Room Usage App

Project Management Plan

Joshua Ang, Tong Siew Wen, Lim Wei Jun, Poong Zui Xiang

TEAM 167

11 October 2018

Contents

Introduction and Purpose

Room Usage App acts as a data collector as well as analyzer for the room utilization in a building. The design of the app is aimed to help organisations understand and make appropriate adjustment on the energy usage and thus prevent energy wastage.

This document intends to give an overview on the design of the app and its basic software design to show how the app interacts internally and with the users. Besides, this document will specify how the tasks are distributed within the team, how the team manages the project and possible risks. Hence, the methodology or practices used in managing the project as well as the software tools used to aid and smoothen the progress is also mentioned in this document.

Project Information

Background and intended use

We are a software development team in *sustainAppility*, who specialises in apps that encourage careful use of resources and awareness of environmental impacts. We are designing an app for Monash University, called Room Usage App, to help them to be greener and make better use of their resources.

It is distributed to users, who are mainly students and staff of Monash University, to record lighting, heating, air-conditioning and room utilisation in each room throughout the campus. Then the data is saved in the local storage and displayed. Also, it analyses, summarizes observations as statistics and detects wasteful observations based on hour or building.

Some assumptions are made, which includes:
- Most students and staff will use the app to record the room utilization, thus no observations is missed and the data covers all observations for every room.
- All users will fill in the correct information, and that the organisation will check through the observation list and delete the faulty observations. Thus, the statistics are accurate.

Scope

There are some limitations of the Room Usage App. For instance, the data accumulates day after day, which will one day reach the storage limit. In other words, there aren't any renewal system that reset the local storage after certain length of time. Also, local storage is stored in individual devices and aren't shared among users. In other words, the management team of the building isn't able to view observations made on user's personal device. This feature can be improved in later time once our team have the knowledge on how to continuously update the data in the cloud.

The app also group data based on hour regardless what date it is recorded, which may be hard for the organisation to manage their resources. This is due to the assumption that the schedule of the campus is similar, i.e. the timetable of students is fixed and the hours of having classes are similar everyday. However, it is recommended to have this feature further improved in later versions of the app.

Moreover, the app does not give offline notification when a wasteful observation is detected so that the organisation can take action immediately. However, this is not very necessary and requires additional coding, therefore this feature is excluded.

The auto-detection of building is also sometimes inaccurate, which can result in faulty observation. This is one of the limits that is beyond our control and capabilities since it is due to external factors.

