Rest APIs Application

Project Requirements Document

CNLjunggren

<https://github.com/CNLjunggren>

Revision 1.0

Date: 2/27/2020

**Table of Contents:**

[**Use Cases:** 2](#_Toc33709734)

[User: 2](#_Toc33709735)

[Admin: 3](#_Toc33709736)

[**System Design:** 4](#_Toc33709737)

[Top-Down Approach: 4](#_Toc33709738)

[Legend for Diagram: 4](#_Toc33709739)

[**Technical Requirements:** 5](#_Toc33709740)

[System Technical Requirements 5](#_Toc33709741)

[Software/Hardware Requirements 5](#_Toc33709742)

[**System Logical Model:** 7](#_Toc33709743)

# **Use Cases:**

## User:

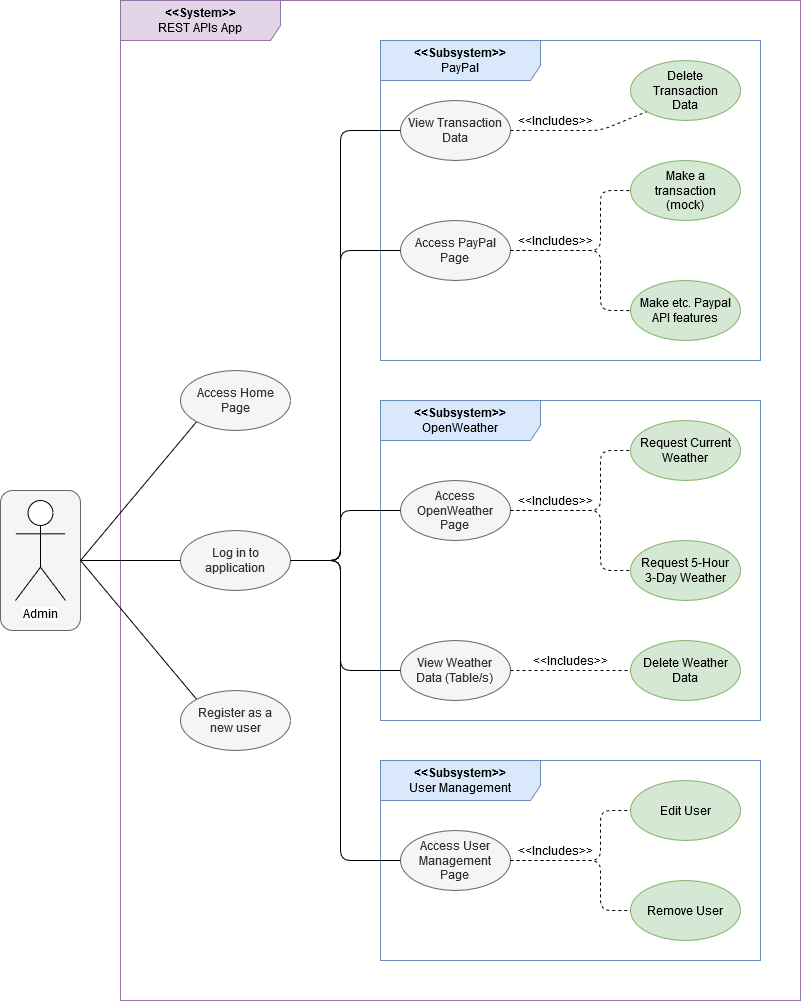
A screenshot of a cell phone

Description automatically generated

This application utilizes two roles that make up the functions and uses for this site. A description of both roles and their available actions is provided below;

* User: The standard role of the application that can log into the application, register as a new user, and access the home page. Once logged in, the User has access to the OpenWeather and PayPal API service pages. From here the user can:
  + Make a mock PayPal transaction.
  + Make requests for etc. PayPal APIs.
* Admin (Diagram on next page): The administrator tole of the application that shares all the actions of a User and more. The administrator actions that only an Admin can take are:
  + Access the User Management Page to view and edit/remove existing users from the database.
  + Access the Weather Management Page to view and remove weather data from the database.
  + Access the Transaction Management Page to view and remove weather data from the database.
    - Not all Transaction data will be visible for privacy/confidentiality reasons.

## Admin:



# **System Design:**

## Top-Down Approach:

A close up of a sign

Description automatically generated

## Legend for Diagram:

* Rounded Rectangles: Webpages/Views.
  + Blue: Standard User access level and higher.
  + Orange: Admin access level only.
* Rectangles: Functions/Methods groups.
  + Green: Standard User access level and higher.
  + Yellow: Admin access level only.

