



## IS2102

### Enterprise Systems Architecture and Design AY2018/2019 Semester 1

Tan Win Phong (A0170152U)  
Arcillas Brian Joshua Pelaez (A0167854N)  
Justin Eng Wei Jie (A0167946L)  
Chen Yi Zhao (A0168759E)  
Lim Zhan Yi (A0168091B)

## **Business Requirements Document (BRD)**

### **1. Executive Summary**

- 1.1. Summary of what this project is about
- 1.2. Key problems the system is addressing
- 1.3. Key components in the solution
- 1.4. Key benefits of this solution
- 1.5. Process taken in order to produce this BRD

### **2. Introduction**

- 2.1. Brief analysis of the cleaning services sector in Singapore
- 2.2. Scope of the project
- 2.3. Introduction to the subsystems/modules in the solution
- 2.4. High-level description of each subsystem/module (e.g. users, description, etc)
- 2.5. List of stakeholders and how they are interacting with the solution
- 2.6. Team (IT business analyst) composition (members in the team, roles and responsibilities of each member) - We are the team to provide the advisory roles

### **3. High-Level System Architecture**

- 3.1. List of subsystems/modules in the solution
  - 3.1.1. In both table/list form and a Visual table of Content (VTOC) form
- 3.2. List of use cases for each subsystem
- 3.3. Class diagram containing all the entities

### **4. Business Requirements Analysis**

In each of the subsystem, include (where applicable):

- 4.1. Detailed description
- 4.2. Business objectives
- 4.3. Business assumptions
- 4.4. Business processes
  - 4.4.1. Consider using an activity diagram or usage scenario to illustrate for complicated use case
- 4.5. Other relevant artifacts
  - 4.5.1. E.g. if reports need to be generated, it would be good to show a sample of how the report could look like

### **5. System Functional Requirement Analysis**

In each of the subsystem, include (where applicable):

- 5.1. Use case diagram
- 5.2. Use case description for each use case
- 5.3. Simplified entity class diagram (includes entities which are relevant to this subsystem)

5.4. Non-entity class diagram (class diagram including all other classes that are not entities)

**5.5. Sequence diagram for the 3 most complicated use cases**

**6. Conclusion**

6.1. Include any references that are cited in the BRD

## Executive Summary

The proposal for the development of a cleaning management information system to manage SG Cleaning Solutions' operation is detailed. The main issue highlighted by SG Cleaning Solutions is the lack of an automated and online system to manage their activities, employees and services. The current system that SG Cleaning Solutions have in place utilizes physical forms and Microsoft Excel to handle their operations. SG Cleaning Solutions is looking to implement a digitized cleaning management system to increase the synchronisation and efficiency of the company operations. The system that Easy Tech proposes utilises a centralised system where all the information will be synchronised at all times and allow the supervisors to effectively manage and track the attendance of cleaning staff, allocation of cleaning staff to different locations as well as upgrade the management of facilities. The cleaning management system also takes into account of the management of cleaning supplies while allowing incident and faulty facility report to be generated and monitored. The integration of different subsystem provides the real time functionality to manage the various components that is required to support a highly efficient and functional cleaning company.

The process involved in preparing the business requirement documents starts off with analysis of the written system requirement for the cleaning management system requested by SG Cleaning Solutions. It is then followed by an interview with the stakeholder, Ms. Ding Yi to clarify the detailed specification requirement for the cleaning management system, and at the same time, proposing ideas for the design and implementation of the system. The IT business analysts from Easy Tech identified the subsystems required with the feedback gathered from consultation and formulated the use case diagrams for the respective subsystem, followed by detailed description for each use cases. Class diagram showing the relationship between entities was produced to support the creation of sequence diagram for three of the most complicated use cases.

# Introduction

After conducting research into the Cleaning Sector of Singapore, it appears that the bulk of the workforce employed by Cleaning Services is comprised of the older generation, who may not be tech-savvy. The cost and effort of training them would be a negative consideration for SG Cleaning Solutions, thus the proposed plan will take this into account and design a system that is accessible and user-friendly.

The scope of this proposal will be to design an information system that will be implemented by the in-house developers to support SG Cleaning Solutions operations. Due to the tight time constraints, the proposal will be mainly focused on the major areas that SG Cleaning Solutions requires to run, namely Access Management, Human Resource Management, Facility Management and Feedback Management. These major areas are split into various subsystems for ease of analysis.

The Access Management System has no subsystems but is split into the Login/Logout module, the Profile Module and the Access Rights Module. The purpose of this system is to allow the supervisors to manage the user accounts, such as to edit the access rights and roles of the accounts. This system also handles the login/logout function.

The Human Resource Management Subsystem has two Modules, the first of which is the Employment Management comprised of the Employee Profile Sub-Module, Employee Payroll Sub-module and Leave/MC Sub-module, and the second of which is the Attendance Management Module with the Attendance Sub-module. The Employment Management Module allows the supervisors to handle the employee related tasks, such as creating staff records, editing staff details, managing leave for the staff, and tracking payment matters. The Attendance Management Subsystem will make use of a card-based check in/out system for ease of use and facilitate the recording of attendance and breaks taken by staff.

The Asset Management Subsystem has four modules, Inventory, Supplier, Procurement, and Inventory Report. The Asset Management Subsystem allows seamless tracking of cleaning supplies between the Cleaning Staff and the Supervisors, as well as generating reports, supply procurement orders and liaising with suppliers.

The Area Management Subsystem has the Location, Venue and Facility Module and the Faulty Facility Reporting Module. The Area Management Subsystem facilitates the management of venues and facilities under the jurisdiction of SG Cleaning Services, and ties in to making the reporting and employee management more convenient.

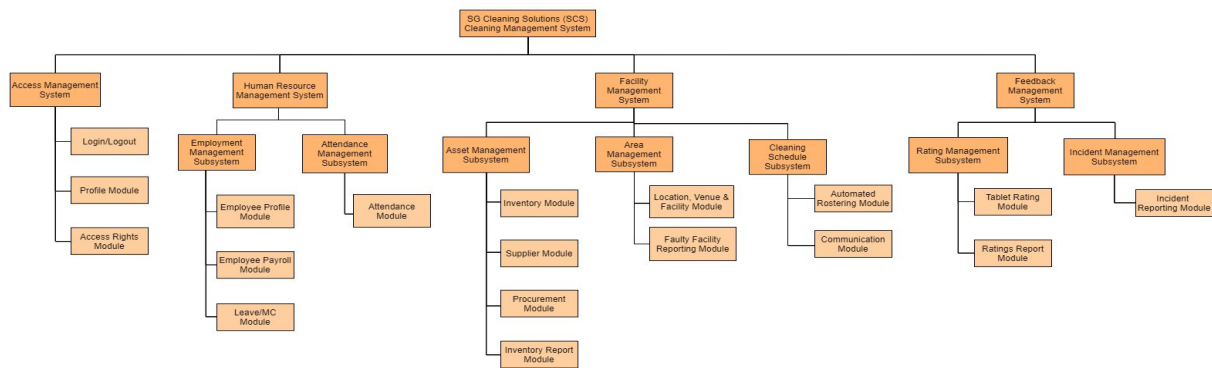
The Cleaning Schedule Subsystem has the Automated Rostering Module and the Communication Module. The Cleaning Schedule Subsystem allows the company to automate the rostering system of the Cleaning Staff and the venues and facilities that they will work at, freeing up valuable manpower hours for more important tasks.

Finally, the Feedback Management Subsystem is divided into the Rating Management Module and the Incident Management Module. The Rating Management Module has the Tablet Rating Sub-Module and Ratings Report Sub-Module, and the Incident Management Module has the Incident Reporting Sub-Module. The Rating Management Module is primarily catered towards patrons of the toilets under the supervision of SG Cleaning Solutions to generate feedback, and for the supervisors and staff of the company to view the feedback quickly and easily. The Incident Management Module helps the cleaning staff to report equipment damage to their superiors back in the main office easily and signal if the problem has not yet been resolved.

The team in charge of this project is comprised of Mr. Ajax Chen, Mr. Tan Win Phong, Mr. Brian Arcillas, Mr. Lim Zhan Yi and Mr. Justin Eng. Mr. Chen and Mr. Lim will be in charge of designing the Facility Management Subsystem, Mr. Tan will be in charge of designing the Human Resource Management Subsystem, Mr. Arcillas will be in charge of designing the Access Management System, and Mr. Eng will be in charge of designing the Feedback Management Subsystem.

