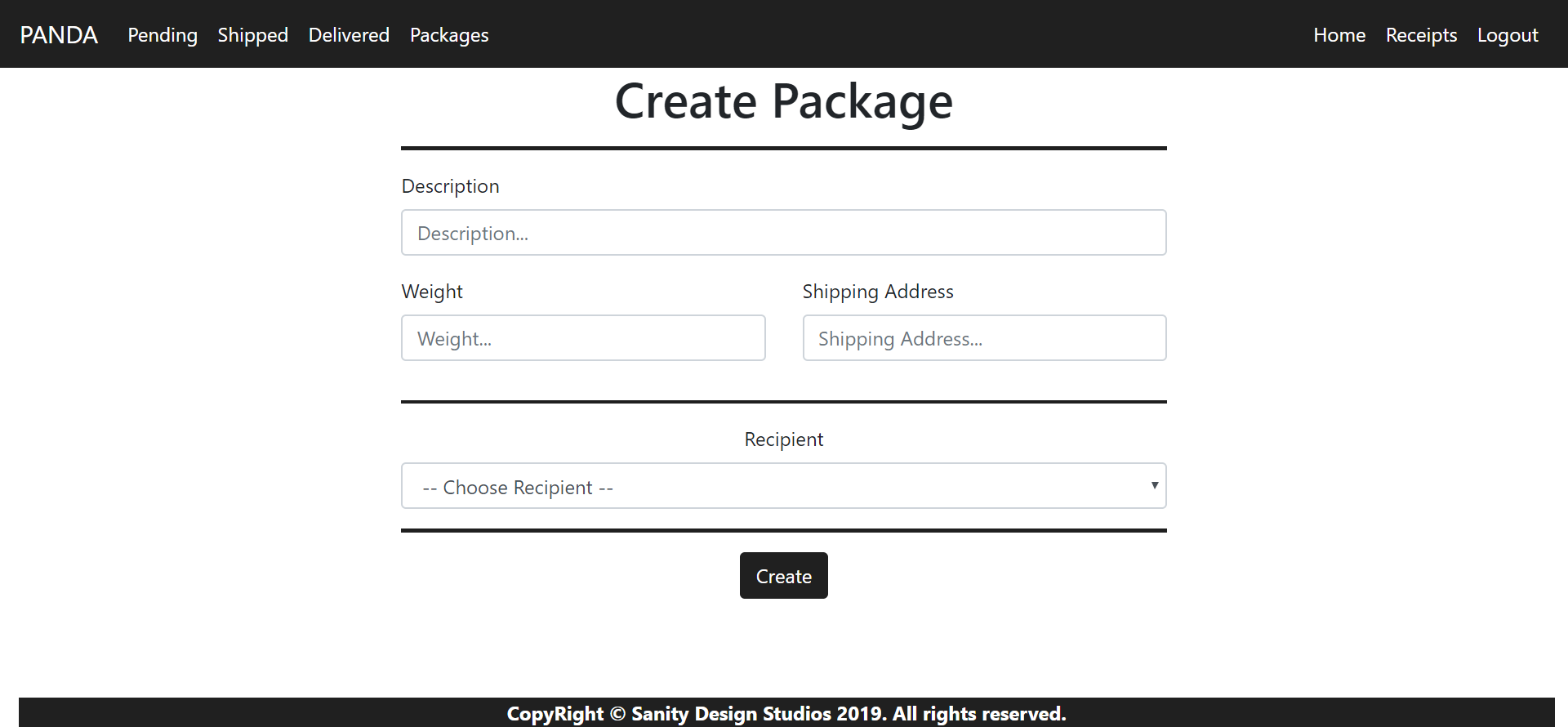
#### LoggedIn Index Page (route="/Home/Index") (logged in user)



