**Megha Sharma**

**Pavani Sree Lakshmi Penumathsa**

**Reshma Reddy Thumma**

**Vinutha Nuchimaniyanda**

**Project Report E-Smart Application**

CS-5551 ASE

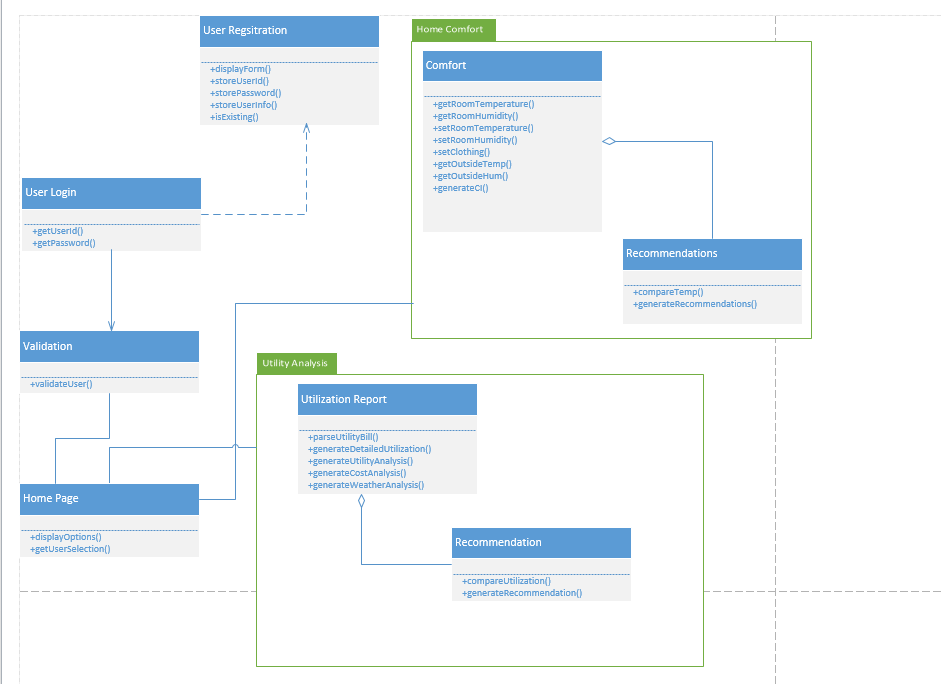
We have been able to implement the following features as part of the first increment of the project.

1. User login, registration & validation
2. Home Page
3. Energy Analyzer includes energy utilization analysis with weather conditions, cost analysis with consumption and detailed utilization over a period of one year.
4. Home Comfort and Recommendation system

**Web Client/Mobile UI Design**

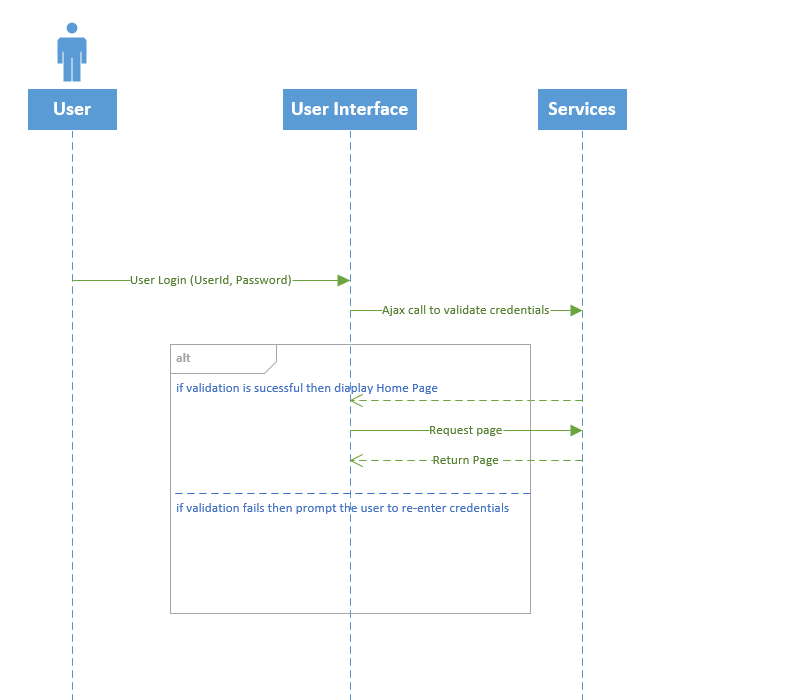
To design the UI, we have used a combination JQuery, HTML5 and CSS. Most of the features have been supported by google and other web apis.

**Class Diagram**

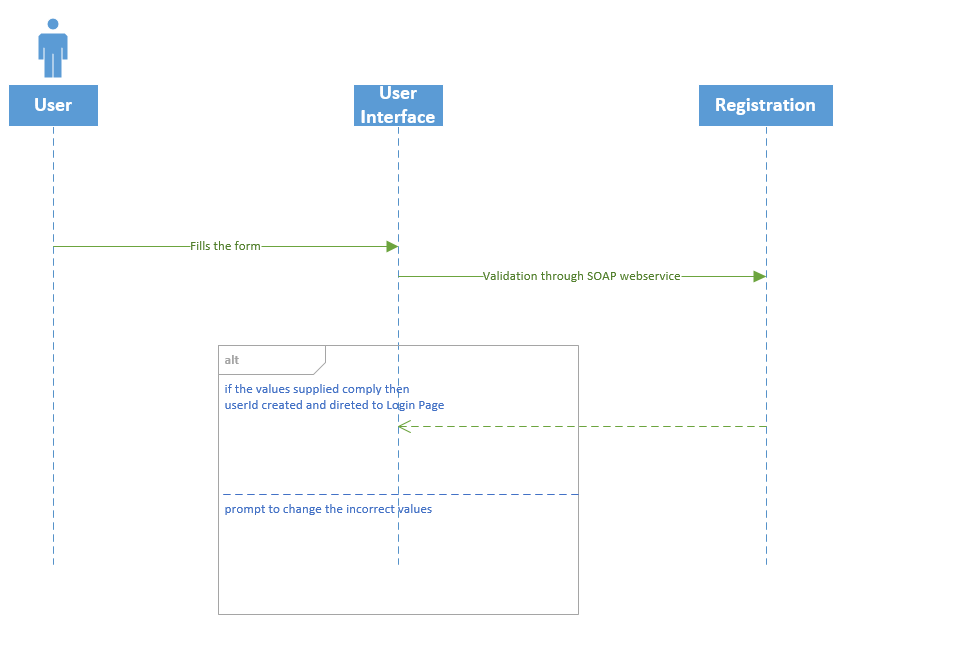


**Sequence Diagrams**

1. User Login and Service request



1. User Registration



**Implementation Details**

The details of the features are below:

**1) User Login & Registration**

**Webservices/API:**

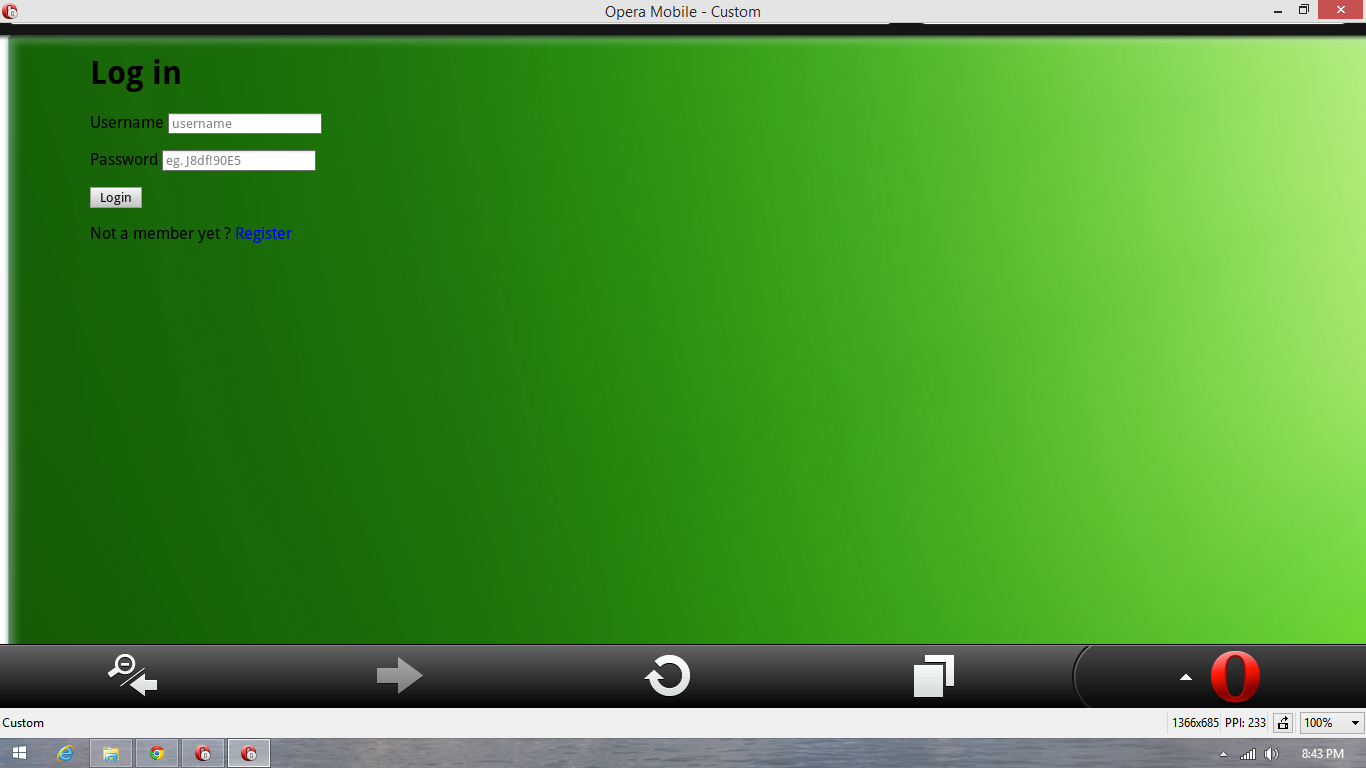
* We plan to use SOAP webservices to perform verification of user credentials and for storing user credentials in the database.

**Design & Implementation details:**

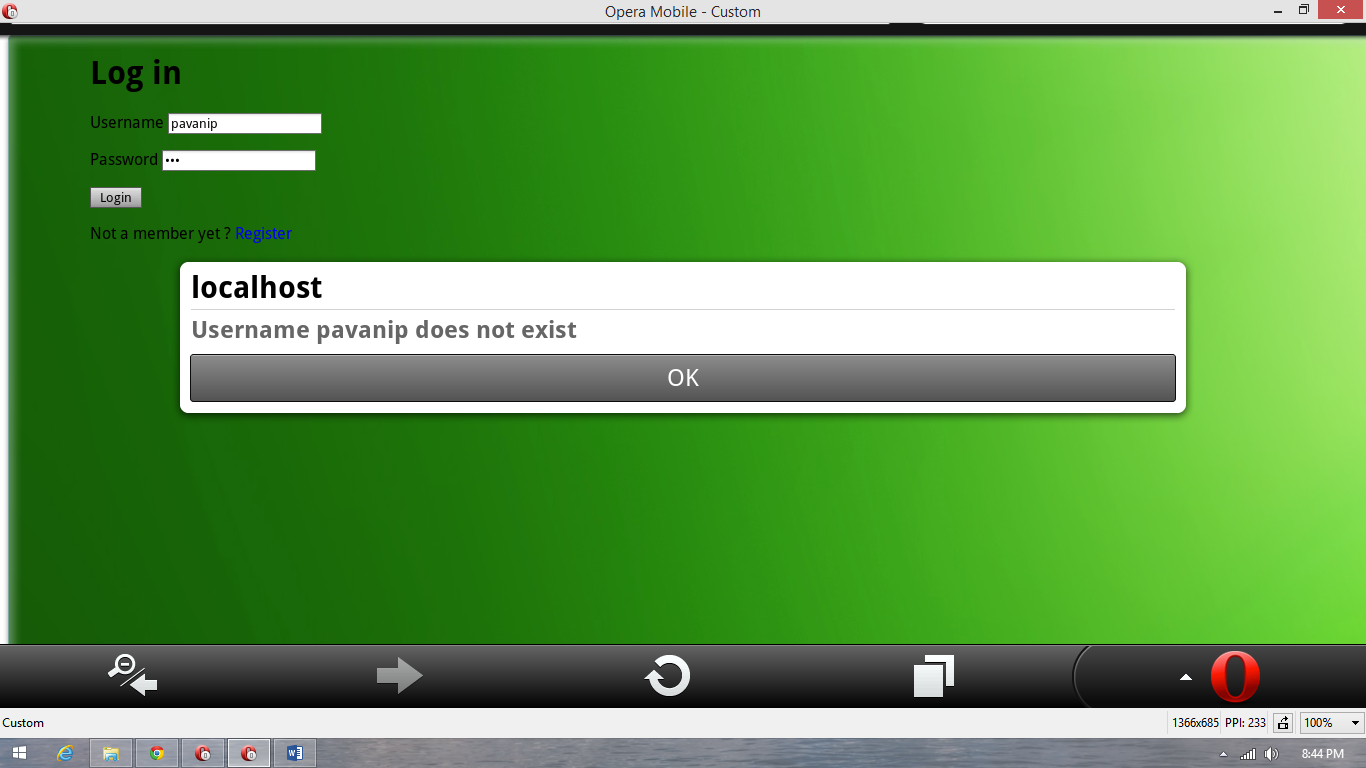
Local HTML storage has been used to store the credentials over sessions. We plan to use SOAP webservices to perform verification of user credentials and for storing user credentials in the database.