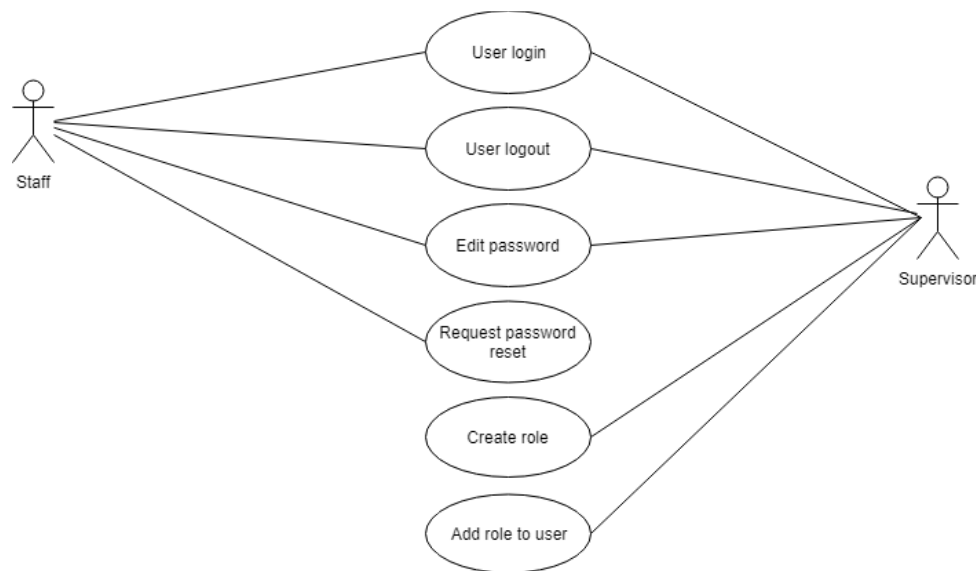
# High Level System Architecture

## Visual Table of Content



## List of use cases for each subsystem

### ***Access Management Subsystem***

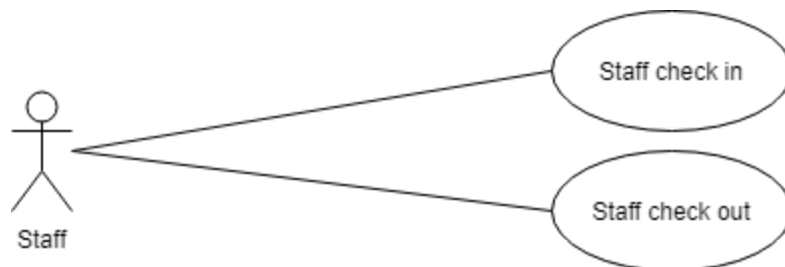


# Human Resource Management Subsystem

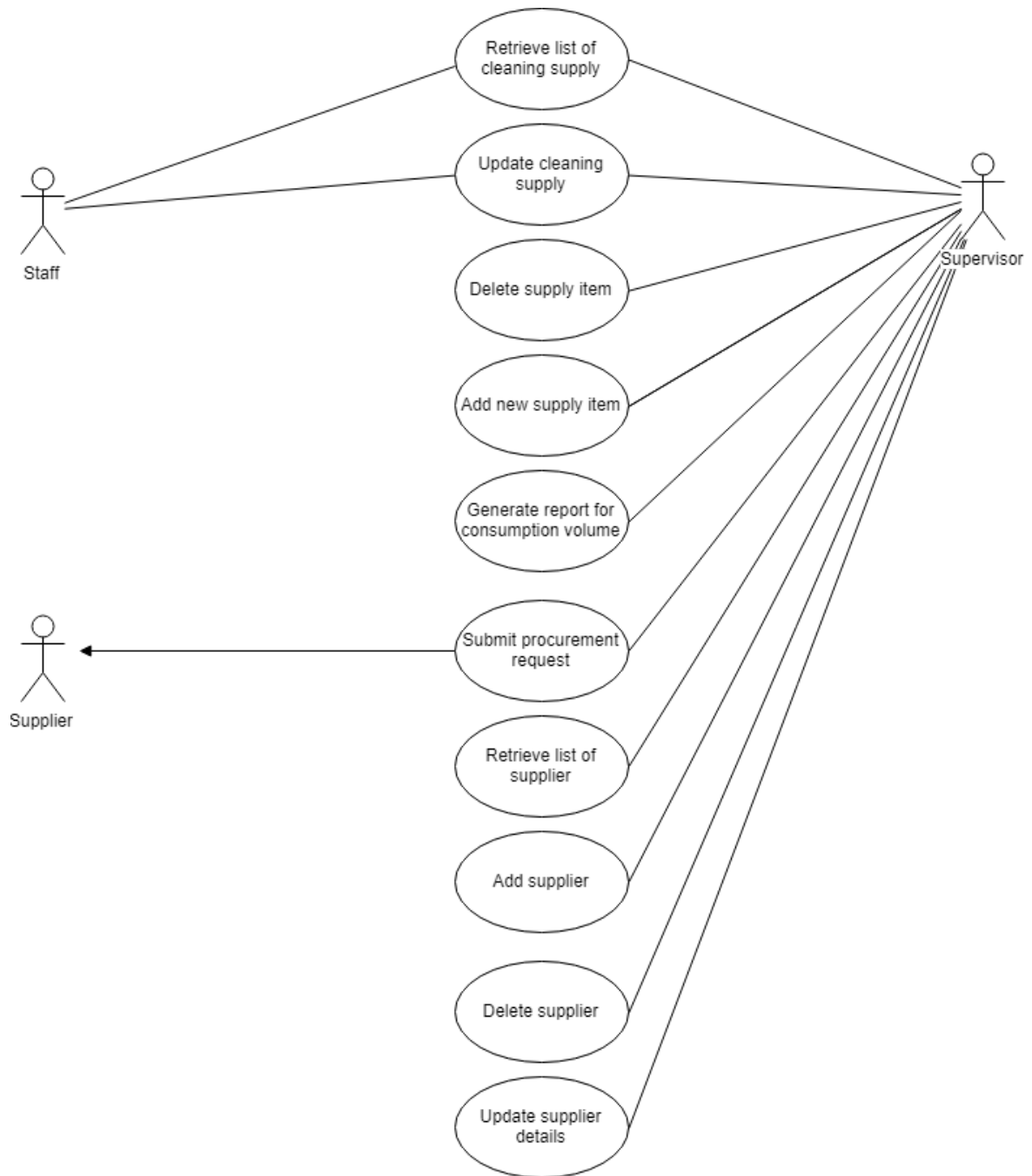
## Employment Management Module



## Attendance Management Module

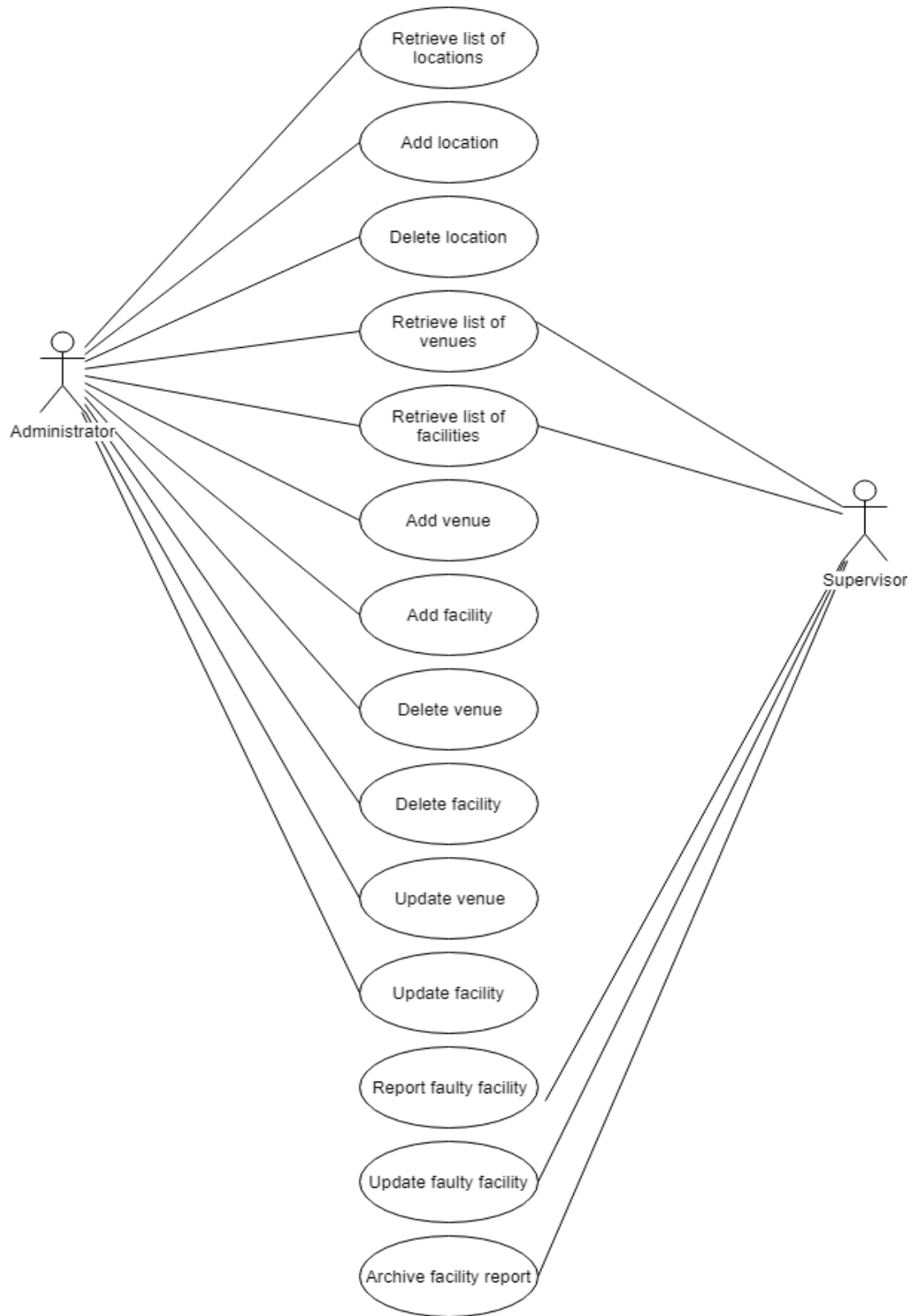


## Asset Management Subsystem

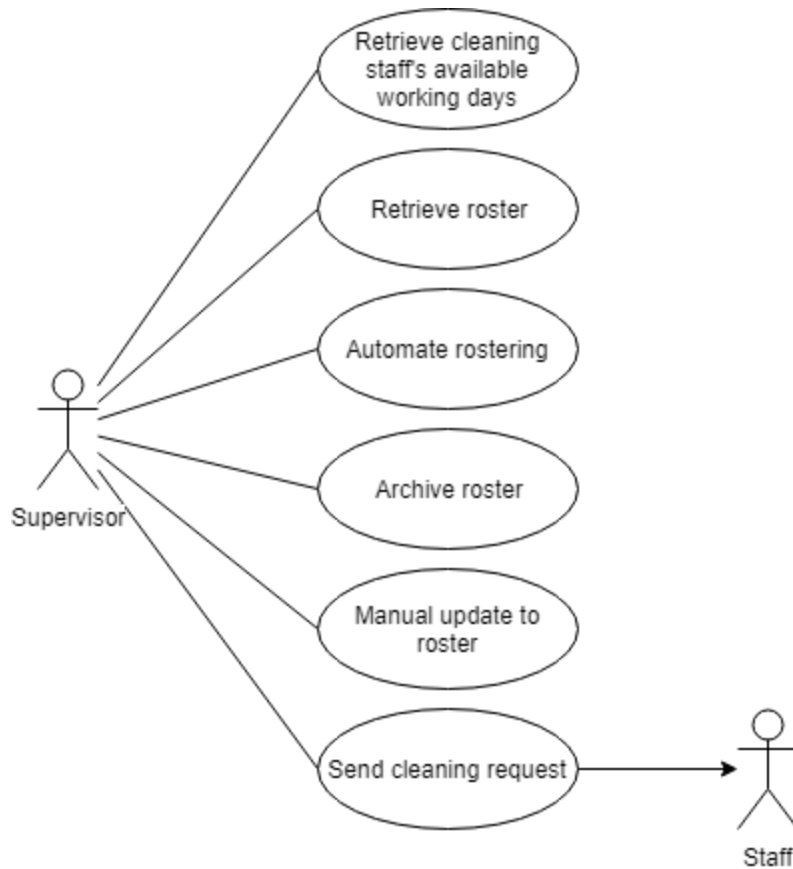




## Area Management Subsystem

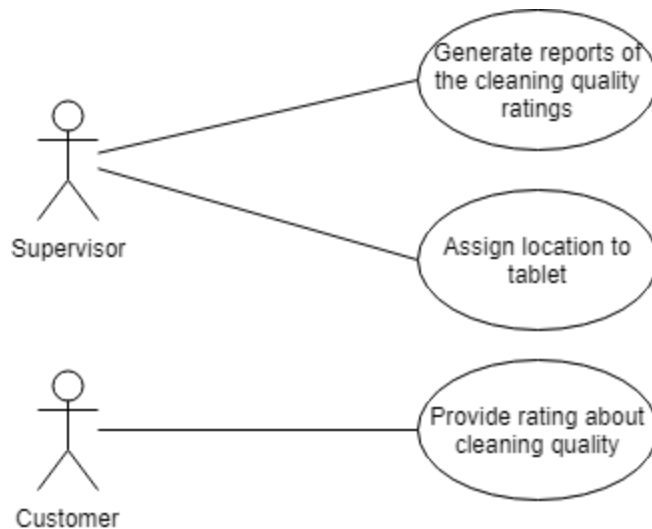


## ***Cleaning Schedule Subsystem***

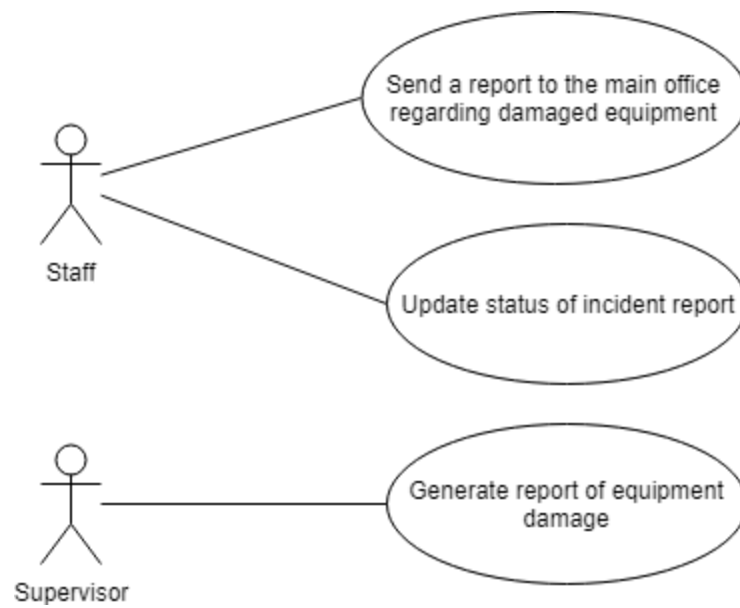


## ***Feedback Management Module***

### *Rating Management Module*



### *Incident Management Module*



Class diagrams containing all entities

*Refer to attachment 1A and attachment 1B*

# Business Requirement Analysis

## **Access management subsystem**

Key objectives of the system:

- 1) Subsystem access restriction to existing user profiles
- 2) Function access restriction based on existing user's access rights
- 3) Functions for easy manipulation of user profile's access attributes
- 4) Function to resolve access issues due to forgotten passwords

To facilitate a comprehensive approach toward related operations, employees will be mapped to a unique user profile.

The administrator subsystem consists of modules that handle the manipulation of said profile for employees within the company database. Access to the subsystem's functionalities are locked behind login/logout which make use of profile attributes to verify proper access. Access to functionalities that allow for the creation, deletion and manipulation of the profile's access attributes are restricted based on the user's role which contains access rights, which can be made and updated within the system as well.

## **Human resource management subsystem**

Key objectives of the system

- 1) Subsystem access restriction to user with "SUPERVISOR" role and higher
- 2) Handle first step of payroll verification. Mitigates unchecked changes to employee payroll
- 3) Automatic change of state and timer operations to be mapped towards card tap
- 4) Registration of card tap action only after option is chosen to prevent accidental changes

The Human Resource management subsystem is split into 2 main modules: the employment management module which deals with employee records and the attendance management module which deals with the tracking of attendance and worked hours.

