Room Usage Web Application

Project Management Plan

VERSION 2.0.0

XINYANG FANG, CHRISTIAN YAACOUB, SURYA KANNAN

TEAM 082

11 September 2018 - 11 October 2018

Contents

Introduction and Purpose

Monash University has requested *SustainAppility* to build a prototype for a location-awareness web application in one month. This prototype is due on the 12th of October, 2018.

The main function of the application is to track room usage around Monash University. It is designed so that users, consisting of students, are able to fill in a form that records occupancy and certain utility usages for given rooms at different times of the day. Using all the data collected, useful information regarding occupancy and building statistics are created and reported back to Monash University so that they are able to allocate their resources more efficiently in the future.

So far, the app is only limited to storing information that includes: location/address of a room, number of seats in use out of a total number of seats and whether heating/cooling or lighting facilities are being used. The limited amount of data that is able to be inputted by the user reflects on the state of the app being a prototype.

By first using and testing simplistic data that could fit the characteristics of most rooms in the Monash campus, *SustainAppility*'s success in delivering a prototype would mean that we would be able to add a wider range of data types regarding resource use in an official release.

This project management plan (PMP) is designed to provide a baseline of what is to be achieved by *SustainAppility* and how the team will achieve it in the process of completing this project. The PMP will: elaborate on how the app is to be used by users, the scope of the web application including limitations, team members and their roles , a timeline of deliverables, design features, communication methods, decision making processes and risk management.

Background and Intended Use

Monash University, our client, has requested for *SustainAppility*'s services, a company that promotes careful use of resources and awareness of environmental impacts. The aim of the project is to build a prototype for a location-awareness web application because Monash University are looking to become greener and want to make better use of their resources.

The web app is only available via a url, given a stable internet connection, and has a simple user interface that consists of four webpages. The app is built on assumptions. These are that all rooms are of the same type of layout, indicated by the type of data accepted by the web app, and only two sources of energy usages are accounted for, heating/cooling and lighting.

The first web page is a form that is meant to be filled by the user, in this case, students or staff of the university. After the user inputs information regarding: location, heating/cooling and lighting facilities and how many seats are being used, the information is stored by the browser. Additionally, they have the option to clear all entries and automatically determine their location using a web request.

The second web page allows the user to view all existing observations. They have the option to: search for an observation using an address, delete an observation or simply scroll to view all

recorded observations. The webpage also will display the total number of observations and will change to display the total number of common observations when the user uses the search function.

The third webpage uses all the observational data to display observations grouped by hour of the day between (and including) 8am and 6pm. It displays the worst five room observations in terms of occupancy or all observations if there are less than five for the time slot.

The fourth webpage uses all the observational data to display observations grouped by a common address/ building. Each group contains: the number of associated observations, the number of wasteful observations (those where there are no seats used but the lights or heating/cooling are on), and average ultilsations of seats, lighting and heating/cooling. Furthermore, for any one or more wasteful observations, the building name will be highlighted by a red background to alert the user.

Scope

The web application creates room usage statistics based on data provided by the users. However this data is restricted to a certain number of data types. As it is a prototype that is being designed, it is made to take less specific inputs initially. This is done to test the applications ability to create useful statistics based on common data types.

Given that the app is built for a university setting, the prototype's initial data inputs reflect and accommodate the majority of the room layouts found. These typical room layouts chosen usually also require more energy and resource use.

Currently the only types of data accepted by the form are: The building address, room number, whether heating/cooling or lighting are being used, the number of seats and the total number of seats.

Although more specific data types such as: power plug points, water usage and number of lights are not requested as an input from the user, these can be added to the application after the prototype is shown to provide correct statistics that reflect the current data recorded.

Deliverables/due dates

Task	Description	Due Date
Feature 1	Creating a RoomUsage Class in shared.js.	13/09/18
Feature 2	Creating a RoomUsageList Class in shared.js.	14/09/18
Feature 3	Setting up a form webpage to store instances, with save and clear functions - dependant on features **1** and **2**	15/09/18
Feature 4	Automatically determining user location using a reverse geocoding API	17/09/18
Feature 5	Storing roomUsageList in localStorage - dependant on features **1** and **2**	20/09/18
Feature 6	Retrieving roomUsageList in localStorage - dependant on features **1** and **2**	20/09/18
Feature 7	Displaying all observations from most recent using observations.html - dependant on features **1**,**2**,**5** and **6**	21/09/18
Feature 8	Deleting observations from observations.html webpage - dependant on features **1**,**2**,**5**,**6** and **7**	23/09/18
Feature 9	Searching observations from observations.html webpage - dependant on features **1**,**2**,**5**,**6** and **7**	23/09/18
Feature 10	Creating aggregateBy method for bucketing observations - dependant on features **1**,**2**,**5** and **6**	25/09/18
Feature 11	Using aggregateBy to sort data in order to display worst occupancy by hour on occupancy.html - dependant on features **1**,**2**,**5**,**6** and **10**	28/09/18
Feature 12	Using aggregateBy to sort data in order to display building statistics on buildingStats.html - dependant on features **1**,**2**,**5**,**6** and **10**	01/10/18
Feature 13	Highlighting wasteful observations in buildingStats.html - dependant on feature **12**	01/10/18
FINAL TESTING	Testing code for bugs	02-10/10/18
DUE DATE	Final prototype submission	11/10/18

