TravelRate Documentation

1498 Words --- 311205F

# Problem​ ​definition​​

Recreational, international travel is a large aspect of the worldwide tourism industry, with 1.235 billion international tourists travelling in 2016 (UNWTO, 2017). However, some individuals can have trouble identifying an appropriate and respected travel location to travel to. If this problem were solved, individuals would have safer and happier holidays abroad, and the tourism industry would grow due to having more tourists willing to travel.

The web application that will be developed will be designed to address these issues. A list of countries should be available to users, who will then be able to select a city, and then read comments and ratings for that city. Furthermore, statistics about each city and country (such as population, temperature and average rating) should be available to allow users to make informed decisions.

As well as viewing all of the data, users should be able to add their own countries, cities and comments to the website, delete ones that are already there, and edit some aspects (such as the display image) of each country and city.

# Analysis​ ​of​ ​the​ ​problem

There are three categories of information that will need to be stored for this project: information about the countries, about the cities, and about the comments.

For each country, a name, national language, capital city and currency should be stored in a database so that it can be displayed for the users of the website. Also, an image URL should be stored so that the element displaying each city can have an appropriate background.

Cities will be processed similarly, with each city’s name, population, average temperature and image URL being stored in the database. Which country each city belongs to should be stored, so that the website can sort the cities by what country they belong to, and display that information to the user. Also, the average (mean) rating of the city’s comments should be calculated dynamically when the page is requested, and displayed alongside the other information.

Each comment will have a name and rating from the user, a time that the rating was placed, and the city that the rating belongs too. Alongside the comments, a graph showing the distribution of ratings should also be available.

Finally, forms will need to be available to allow users to add countries, cities and comments to the database, and processes will need to be implemented to allow users to delete entries, and to edit the images or the countries and cities.

# Design​ ​for​ ​the​ ​solution​​

Firstly, the appropriate information in the database should be displayed to the user. A horizontally-scrolling div will be used to hold all of the country blocks, with each country block displaying the country’s name, language, capital and currency (as retrieved from the database). Similarly, each country will have a horizontally-scrolling ‘div’ to hold all of its city blocks, which will display the city’s name, population, average temperature and the calculated average rating (as determined from the city’s ratings). As one of these scrolling ‘div’s will be created for every country, they should remain hidden until the user clicks the appropriate country, to avoid excess information being displayed on the screen. This will be done using JavaScript that identifies the specific country pressed (using its ID), and sets the ‘display’ style property of the city display ‘div’s on the page appropriately. Finally, each city will have a horizontally-scrolling div to hold all of its comment blocks, which will display the commenter’s name, comment, rating, and time of comment. As with the city ‘div’s, these will need to be displayed and hidden appropriately as the users clicks around using JavaScript.

All of the information being displayed will be retrieved using RedBean, which is an object relational mapping system that allows for the treatment of each country, city and comment as a PHP object instead of rows and columns in a database.

The comment div of each city should also contain a graph, generated to display the distribution of ratings and the mean rating for that city. The graph will be displayed as an image with the image source URL being a PHP page with the appropriate city ID as a URL parameter. RedBean will be used to retrieve the city’s comments from the database, and PHP image functions will be used to generate the image.