The employment module handles the supervisor's day to day operations and management of employees. The module is divided into 3 sub modules:

- 1) The viewing and manipulation of employee records which include their personal details and leave records
- 2) Handling of administrative procedures related to attendance such as handling of leave applications, sick leaves and attendance corrections
- 3) Handling of administrative procedures related to salary including but not limited to verification, correction and calculation of an employee's payroll

The attendance module handles the recording of employee attendance. To facilitate this, employees' interact with an attendance terminal through an attendance card. Hours worked for

a shift are calculated based on a timer started upon the first time in of the day and stopped at time out. Tapping during start of break and end of break are treated as time out and time in respectively. Hence, in normal situations the duration of the pre break timer and the post break timer are used to calculate shift hours.

The attendance record of an employee is displayed in a calendar view. Employees are counted as having attended for a specified date if a timer has been started on the specified date.

For the actual recording of attendance, each employee is provided with a employee card which they have to tap on the terminal upon the start and end of their shifts and breaks. Each employee card has a unique id and is mapped to one employee only and will trigger timers depending on which option (start shift, start break, end shift and end break) is chosen before tapping. Tapping without choosing an option will not trigger the change to prevent accidental changes.

### **Asset management subsystem**

Key objectives of the system:

- 1) Recording and manipulation of supply inventory
- 2) Facilitate ordering of supplies from suppliers
- 3) Handles supplier related issues
- 4) Inventory incident report for procurement request

The subsystem's main purpose is to handle and track the supplies within venues. Recording and manipulation of supply levels form the basic requirement to fulfill this purpose. Additionally, a supply procurement module which involves the ordering of supplies from known suppliers as well as the addition of suppliers exist to maintain optimal supply levels. To achieve this optimisation, the system also handles any inventory reports to ensure that the each location is appraised of the supply levels of their respective venues and facilities making it easier to ensure that venues are well supplied.

Sample report

*Refer to attachment 5F*

## Area management subsystem

Key objectives of the system:

- 1) Manipulation and archiving of cleaning locations and their respective venues and facilities
- 2) Generation and sending of fault incident reports

Subsystem's main purpose is to allow for the creation and manipulation of location entities and their related sub-entities in order to facilitate rostering of employees in the cleaning schedule management. The venues will also be used for the feedback system in order to better keep track of quality of work. Additionally, the subsystem will handle any faulty facility reports to keep the location's main office and management well updated of the state of each venue and facility within the location.

## Cleaning schedule subsystem

Key objectives of subsystem:

- 1) Automate rostering, taking into account of cleaning locations, employee locations and available time slots
- 2) Manual roster editing to account for last minute changes

The subsystem's main purpose is to facilitate automated rostering of employee based on their location and available time slots. Using the employee's available leave records as well as their locations, the system creates an optimised roster to ensure that the employees are rostered into the most convenient schedule for them. A manual edit of the roster is in place in case of any last minute changes to an employee's schedule, such as sick leave or urgent leave. This is to ensure that each location can have a functional roster and that there is minimal downtime of available cleaners.

There is also a communication module that allows for cleaning requests to be sent to the cleaning staff in order to efficiently handle any urgent cleaning requests that may occur.

Scenario: Supervisor automates rostering
<ol style="list-style-type: none"><li>1. Supervisor clicks on "manage locations" button on logged in page</li><li>2. Supervisor selects the particular location to retrieve the list of venues under that location</li><li>3. Supervisor selects the particular venue to conduct rostering</li><li>4. Supervisor clicks on "cleaning management schedule" button</li><li>5. Supervisor clicks on "add rostering" button</li><li>6. System detected a rostering conflict and prompts Supervisor by stating "rostering has already been conducted"</li><li>7. Supervisor opt for manual updates to the existing roster instead</li><li>8. System did not detect any rostering conflict</li><li>9. System retrieves all the cleaning staff records under the particular location</li><li>10. System retrieves the gender requirement for the venue</li><li>11. System filters the cleaning staff records based on the gender requirement</li><li>12. System retrieves the cleaning staff's addresses and categorize it corresponding to its town council</li></ol>

13. System categorizes the location's address corresponding to its town council
14. System compares the adjacency of the location's town council and cleaning staff's town council and assigns a score to the cleaning staff (the lower the adjacency, the higher the score)
15. System retrieves the cleaning staff's number of years of past experiences for that location and assign a score accordingly (1 year = 1 point) and assign the score to the cleaning staff
16. System accumulates the scores for each cleaning staff
17. System retrieves the cleaning staff's available working days for the month of rostering to factor in Off, Leaves, and MCs
18. System retrieves the number of staff required for the venue
19. System sorts the cleaning staff based on their accumulated scores (Descending order)
20. System randomly assigns the cleaning staff to the time slots according to the cleaning staff's available working day until the number of required staff has been fulfilled
21. System detected that there are still empty time slots not assigned to any cleaning staff, prompts users by highlighting the empty time slots and displaying "System was unable to fully fill the rostering". System creates new rostering and assigns a new roster Id.
22. System did not detect any empty time slots, displays "System successfully filled up the rostering". System creates new rostering and assigns a new roster Id

### **Feedback subsystem**

Key objectives of system:

- 1) Location based rating system and rating report generation
- 2) Incident report sending as well as updating

To handle feedback from guests, a simple rating management module is installed at various cleaning points similar to other establishments. Ratings terminals will be tablets which will have their locations tracked by the management computer in order to accurately track quality of work. A report can be generated by the tablet and sent to the management computer to better keep track of employee work quality.

Additionally, there is an incident management module can send "incident" reports from staff to the management computer in order to catalogue faulty equipments or accidents at each venue. This module can also track the progress of any work done to fix these incidents.

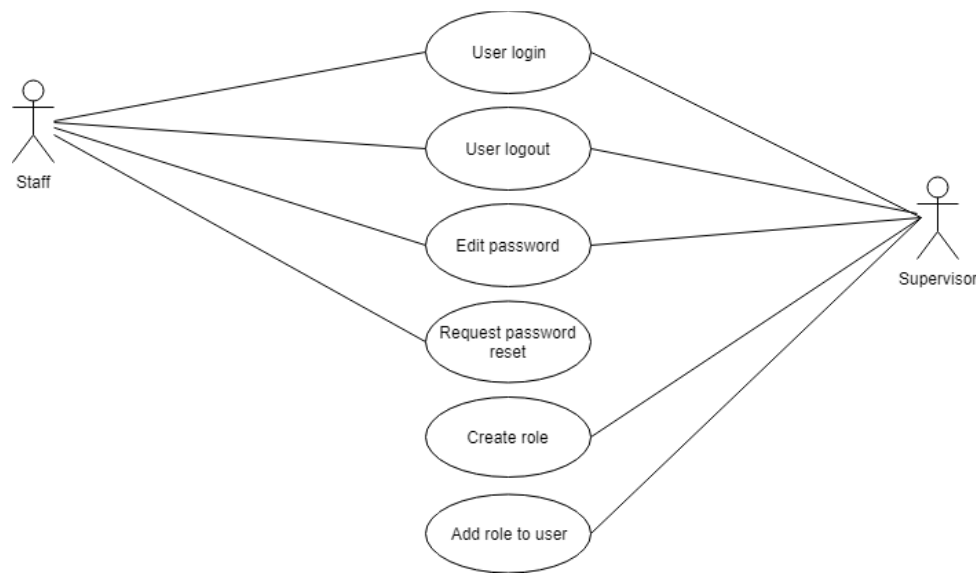
### **Sample report**

*Refer to attachment 7F*

# System Functional Requirements Analysis

## Access Management Subsystem

Use case diagram



*Use case diagram for access management subsystem*



## Use case description for access management subsystem

Use case:	User login					
Triggering case:	User press login button located on toolbar					
Brief description:	User key in username and password to login to the system					
Actors:	Staff/Supervisor					
Pre-conditions:	1. There must not be any logged in account on the computer accessing the system. 2. User must already have their account registered in the system.					
Post-conditions:	Log in user to the system and direct user to the logged in home page.					
Flow of events:	<table><tr><th>Staff / Supervisor</th><th>System</th></tr><tr><td>1. Key in staffId / supervisorId and password into the input field on the login page and click log in.</td><td>1.1 System verifies the validity of the respective Id and password with database.  1.2 If credential fails, go back to login page and go back to step 1.  1.2a. If credential verification is successful, direct to logged in page.</td></tr></table>		Staff / Supervisor	System	1. Key in staffId / supervisorId and password into the input field on the login page and click log in.	1.1 System verifies the validity of the respective Id and password with database.  1.2 If credential fails, go back to login page and go back to step 1.  1.2a. If credential verification is successful, direct to logged in page.
Staff / Supervisor	System					
1. Key in staffId / supervisorId and password into the input field on the login page and click log in.	1.1 System verifies the validity of the respective Id and password with database.  1.2 If credential fails, go back to login page and go back to step 1.  1.2a. If credential verification is successful, direct to logged in page.					
Exception conditions:	Nil					

Use case:	User logout					
Triggering case:	User press log out button located on toolbar					
Brief description:	User is logged out of system.					
Actors:	Staff/Supervisor					
Pre-conditions:	A user is currently logged in					
Post-conditions:	User is logged out of the system System cache is cleared					
Flow of events:	<table><tr><th>Staff/Supervisor</th><th>System</th></tr><tr><td>1. User presses log out.</td><td>1.1 User is logged out of system.</td></tr></table>		Staff/Supervisor	System	1. User presses log out.	1.1 User is logged out of system.
Staff/Supervisor	System					
1. User presses log out.	1.1 User is logged out of system.					
Exception conditions:	Nil					

Use case:	Edit password					
Triggering case:	Edit password is chosen in profile menu					
Brief description:	Current password is changed, and action is verified by inputting previous password					
Actors:	Staff/Supervisor					
Pre-conditions:	User is logged in					
Post-conditions:	Password associated with user is changed and change is reflected in relevant databases					
Flow of events:	<table><tr><th>Staff/Supervisor</th><th>System</th></tr><tr><td>1. User inputs old (current) password, new password and repeats input of new password</td><td>1.1 Old password is verified by previous database  1.2 New password and repeat are verified to be equivalent. If not equivalent, repeat step 1  1.2 Display message confirming the change in password to user and current shift supervisor</td></tr></table>		Staff/Supervisor	System	1. User inputs old (current) password, new password and repeats input of new password	1.1 Old password is verified by previous database  1.2 New password and repeat are verified to be equivalent. If not equivalent, repeat step 1  1.2 Display message confirming the change in password to user and current shift supervisor
Staff/Supervisor	System					
1. User inputs old (current) password, new password and repeats input of new password	1.1 Old password is verified by previous database  1.2 New password and repeat are verified to be equivalent. If not equivalent, repeat step 1  1.2 Display message confirming the change in password to user and current shift supervisor					
Exception conditions:	1.1 If old password is not identical to password associated with user in database, change in password fails					

Use case:	Request password reset							
Triggering case:	Forgot password is clicked in login screen							
Brief description:	Staff requests for a reset of password from supervisor							
Actors:	Staff/Supervisor							
Pre-conditions:	User is not logged in							
Post-conditions:	Password associated with user is reset to a new randomly generated password that will be provided to user							
Flow of events:	<table><tr><th>Staff / Supervisor</th><th>System</th></tr><tr><td>1. User enters staffId / supervisorId and clicks reset password</td><td>1.1 System sends notification to supervisor about reset request 1.2 Send email to email address associated with staff to confirm reset</td></tr><tr><td>2 User confirms reset from sent email</td><td>2.1 Generate new random password 2.2 Change associated password with generated password 2.3 Send email to staff containing new generated password</td></tr></table>		Staff / Supervisor	System	1. User enters staffId / supervisorId and clicks reset password	1.1 System sends notification to supervisor about reset request 1.2 Send email to email address associated with staff to confirm reset	2 User confirms reset from sent email	2.1 Generate new random password 2.2 Change associated password with generated password 2.3 Send email to staff containing new generated password
Staff / Supervisor	System							
1. User enters staffId / supervisorId and clicks reset password	1.1 System sends notification to supervisor about reset request 1.2 Send email to email address associated with staff to confirm reset							
2 User confirms reset from sent email	2.1 Generate new random password 2.2 Change associated password with generated password 2.3 Send email to staff containing new generated password							
Exception conditions:	Staff does not follow email confirmation. Reset password is cancelled							

