## 

## 

**Renewable Home**

**Software Design Document**

Version 2

12/6/17

Michael Dardis

Lead Software Engineer

Prepared for

COT6931- Computer Science Project

Fall 2017

## 

**TABLE OF CONTENTS**

**Revision History……………………………………………………………………………………………………..3**

**Document Approval…………………………………………………………………………………………………3**

**1. Introduction………………………………………………………………………………………………………..3**

1.1 Purpose of this Document……………………………………………………………………………………3

1.2 Scope of the Development project…………………………………………………………………………..3

1.3 Objectives…………………………..………………………………………………………………………….3

1.4 Definitions, Acronyms, Abbreviations……………………………………………………………………….4

1.5 References..........................................................................................................................................4

**2. System Architecture Description……………………………………………………………………………...5**

2.1 Overview of modules/components…………………………………………………………………………...5

2.2 Structure and relationships…………………………………………………………………………………....5

2.3 Data Repository design………………………………………………………………………………………..6

2.4 Object-Oriented design..........................................................................................................................7

2.5 User interface........................................................................................................................................8

**3. Detailed description of components.......................................................................................................9**

3.1 Component descriptions.......................................................................................................................10

3.2 Data Flow Diagram...............................................................................................................................13

**4. Test documentation..................................................................................................................................14**

4.1 Interface testing.....................................................................................................................................14

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| **7/16** | **1.1** | **Created document for use** | **M. Dardis** |
| **7/23** | **1.2** | **Added in descriptions of components** | **M. Dardis** |
| **8/1** | **1.3** | **Implemented diagrams and tables with test plan** | **M. Dardis** |
| **12/4** | **2.0** | **Added finalized changes to design** | **M. Dardis** |

**1. Introduction**

*1.1* *Purpose of this Document*

This document provides a low-level and high-level description of the Renewable Home web application. This application allows a user to see the possible effects of using or investing into renewable energy resources for their home. This is done by the user inputting information about their home, and then seeing how different energy sources can affect their utility cost. The use of this document can be utilized by software developers and project managers.

*1.2 Scope of the Development Project*

The Renewable Home application will be hosted online and can be viewed by any user accessing it. On the start screen, the user will begin by entering in information about their home, including the total square feet, as well as their monthly utility consumption. There is a helpful dropdown showing the consumption rates by state, in case the user doesn’t have their information on hand, or if they are curious as to how other states match up. There will then be a screen depicting various forms of energy use, with a slider to indicate the percentage use. A button will be displayed besides each source to provide helpful information, and to also provide links to outside websites in case the user wants to learn more about each source of energy. The final page will use the energy indicators to determine the final calculation on the next screen. This will display their yearly energy cost and the cost per square foot for the user to see how much a sustainable energy lifestyle would cost.

*Objectives*

The objectives of the overall application include an increased awareness on renewable energy resources to help promote sustainability within residential communities. One way to do that is to have an application to show different renewable energy sources on the market to see how current trends can shape the costs of the user. Information about each source will be provided and helpful links will also be provided to raise awareness about each source of energy. An important objective includes having a clean interface to easily navigate through the different pages. The user interface will also be able to adapt to different devices and screen sizes to maximize its usability.

*1.3 Definitions, Acronyms, Abbreviations*

* User - Any person accessing the web application
* KW - Kilowatt
* CO2 - Carbon Dioxide
* UI – User Interface
* ASP.NET – Framework used for application development
* Visual Studio – Microsoft’s integrated development environment