This page will allow the user to login to the E-Smart website. The validation is done based on the username and password entered by the user. If the validation is successful, it navigates to the website’s Homepage. If either username or password doesn’t match it shows the error message and will not allow the user to login to the website.

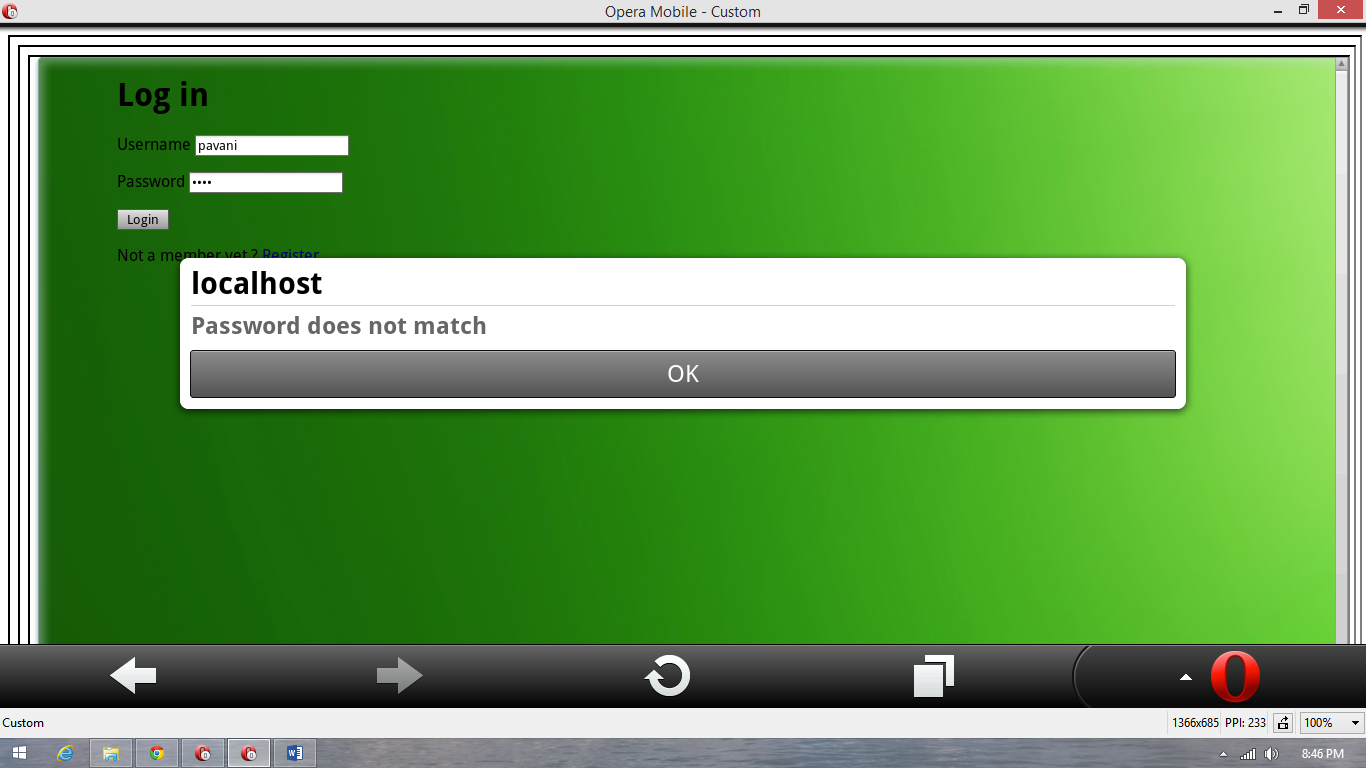
Below is the login screen.



If the user doesn’t exist then an alert is generated.