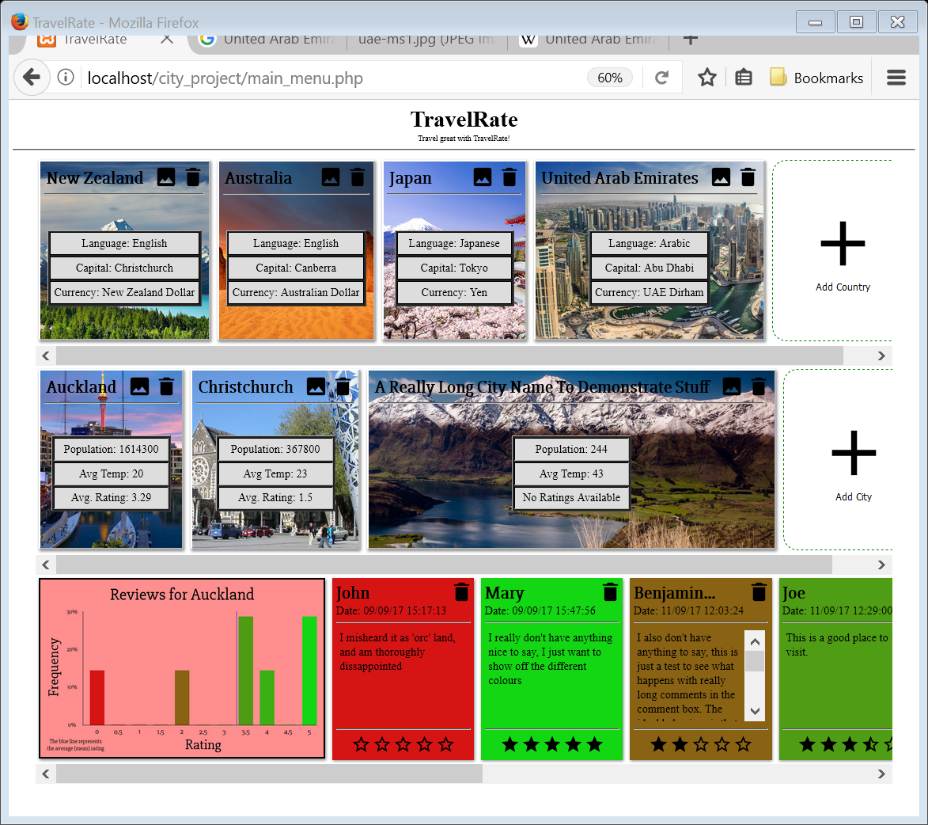
Secondly, users need to have a process for adding information to the webpage. Post forms will be used to allow the user to fill in the appropriate information about their city, country or comment, before a PHP page will use RedBean to add this information to the appropriate databases. Web standard input elements of the appropriate type (such as number, text and select inputs) will be used in the post forms for all user inputted information, and the time of comment submission will be determined in the PHP. On the generation of the main page, buttons will be created using PHP that will bring up a screen-wide, fixed, black and partially transparent div, which will contain a child element with the post form on it for users to add entries with.