Use case:	Create role									
Triggering case:	Supervisor selects “Create a role”									
Brief description:	A new role, applicable to staff, is created along with associated access rights									
Actors:	Supervisor									
Pre-conditions:	Role with exact same access rights/name does not exist in database									
Post-conditions:	New role with chosen access rights is created									
Flow of events:	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Supervisor chooses create a role</td><td>1.1 Displays a list of all access rights that can be given</td></tr><tr><td>2. Supervisor chooses which access rights to add to Role</td><td></td></tr><tr><td>3. Supervisor inputs name of role</td><td>3.1 System verifies that current roles’ names are not identical to the input  3.1a If any existing role has the same name as the input name, prompt user to repeat step 3  3.2 New role with chosen access rights is created and saved.</td></tr></table>		Supervisor	System	1. Supervisor chooses create a role	1.1 Displays a list of all access rights that can be given	2. Supervisor chooses which access rights to add to Role		3. Supervisor inputs name of role	3.1 System verifies that current roles’ names are not identical to the input  3.1a If any existing role has the same name as the input name, prompt user to repeat step 3  3.2 New role with chosen access rights is created and saved.
Supervisor	System									
1. Supervisor chooses create a role	1.1 Displays a list of all access rights that can be given									
2. Supervisor chooses which access rights to add to Role										
3. Supervisor inputs name of role	3.1 System verifies that current roles’ names are not identical to the input  3.1a If any existing role has the same name as the input name, prompt user to repeat step 3  3.2 New role with chosen access rights is created and saved.									
Exception conditions:	(3) User repeatedly does not input unique role name. Create role cancelled									

Use case:	Add role to user					
Triggering case:	Receive notification to add rights to existing user					
Brief description:	Existing user’s role is replaced by new role					
Actors:	Supervisor					
Pre-conditions:	User to change already exists in database Role to be added to user already exists					
Post-conditions:	User granted chosen role in database					
Flow of events:	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include “View Staff Details”  2. User selects new role to confer from options displayed</td><td>2.1 System sends email to company HR to confirm grant access  2.2 After receiving confirmation, grants identified user access rights</td></tr></table>		Supervisor	System	1. Include “View Staff Details”  2. User selects new role to confer from options displayed	2.1 System sends email to company HR to confirm grant access  2.2 After receiving confirmation, grants identified user access rights
Supervisor	System					
1. Include “View Staff Details”  2. User selects new role to confer from options displayed	2.1 System sends email to company HR to confirm grant access  2.2 After receiving confirmation, grants identified user access rights					
Exception conditions:	Nil					

Simplified entity class diagram

*Refer to attachment 2A*

Non-entity class diagram

*Refer to attachment 2B*

Sequence diagrams

*Refer to attachment 2C, 2D and 2E*

# Human Resource Management Subsystem

## Employment Management Module

Use case diagram



*Use case diagram for employment management module*

## Use case descriptions for employment management module

Use case:	Create new staff record					
Triggering case:	Creating new account for new hire					
Brief description:	Supervisor add new staff record with personal details into the system's database.					
Actors:	Supervisor					
Pre-conditions:	1. Supervisor must be logged in to the system 2. The computer accessing the system must be connected to an active printer					
Post-conditions:	Add the staff into system's database and generate staff ID and default password for staff.					
Flow of events:	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Click on "Create new staff record" on logged in page.  2. Insert new staff name, age, gender, identification card number, nationality, race, religion, marital status, email address, house address, contact number, bank account number, associated bank, past experiences into the system.</td><td>1.1 Display create new staff page  2.1 Verify that there is no existing record for email address, contact number, bank account number along with associated bank, and identification card number.  2.2 If verification is successful, create new staff record in the staff database.  2.2a. If verification fails, display error message "The <i>information</i>* already exist". Go back to step 1.1.  2.3 Generate staff ID by incrementing the latest staff ID by 1 and generate a random password for the staff.</td></tr></table>		Supervisor	System	1. Click on "Create new staff record" on logged in page.  2. Insert new staff name, age, gender, identification card number, nationality, race, religion, marital status, email address, house address, contact number, bank account number, associated bank, past experiences into the system.	1.1 Display create new staff page  2.1 Verify that there is no existing record for email address, contact number, bank account number along with associated bank, and identification card number.  2.2 If verification is successful, create new staff record in the staff database.  2.2a. If verification fails, display error message "The <i>information</i> * already exist". Go back to step 1.1.  2.3 Generate staff ID by incrementing the latest staff ID by 1 and generate a random password for the staff.
Supervisor	System					
1. Click on "Create new staff record" on logged in page.  2. Insert new staff name, age, gender, identification card number, nationality, race, religion, marital status, email address, house address, contact number, bank account number, associated bank, past experiences into the system.	1.1 Display create new staff page  2.1 Verify that there is no existing record for email address, contact number, bank account number along with associated bank, and identification card number.  2.2 If verification is successful, create new staff record in the staff database.  2.2a. If verification fails, display error message "The <i>information</i> * already exist". Go back to step 1.1.  2.3 Generate staff ID by incrementing the latest staff ID by 1 and generate a random password for the staff.					



	<div> <div></div> <div> 2.4 Print out the staff ID and default password using the connected printer.   2.5 Send a copy of the email containing the staff ID and password to the email address obtained from the new staff record. </div> </div> <p><i>*The word information is to be substituted with any of one the details that are being captured in the system</i></p>
Exception conditions:	Nil

Use case:	View staff list					
Triggering case:	Supervisor to view the list of staff					
Brief description:	Supervisor view the list of staff record					
Actors:	Supervisor					
Pre-conditions:	1. Supervisor must be logged into the system					
Post-conditions:	A list of staff record will be retrieved from database and displayed.					
Flow of events	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Click “View staff list”.</td><td>1.1 Retrieve staff information from database and display a list of all staff record with only staff ID, name, contact number and identification card number.</td></tr></table>		Supervisor	System	1. Click “View staff list”.	1.1 Retrieve staff information from database and display a list of all staff record with only staff ID, name, contact number and identification card number.
Supervisor	System					
1. Click “View staff list”.	1.1 Retrieve staff information from database and display a list of all staff record with only staff ID, name, contact number and identification card number.					
Exception conditions:	Nil					

Use case:	View staff details													
Triggering case:	Supervisor to view details of a particular staff record													
Brief description:	Supervisor provide staff ID or choose from a list of staff record to retrieve staff details from database													
Actors:	Supervisor													
Pre-conditions:	1. Supervisor must be logged into the system													
Post-conditions:	Staff details will be retrieved from database and displayed.													
Flow of events	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include View staff list</td><td></td></tr><tr><td>2. Enter staff ID into the search bar to search for the staff.</td><td>2.1 Display the staff record that matches the staff ID.</td></tr><tr><td></td><td>2.1a) No staff record found, check for typographical error and go back to step 2.2.</td></tr><tr><td>2a. Search for the desired staff record from the list of staff records.</td><td></td></tr><tr><td>3. Click on the desired staff record.</td><td>3.1 Go to selected staff's details page. Retrieve and display full details of the selected staff record.</td></tr></table>		Supervisor	System	1. Include View staff list		2. Enter staff ID into the search bar to search for the staff.	2.1 Display the staff record that matches the staff ID.		2.1a) No staff record found, check for typographical error and go back to step 2.2.	2a. Search for the desired staff record from the list of staff records.		3. Click on the desired staff record.	3.1 Go to selected staff's details page. Retrieve and display full details of the selected staff record.
Supervisor	System													
1. Include View staff list														
2. Enter staff ID into the search bar to search for the staff.	2.1 Display the staff record that matches the staff ID.													
	2.1a) No staff record found, check for typographical error and go back to step 2.2.													
2a. Search for the desired staff record from the list of staff records.														
3. Click on the desired staff record.	3.1 Go to selected staff's details page. Retrieve and display full details of the selected staff record.													
Exception conditions:	Nil													

Use case:	Update staff details					
Triggering case:	Change in staff personal details					
Brief description:	Supervisor update staff details into the system database.					
Actors:	Supervisor					
Pre-conditions:	1. Supervisor must be logged into the system 2. Staff record of the selected staff must exist in the database					
Post-conditions:	The staff details updated in the database.					
Flow of events	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include View staff details  2. Click “Update” button  3. Update the detail fields accordingly.</td><td>2.1 Display staff details in editable format.  3.1 Verify that there is no existing record for the new email address, contact number, bank account number and identification card number.  3.2 If verification is successful, update the staff record with the respective updated details into the database.  3.2a. If verification fails, display error message “The <i>information</i>* already exist”. Go back to step 3.2.  3.3 Display the updated staff details.  3.4 Send a copy of the update(s) to the email address of the staff to notify him/her that the details have been updated.</td></tr></table> <p><i>*The word information is to be substituted with any of one the details that are being captured in the system</i></p>		Supervisor	System	1. Include View staff details  2. Click “Update” button  3. Update the detail fields accordingly.	2.1 Display staff details in editable format.  3.1 Verify that there is no existing record for the new email address, contact number, bank account number and identification card number.  3.2 If verification is successful, update the staff record with the respective updated details into the database.  3.2a. If verification fails, display error message “The <i>information</i> * already exist”. Go back to step 3.2.  3.3 Display the updated staff details.  3.4 Send a copy of the update(s) to the email address of the staff to notify him/her that the details have been updated.
Supervisor	System					
1. Include View staff details  2. Click “Update” button  3. Update the detail fields accordingly.	2.1 Display staff details in editable format.  3.1 Verify that there is no existing record for the new email address, contact number, bank account number and identification card number.  3.2 If verification is successful, update the staff record with the respective updated details into the database.  3.2a. If verification fails, display error message “The <i>information</i> * already exist”. Go back to step 3.2.  3.3 Display the updated staff details.  3.4 Send a copy of the update(s) to the email address of the staff to notify him/her that the details have been updated.					
Exception conditions:	Nil					

Use case:	Archive staff record					
Triggering case:	Staff leaves the company					
Brief description:	Supervisor archive staff record after the staff leaves the company.					
Actors:	Supervisor					
Pre-conditions:	1. Supervisor must be logged into the system 2. Staff record of the selected staff must exist in the database					
Post-conditions:	Staff details will be archived and removed from primary storage.					
Flow of events	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include View staff details.  2. Click on “Archive” button.</td><td>2.1 Archive the selected staff record and remove the record from primary storage.  2.2 Display “Staff record successfully archived” message. Redirect back to the page after clicking “View staff details”.  2.3 Retrieve staff information from database and display a updated list of all staff record with only staff ID, name, contact number and identification card number.</td></tr></table>		Supervisor	System	1. Include View staff details.  2. Click on “Archive” button.	2.1 Archive the selected staff record and remove the record from primary storage.  2.2 Display “Staff record successfully archived” message. Redirect back to the page after clicking “View staff details”.  2.3 Retrieve staff information from database and display a updated list of all staff record with only staff ID, name, contact number and identification card number.
Supervisor	System					
1. Include View staff details.  2. Click on “Archive” button.	2.1 Archive the selected staff record and remove the record from primary storage.  2.2 Display “Staff record successfully archived” message. Redirect back to the page after clicking “View staff details”.  2.3 Retrieve staff information from database and display a updated list of all staff record with only staff ID, name, contact number and identification card number.					
Exception conditions:	Nil					

Use case:	View leave application record					
Triggering case:	Supervisor check leave for the selected staff					
Brief description:	Supervisor check leave application record of the selected staff					
Actors:	Supervisor					
Pre-conditions:	1. Supervisor must be logged in to the system 2. Staff record of the selected staff must exist in the database					
Post-conditions:	A list of leave application record will be displayed					
Flow of events	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include View staff details.  2. Click on “View leave” button.</td><td>2.1 Go to leave application record page. Retrieve all leave application record and display all past leave application of the selected staff.*</td></tr></table>		Supervisor	System	1. Include View staff details.  2. Click on “View leave” button.	2.1 Go to leave application record page. Retrieve all leave application record and display all past leave application of the selected staff.*
Supervisor	System					
1. Include View staff details.  2. Click on “View leave” button.	2.1 Go to leave application record page. Retrieve all leave application record and display all past leave application of the selected staff.*					
Exception conditions:	Nil					

Use case:	Apply leave for staff					
Triggering case:	Staff applies for leave					
Brief description:	Supervisor help staff to apply for leave					
Actors:	Supervisor					
Pre-conditions:	1. Supervisor must be logged in to the system 2. Staff record for the staff applying for leave must exist in the database					
Post-conditions:	Leave application record will be added into the database.					
Flow of events	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include View staff details.  2. Include View leave application record  3. Click on “Apply” button.  4. Select required information accordingly using date and time picker and submit.</td><td>3.1 Display a pop-up window with date and time picker for supervisor to select starting date, ending date, starting time, ending time, type of leave and reason for applying leave.  4.1 Check that there is no exact same leave application record in the database by verifying the corresponding date and time.  4.2 Exact record already exist in database. Display message “Leave application record already submitted” and go back to step 4.  4.2a Record does not exist in database. Verify that the ending time and, or date is bigger than starting time and, or date. At the same time, verify that the starting date is bigger than the current system date.</td></tr></table>		Supervisor	System	1. Include View staff details.  2. Include View leave application record  3. Click on “Apply” button.  4. Select required information accordingly using date and time picker and submit.	3.1 Display a pop-up window with date and time picker for supervisor to select starting date, ending date, starting time, ending time, type of leave and reason for applying leave.  4.1 Check that there is no exact same leave application record in the database by verifying the corresponding date and time.  4.2 Exact record already exist in database. Display message “Leave application record already submitted” and go back to step 4.  4.2a Record does not exist in database. Verify that the ending time and, or date is bigger than starting time and, or date. At the same time, verify that the starting date is bigger than the current system date.
Supervisor	System					
1. Include View staff details.  2. Include View leave application record  3. Click on “Apply” button.  4. Select required information accordingly using date and time picker and submit.	3.1 Display a pop-up window with date and time picker for supervisor to select starting date, ending date, starting time, ending time, type of leave and reason for applying leave.  4.1 Check that there is no exact same leave application record in the database by verifying the corresponding date and time.  4.2 Exact record already exist in database. Display message “Leave application record already submitted” and go back to step 4.  4.2a Record does not exist in database. Verify that the ending time and, or date is bigger than starting time and, or date. At the same time, verify that the starting date is bigger than the current system date.					