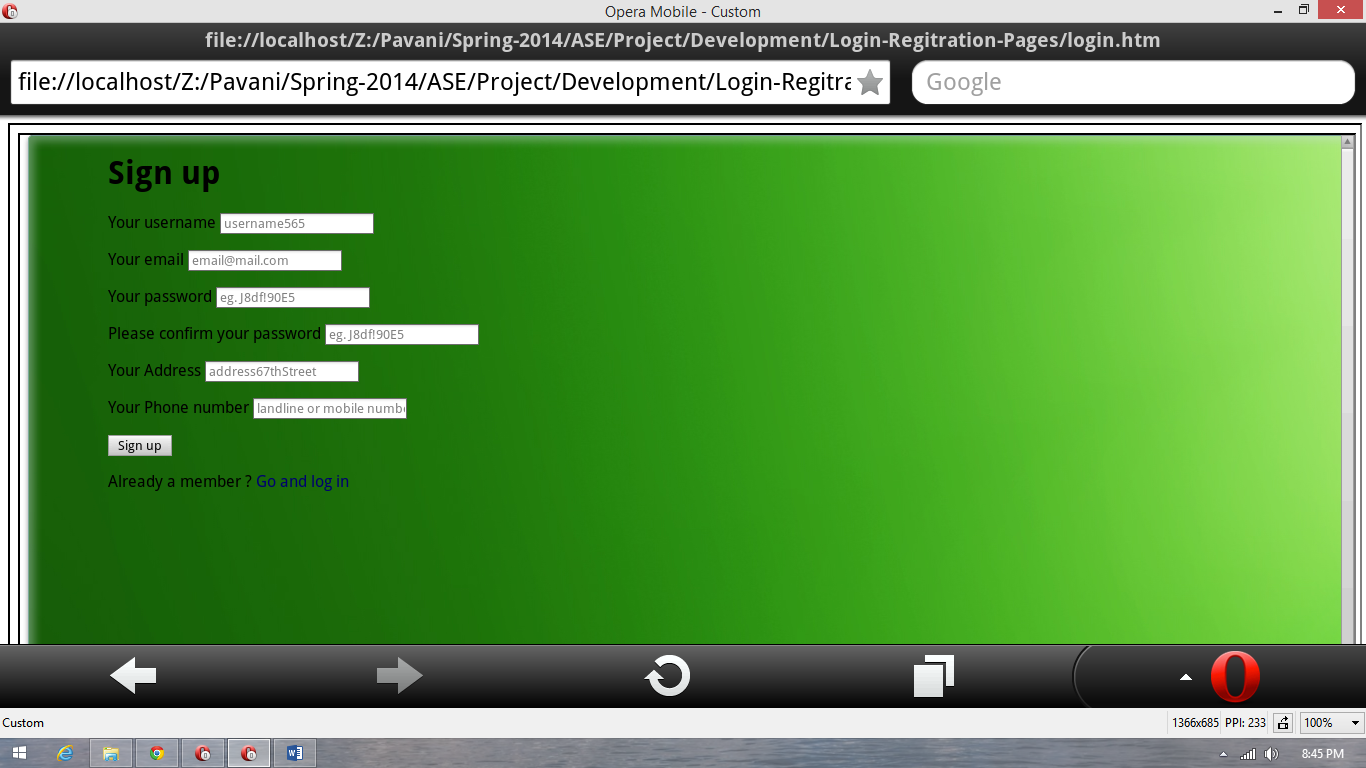
If the password doesn’t match then the below alert is generated.



On successful login the homepage is displayed.

Below is the user registration page.

This page will allow the user to register for accessing the E-Smart website. The **Register** button on the Login page navigates the users to this registration page. The user has to fill in his details, select username and password for this website. Once the registration is successful, it will navigate to the login page. The user can click on **Go and login** buttononthis page to go back to login page.



**2) Energy Analyzer**

**Webservices/API:**

* Google Visualization API Source
* Visualization ComboChart for Energy Analysis Monthly data.
* Visualization AreaChart for Cost difference data

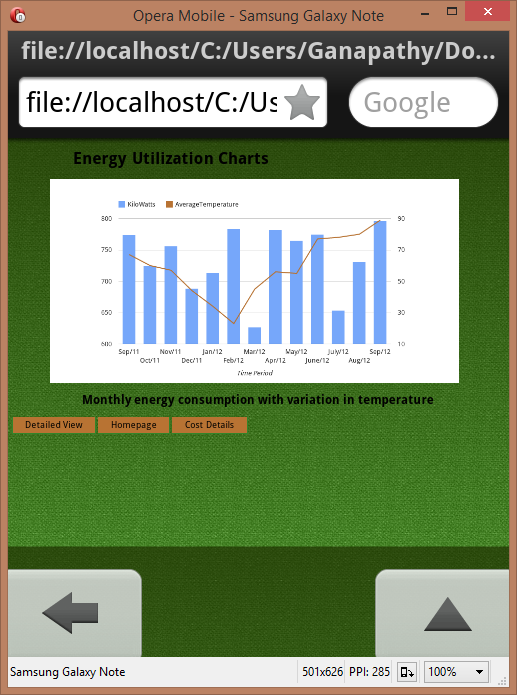
**Design & Implementation details:**

This service takes in the user’s utility bill in green button format and parses it to get the energy utilization details (inputting and parsing of green button utility bill is yet to be implemented). A number of analysis reports and charts are then generated with the utilization data.

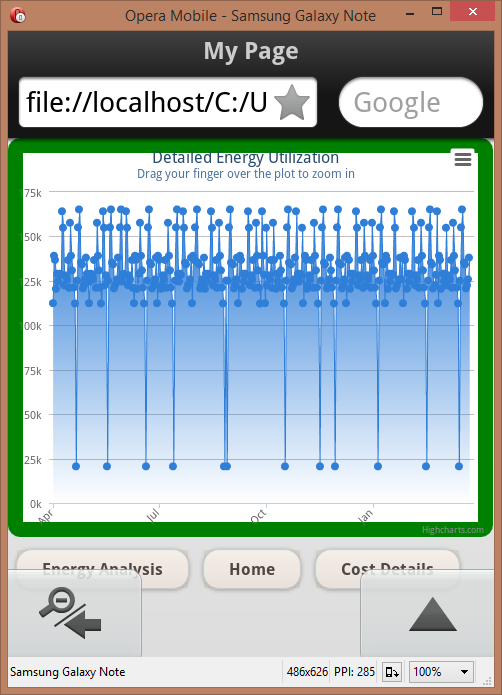
These charts give a complete picture of how much energy was used that allows the user to observe his energy consumption. Based on these charts, he can then use the other features in the website such discussion board or tips for the day to cut down his bill.

There are three charts.