Brief Overview of the design of the app

Menu of the app and user can choose which page they wanted to access
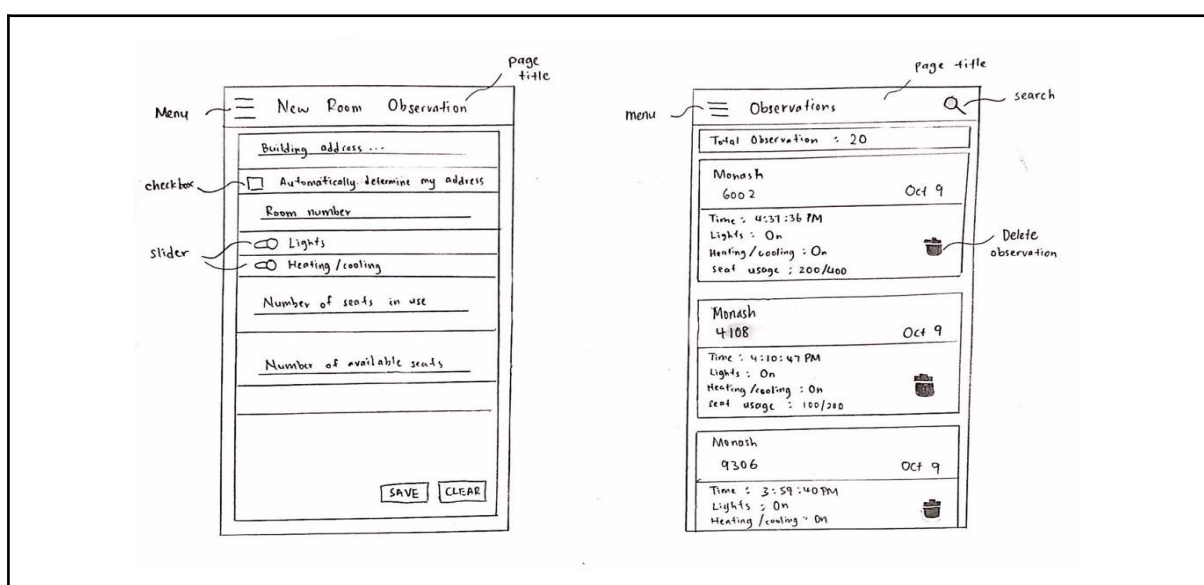