#### Receipts Page (route="/Receipts/Index") (logged in user)

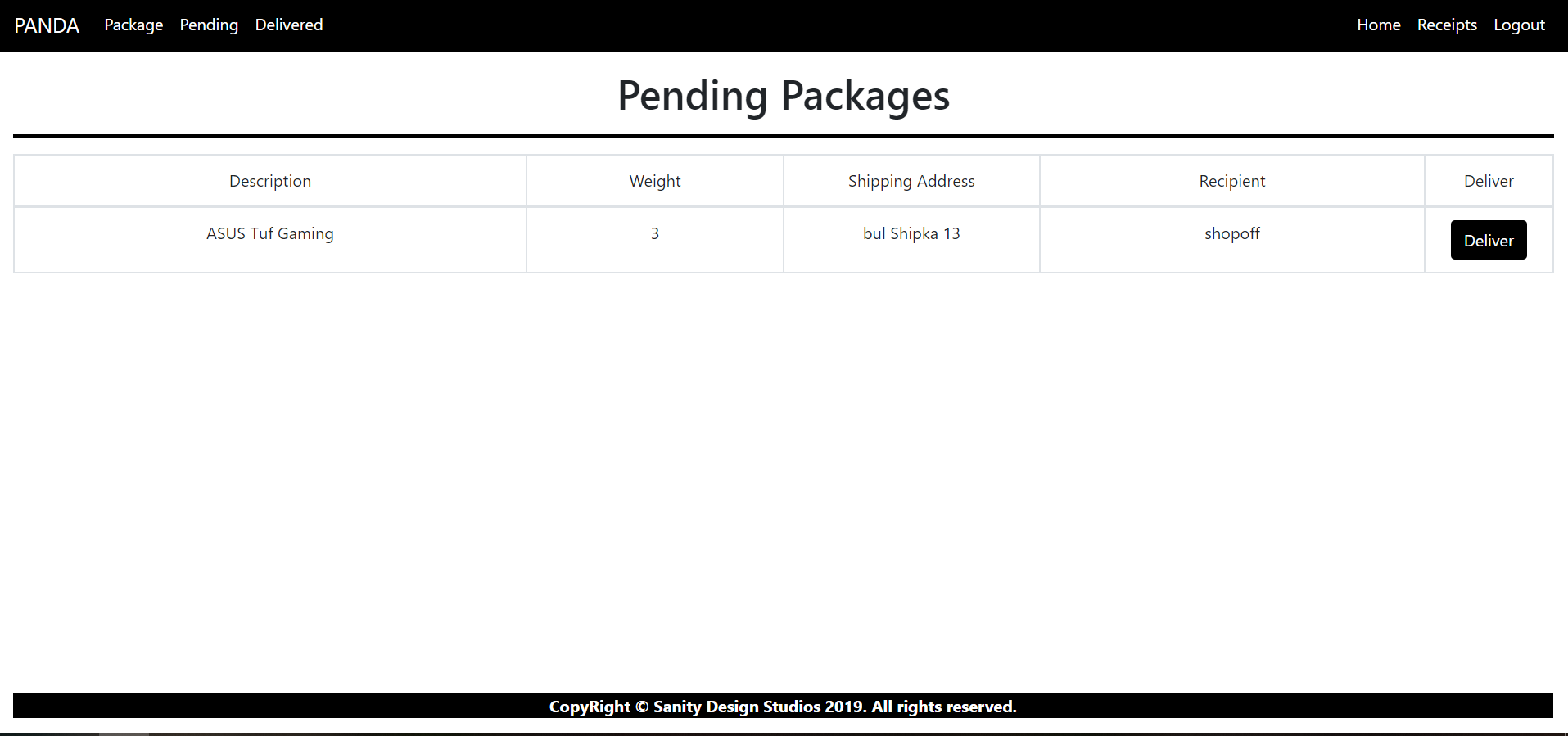


#### Create Package Page (route="/Packages/Create") (logged in)

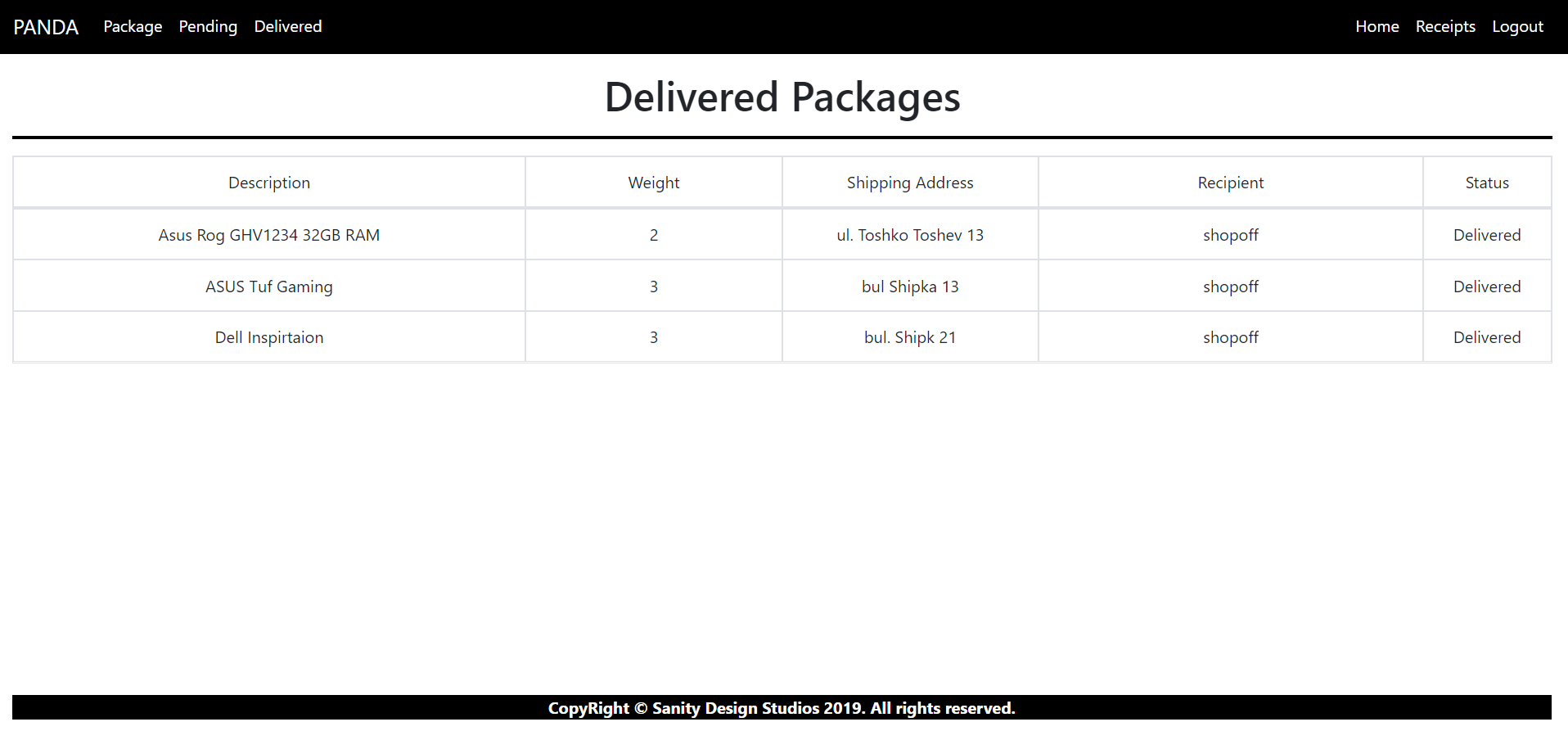


**NOTE**: The **Recipients** field is a **dropdown** **menu**, with **all the registered users** in the **application** (including **yourself**). The **primarily selected** **option** should be “-- Choose Recipient --" and it should be **disabled**, so that it **cannot** be **selected**.