1.4 *References*

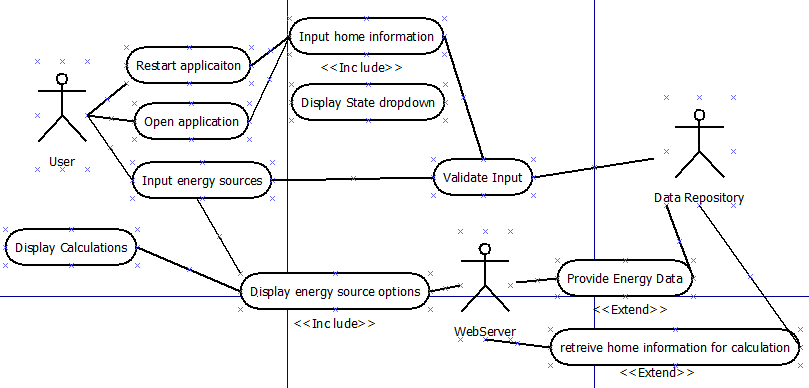
Rozenblat, Lazar. "Your Guide to renewable Energy." Renewable Energy Sources: Cost Comparison. N.p., n.d. Web. 11 June 2017. <<http://www.renewable-energysources.com/>>

Simmons, C. Instant Responsive Web Design : Learn the Important Components of Responsive Web Design and Make Your Websites Mobile-friendly. Birmingham, UK : Packt Publishing, 2013

Amin, Adnan Z. "How renewable energy can be cost-competitive." UN Chronicle P8 52.3 (2015): 8-11. *UWF Libraries.* Web. 4 June 201*7*

Bellomy, Ian1. "The Grain of the Browser: What Designers Should Know about the Craft of Web Design." International Journal of Visual Design, vol. 11, no. 1, Mar. 2017, pp. 1-15. EBSCOhost

**2. System Architecture Description**



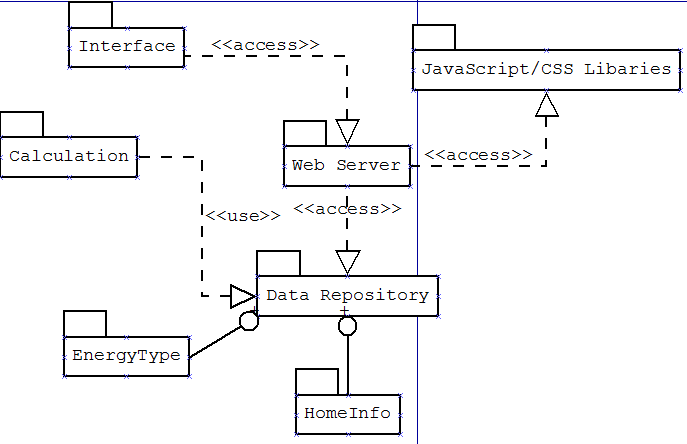
**Figure 1.** Renewable Home UML Use Case Diagram

2.1 *Overview of Modules/Components*

The application is built in Visual Studio, and has the ability o be deployed on any web hosting service, such as Microsoft’s Azure cloud service. The data repository is used to provide the current information about a type of energy. This is then tied to the energy calculation table to provide the specified outcome on the final page. The main class for the application will create instances of the HomeInfo class, the EnergyType class, and the HomeController class to handle all get and post requests to the application. These instances are created during runtime. During the application, the user will enter in information which includes their home information, as well as their chosen combination of energy. The web server will provide validation checks to make sure the user is entering the correct type of data. The final results page will display the yearly cost of the chosen resource allocation, and it will include cost per square feet.

*2.2 Structure and relationships*

The application interface will handle user commands and will access the web server being used to then access the data repository for information. The repository will then provide the application with information to initialize objects to use for the energy calculation for the user.



**Figure 2.** Renewable Home UML Package Diagram

Data Repository Documentation

* EnergyData
  1. EnergyData
     1. Initializes the energy data upon application start up
     2. Data structure includes a list of EnergyTypes
        1. EnergyTypes(name, description, cost, state(optional))
           1. Solar
           2. Wind
           3. Hydropower
           4. Biomass
           5. Geothermal
* EnergySourceRepo
  1. Creates instance of Energy Data, and enables the program to retrieve specific information on each source at runtime