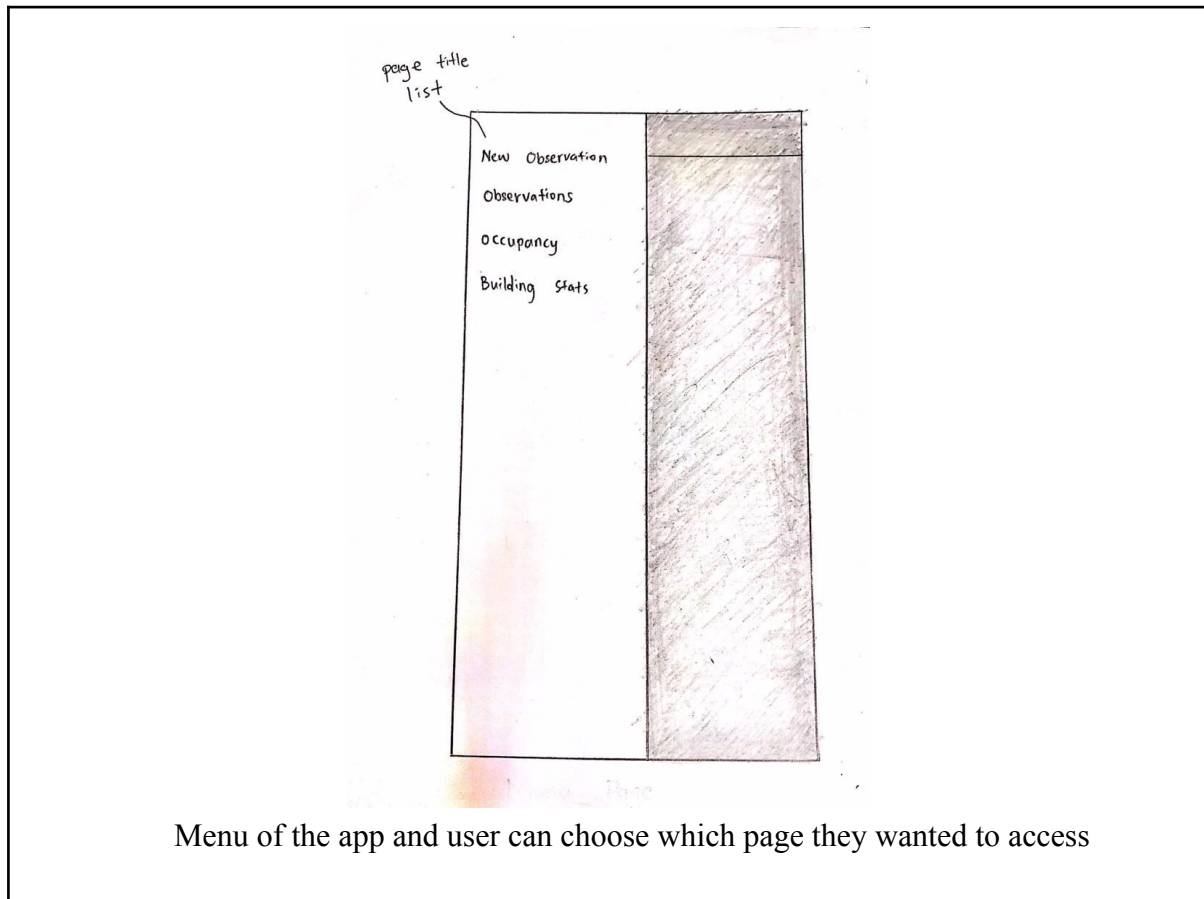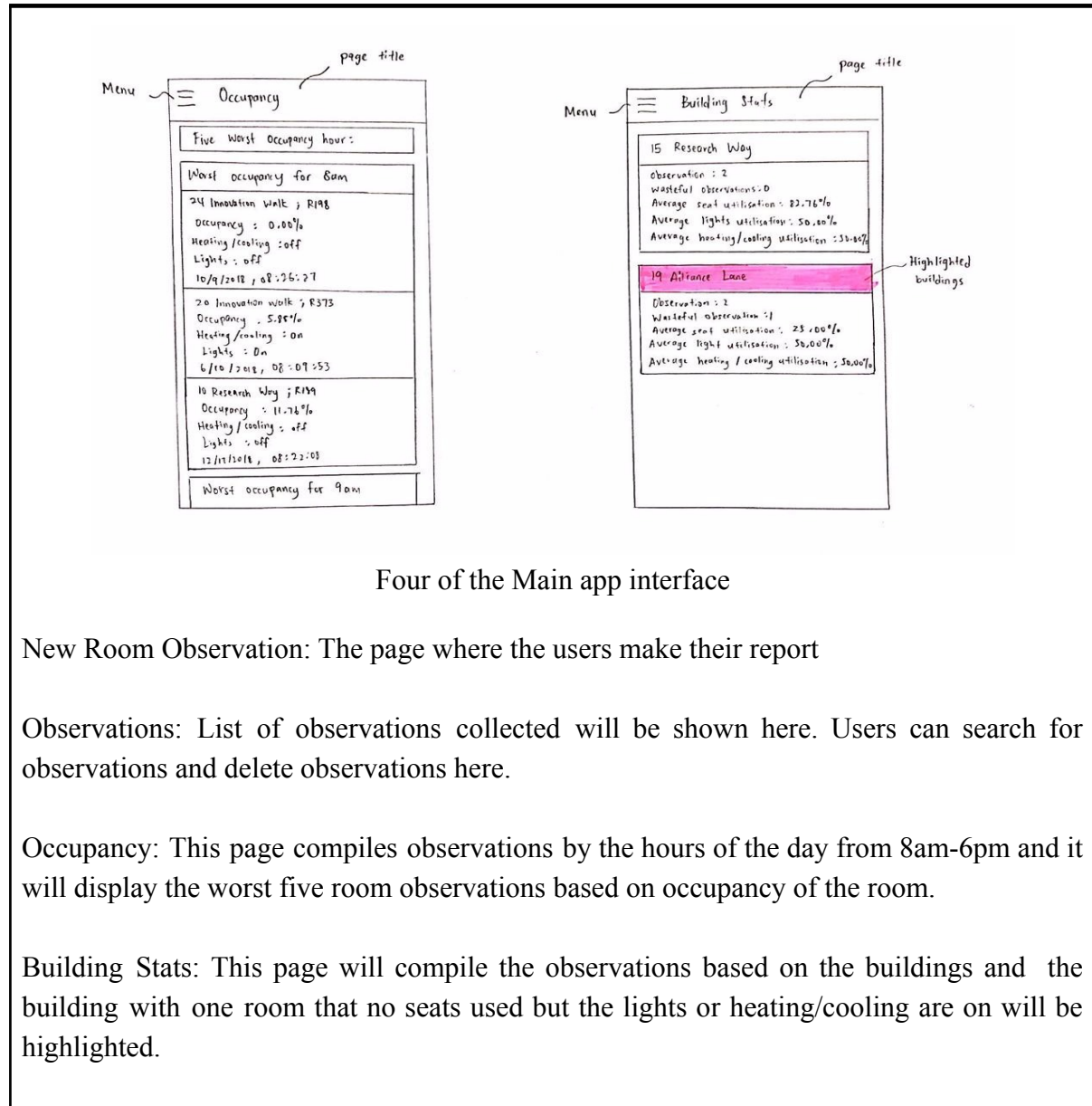
Four of the Main app interface