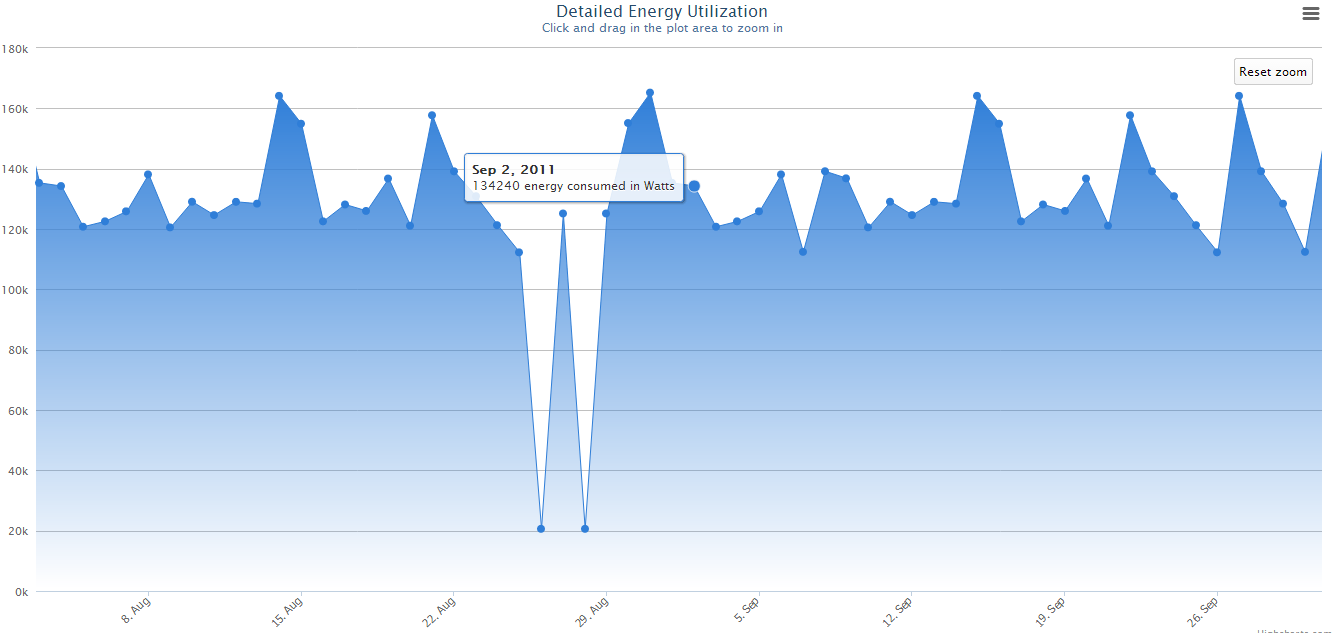
1. Energy Utilization chart: This chart provides information of energy utilization over period of 12 months. This chart shows the variation in energy utilization based on the temperature outside.



2. Detailed Utilization chart: This chart provides detailed analysis about how much energy was utilized each day of the month.



User has option to zoom in on the particular date to get detailed numbers. Zoom in feature does not work in opera. Below is the screenshot in the webclient.



1. This chart shows the difference in cost which is related to kilowatt data in Energy Analysis page



**3) Home Comfort**

**Webservices/API:**

* The geocode function supported by the browser has been used to detect user’s current location.
* The weather webservice by wunder “api.wunderground.com” has been used to populate the outside temperature and humidity.
* The jQuery UI sliders have been used for creating sliders on the home comfort page.
* The “google.visualization.Gauge” has been used to create the comfort index gauge.

**Design & Implementation details:**

On successful login the user is directed to the E-Smart homepage.