	<div> <div></div> <div> <p>4.2a.1 Verification successful, add in leave application record in the database. Set the status of the application as “Pending for approval”.</p> <p>4.2a.1a Verification unsuccessful, display error message and go back to step 4.</p> <p>4.3 Go back to leave application record page and display all leave application records.</p> </div> </div>
Exception conditions:	Nil

Use case:	Modify leave for staff					
Triggering case:	Staff change the date or date for leave application					
Brief description:	Supervisor help staff to modify the time or date for the applied leave					
Actors:	Supervisor					
Pre-conditions:	<div>1. Supervisor must be logged in to the system.</div> <div>2. Staff record for the staff applying for leave must exist in database.</div> <div>3. The leave application status must be pending for verification at the point of modification.</div>					
Post-conditions:	Leave application record will be updated into the database.					
Flow of events	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td><div>1. Include View staff details.</div><div>2. Include View leave application record</div><div>3. Check the selection button beside the leave application record that is to be modified and click the “Modify” button.</div><div>4. Change the starting and ending date and time accordingly. Click “Confirm” to apply the changes.</div></td><td><div>3.1 Display a pop-up window with the applied starting date, time and ending date, time which allows modifications.</div><div>4.1 Verify that the ending time and, or date is bigger than starting time and, or date. At the same time, verify that the starting date is bigger than the current system date.</div><div>4.2 Verification successful, update the corresponding date and time in the database.</div><div>4.2a. Verification unsuccessful, display error message “Ending date/time should be after starting date/time.” or “Starting date/time should be after current system date/time” and go back to step 4.</div></td></tr></table>		Supervisor	System	<div>1. Include View staff details.</div> <div>2. Include View leave application record</div> <div>3. Check the selection button beside the leave application record that is to be modified and click the “Modify” button.</div> <div>4. Change the starting and ending date and time accordingly. Click “Confirm” to apply the changes.</div>	<div>3.1 Display a pop-up window with the applied starting date, time and ending date, time which allows modifications.</div> <div>4.1 Verify that the ending time and, or date is bigger than starting time and, or date. At the same time, verify that the starting date is bigger than the current system date.</div> <div>4.2 Verification successful, update the corresponding date and time in the database.</div> <div>4.2a. Verification unsuccessful, display error message “Ending date/time should be after starting date/time.” or “Starting date/time should be after current system date/time” and go back to step 4.</div>
Supervisor	System					
<div>1. Include View staff details.</div> <div>2. Include View leave application record</div> <div>3. Check the selection button beside the leave application record that is to be modified and click the “Modify” button.</div> <div>4. Change the starting and ending date and time accordingly. Click “Confirm” to apply the changes.</div>	<div>3.1 Display a pop-up window with the applied starting date, time and ending date, time which allows modifications.</div> <div>4.1 Verify that the ending time and, or date is bigger than starting time and, or date. At the same time, verify that the starting date is bigger than the current system date.</div> <div>4.2 Verification successful, update the corresponding date and time in the database.</div> <div>4.2a. Verification unsuccessful, display error message “Ending date/time should be after starting date/time.” or “Starting date/time should be after current system date/time” and go back to step 4.</div>					
Exception conditions:	Nil					



Use case:	View all pending approval leave applications					
Triggering case:	Staff wants to view leave application that is pending for approval					
Brief description:	Supervisor retrieve the list of leaving applications that are pending for verification					
Actors:	Supervisor					
Pre-conditions:	1. Supervisor must be logged in to the system. 2. Staff record must exist in the database. 3. Staff must have successfully applied for leave previously.					
Post-conditions:	Leave application that is pending for approval will be retrieved and displayed.					
Flow of events	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Click “View pending leave application” button on the logged in home page.</td><td>1.1 Retrieve and display all leave application record that is pending for approval.</td></tr></table>		Supervisor	System	1. Click “View pending leave application” button on the logged in home page.	1.1 Retrieve and display all leave application record that is pending for approval.
Supervisor	System					
1. Click “View pending leave application” button on the logged in home page.	1.1 Retrieve and display all leave application record that is pending for approval.					
Exception conditions:	Nil					

Use case:	Approve leave application for staff											
Triggering case:	Staff applied for leave and waiting for approval from supervisor											
Brief description:	Supervisor will determine whether to approve the leave based on the amount of manpower available during the applied leave period as well as past application record of the staff.											
Actors:	Supervisor											
Pre-conditions:	1. Supervisor must be logged in to the system. 2. Staff record must exist in the database. 3. Staff must have successfully applied for leave previously.											
Post-conditions:	Leave application status of the staff is updated and email as well as sms notification is sent to the staff.											
Flow of events	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include View all pending leave application</td><td></td></tr><tr><td>2. Consider manpower needed during the applied leave period as well the past application of the staff.</td><td></td></tr><tr><td>3. Check the selection button beside the leave application record that is to be approved and click the “Approve” button.</td><td>3.1 Prompt with a message to confirm the operation.</td></tr><tr><td>4. Click on “Confirm” to approve the leave application.</td><td>4.1 Update the status of leave application from “Pending from approval” to “Approved”.  4.2 Update the attendance of the selected staff by marking as “On leave” for the approved leave period.  4.3 Send email and SMS notification to the staff to notify the staff of the outcome of the leave application.</td></tr></table>		Supervisor	System	1. Include View all pending leave application		2. Consider manpower needed during the applied leave period as well the past application of the staff.		3. Check the selection button beside the leave application record that is to be approved and click the “Approve” button.	3.1 Prompt with a message to confirm the operation.	4. Click on “Confirm” to approve the leave application.	4.1 Update the status of leave application from “Pending from approval” to “Approved”.  4.2 Update the attendance of the selected staff by marking as “On leave” for the approved leave period.  4.3 Send email and SMS notification to the staff to notify the staff of the outcome of the leave application.
Supervisor	System											
1. Include View all pending leave application												
2. Consider manpower needed during the applied leave period as well the past application of the staff.												
3. Check the selection button beside the leave application record that is to be approved and click the “Approve” button.	3.1 Prompt with a message to confirm the operation.											
4. Click on “Confirm” to approve the leave application.	4.1 Update the status of leave application from “Pending from approval” to “Approved”.  4.2 Update the attendance of the selected staff by marking as “On leave” for the approved leave period.  4.3 Send email and SMS notification to the staff to notify the staff of the outcome of the leave application.											

	<table border="1"> <tr> <td></td><td>4.4 Go back to pending leave application page and display all leave application record that is pending for approval.</td></tr> </table>		4.4 Go back to pending leave application page and display all leave application record that is pending for approval.
	4.4 Go back to pending leave application page and display all leave application record that is pending for approval.		
Exception conditions:	Nil		

Use case:	Reject leave application for staff					
Triggering case:	Staff applied for leave and waiting for approval from supervisor					
Brief description:	Supervisor will determine whether to approve the leave based on the amount of manpower available during the applied leave period as well as past application record of the staff.					
Actors:	Supervisor					
Pre-conditions:	<div>1. Supervisor must be logged in to the system.</div> <div>2. Staff record must exist in the database.</div> <div>3. Staff must have successfully applied for leave previously.</div>					
Post-conditions:	Leave application status of the staff is updated and email as well as sms notification is sent to the staff.					
Flow of events	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td><div>1. Click “View all leave application” button on the logged in home page.</div><div>2. Take into consideration of the amount of manpower needed during the applied leave period as well the past application of the staff.</div><div>3. Check the selection button beside the leave application record that is to be rejected</div></td><td><div>1.1 Retrieve and display all leave application records pending for approval.</div><div>3.1 Prompt with a message to confirm the operation.</div></td></tr></table>		Supervisor	System	<div>1. Click “View all leave application” button on the logged in home page.</div> <div>2. Take into consideration of the amount of manpower needed during the applied leave period as well the past application of the staff.</div> <div>3. Check the selection button beside the leave application record that is to be rejected</div>	<div>1.1 Retrieve and display all leave application records pending for approval.</div> <div>3.1 Prompt with a message to confirm the operation.</div>
Supervisor	System					
<div>1. Click “View all leave application” button on the logged in home page.</div> <div>2. Take into consideration of the amount of manpower needed during the applied leave period as well the past application of the staff.</div> <div>3. Check the selection button beside the leave application record that is to be rejected</div>	<div>1.1 Retrieve and display all leave application records pending for approval.</div> <div>3.1 Prompt with a message to confirm the operation.</div>					

	<p>and click the “Reject” button</p> <p>4. Click on “Confirm” to reject the leave application.</p>	<p>4.1 Update status of leave application from “Pending approval” to “Rejected”.</p> <p>4.2 Send email and SMS notification to the staff to notify the staff of the outcome of the leave application.</p> <p>4.3 Go back to pending leave application page and display all leave application records pending approval.</p>
Exception conditions:	Nil	

Use case:	Cancel leave for staff
Triggering case:	Staff cancel applied leave
Brief description:	Supervisor help to cancel leave that the staff applied previously
Actors:	Supervisor
Pre-conditions:	<ol style="list-style-type: none"> <li>1. Supervisor must be logged into the system.</li> <li>2. Staff record must exist in the database.</li> <li>3. Staff must have successfully applied for leave previously.</li> </ol>
Post-conditions:	Leave application record will be removed from the database.
Flow of events	
Exception conditions:	Nil

Use case:	Submit medical certificate for staff													
Triggering case:	Staff submit document as prove for sick leave													
Brief description:	Supervisor help to submit the medical certificate into the system													
Actors:	Supervisor													
Pre-conditions:	1. Supervisor must be logged into the system. 2. Staff record must exist in the database.													
Post-conditions:	New leave application record of sick leave type is created with attached medical certificate													
Flow of events	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include View staff details. 2. Include View leave application record</td><td></td></tr><tr><td>3. Click “Submit MC” button</td><td>3.1 Display a pop up window with date and time picker for supervisor to select starting date, ending date, starting time, ending time, reason for sick leave and upload medical certificate field.</td></tr><tr><td>4. Upload the soft copy of MC to the system.</td><td></td></tr><tr><td>5. Click the “Submit” button.</td><td>5.1 Create a leave application record in the database and mark as sick leave.  5.2 Update the attendance of the selected staff by marking “Leave with MC” for the period approved by medical certificate.</td></tr><tr><td></td><td>5.3 Go back to leave application record page and display all leave application records.</td></tr></table>		Supervisor	System	1. Include View staff details. 2. Include View leave application record		3. Click “Submit MC” button	3.1 Display a pop up window with date and time picker for supervisor to select starting date, ending date, starting time, ending time, reason for sick leave and upload medical certificate field.	4. Upload the soft copy of MC to the system.		5. Click the “Submit” button.	5.1 Create a leave application record in the database and mark as sick leave.  5.2 Update the attendance of the selected staff by marking “Leave with MC” for the period approved by medical certificate.		5.3 Go back to leave application record page and display all leave application records.
Supervisor	System													
1. Include View staff details. 2. Include View leave application record														
3. Click “Submit MC” button	3.1 Display a pop up window with date and time picker for supervisor to select starting date, ending date, starting time, ending time, reason for sick leave and upload medical certificate field.													
4. Upload the soft copy of MC to the system.														
5. Click the “Submit” button.	5.1 Create a leave application record in the database and mark as sick leave.  5.2 Update the attendance of the selected staff by marking “Leave with MC” for the period approved by medical certificate.													
	5.3 Go back to leave application record page and display all leave application records.													