New Room Observation: The page where the users make their report

Observations: List of observations collected will be shown here. Users can search for observations and delete observations here.

Occupancy: This page compiles observations by the hours of the day from 8am-6pm and it will display the worst five room observations based on occupancy of the room.

Building Stats: This page will compile the observations based on the buildings and the building with one room that no seats used but the lights or heating/cooling are on will be highlighted.

When the response from the user was recorded

Our app will pop up a message, when the location of the user was not detected accurately

This is the error message that will display on the webpage. It will notify user that there was an error detected before submitting their observation

When the delete button is pressed, the observation will be deleted

This is the search function of the observation page

Deliverables/due dates.
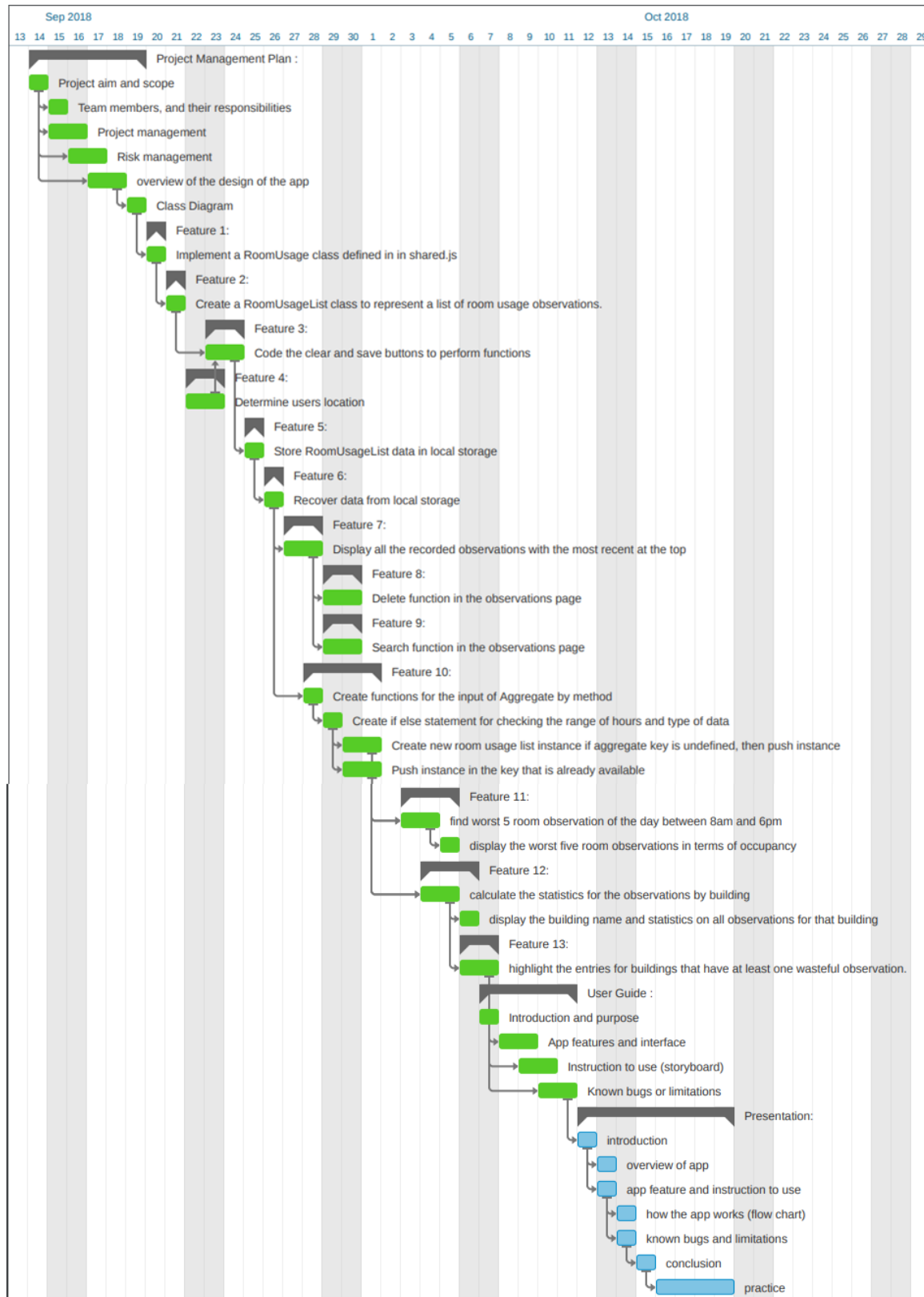
The Gantt Chart above shows the planned progress of each task in the project. The Project Management Plan (PMP) is placed first to be done, so that every member in the team understands the project aim and agrees on the methodology, the task assigned as well as the software tools used before we start coding.