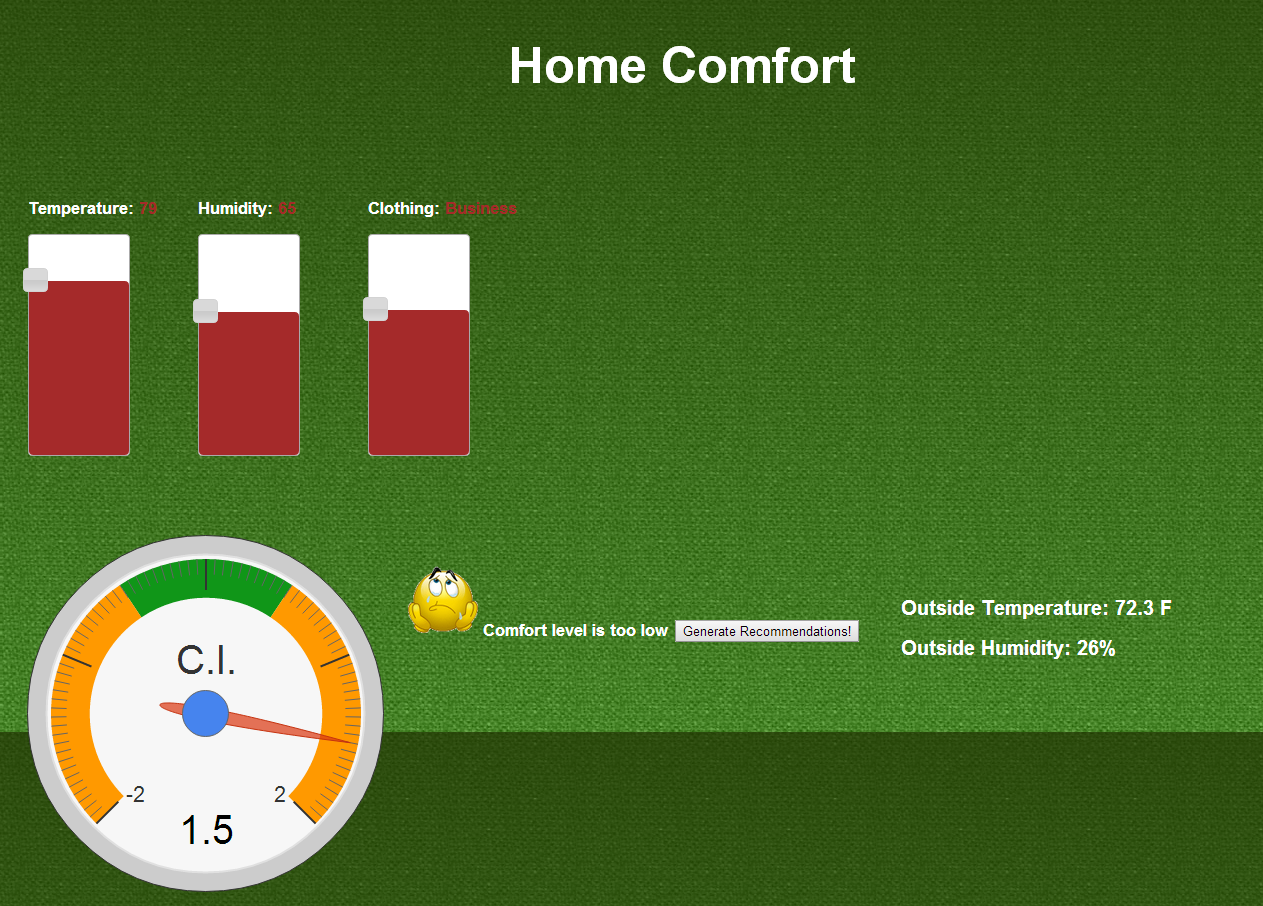


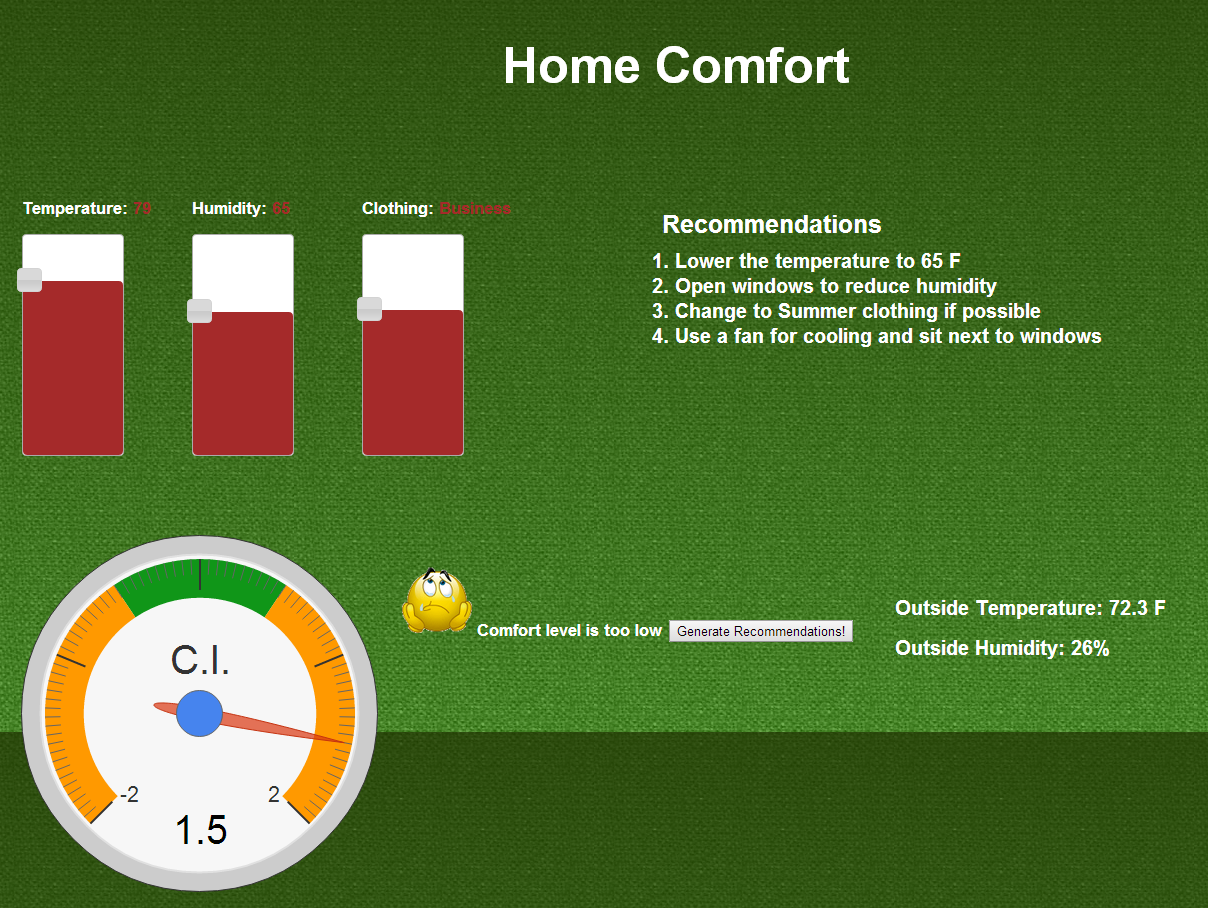
User Clicks on the Home Comfort Image.



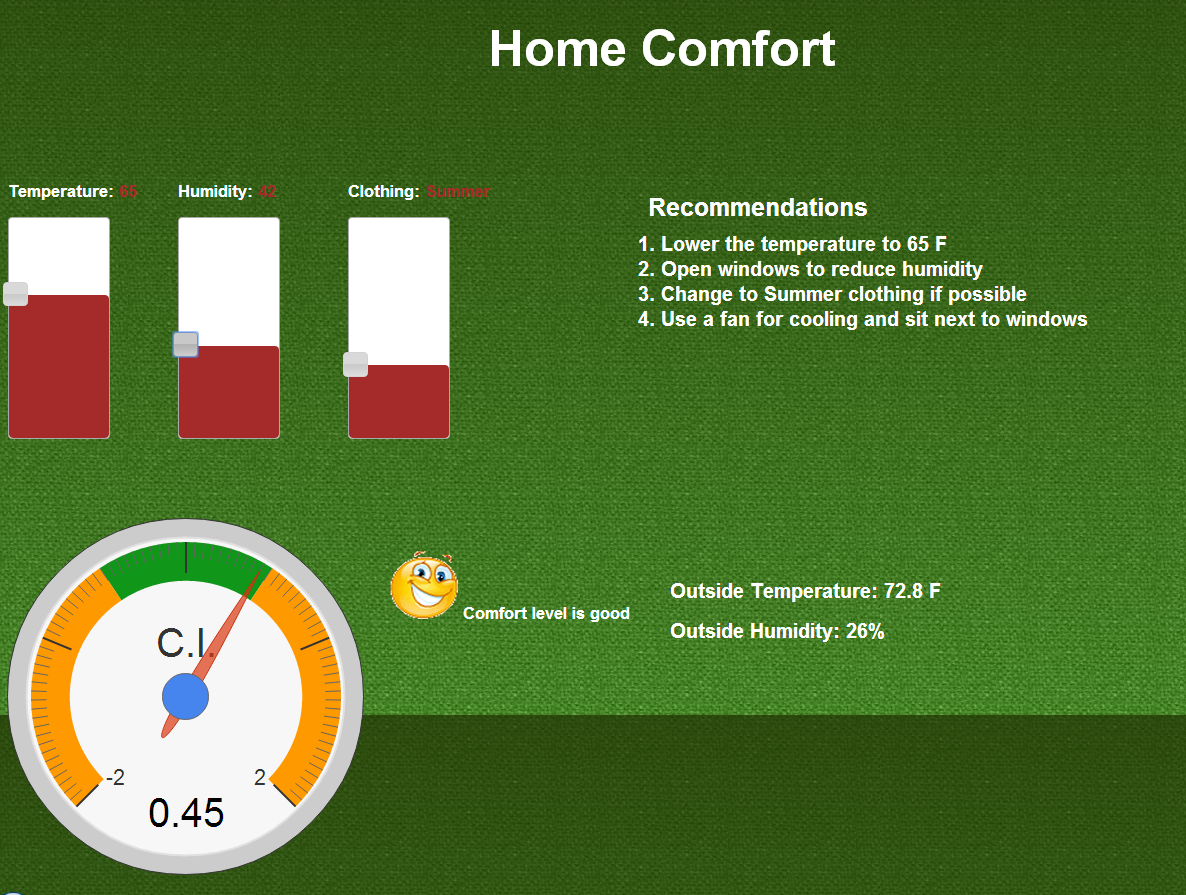
The Page displays the user’s current home temperature and humidity and computes the Comfort Index (Comfort Index computation using formulae is yet to be implemented) of his home environment. This page also shows the outside temperature and humidity and displays a message providing an explanation of the comfort index.