# **Technical Requirements:**

|  |  |
| --- | --- |
| System Technical Requirements | |
| Requirement | Description |
| Database | This project depends on a database that can store and allow for CRUD management of its entries for at least users, weather response data, and transaction data. |
| OpenWeather APIs Page | This web project requires at least one data validation supported form page for the OpenWeather APIs that allows for a user to select the type of weather call they want to request along with the location and measurement system used for the data returned. There will at least be one response page dedicated for displaying this data and error messages. |
| PayPal APIs Page | This web project requires at least one data validation supported form page for PayPay’s Sandbox REST API that allows for a user to Make and process a mock transaction. There will at least be one response page dedicated for displaying a confirmation or error message. |
| Data Management Pages | This web project requires at least three pages dedicated for database data viewing and modification through CRUD functionality. This includes a table and CRUD support each for users, transaction history, and weather data. Sensitive information will not be displayed or viewable by admins for security reasons. |

|  |  |
| --- | --- |
| Software/Hardware Requirements | |
| Requirement | Description |
| Microsoft Visual Studio  (C# & AspNet packages) | Microsoft Visual Studio is an IDE made by Microsoft that supports various languages and functionalities. For this project, I will be utilizing the C# language with AspNet’s .NET MVC framework. |
| Microsoft.AspNet.MVC (v5.2.4) | Microsoft .Net MVC is an MVC framework for C# that provides the required aspects for an MVC web application. This includes components such as Razor, web page functionality, and web optimization. |
| Microsoft.AspNet.Razor  (v3.2.4) | Razor is a portion of .Net MVC that is used to create the project’s website pages. |
| Microsoft.AspNet.WebPages  (v3.2.4) | WebPages is a portion of .Net MVC that that contains the core assemblies for .Net MVC web pages. |
| Microsoft.Bcl.AsyncInterfaces  (v1.1.0) | Provides the IAsyncEnumerable<T> and IAsyncDisposable interfaces and helper types for .NET Standard 2.0. |
| Microsoft.CodeDom.Providers.  DotNetCompilerPlatform  (v2.0.0) | Replacement CodeDOM providers that utilize the new .NET Compiler Platform (“Roslyn”) compiler as a service APIs. |
| Microsoft.Web.Infrastructure  (v1.0.0) | Microsoft Web infrastructure is an assembly that is used in web projects for dynamic HTTP module registering at run time. |
| System.Buffers (v4.5.0) | Provides resource pooling of any type for performance-critical applications that allocate and deallocate objects frequently. |
| System.Memory (v4.5.3) | Provides types for efficient representation and pooling of managed, stack, and native memory segments and sequences of such segments, along with primitives to parse and format UTF-8 encoded text stored in those memory segments. |
| System.Numerics.Vectors  (v4.5.0) | Provides hardware-accelerated numeric types, suitable for high-performance processing and graphics applications. |
| System.Runtime.  CompilerServices.Unsafe  (v4.7.0) | Provides the System.Runtime.CompilerServices.Unsafe class, which provides generic, low-level functionality for manipulating pointers. |
| Systems.Text.Encoding.Web  (v4.7.0) | Provides types for encoding and escaping strings for use in JavaScript, HyperText Markup Language (HTML), and uniform resource locators (URL). |
| System.Threading.Tasks.  Extensions (v4.5.3) | Provides additional types that simplify the work of writing concurrent and asynchronous code. |
| System.Value.Tuple (v4.5.0) | Provides the System.ValueTuple structs, which implement the underlying types for tuples in C# and Visual Basic. |
| Bootstrap (v4.4.1) | Bootstrap is going to be utilized alongside razor to create the visual appearances of the project’s website. Bootstrap is a library/framework for CSS. |
| popper.js (v1.16.0) | A requirement when installing version 4.4.1 of Bootstrap. Popper serves as a library that allows you to manage your poppers. |
| Modernizr (v2.6.2) | Modernizr is a JavaScript library used for CSS3 and HTML 5 technologies which is a big help in creating HTML web pages. |
| jQuery (v3.4.1) | jQuery is a JavaScript library used for various HTML, event handling, etc. simplification and ease of use applications. |
| jQuery Validation (v1.19.1) | The Validation jQuery plugin is used for data validation simplification and customization. |
| Microsoft.jQuery.Unobtrusive.  Validation (v3.2.11) | jQuery plugin that unobtrusively sets up jQuery.Validation. |
| Newtonsoft.Json (v12.0.3) | A high-performance JSON framework for use with the .NET frameworks. This framework will be used for JSON file deserialization from API responses, and possibly to serialize data into JSON files. |
| Microsoft SQL Server | SQL Server is going to be the database used for this project as it is a built-in database system in Visual Studio. |

# **System Logical Model:**