After the PMP is documented, we will move on to writing the code for the app. The time given to complete each feature is based on the complexity of that feature. Since most feature have dependency on other feature, the whole coding task will take greatest amount of time, if compared to the documentation and the presentation.

User guide is done after all coding tasks is finished. This is to ensure that the interactions shown in the storyboards matches with the actual reaction of the app. After that, we will work on our presentation slides since we will know how well the app is in line with the aim, as well as the limitations and bugs after documenting the user guide. Last but not least, we will practice for few days to give a flawless presentation to the representative for Monash University, Susy Tayn How in order to leave a better impression to him.

Personnel/HR Management

Member Name	Email	Responsibility
Joshua Ang	jang0009@student.monash.edu	- In the Project Management Plan (PMP) documentation: - In charge of updating the risk management regarding this project - Wrote code regarding the following features: - Set up the RoomUsage and RoomUsageList classes with their getter/setter and other relevant methods - Created the save function that stores the input data if it is successfully validated - Wrote the storeData function that adds relevant RoomUsage data to the RoomUsageList array and stores it in local storage - Wrote code to determine the 5 worst occupancy cases every hour that has been recorded from observations - In user guide: - Discussed the app features and interface - Provided screenshots and description for "instructions to use app" - In preparing for presentation: - Help in preparing the presentation slides - Practice presentation with team members
Tong Siew Wen	ston0005@student.monash.edu	- In the Project Management Plan (PMP) documentation: - Prepare the fragments for introduction, purpose and the project management - Write codes to handle the following

		features::
		○ automatically detects user's current location to aid in filling new observations ○ create a search filter when user search an observations using keywords ○ enable user to delete unwanted or faulty observations ○ aggregate and group observations by hour and by building name ○ calculate statistic for the usage of resources in each building - in User Guide documentation: ○ discussed the limitations of the app - in preparing for presentation: ○ help in preparing the presentation slides ○ practice to present with team members
Lim Wei Jun	wlim0014@student .monash.edu	- In the Project Management Plan (PMP) documentation: ○ In charge of the drawing of storyboard and the brief design of the app - Wrote code regarding the following feature: ○ Parts of the store data function ○ The retrieve data function which will take the stored data from the local storage ○ Display the building name and observations on all observations for that building ○ Highlight the buildings that have at least one wasteful observation - In User Guide documentation: ○ Writing the instructions on how to use our app - In preparing the presentation: ○ Help in preparing the presentation slides

		○ Practice presentation with team members ○

Poong Zui Xiang	zui0001@student.monash.edu	● In the Project Management Plan (PMP) documentation:
		  ○ In charge of making the class diagram
		● Wrote code regarding the following feature:
		  ○ The display data function which will take the retrieved data from the local storage .
		  ○ the display of the worst occupancy hour.
		● In User Guide documentation:
		  ○ writing the introduction and purpose of the user guide
		● In preparing the presentation:
		  ○ Help in preparing the presentation slides
		  ○ Practice presentation with team members

Decisions on Processes

In order to be productive and systematic, our team has decided to utilise some software tools to store our work and to communicate.

Asana is chosen to be the team's communication platform. This is due to its ability to group discussions on a task at the same place and keep track on the team's overall progression. Asana also can list and distribute subtask in a project, which is very convenient and useful.

On the other hand, codes are shared among team members using Git, as Git is a version control system that enable us to do work locally, while being able to update and commit changes to the original code whenever we want.

A version control system is also needed for documentation; hence Google Drive would be the best option. Google drive enables us to have only one final version of document, while being able to view the changes made by other members. It also increases productivity since all of us can do our part simultaneously.

With these three software tools, the work can be facilitated in a more effective and easier way.