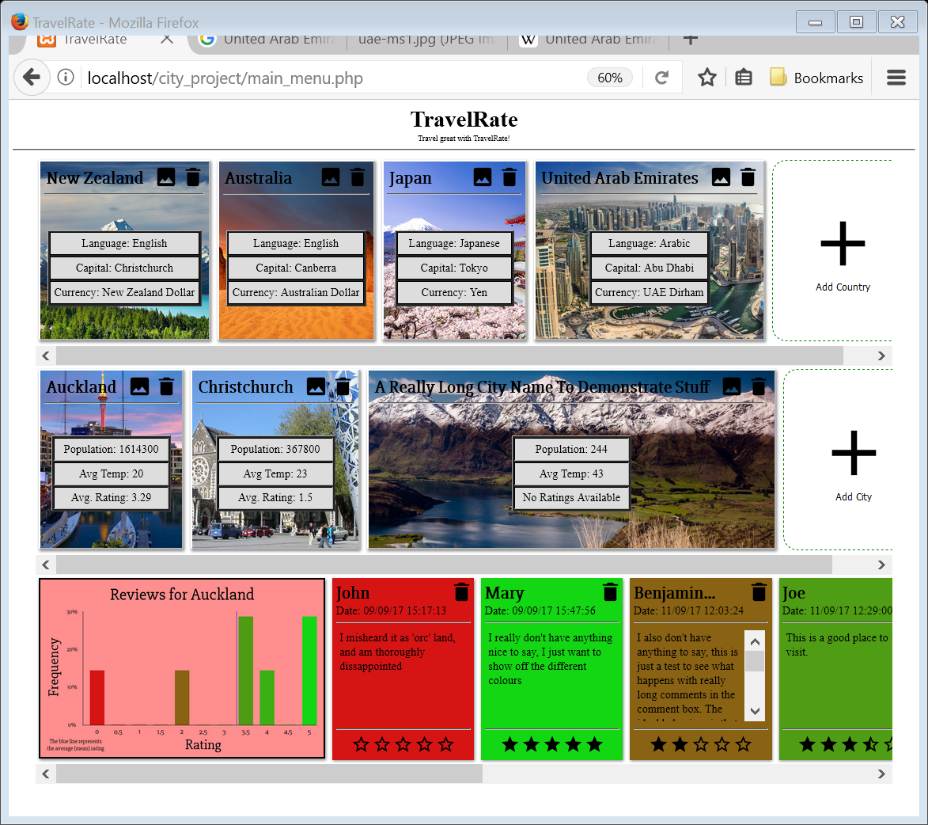
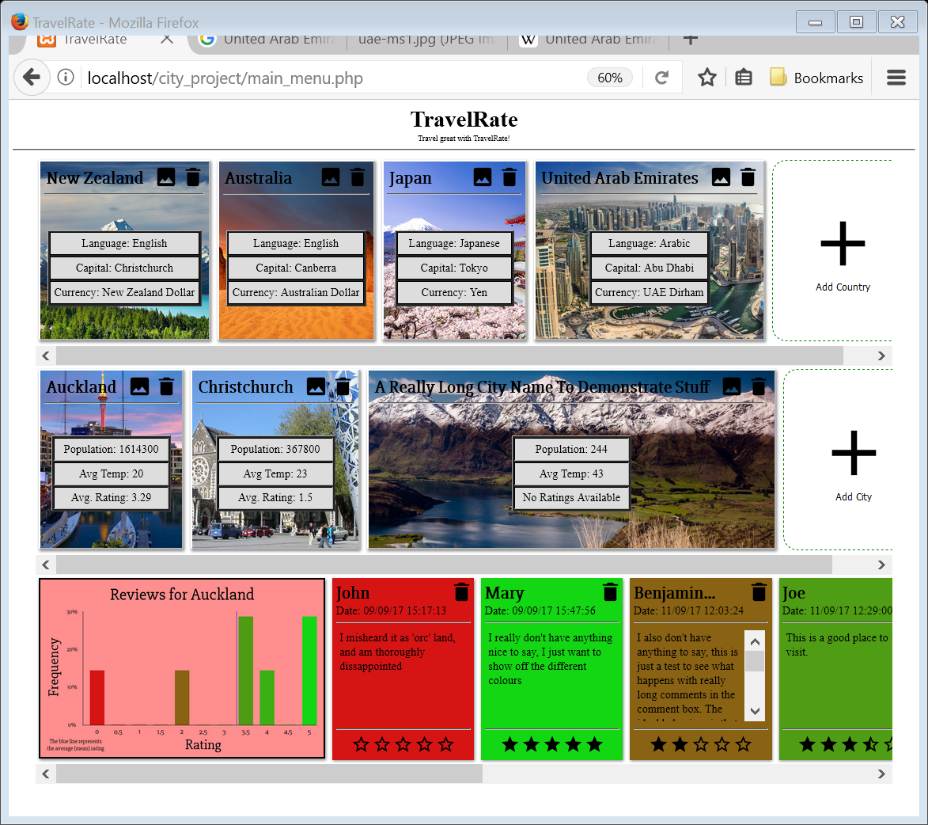
Thirdly, the user should be able to delete entries on the page, using a delete button on each country, city and comment box. Two differing techniques could be used for this functionality; each delete button could point to a URL with the appropriate entry to delete as a URL parameter, or each button could bring up a post form with a pre-filled entry with what entry to delete. The latter of these options will be used, as it will prevent users accidentally deleting entries, and allow them to change what entry they want to delete. As with adding to the database, pressing the appropriate delete ‘trashcan’ icon will bring up a black, partially transparent div that will allow the user to submit which entry they want to delete, and then a PHP script will delete it and redirect back to the main page.