Exception conditions:	Nil
-----------------------	-----

Use case:	View attendance of staff					
Triggering case:	Supervisor wants to check the attendance of staff					
Brief description:	Supervisor access the system to check attendance of staff					
Actors:	Supervisor					
Pre-conditions:	1. Supervisor must be logged into the system. 2. Staff record must exist in the database.					
Post-conditions:	The attendance of the selected staff will be displayed in a monthly calendar view.					
Flow of events	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include View staff details.  2. Click “View attendance”</td><td>2.1 Retrieve and display the attendance record of the selected staff in monthly calendar view of the current month, specifying the start and end time of working hour of each day.</td></tr></table>		Supervisor	System	1. Include View staff details.  2. Click “View attendance”	2.1 Retrieve and display the attendance record of the selected staff in monthly calendar view of the current month, specifying the start and end time of working hour of each day.
Supervisor	System					
1. Include View staff details.  2. Click “View attendance”	2.1 Retrieve and display the attendance record of the selected staff in monthly calendar view of the current month, specifying the start and end time of working hour of each day.					
Exception conditions:	Nil					

Use case:	Modify attendance of staff					
Triggering case:	Error in attendance taking procedure					
Brief description:	Supervisor manually modify the attendance of the staff for cases such as forgetting to tap in or tap out, or due to system error.					
Actors:	Supervisor					
Pre-conditions:	1. Supervisor must be logged into the system. 2. Staff record must exist in the database.					
Post-conditions:	The attendance of the selected staff will be modified and updated in the database.					
Flow of events	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include View staff details.  2. Include View attendance of staff.  3. Click “Modify” button  4. Enter password.   5. Click on the day on the calendar which modification to the working hour of the selected staff is to be modified.  6. Modify and submit the starting and ending time accordingly and correctly.</td><td>3.1 System prompt user to enter password of the currently logged in account for verification.  4.1 Verify password  4.2 If verification unsuccessful, display error message “Incorrect password” and go back to step 3.1.  4.2a. If verification successful, display the attendance in calendar view where all the days of the month will be hyperlinked and can be further accessed. 5.1 Display a pop-up window showing the starting and ending time of work which allows modification.  6.1 Verify that the ending time is after the starting time.  6.2 If verification is unsuccessful,</td></tr></table>		Supervisor	System	1. Include View staff details.  2. Include View attendance of staff.  3. Click “Modify” button  4. Enter password.   5. Click on the day on the calendar which modification to the working hour of the selected staff is to be modified.  6. Modify and submit the starting and ending time accordingly and correctly.	3.1 System prompt user to enter password of the currently logged in account for verification.  4.1 Verify password  4.2 If verification unsuccessful, display error message “Incorrect password” and go back to step 3.1.  4.2a. If verification successful, display the attendance in calendar view where all the days of the month will be hyperlinked and can be further accessed. 5.1 Display a pop-up window showing the starting and ending time of work which allows modification.  6.1 Verify that the ending time is after the starting time.  6.2 If verification is unsuccessful,
Supervisor	System					
1. Include View staff details.  2. Include View attendance of staff.  3. Click “Modify” button  4. Enter password.   5. Click on the day on the calendar which modification to the working hour of the selected staff is to be modified.  6. Modify and submit the starting and ending time accordingly and correctly.	3.1 System prompt user to enter password of the currently logged in account for verification.  4.1 Verify password  4.2 If verification unsuccessful, display error message “Incorrect password” and go back to step 3.1.  4.2a. If verification successful, display the attendance in calendar view where all the days of the month will be hyperlinked and can be further accessed. 5.1 Display a pop-up window showing the starting and ending time of work which allows modification.  6.1 Verify that the ending time is after the starting time.  6.2 If verification is unsuccessful,					

	<div> display error message “Ending time should be after starting time” and go back to step 6.   6.2a. If verification is successful, update the changes in the database.   6.3 Close the pop-up window. Retrieve and display the updated working hour information on the calendar. </div>
Exception conditions:	Nil

Use case:	Verify payroll of staff					
Triggering case:	End of the month					
Brief description:	Supervisor verify the payroll of the staff based on the attendance before submitting the human relations department					
Actors:	Supervisor					
Pre-conditions:	1. Supervisor must be logged into the system. 2. Staff record must exist in the database.					
Post-conditions:	The attendance of the staff for the month will be marked as verified and sent to the human relations department for salary crediting.					
Flow of events	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include View staff details.  2. Include View attendance of staff.  3. Click “Payroll” button</td><td>3.1 System prompt user to enter password of the currently logged in account for verification.</td></tr></table>		Supervisor	System	1. Include View staff details.  2. Include View attendance of staff.  3. Click “Payroll” button	3.1 System prompt user to enter password of the currently logged in account for verification.
Supervisor	System					
1. Include View staff details.  2. Include View attendance of staff.  3. Click “Payroll” button	3.1 System prompt user to enter password of the currently logged in account for verification.					



	<p>4. Enter password.</p> <p>5. Verify that the attendance and the salary is accurate.</p> <p>6. Click “Verify payroll” button</p> <p>7. Click “Confirm” button to verify the payroll.</p>	<p>4.1 Verify password</p> <p>4.2 If verification is unsuccessful, display error message “Incorrect password” and go back to step 3.1.</p> <p>4.2a. If verification is successful, display the attendance in calendar view and a column of total number of working hours and salary for the particular month.</p> <p>6.1 Prompt with a message to confirm the operation.</p> <p>7.1 Update the status of the calendar for the particular month as verified and disable all updating features for that particular month.</p> <p>7.2 Send the verified information to the human relations department.</p> <p>7.3 Update the salary credit status from default “Pending verification” to “Processing”.</p>
Exception conditions:	Nil	

Use case:	Check salary credit status for staff					
Triggering case:	Staff enquiry about the crediting of salary					
Brief description:	Supervisor check the salary credit status for staff					
Actors:	Supervisor					
Pre-conditions:	1. Supervisor must be logged into the system. 2. Staff record must exist in the database. 3. The attendance for the selected staff must have been verified.					
Post-conditions:	The salary credit status of the selected staff will be displayed.					
Flow of events	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include View staff details.  2. Include View attendance of staff.  3. Click “Payroll” button  4. Enter password.    5. Click on “Salary credit status” button</td><td>3.1 System prompt user to enter password of the currently logged in account for verification.  4.1 Verify password  4.2 If verification is unsuccessful, display error message “Incorrect password” and go back to step 3.1.  4.2a. If verification is successful, display the attendance in calendar view and a column of verified total number of working hours and salary for the particular month.  5.1 Retrieve information of salary credit status from database and display.</td></tr></table>		Supervisor	System	1. Include View staff details.  2. Include View attendance of staff.  3. Click “Payroll” button  4. Enter password.    5. Click on “Salary credit status” button	3.1 System prompt user to enter password of the currently logged in account for verification.  4.1 Verify password  4.2 If verification is unsuccessful, display error message “Incorrect password” and go back to step 3.1.  4.2a. If verification is successful, display the attendance in calendar view and a column of verified total number of working hours and salary for the particular month.  5.1 Retrieve information of salary credit status from database and display.
Supervisor	System					
1. Include View staff details.  2. Include View attendance of staff.  3. Click “Payroll” button  4. Enter password.    5. Click on “Salary credit status” button	3.1 System prompt user to enter password of the currently logged in account for verification.  4.1 Verify password  4.2 If verification is unsuccessful, display error message “Incorrect password” and go back to step 3.1.  4.2a. If verification is successful, display the attendance in calendar view and a column of verified total number of working hours and salary for the particular month.  5.1 Retrieve information of salary credit status from database and display.					
Exception conditions:	Nil					

Simplified entity class diagram

*Refer to attachment 3A*

Non-entity class diagram

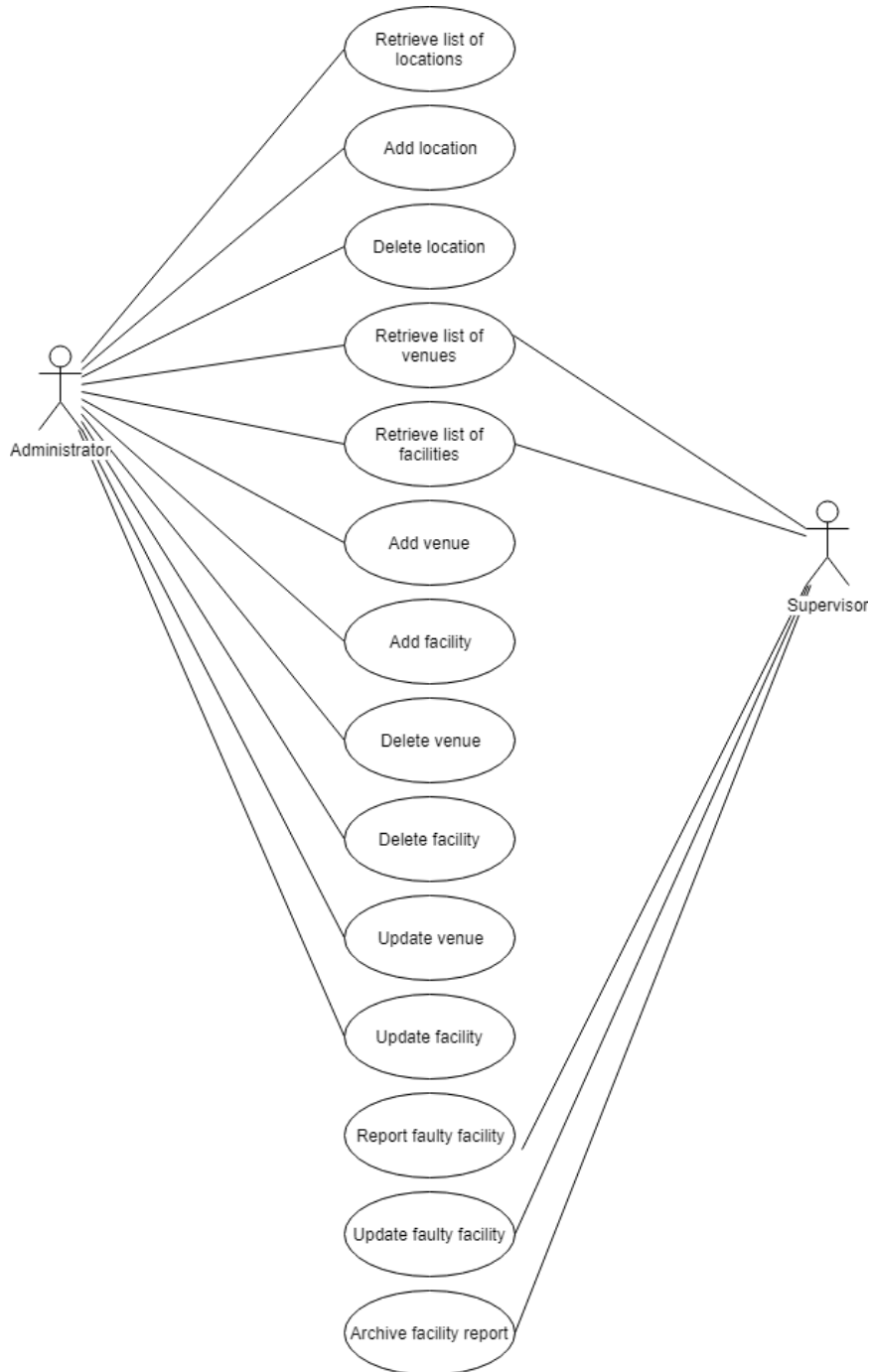
*Refer to attachment 3B*

Sequence diagrams

*Refer to attachment 3C, 3D and 3E*

# Area Management Subsystem

Use case diagram



*Use case diagram for area management subsystem*

## Use case descriptions for area management module use cases

Use case:	Retrieve list of locations					
Triggering case:	Administrator wants to retrieve the list of locations					
Brief description:	Administrator access system to retrieve the list of locations					
Actors:	Administrator					
Pre-conditions:	1. Administrator must be logged in to the system					
Post-conditions:	A list of existing locations will be displayed from the system's database					
Flow of events:	<table><tr><th>Administrator</th><th>System</th></tr><tr><td>1. Click on "manage locations" on logged in page.</td><td>1.1 Retrieves the list of existing locations stored in the system's database.  1.2 Displays manage locations page with the list of existing locations</td></tr></table>		Administrator	System	1. Click on "manage locations" on logged in page.	1.1 Retrieves the list of existing locations stored in the system's database.  1.2 Displays manage locations page with the list of existing locations
Administrator	System					
1. Click on "manage locations" on logged in page.	1.1 Retrieves the list of existing locations stored in the system's database.  1.2 Displays manage locations page with the list of existing locations					
Exception conditions:	Nil					