Design

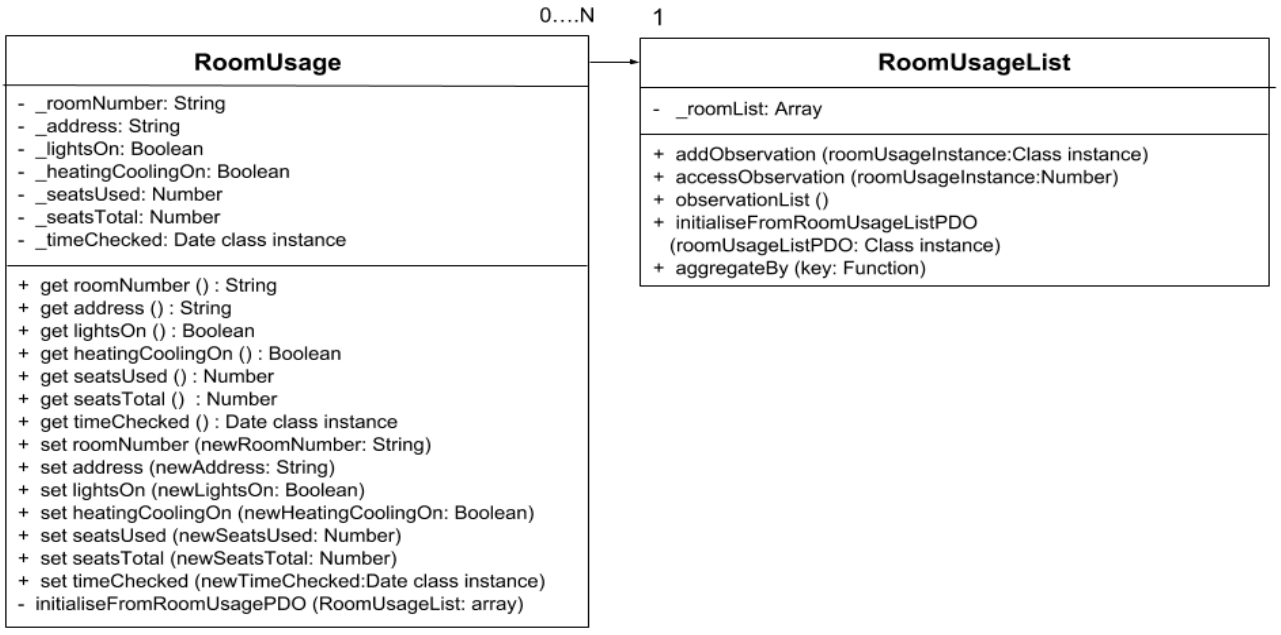
The web application utilises two classes.

The first is the RoomUsage class. This class contains attributes that are used for room usage statistics (seats used, timeChecked etc). This class is used as storage for data entries made by the user by creating a new instance of the class. These instances are created after the user clicks on the “SAVE” button located on the form webpage which calls a save function that uses the addObservation method of RoomUsageList.

The second class is RoomUsageList which only contains one attribute, roomList. This attribute stores all instances of RoomUsage. Through indexing, information about a given instance can be accessed and this is how statistics are mainly created and shown on the occupancy and building statistics webpages.

When the webpage is loaded, if a roomUsageList instance is not found, a new instance is created. Otherwise, a current roomUsageList is loaded from local storage and is initialised using the intialiseFromPDO methods of each class.

aggregateBy, a method of RoomUsageList, is used to bucket instances found in roomList by either hour or address. This, along with extracting correct data from RoomUsage instances, is how all room usage statistics are created.