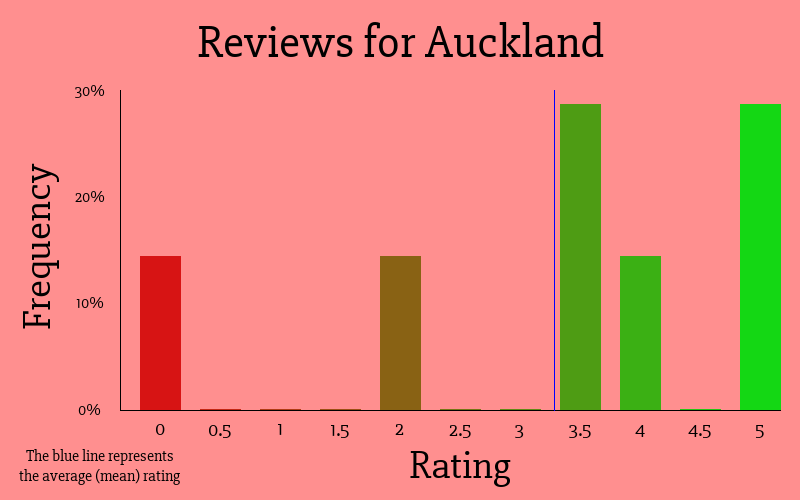
Finally, the user should be able to edit the background images of the countries and cities in the database. A similar process to entry deletion will be used, with every country and city having an edit button, which will show a pop-up menu on the page with the country/city to edit pre-filled in, and a text input for the new URL. Then, after submission of the form, RedBean will be used in a PHP script to find the appropriate entry in the database (given the country or city id), and the background URL will be changed, before redirecting back to the main page.