Communications Management

Name	Type	Audience	Purposes	Requirement	Strategies to make the communication effective
Project Manage ment Plan	External documen t	New or existing members	Communicate procedures and information to team members	"Closed" and "context independent "	Needs to fully explain the overview of the app design, code design, as well as how the team manages the project, communication and possible risks.
Client presentat ion	External message	The client	To hand over the maintenance of the project and code	"Open" and "context-dep endent"	Every team member must present equally without swapping too frequently Intention and context must be clearly explained: structure and functionality of application, design decisions and the limitations of the app
Intervie w with demonstr ator	External message	Demonstr ator	To explain and demonstrate individual understanding	"Open" and "context-dep endent"	Every team member must understand every part of the code, so that they are able to answer all questions asked individually Team members should ask each other questions if anyone of them don't understand part of codes which is written by another member.
Asana entries	Internal message	Team members	Update progress and discuss problem in each feature	"Open" and "context-dep endent"	Every team members must update their progress regularly (every few days) so that other team members are always on the same track Conversation shall only be related to the project

Face to face meeting among team members	Internal message	Team members	Update progress and discuss problem in each feature	"Open" and "context-dep endent"	Team members should be involve in discussion and team coding during face to face meeting Conversation shall only be related to the project
Git commits	Internal message	Team members	Update progress and discuss problem in each feature	"Open" and "context-dep endent"	Every team members must update their progress regularly (once his or her part is done) so that other team members have the newest version of the code Commit message should be relevant and understandable
Whatsapp communication	Internal message	Team members	Minor discussion in to decide meeting time etc	"Open" and "context-dep endent"	Every team member should be able to respond in Whatsapp in a short time (within a day) Conversation shall only be related to the project
User guide	External document	App users	Explain to users how to use the app step by step	"Closed" and "context independent"	Every step should be included with screenshots and explanations Any known bugs and limitations should be included, as well as how to avoid or solve them
Stand-up meeting with demonstrator	External message	Team members and demonstrator	Update the progress of team and ask demonstrator questions is there is any doubts or problem faced	"Open" and "context-dep endent"	List of questions should be prepared prior to the stand-up meeting Team member should be aware of the team progress to be reported

Risk Management

Every project comes with risk. It is unavoidable and always had a chance of occuring. Hence, in this section known as risk management, possible issues may be faced is identified and strategies to overcome or avoid them are developed.

The main risk involved in this project is poor time management in the completion of individual tasks. This may occur due to obligations from other units taking up most of a person's time, or the individual lacking the skills to complete a task and prolonging it in the process. From past experience, this was not a problem in our team and has a relatively low chance of occuring. The reason this risk can be easily avoided is due to frequent communication over the progress of our tasks allocated and the willingness to help one another such that it can be completed. However, in the event this risk does occur, we will have to resolve the issue by setting due dates for every features as well as request for help whenever necessary.

Another risk we face is the inability to effectively use the new version control system, resulting in loss of work and time. As this is the team's first time using a version control system, team members may not be very familiar with the way it is operated. Hence, the written code can easily be lost if team members are not careful with the push or pull function of the software. The likelihood of this occurring is quite high as this is unfamiliar territory for the team. A step that can be taken such that this risk factor can be avoided, is making a backup of member's code so that a copy is saved regardless. However, if this issue is experienced, members would have no choice but to resolve it by re-coding the part that was lost.

Adding on, another risk that might be faced is overestimating the amount of time we have. This would cause us to be left behind and potentially not make the submission deadline. This chance of this risk occurring is moderate. Due to the large amount of work we face from other units and our given duration, we could easily get caught up in other work and overestimate the time we have to complete this project. To avoid this issue we should use an effective communication medium to discuss our progress often as well as set deadlines and reminders for each task so everything is kept on track.

Apart from that, failing to understand the desired requirements from the project is another risk we face. Misinterpretation of the desired outcome may cause us to produce something different from what is requested by the client. The likeliness of this risk occurring is moderate. To ensure this does not happen, team members should check with the client often and incorporate a Agile Development practice. Through this, any uncertainty can be cleared up. In the case the risk is faced, we would have to consult with the client on the next course of action to be taken.