Personnel/HR Management

NAME OF MEMBER	CONTACT DETAILS	RESPONSIBILITIES
Christian Yaacoub	Email:cyaa0001@student.monash.edu Phone number: 0402198837	- Chairperson for weekly meetings. In charge of creating agenda and setting meeting time and place. - Task tracker/ manager. Purpose is to help members remain organised with assigned tasks. -Responsible for coding features 3,7,8 and 9. - Provides input for user manual documentation
Xinyang Fang	Email:xfan0009@student.monash.edu Phone number:0452471266	-Responsible for coding features 10,11,12 and 13 - Provides input for user manual documentation - Quality control of code and documentation. Needs to ensure that completed tasks are of a high standard and meet all requirements.
Surya Kannan	Email:skan0017@student.monash.edu Phone number: 0407934563	- Minute taker for weekly meetings. In charge of creating a new document for each meeting to record meeting minutes. Also responsible for informing any absent members of important details discussed during meetings. - Task/ allocation coordinator. In charge of assigning tasks to each member and ensuring allocation is fair. - Responsible for coding features 1,2,4,5 and 6 - Provides input for user manual documentation

Decision on Processes

The project is broken up into two areas. These are: coding the web application and creating useful external documents for users and clients in the form of a user guide.

-**Code**
Brackets, a text editor, is used to write the javascript, CSS and HTML files for the web application. This is how the final code is written. An intermediate code writing space, the Javascript playground, is used to mainly test sections of the final code.

GitKraken, a software version control tool, is used by all members of the team. While individually, members may write code on their local machine, a local repository is shared by all team members where regular commits are made. All final updated code is stored in the Git space. Team members are expected to provide brief but useful summary and descriptions of each commit to signal to other members any areas of the code that need refining or are fully completed.

-**Documentation**
A google drive, shared by all team members, is used to store and edit all external documentation. It is expected that the drive is used regularly as certain areas of the external documentation link up to the code being written for the application. This could refer to user interface, functionality or any bugs found in the code that will not be able to be fixed before the prototype is released.

-**Task allocation and management**
Asana is an application that is used to allocate and track progress of assigned tasks. It is the space, outside of weekly meetings, that tasks are assigned to individuals. Asana should be reviewed every two days as new tasks may be completed that allow other dependant features to be written. Team members who complete a task must update their allocated Asana tasks and ensure that it is marked as complete.

Communications Management

As the project will require communication between members on a regular basis, the following forms will be used by the team over the course of one month.

Method of Communication	Purpose	Frequency	Audience	Effective use	If frequency is not met by a member
Git Commits	Update project code and inform team of progress on current task that was assigned.	Per commit	Individual and Full team	Clear and descriptive summary and description per commit	Should warn the member and ask to commit regularly to track progress. Must also check if the member has completed the task by the due date assigned.
Facebook Messenger	Quick or urgent matters that need to be resolved. e.g a small bug found	Ongoing	Up to all team members	Non-relevant information excluded in the group chat. Clear and appropriate language used.	Not required if quick messages are not appropriate for the task or situation.
Email	Direct communication while working on a task.	Ongoing	Two team members	Formal language. Informs the other member about the topic at hand and does not include irrelevant information.	Need to maintain some form of communication, so the member will be asked to explain why they did not choose to communicate.
Face to Face team Meetings	Discuss current progress and issues. Also used for planning and setting goals before the next meeting.	Weekly	Full team	All items on the agenda are addressed. Minutes are well written. Pre-prepared	Member will be notified of their absence and will be sent meeting minutes as well as any additional information that was discussed.

Risk Management

Even though the project has a duration of one month, many risks are involved that may affect the team's ability to work together to complete the project. This may be due to the fault of an individual member, something the team overlooked, or a lack of proper planning. These are outlined below.

- Team members not completing assigned tasks

  - This risk is highly likely to occur in the absence of proper communication and planning. One team member may not be able to or want to communicate their current progress.

  - To solve this issue, during weekly meetings, every team member will be asked of their current progress and an estimated time of completion. If they require assistance, other team members will be assigned to the task to provide help. To ensure this does not occur, all team members must strictly meet the due date of their assigned task.

- Software bugs

  - This risk is highly likely to occur given the short time frame that the app is to be built in. This is especially true in connecting features together written by different members.

  - Using a peer review system, every commit done by a member will be immediately reviewed and read over for any software bugs. This will be done a number of times before the task written meets all the given requirements. After the task written is implemented together with the completed features, the peer review system will once again be used to check the entire code.

- Lack of communication

  - Given the many modes of communication that will be used during the project by the members, this issue is not likely to occur. In one instance however, a team member may not be happy and may choose not to communicate due to a disagreement.

  - Through proper management, the team will be using every mode of communication efficiently and with the correct frequency. This would avoid any lack or miscommunication between members. If a team member chooses not to communicate for other reasons, this will be addressed during the weekly team meeting and the issue will be resolved.