Use case:	Add location									
Triggering case:	SCS obtains a new cleaning services contract at a new location									
Brief description:	Administrator access system to add the new location									
Actors:	Administrator									
Pre-conditions:	1. Administrator must be logged in to the system 2. Location must be new									
Post-conditions:	Adds the new location to the list of existing locations in the system’s database									
Flow of events:	<table><tr><th>Administrator</th><th>System</th></tr><tr><td>1. Include retrieve list of locations</td><td></td></tr><tr><td>2. Click on “add location”</td><td>2.1 Displays add location page and a form with empty fields (name of building, building management email, and address)</td></tr><tr><td>3. Enters the name of building, building management email, and address</td><td>3.1 Verify that there is no existing record with the fields entered  3.2 If verification is successful, create new location record in the location database. Displays message “successfully added new location”.  3.2a. If verification fails, displays error message “The <i>information</i>* already exist”. Go to step 2.1.</td></tr></table> <p><i>*The word information is to be substituted with any of one the details that are being captured in the system</i></p>		Administrator	System	1. Include retrieve list of locations		2. Click on “add location”	2.1 Displays add location page and a form with empty fields (name of building, building management email, and address)	3. Enters the name of building, building management email, and address	3.1 Verify that there is no existing record with the fields entered  3.2 If verification is successful, create new location record in the location database. Displays message “successfully added new location”.  3.2a. If verification fails, displays error message “The <i>information</i> * already exist”. Go to step 2.1.
Administrator	System									
1. Include retrieve list of locations										
2. Click on “add location”	2.1 Displays add location page and a form with empty fields (name of building, building management email, and address)									
3. Enters the name of building, building management email, and address	3.1 Verify that there is no existing record with the fields entered  3.2 If verification is successful, create new location record in the location database. Displays message “successfully added new location”.  3.2a. If verification fails, displays error message “The <i>information</i> * already exist”. Go to step 2.1.									
Exception conditions:	Nil									

Use case:	Delete location	
Triggering case:	Existing cleaning services contract expired	
Brief description:	Administrator access system to delete location	
Actors:	Administrator	
Pre-conditions:	<ol style="list-style-type: none"> <li>1. Administrator must be logged in to the system</li> <li>2. Location must exist in the system</li> </ol>	
Post-conditions:	Remove the location from the list of existing locations in the system's database	
Flow of events:		
	Administrator	System
	1. Include retrieve list of locations	
	2. Click on the delete button next to the location	<p>2.1 Displays 2-step confirmation message stating "You are deleting the "specified location" along with all the venues under this location, do you want to proceed?"</p> <p>2.2 If administrator clicked yes, system removes the location from the database. Displays message "successfully deleted location".</p> <p>2.2a. If administrator clicked no, system navigates back to the manage locations page. Go to step 1.1.</p>
Exception conditions:	Nil	

Use case:	Retrieve list of venues							
Triggering case:	Administrator wants to retrieve the list of venues							
Brief description:	Administrator access system to retrieve the list of venues							
Actors:	Administrator, Supervisor							
Pre-conditions:	1. Administrator/Supervisor must be logged in to the system							
Post-conditions:	A list of existing venues will be displayed from the system's database							
Flow of events:	<table><tr><th>Administrator, Supervisor</th><th>System</th></tr><tr><td>1. Include retrieve list of locations</td><td></td></tr><tr><td>2. Click on the particular location to retrieve a list of venues</td><td>2.1 Retrieves the list of venues under the particular location  2.2 Displays the location details page with a list of existing venues</td></tr></table>		Administrator, Supervisor	System	1. Include retrieve list of locations		2. Click on the particular location to retrieve a list of venues	2.1 Retrieves the list of venues under the particular location  2.2 Displays the location details page with a list of existing venues
Administrator, Supervisor	System							
1. Include retrieve list of locations								
2. Click on the particular location to retrieve a list of venues	2.1 Retrieves the list of venues under the particular location  2.2 Displays the location details page with a list of existing venues							
Exception conditions:	Nil							



Use case:	Retrieve list of facilities									
Triggering case:	Administrator wants to retrieve the list of facilities									
Brief description:	Administrator access system to retrieve the list of facilities									
Actors:	Administrator, Supervisor									
Pre-conditions:	2. Administrator/Supervisor must be logged in to the system									
Post-conditions:	A list of existing facilities will be displayed from the system's database									
Flow of events:	<table><tr><th>Administrator, Supervisor</th><th>System</th></tr><tr><td>1. Include retrieve list of locations</td><td></td></tr><tr><td>2. Click on the particular location to retrieve a list of venues</td><td>2.1 Retrieves the list of venues under the particular location  2.2 Displays the location details page with a list of existing venues</td></tr><tr><td>3. Click on the particular venue to retrieve a list of facilities</td><td>3.1 Retrieves the list of facilities under the particular venue  3.2 Displays the venue details page with a list of existing facilities</td></tr></table>		Administrator, Supervisor	System	1. Include retrieve list of locations		2. Click on the particular location to retrieve a list of venues	2.1 Retrieves the list of venues under the particular location  2.2 Displays the location details page with a list of existing venues	3. Click on the particular venue to retrieve a list of facilities	3.1 Retrieves the list of facilities under the particular venue  3.2 Displays the venue details page with a list of existing facilities
Administrator, Supervisor	System									
1. Include retrieve list of locations										
2. Click on the particular location to retrieve a list of venues	2.1 Retrieves the list of venues under the particular location  2.2 Displays the location details page with a list of existing venues									
3. Click on the particular venue to retrieve a list of facilities	3.1 Retrieves the list of facilities under the particular venue  3.2 Displays the venue details page with a list of existing facilities									
Exception conditions:	Nil									

Use case:	Add venue									
Triggering case:	SCS obtains a new cleaning services contract at a new location with specified venue(s)									
Brief description:	Administrator access system to add the venue(s)									
Actors:	Administrator									
Pre-conditions:	1. Administrator must be logged in to the system 2. Location must exist in the system 3. Venue(s) must be new									
Post-conditions:	Add the list of venues under an existing location record in the system's database									
Flow of events:	<table><tr><th>Administrator</th><th>System</th></tr><tr><td>1. Include retrieve list of venues</td><td></td></tr><tr><td>2. Click on “add venues” on the location details page</td><td>2.1 Displays add venues page and a form with empty fields [name of venue, level, unit number (if applicable), gender requirement, type of venue, and number of staff required.</td></tr><tr><td>3. Enters the name of the venue, level, unit number (if applicable), gender requirement, type of venue, and the number of staff required</td><td>3.1 Verify that there is no existing record with the fields entered  3.2 If verification is successful, create venue record under the existing location record in the database. Displays message “successfully added venues”.  3.2a. If verification fails, displays error message “The <i>information</i>* already exist”. Go to step 2.1.</td></tr></table> <p><i>*The word information is to be substituted with any of one the details that are being captured in the system</i></p>		Administrator	System	1. Include retrieve list of venues		2. Click on “add venues” on the location details page	2.1 Displays add venues page and a form with empty fields [name of venue, level, unit number (if applicable), gender requirement, type of venue, and number of staff required.	3. Enters the name of the venue, level, unit number (if applicable), gender requirement, type of venue, and the number of staff required	3.1 Verify that there is no existing record with the fields entered  3.2 If verification is successful, create venue record under the existing location record in the database. Displays message “successfully added venues”.  3.2a. If verification fails, displays error message “The <i>information</i> * already exist”. Go to step 2.1.
Administrator	System									
1. Include retrieve list of venues										
2. Click on “add venues” on the location details page	2.1 Displays add venues page and a form with empty fields [name of venue, level, unit number (if applicable), gender requirement, type of venue, and number of staff required.									
3. Enters the name of the venue, level, unit number (if applicable), gender requirement, type of venue, and the number of staff required	3.1 Verify that there is no existing record with the fields entered  3.2 If verification is successful, create venue record under the existing location record in the database. Displays message “successfully added venues”.  3.2a. If verification fails, displays error message “The <i>information</i> * already exist”. Go to step 2.1.									
Exception conditions:	Nil									

Use case:	Add facility									
Triggering case:	SCS obtains a new cleaning services contract at a new location with specified venues and the facilities under each venue									
Brief description:	Administrator access system to add facilities under the venue									
Actors:	Administrator									
Pre-conditions:	1. Administrator must be logged in to the system 2. Location and venue must exist in the system 3. Facility must be new									
Post-conditions:	Add the list of facilities under an existing venue record in the system's database									
Flow of events:	<table><tr><th>Administrator</th><th>System</th></tr><tr><td>1. Include retrieve list of facilities</td><td></td></tr><tr><td>2. Click on "add facilities" on the venue details page</td><td>2.1 Displays add facilities page and a form with empty fields [name of facility]</td></tr><tr><td>3. Enters the name of the facilities</td><td>3.1 Verify that there is no existing record with the fields entered  3.2 If verification is successful, create facility record under the existing venue record in the database. Displays message "successfully added facilities".  3.2a. If verification fails, displays error message "The <i>information</i>* already exist". Go to step 2.1.</td></tr></table> <p><i>*The word information is to be substituted with any of one the details that are being captured in the system</i></p>		Administrator	System	1. Include retrieve list of facilities		2. Click on "add facilities" on the venue details page	2.1 Displays add facilities page and a form with empty fields [name of facility]	3. Enters the name of the facilities	3.1 Verify that there is no existing record with the fields entered  3.2 If verification is successful, create facility record under the existing venue record in the database. Displays message "successfully added facilities".  3.2a. If verification fails, displays error message "The <i>information</i> * already exist". Go to step 2.1.
Administrator	System									
1. Include retrieve list of facilities										
2. Click on "add facilities" on the venue details page	2.1 Displays add facilities page and a form with empty fields [name of facility]									
3. Enters the name of the facilities	3.1 Verify that there is no existing record with the fields entered  3.2 If verification is successful, create facility record under the existing venue record in the database. Displays message "successfully added facilities".  3.2a. If verification fails, displays error message "The <i>information</i> * already exist". Go to step 2.1.									
Exception conditions:	Nil									

Use case:	Delete venue							
Triggering case:	Venue no longer exist or has become invalid							
Brief description:	Remove the venue from the list of existing venues in the system’s database							
Actors:	Administrator							
Pre-conditions:	1. Administrator must be logged in to the system 2. Venue must exist in the system							
Post-conditions:	Remove the venue from the list of existing venues in the system’s database along with the facilities that are associated with the venue							
Flow of events:	<table><tr><th>Administrator</th><th>System</th></tr><tr><td>1. Include retrieve list of venues</td><td></td></tr><tr><td>2. Click on the “delete” button next to the venue</td><td>2.1 Displays 2-step confirmation message stating “You are deleting the “specified venue” along with all the facilities under this venue, do you want to proceed?”  2.2 If administrator clicked yes, system removes the venue from the database. Displays message “successfully deleted venue”.  2.2a. If administrator clicked no, system navigates back to the location details page. Go to step 2.2.</td></tr></table>		Administrator	System	1. Include retrieve list of venues		2. Click on the “delete” button next to the venue	2.1 Displays 2-step confirmation message stating “You are deleting the “specified venue” along with all the facilities under this venue, do you want to proceed?”  2.2 If administrator clicked yes, system removes the venue from the database. Displays message “successfully deleted venue”.  2.2a. If administrator clicked no, system navigates back to the location details page. Go to step 2.2.
Administrator	System							
1. Include retrieve list of venues								
2. Click on the “delete” button next to the venue	2.1 Displays 2-step confirmation message stating “You are deleting the “specified venue” along with all the facilities under this venue, do you want to proceed?”  2.2 If administrator clicked yes, system removes the venue from the database. Displays message “successfully deleted venue”.  2.2a. If administrator clicked no, system navigates back to the location details page. Go to step 2.2.							
Exception conditions:	Nil							