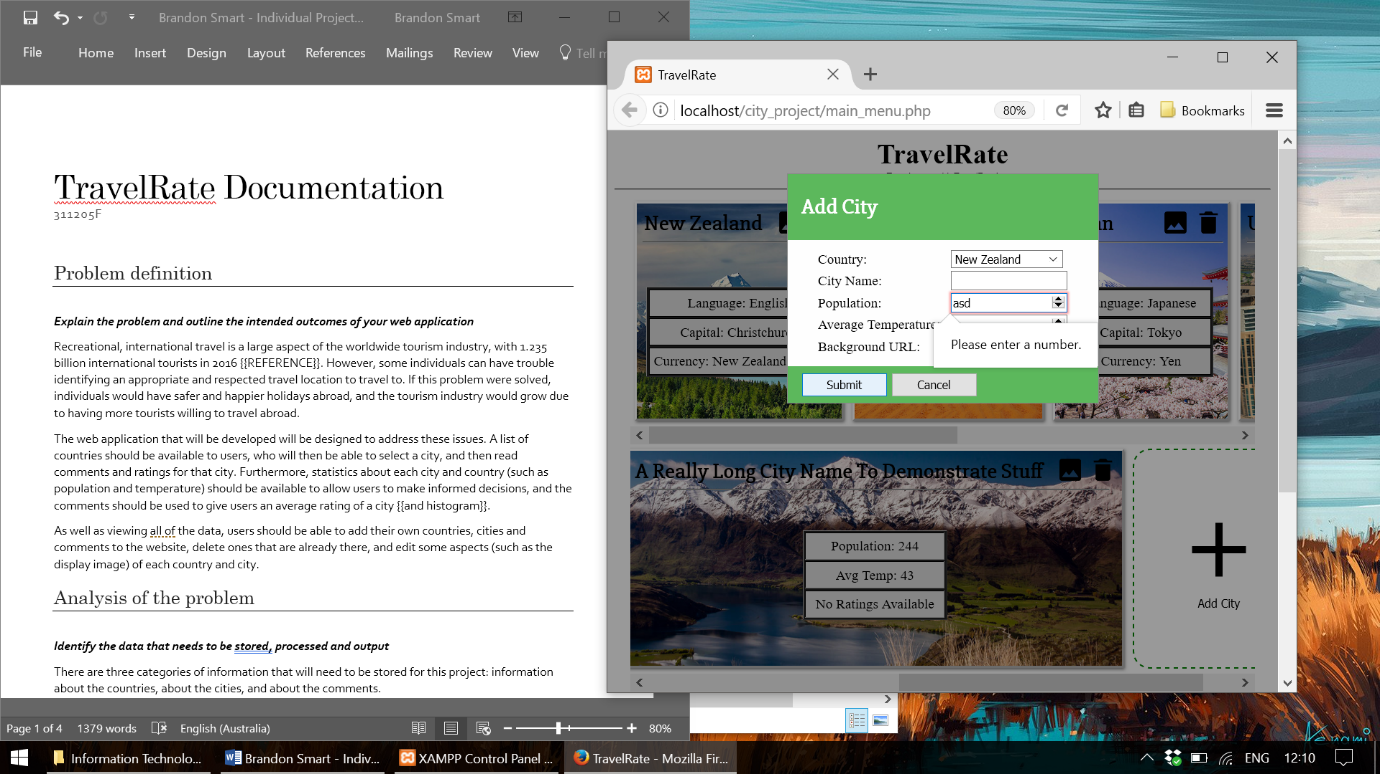
# Validation​​

In order to validate the functioning of the website, the functionality of each feature will be confirmed.

****The horizontally scrolling ‘div’s scale and work accurately, hiding and showing correctly depending on what country and city has been clicked.

****The display blocks for each country and city work correctly, with the background image scaling and the width of the element changing correctly, as well as the title, information, delete and edit buttons all being positioned in the right places. An accurate mean rating for a city’s comments is also displayed (as checked by hand), and a ‘No Ratings Available’ message is displayed when applicable. The comments also display accurately, with the background colour accurately reflecting the rating between 0 and 5, an accurate number of stars displaying at the button, the commenter’s name ending in ellipses if too long, and the date displaying accurately for the Adelaide time zone.

The ratings graph works correctly, with each image correctly displaying the correct title and data, with a blue line representing the average rating, a scaling vertical axis depending on the maximum frequency, each frequency bar being coloured according to its rating, and a ‘there are no reviews’ message when there are no reviews.

The add, delete and edit systems all work. The add, delete and edit buttons all bring up a correct menu with a post form that uses web standard input boxes to validate the information being sent, and the correct country or city filled in if applicable. Clicking the cancel button or outside of the box correctly closes the menu, and clicking submit will correctly submit the form. Server-side validation of the inputs is also completed to ensure all numeric inputs are numeric, and all country and city ids used exist in the database.