If the comfort index is outside the comfort zone in green then it also shows a generate recommendation button (dynamic recommendation generation is yet to be implemented)

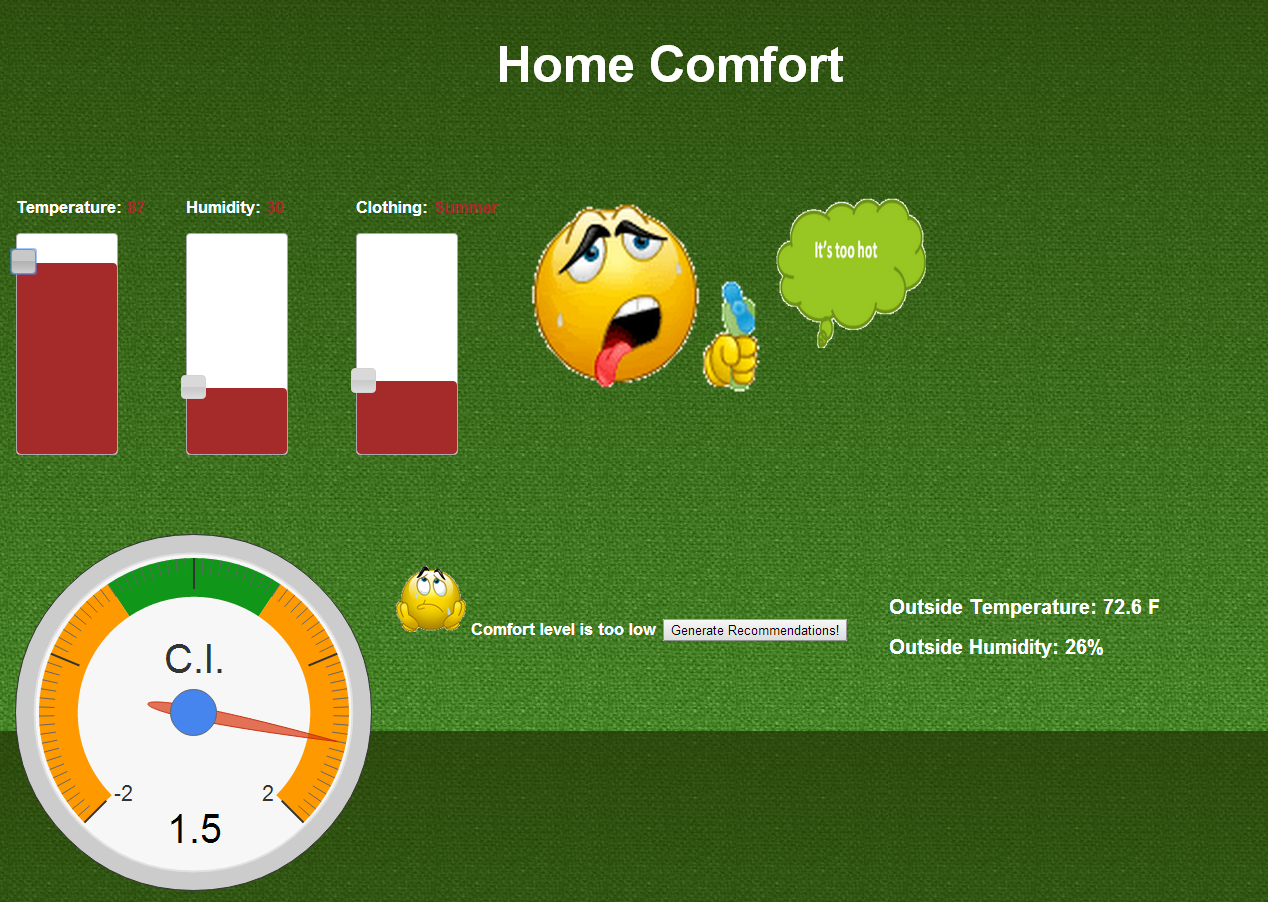


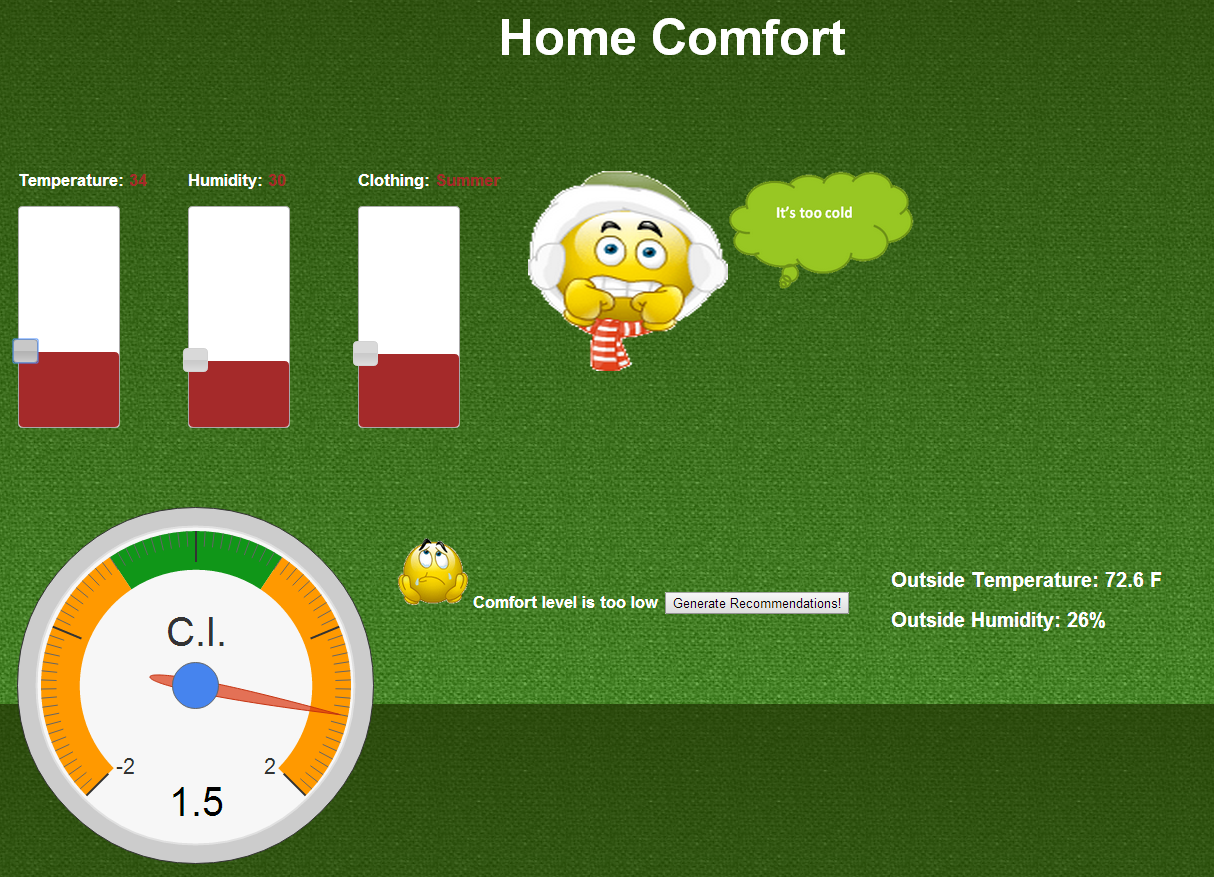
When the user clicks on the Generate Recommendations a list of recommendations are displayed. These recommendations are created by performing computation on his current home environment.