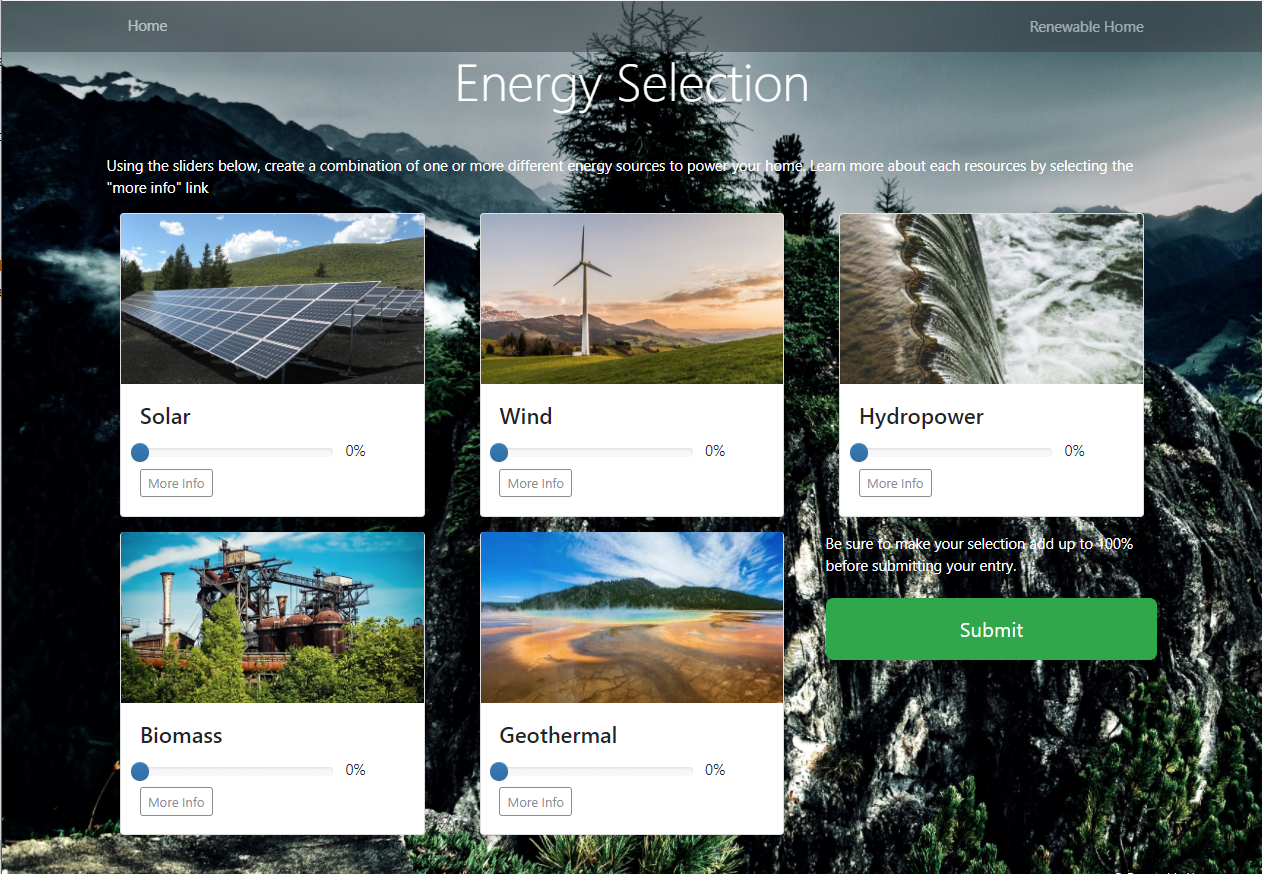
2.4 *Object-Oriented Design*

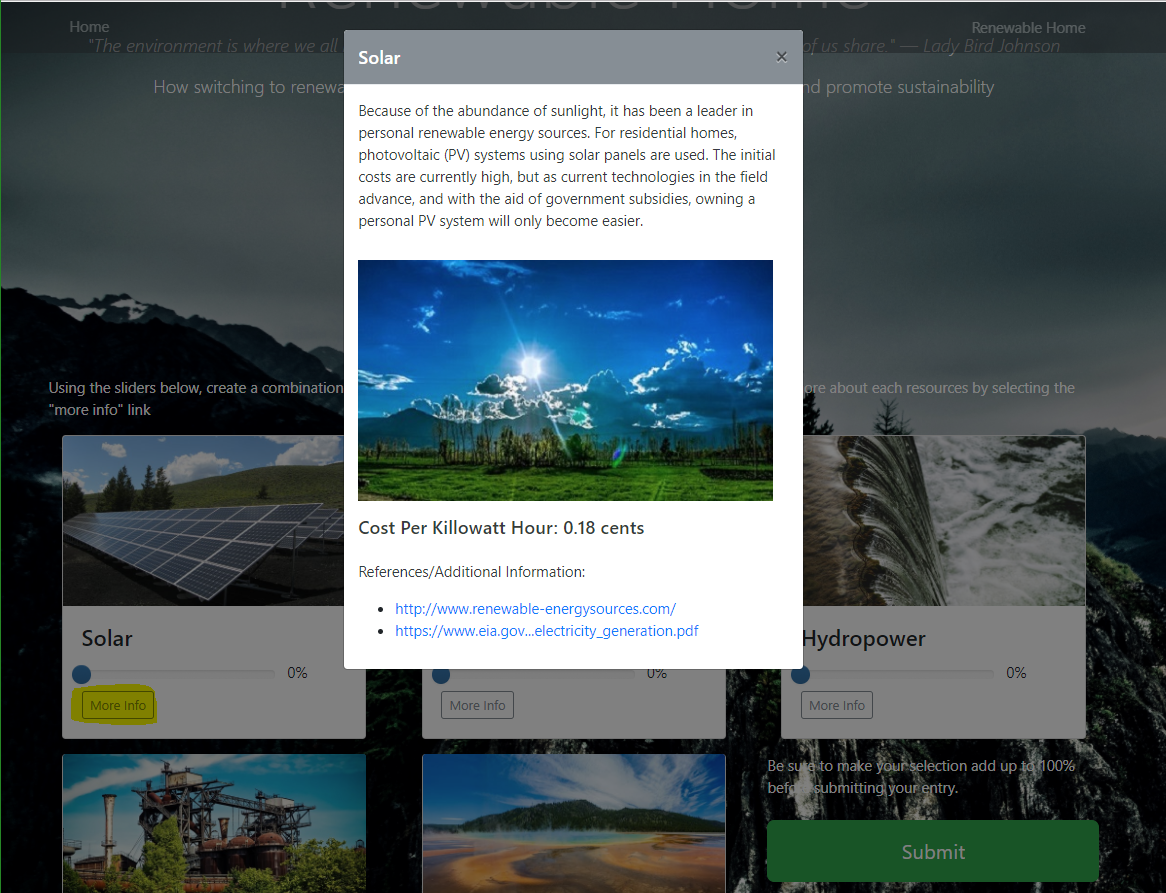


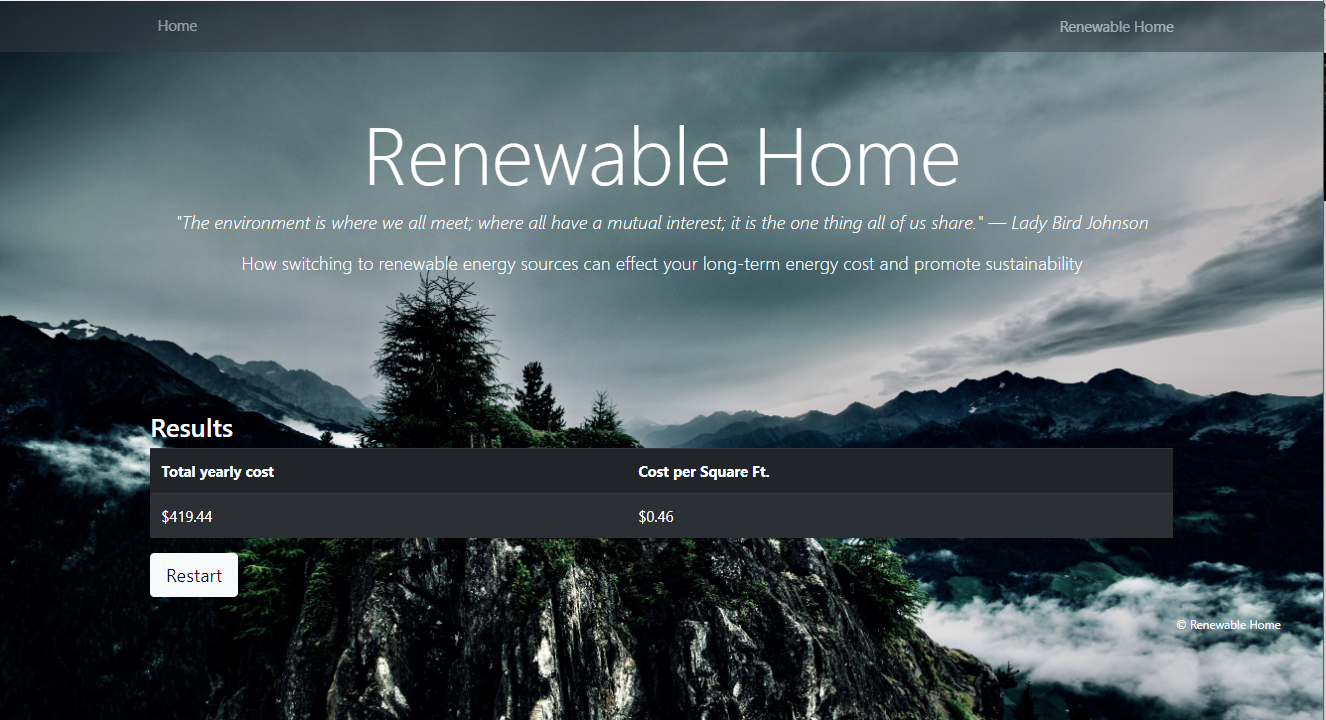
**Figure 4.** Renewable Home Classes

2.5 *User Interface*









*2.6 Mobile Interface*

|  |  |  |
| --- | --- | --- |
|  |  |  |

**3. Detailed description of components**

***HomeController Class***

*Component identifier*

HomeController Class

*Purpose*

This class is to handle the post and get requests during the web application at runtime. It is also used ot initialize all variables and the data repository stored in memory. When handling post requests, server-side validation methods handle user input.

*Function*

This class handles get and post requests from the view and uses the model to post data back to the client.

*Subordinates*

This class is required for a runtime of the program, and is extended from the Controller namespace.