Software design of the app

RoomUsage

- _roomNumber : String
- _address : String
- _lightsOn : Boolean
- _heatingCoolingOn : Boolean
- _seatsUsed : Number
- _seatsTotal : Number
- _timeChecked : Date class instance

+ RoomUsage
(roomNumber:String,address:String,lightsOn:Bo
olean,heatingCoolingOn:Boolean,seatsUsed:Nu
mber,seatsTotal:Number,timeChecked:Date
class instance)
+ set roomNumber(roomNum:String)
+ set address(address:String)
+ set lightsOn(truth:Boolean)
+ set heatingCoolingOn(truth:Boolean)
+ set seatsUsed(usedSeats:Number)
+ set seatsTotal(totalSeats:Number)
+ set timeChecked(timeChecked:String)
+
initialiseFromRoomUsagePDO(RoomUsageObj
ect:RoomUsageInstance)
+ get roomNumber():String
+ get address():String
+ get lightsOn():Boolean
+ get heatingCoolingOn():Boolean
+ get seatsUsed():Number
+ get seatsTotal():Number
+ get timeChecked():Date class instance

Room Usage List

- _roomList:Array

+ pushEntry(roomList:Array)
+
initialiseFromRoomUsageListPDO(roomListPD
O:Public Data Object)
+ aggregateBy(func:Function)
+ set roomList(newRoomList:RoomUsage
Instance)
+get roomList():RoomUsage Instance

1   0..*

Table 1.0: class diagram

Appendix

Minute meeting 1		
Date: 14/9/2018	Time: 9-10 am	Location: computer lab
Agenda	**Discussion**	
Understand requirements for assignment	All of us read through the instructions from moodle together	
Delegate tasks to each group members for feature 1 to 4	Feature 1, 2 and 3 will be coded by Joshua. Feature 4 will be coded by Siew Wen. All of us agree to help each other when coding.	
Set due date for each features	We aim to finish until feature 4 before mid sembreak commences.	
Solve problems (if any)	-	

Minute meeting 2		
Date: 20/9/2018	Time: 9-10 am	Location: 9305
Agenda	**Discussion**	
Understand codes for feature 1 to 4	Joshua explained the code for feature 1 to 3. He struggles on the clear function. All of us discussed on the pseudocode for the clear function and Siew Wen agreed to take over that function. Siew Wen explained the codes for feature 4.	
Delegate tasks to each group members for feature 5 to 9	Feature 5 and 6 will be coded by Wei Jun (main) and Joshua. Feature 7 will be coded by Zui Xiang, while feature 8 and 9 will be coded by Zui Xiang and Siew Wen. All of us agree to meet up to do the coding during mid sem break.	
Set due date for each features	We aim to finish feature 10 before week 10 commends.	
Solve problems (if any)	Joshua faces problem to push his files onto gitkraken. Since we are not familiar with the app, we couldn't detect the reason. So, we will consult the lecturer, Mr. Teo about this issue.	

Minute meeting 3		
Date: 29/9/2018	Time: 10 am - 5pm	Location: 9305
Agenda	**Discussion**	
Understand codes for feature 5 to 9	Wei Jun and Joshua explained their code on feature 5 and 6, then Siew Wen and Zui Xiang explained their code on feature 7, 8 and 9. All members understand how the codes work.	
Solve problems (if any)	The room usage list stored in local storage is in a very confusing way, as both room usage list and each room usage is in a nested array. We found that the problem is in the save function. Thus, we solve this by pushing the new room usage class instance into the existing room usage list, which then is updated in the local storage. Also, the data cannot be retrieved in when displaying the observations. After debugging using inspect, we found that the problem is the setter method due to wrong capitalisation. After solving this issue, the observation page can retrieve and display the data.	
Do feature 10 together	All of us try to understand the instructions and managed to finish writing the codes for feature 10 within the meeting	
Distribute feature 11 to feature 13	Feature 11 will be coded by Zui Xiang and Joshua. Feature 12 will be coded by Wei Jun and Siew Wen. Feature 13 will be coded by Wei Jun. All of us agree to meet up after all coding is done, so that we can finalise and solve any integration problems for all features.	
Set due date feature 11 to 13	Due date is set to be end of week 10	

Minute meeting 4		
Date: 10/10/2018	Time: 6-7pm	Location: 9305
Agenda	**Discussion**	
Understand codes for feature 11 to 13	Zui Xiang and Joshua explained their code on feature 11, then Siew Wen and Wei Jun explained their code on feature 12. Wei Jun also explained his code for feature 13	
Solve problem (if any)	There is no problem in the final version of the app	