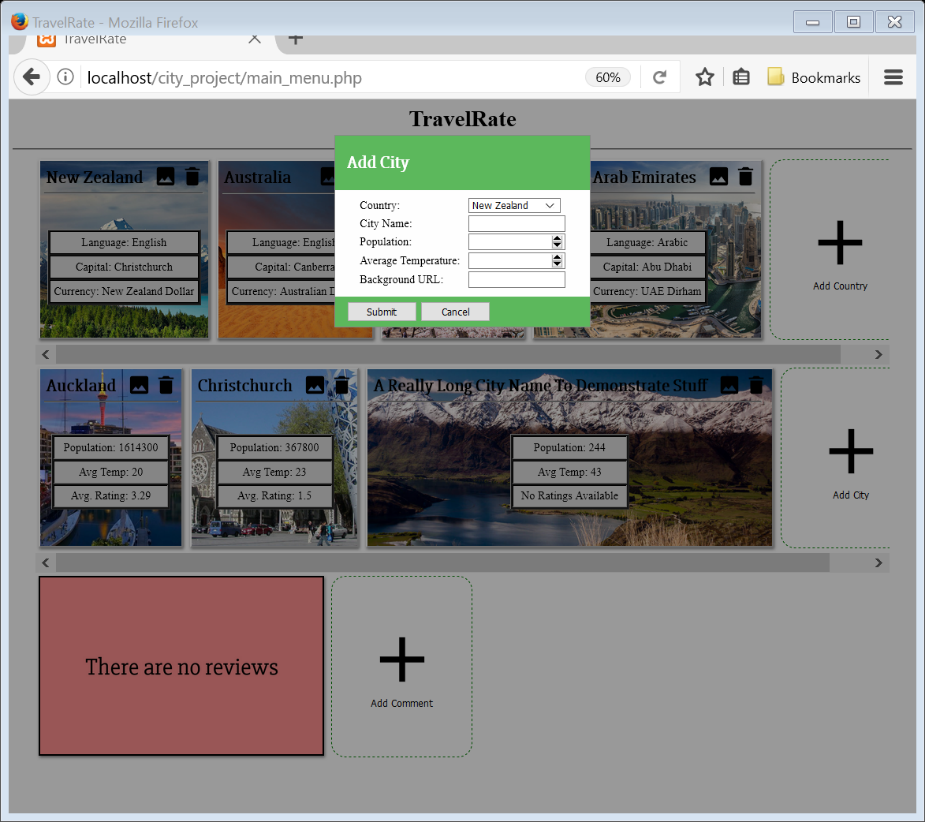
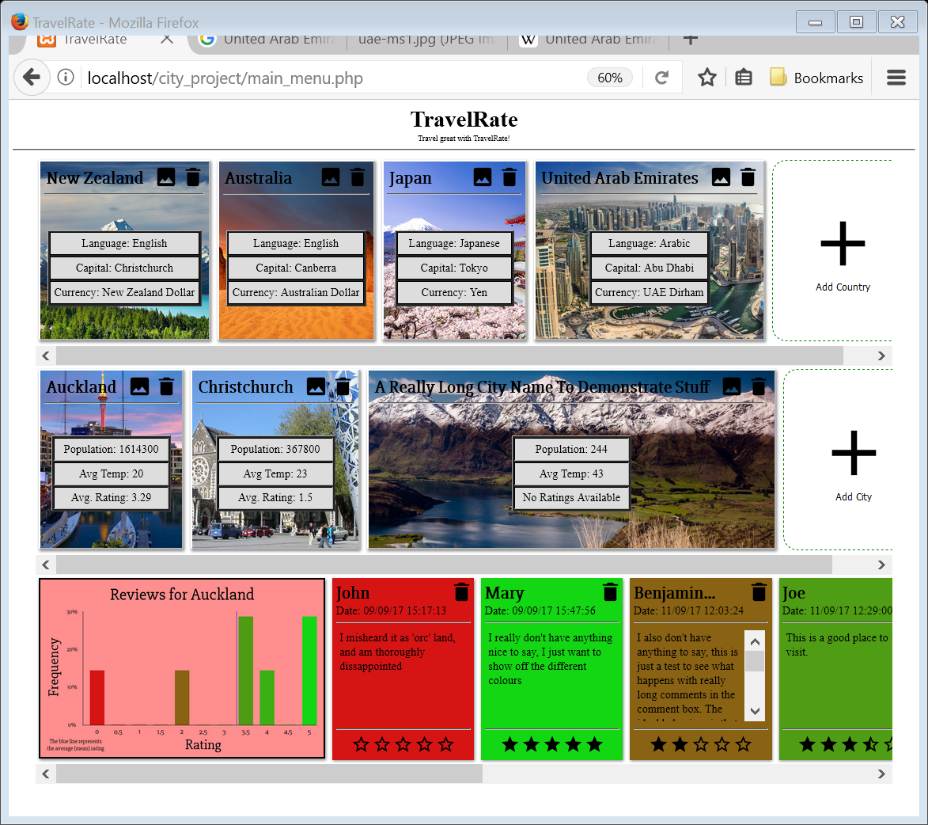
The dynamic website created successfully displays the information required and handles all user inputs accurately.

# Evaluation​​

I believe that the website developed works well in addressing the problem outlined in the beginning. The webpage developed gives users the ability to browse cities and countries, and make informed decisions about where to go based on information about these places, and the ratings and ratings graph for the cities. The framework developed allows for users to add and delete entries in the database, and for the images of the countries and cities to be changed, allowing the system to be extended and used into the future. I believe the website has been successful, as it can provide users with the opportunity to have safer and happier holidays, and a greater willingness to travel abroad.

Reference List

* RedBean, 2017, RedBean: The Power ORM, Viewed 21st September 2017, <http://redbeanphp.com/>
* UNWTO, 2017, UNWTO Tourism Highlights, Viewed 21st September 2017, <http://www.e-unwto.org/doi/pdf/10.18111/9789284419029>

****