*Dependencies*

The MVC framework library to use the Controller class extension.

*Interfaces*

The Controller interface from the MVC framework of .NET

*Data*

This class utilizes the model classes of EnergyData and EnergySourcesRepo. They are used to gather and initialize the data for the application to use.

**HomeInfo Class**

*Component Identifier*

HomeInfo Class

*Purpose*

This class is used to create a Home object to belong to a User, and also to be used for the final energy calculation.

*Function*

This class stores details about a User’s home. This includes the size of the home in square feet, and the amount of energy consumption per month in Kilowatt hours.

*Subordinates*

This class relies on the EnergyType class for home energy use classification and the User for assignment. There can be one or many different EnergyTypes associated with a Home for energy calculation.

*Dependencies*

Each state is tied to a Kilowatt usage number, and can be assigned to a home, if the user choses to use that functionality.

*Interfaces*

This class does not utilize any interfaces.

*Data*

Instance variables are used to hold information about the square footage of the home. This is passed along during the application using the controller as the user goes through the steps to finalize the calcualtion.

***EnergyType Class***

*Component identifier*

EnergyType Class

*Purpose*

This class is designed to hold instances of different types of energy for use in a home energy usage calculation.

*Function*

This class is required for the final energy calculation, and is tied to a Home object. A Home can contain one or many EnergyType objects to use for its energy usage when performing the final calculation of results.