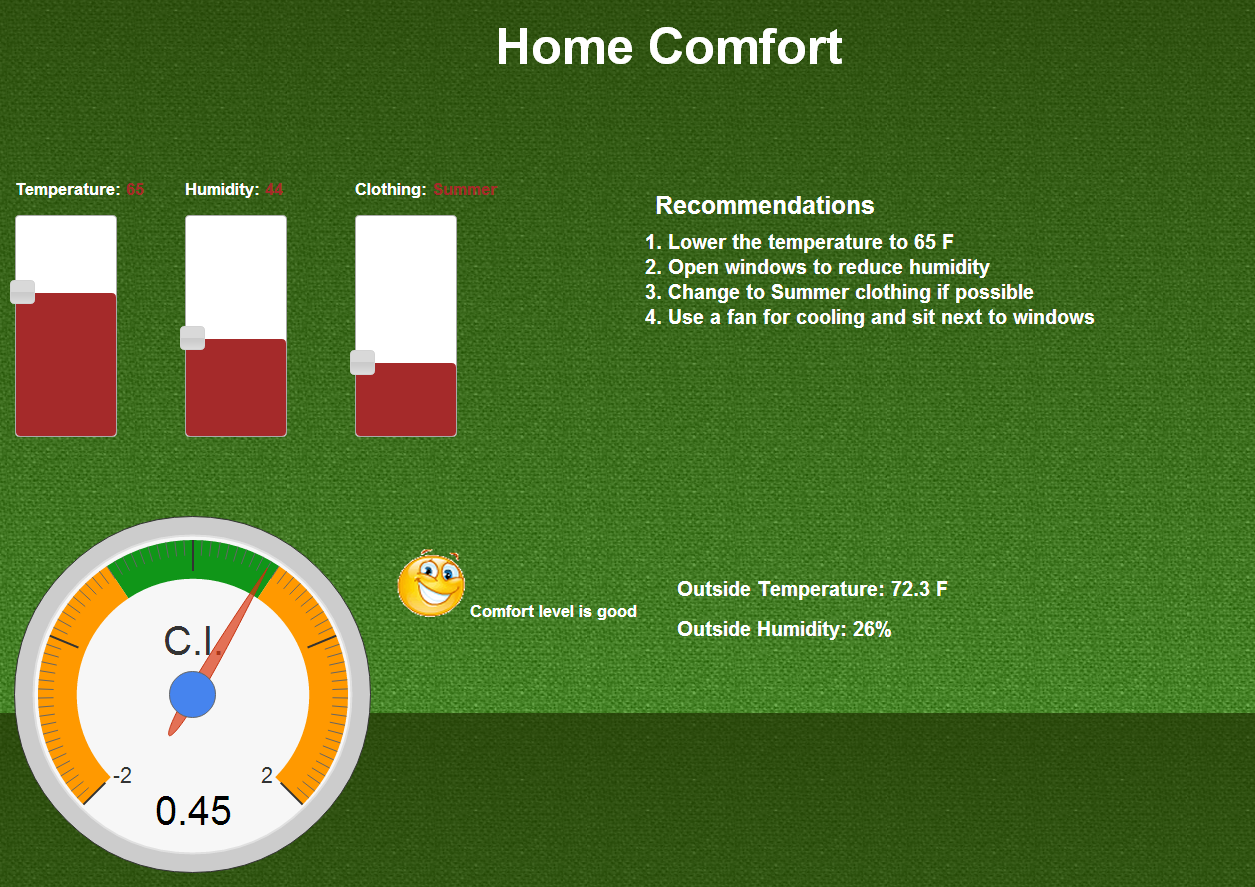
The user then tries to change the temperature, humidity and clothing based on the recommendations and the new comfort index is computed and displayed.



If the user sets the temperature too high or too low then the system prompts an alert message.







The comfort index is now in the green zone so the user’s home environment is now comfortable.

**Testing**

The unit testing has been performed manually as most of the webservices used were external.

**Distribution of Work**

|  |  |  |
| --- | --- | --- |
| Responsibility | Time Taken | Person |
| Login & Registration | 20 | Pavani |
| Homepage | 4 | All |
| Energy Analyzer | 20 | Vinutha, Reshma |
| Home Comfort | 20 | Megha |
| Reporting & Unit Testing | 5 | All |
| Scrum Do | 2 | Vinutha |

**Git Hub Repository:**

[**https://github.com/mszzf/Megha-Repository/tree/master/Project-Phase1/assets**](https://github.com/mszzf/Megha-Repository/tree/master/Project-Phase1/assets)

**ScrumDo:**

[**https://www.scrumdo.com/projects/project/esmart/iteration/92362**](https://www.scrumdo.com/projects/project/esmart/iteration/92362)