#### Pending Packages Page (route="/Packages/Pending") (logged in)



#### Delivered Packages Page (route="/Packages/Delivered") (logged in)



Some of the templates have been given to you in the application skeleton, but the others will be for you to implement, so make sure you implement them correctly. You can use the given ones as helpers.

**NOTE**: The templates should look **EXACTLY** as shown above.

**NOTE**: The templates do **NOT** **require** **additional** **CSS**. Only **bootstrap** and the given **style.css** are enough.

**NOTE**: In the given **style.css** you’ll see some helpful classes "bg-panda", "text-panda", "border-panda". “border-panda” will help you with the **border** of the **blocks** on the **Index Pages**.

## Functional Requirements

The Functionality Requirements describe the functionality that the **Application** must support.

The **application** should provide Guest (not logged in) users with the functionality to:

* Login
* Register
* View the Guest Index page

The **application** should provide Users (logged in) with the functionality to:

* Logout
* View their Packages
* View their Receipts
* View all Pending Packages
* View all Delivered Packages
* Deliver Packages

### Users

Users have Packages. Users can **create** Packages. By default, the status of a package is "**Pending**". When a Package is **delivered**, status is change from "**Pending**" to "**Delivered**" and Receipt is created.

**Users** can also **create** Packages for a specific User.

* They can also **view** all of his Pending Packages.
* They can also **view** all of his Delivered Packages.
* They can also **view** all of his Receipt.

### Packages

When Packages are **created**, they are **created** with a Description, a Weight, a Shipping Address and a Recipient User.

* Upon creation, the Status of a Package should be set to Pending.

##### Pending Packages

A Pending Package, can be Delivered by clicking on the [Deliver] button from the Pending Packages Page. At that moment the Package Status becomes "Delivered" and receipt is **generated** to the User for that Package

* All Pending Packages are presented on the Pending Packages Page.

##### Delivered Packages

* All Delivered Packages are presented on the Delivered Packages Page.

### Receipts

Receipts are just data entities. They are **created** when a Package is Delivered by its Recipient User.   
A Receipt should be **created** with a Package and a Recipient User.

Upon creation, a Receipt's Fee should be **set** to the Package's Weight **multiplied** (\*) by 2.67.

Upon creation, a Receipt**'**s IssuedOn should be set to the **current moment**.

## Security Requirements

The Security Requirements are mainly access requirements. Configurations about which users can access specific functionalities and pages.

* Guest (not logged in) users can access Index page and functionality.
* Guest (not logged in) users can access Login page and functionality.
* Guest (not logged in) users can access Register page and functionality.
* Users (logged in) can access User LoggedIn Index page and functionality.
* Users (logged in) can access User Receipts page and functionality.
* Users (logged in) can access Logout functionality.
* Users (logged in) can access the Package Create page and functionality.
* Users (logged in) can access the Pending Packages page and functionality.
* Users (logged in) can access the Delivered Packages page and functionality.
* Users (logged in) can access the Package Deliver functionality.

## Code Quality

Make sure you provide the best architecture possible. Structure your code into different classes, follow the principles of high-quality code (**Single Responsibility**, **IoC** and **Interface Segregation**). You will be scored for the Code Quality and Architecture of your project.

**NOTE:** The **Service Layer** is **required**

## Data Validation

* In case of invalid data during the **login**, redirect to "**Users/Login**"
* In case of invalid data during the **registration**, redirect to "**Users/Register**"
* In case of invalid data during the **package creation**, redirect to "**Package/Create**"

**NOTE:** Use the ValidationSisAttributes

## Scoring

This section describes how the scoring of the Exam will be made.

### Database Requirements – 10 points.

### Template Requirements – 10 points.

### Functionality – 40 points.

### Security – 10 points.

### Application Structure – 10 points.

### Data Validation – 20 points.