*Subordinates*

There are no subordinates for the EnergyType class

*Dependencies*

The EnergyType class is dependent on the initalization of the energy data repository. Energy types must be assigned to the repository class and is used to display the information on the energy selection page. This page

*Interfaces*

This class relies on the user interaction from the web interface. Depending on the chosen spectrum of Energy types, EnergyType cost information will be created and assigned to that Home.

***EnergyData Class***

*Component identifier*

EnergyData Class

*Purpose*

This class holds the infomraiton on each type of energy use. It uses the EnergyType class to build and contain a series of objects to use for the application.

*Function*

This class gives the controller data about each type of energy and helps with the final calculation.

*Subordinates*

The class EnergySourceRepo uses this class to pass data to the controller.

*Dependencies*

The EnergyData class is dependent on the EnergyType class to be able to build the list of entries. The class then has its data initialized by the EnergySourceRepo class.

*Interfaces*

This class does not utilize any interfaces.

*Data*

This class uses EnergyType objects to create the storage for the data repository. These individual objects hold the information on each type of energy that is then used for the application.

***EnergySourceRepo Class***

*Component identifier*

EnergySourceRepo class

*Purpose*

This class initializes the application’s data from the EnergyData class.

*Function*

This class gives the controller data about each type of energy by retrieving the list of energy sources and provides helpful getter methods about each source of energy.

*Subordinates*

This class has no subordinates.

*Dependencies*

This class is dependent on the EnergyData class to have data to use.

*Interfaces*

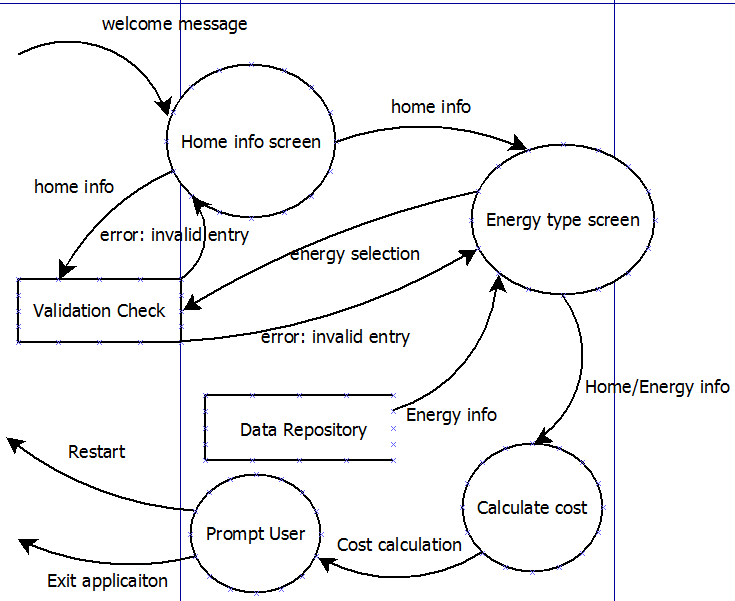
This class does not utilize any interfaces.

*Data*

This class gets the different types of energy by calling the methods contained in the EnergyData class. It is then used to pass this data back to the controller

*3.2 Data Flow Diagram*

The following diagram illustrates the flow of information through the Renewable Home application, beginning with the validation of the energy consumption and the square feet of the home that is input by the user. The data then flows to ending with displaying the energy calculation result. The user then has the option to restart from the beginning to see how different variations could affect the final total.



**Figure 5: Data-Flow Diagram**

**8. Test Documentation**

*8.1 Requirements Review*

During development, there may be changes to requirements. Final testing involves comparing the final set of requirements to the final design. The review is with the client to sign off on the final changes and to accept the handoff of the project.

8.3 *Interface Testing / Calculation Logic Testing*

The following test cases are used for the web interface and how validation would work for user input. Different data points are then inputted to see how the pages respond. Different sizes of web browsers are also used to make sure the web application is responsive and still maintains its functionality and view.



**Figure 6.** Renewable Home Interface Test Cases