Use case:	Delete facility							
Triggering case:	Facility no longer exist or has become invalid							
Brief description:	Remove the facility from the list of existing facilities in the system's database							
Actors:	Administrator							
Pre-conditions:	3. Administrator must be logged in to the system 4. Facility must exist in the system							
Post-conditions:	Remove the facility from the list of existing facilities in the system's database							
Flow of events:	<table><tr><th>Administrator</th><th>System</th></tr><tr><td>1. Include retrieve list of facilities</td><td></td></tr><tr><td>2. Click on the "delete" button next to the facility</td><td>2.1 Displays 2-step confirmation message stating "You are deleting <i>"name of facility"</i>, do you want to proceed?"  2.2 If administrator clicked yes, system removes the facility from the database. Displays message "successfully deleted facility".  2.2a. If administrator clicked no, system navigates back to the venue details page. Go to step 3.2.</td></tr></table>		Administrator	System	1. Include retrieve list of facilities		2. Click on the "delete" button next to the facility	2.1 Displays 2-step confirmation message stating "You are deleting <i>"name of facility"</i> , do you want to proceed?"  2.2 If administrator clicked yes, system removes the facility from the database. Displays message "successfully deleted facility".  2.2a. If administrator clicked no, system navigates back to the venue details page. Go to step 3.2.
Administrator	System							
1. Include retrieve list of facilities								
2. Click on the "delete" button next to the facility	2.1 Displays 2-step confirmation message stating "You are deleting <i>"name of facility"</i> , do you want to proceed?"  2.2 If administrator clicked yes, system removes the facility from the database. Displays message "successfully deleted facility".  2.2a. If administrator clicked no, system navigates back to the venue details page. Go to step 3.2.							
Exception conditions:	Nil							

Use case:	Update venue									
Triggering case:	Changes made to existing venue									
Brief description:	Administrator access system to update the venue									
Actors:	Administrator									
Pre-conditions:	1. Administrator must be logged in to the system 2. Venue must exist in the system									
Post-conditions:	Update the venue record in the system's database									
Flow of events:	<table><tr><th>Administrator</th><th>System</th></tr><tr><td>1. Include retrieve list of venues</td><td></td></tr><tr><td>2. Click on the update button next to the particular venue</td><td>2.1 Displays a form with the existing venue records from the database for editing</td></tr><tr><td>3. Enters the updated information for the venue</td><td>3.1 Verify that there is no existing record with the fields entered  3.2 If verification is successful, updates venue record under the existing location record in the database. Displays message "successfully updated venue".  3.2a. If verification fails, displays error message "The <i>information</i>* already exist". Go to step 3.1.</td></tr></table> <p><i>*The word information is to be substituted with any of one the details that are being captured in the system</i></p>		Administrator	System	1. Include retrieve list of venues		2. Click on the update button next to the particular venue	2.1 Displays a form with the existing venue records from the database for editing	3. Enters the updated information for the venue	3.1 Verify that there is no existing record with the fields entered  3.2 If verification is successful, updates venue record under the existing location record in the database. Displays message "successfully updated venue".  3.2a. If verification fails, displays error message "The <i>information</i> * already exist". Go to step 3.1.
Administrator	System									
1. Include retrieve list of venues										
2. Click on the update button next to the particular venue	2.1 Displays a form with the existing venue records from the database for editing									
3. Enters the updated information for the venue	3.1 Verify that there is no existing record with the fields entered  3.2 If verification is successful, updates venue record under the existing location record in the database. Displays message "successfully updated venue".  3.2a. If verification fails, displays error message "The <i>information</i> * already exist". Go to step 3.1.									
Exception conditions:	Nil									

Use case:	Update facility									
Triggering case:	Changes made to existing facility									
Brief description:	Administrator access system to update the facility									
Actors:	Administrator									
Pre-conditions:	1. Administrator must be logged in to the system 2. Facility must exist in the system									
Post-conditions:	Update the facility record in the system's database									
Flow of events:	<table><tr><th>Administrator</th><th>System</th></tr><tr><td>1. Include retrieve list of facilities</td><td></td></tr><tr><td>2. Click on the update button next to the particular facility</td><td>2.1 Displays a form with the existing facility record from the database for editing</td></tr><tr><td>3. Enters the updated information for the facility</td><td>3.1 Verify that there is no existing record with the fields entered  3.2 If verification is successful, updates facility record under the existing venue record in the database. Displays message "successfully updated facility".  3.2a. If verification fails, displays error message "The <i>information</i>* already exist". Go to step 4.1.</td></tr></table> <p><i>*The word information is to be substituted with any of one the details that are being captured in the system</i></p>		Administrator	System	1. Include retrieve list of facilities		2. Click on the update button next to the particular facility	2.1 Displays a form with the existing facility record from the database for editing	3. Enters the updated information for the facility	3.1 Verify that there is no existing record with the fields entered  3.2 If verification is successful, updates facility record under the existing venue record in the database. Displays message "successfully updated facility".  3.2a. If verification fails, displays error message "The <i>information</i> * already exist". Go to step 4.1.
Administrator	System									
1. Include retrieve list of facilities										
2. Click on the update button next to the particular facility	2.1 Displays a form with the existing facility record from the database for editing									
3. Enters the updated information for the facility	3.1 Verify that there is no existing record with the fields entered  3.2 If verification is successful, updates facility record under the existing venue record in the database. Displays message "successfully updated facility".  3.2a. If verification fails, displays error message "The <i>information</i> * already exist". Go to step 4.1.									
Exception conditions:	Nil									

Use case:	Report faulty facility											
Triggering case:	Supervisor wants to report faulty facility											
Brief description:	Supervisor access the system to report on faulty facility											
Actors:	Supervisor											
Pre-conditions:	1. Supervisor must be logged in to the system 2. Facility must exist in the system's database											
Post-conditions:	Sends repair request to the building's management											
Flow of events:	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include retrieve list of facilities</td><td></td></tr><tr><td>2. Click on the particular facility to report</td><td>2.1 Retrieves the report under the facility  2.2 Displays the facility report page with the report under the facility</td></tr><tr><td>3. If there is no existing report for the facility, click on the "add new report" button</td><td>3.1 Displays report faulty facility page and a form with empty fields [report date, title, description]</td></tr><tr><td>4. Enters the report date, title and description</td><td>4.1 Creates a new report record under the facility record in the database.  4.2 Email the repair request to the respective building's management.  4.3 Displays message "successfully created report for the facility, and email sent to the building's management".</td></tr></table>		Supervisor	System	1. Include retrieve list of facilities		2. Click on the particular facility to report	2.1 Retrieves the report under the facility  2.2 Displays the facility report page with the report under the facility	3. If there is no existing report for the facility, click on the "add new report" button	3.1 Displays report faulty facility page and a form with empty fields [report date, title, description]	4. Enters the report date, title and description	4.1 Creates a new report record under the facility record in the database.  4.2 Email the repair request to the respective building's management.  4.3 Displays message "successfully created report for the facility, and email sent to the building's management".
Supervisor	System											
1. Include retrieve list of facilities												
2. Click on the particular facility to report	2.1 Retrieves the report under the facility  2.2 Displays the facility report page with the report under the facility											
3. If there is no existing report for the facility, click on the "add new report" button	3.1 Displays report faulty facility page and a form with empty fields [report date, title, description]											
4. Enters the report date, title and description	4.1 Creates a new report record under the facility record in the database.  4.2 Email the repair request to the respective building's management.  4.3 Displays message "successfully created report for the facility, and email sent to the building's management".											
Exception conditions:	Nil											



Use case:	Update faulty facility											
Triggering case:	Supervisor wants to update faulty facility											
Brief description:	Supervisor access the system to update faulty facility											
Actors:	Supervisor											
Pre-conditions:	1. Supervisor must be logged in to the system 2. Facility must exist in the system's database 3. Report must exist in the system's database											
Post-conditions:	Update the report record in the system's database											
Flow of events:	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include retrieve list of facilities</td><td></td></tr><tr><td>2. Click on the particular facility to update</td><td>2.1 Retrieves the report under the facility  2.2 Displays the facility report page with the report under the facility</td></tr><tr><td>3. If there is an existing report for the facility, click on update button next to the report</td><td>3.1 Displays the form with pre-filled fields from the existing report record (title, description for the report)</td></tr><tr><td>4. Updates the relevant fields and click ok</td><td>4.1 Updates report record under the facility record in the database  4.2 Displays prompt message to the user: "Do you wish to email the updated report to the building's management?"  4.2a If the user selects yes, email the updated repair request to the respective building's management</td></tr></table>		Supervisor	System	1. Include retrieve list of facilities		2. Click on the particular facility to update	2.1 Retrieves the report under the facility  2.2 Displays the facility report page with the report under the facility	3. If there is an existing report for the facility, click on update button next to the report	3.1 Displays the form with pre-filled fields from the existing report record (title, description for the report)	4. Updates the relevant fields and click ok	4.1 Updates report record under the facility record in the database  4.2 Displays prompt message to the user: "Do you wish to email the updated report to the building's management?"  4.2a If the user selects yes, email the updated repair request to the respective building's management
Supervisor	System											
1. Include retrieve list of facilities												
2. Click on the particular facility to update	2.1 Retrieves the report under the facility  2.2 Displays the facility report page with the report under the facility											
3. If there is an existing report for the facility, click on update button next to the report	3.1 Displays the form with pre-filled fields from the existing report record (title, description for the report)											
4. Updates the relevant fields and click ok	4.1 Updates report record under the facility record in the database  4.2 Displays prompt message to the user: "Do you wish to email the updated report to the building's management?"  4.2a If the user selects yes, email the updated repair request to the respective building's management											
Exception conditions:	Nil											

Use case:	Archive facility's report									
Triggering case:	Supervisor wants to archive faulty facility report									
Brief description:	Supervisor access the system to archive faulty facility report									
Actors:	Supervisor									
Pre-conditions:	1. Supervisor must be logged in to the system 2. Facility must exist in the system's database 3. Report must exist in the system's database									
Post-conditions:	Archive the report record in the system's database									
Flow of events:	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include retrieve list of facilities</td><td></td></tr><tr><td>2. Click on the particular facility to archive the facility's report</td><td>2.1 Retrieves the report under the facility  2.2 Displays the facility report page with the report under the facility</td></tr><tr><td>3. If there is an existing report for the facility, click on archive button next to the report</td><td>3.1 Archive the selected report record to the list of archived reports in the database.  3.2 Displays message "successfully archived report"</td></tr></table>		Supervisor	System	1. Include retrieve list of facilities		2. Click on the particular facility to archive the facility's report	2.1 Retrieves the report under the facility  2.2 Displays the facility report page with the report under the facility	3. If there is an existing report for the facility, click on archive button next to the report	3.1 Archive the selected report record to the list of archived reports in the database.  3.2 Displays message "successfully archived report"
Supervisor	System									
1. Include retrieve list of facilities										
2. Click on the particular facility to archive the facility's report	2.1 Retrieves the report under the facility  2.2 Displays the facility report page with the report under the facility									
3. If there is an existing report for the facility, click on archive button next to the report	3.1 Archive the selected report record to the list of archived reports in the database.  3.2 Displays message "successfully archived report"									
Exception conditions:	Nil									

Use case:	Retrieve cleaning staff's available working days									
Triggering case:	Supervisor wants to retrieve cleaning staff's available working days									
Brief description:	Supervisor access system to retrieve cleaning staff's available working days									
Actors:	Supervisor									
Pre-conditions:	<div>1. Supervisor must be logged in to the system</div> <div>2. Offs, Leaves and Medical Certificates have been approved and recorded in the system</div>									
Post-conditions:	Displays the list of cleaning staff's available working days									
Flow of events:	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include retrieve list of venues</td><td></td></tr><tr><td>2. Click on "cleaning schedule management"</td><td><div>2.1 Retrieves the list of cleaning staff assigned to the location</div><div>2.2 Displays cleaning schedule management page with the list of cleaning staff assigned to the location</div></td></tr><tr><td>3. Click on retrieve cleaning staff's available working days</td><td><div>3.1 Retrieves all of the staff's approved off, leaves and MC</div><div>3.2 Displays a variety of cleaning staff's available working days for the next month in a calendar table view (grey out working days which the staff has applied for off, leaves, and MC)</div></td></tr></table>		Supervisor	System	1. Include retrieve list of venues		2. Click on "cleaning schedule management"	<div>2.1 Retrieves the list of cleaning staff assigned to the location</div> <div>2.2 Displays cleaning schedule management page with the list of cleaning staff assigned to the location</div>	3. Click on retrieve cleaning staff's available working days	<div>3.1 Retrieves all of the staff's approved off, leaves and MC</div> <div>3.2 Displays a variety of cleaning staff's available working days for the next month in a calendar table view (grey out working days which the staff has applied for off, leaves, and MC)</div>
Supervisor	System									
1. Include retrieve list of venues										
2. Click on "cleaning schedule management"	<div>2.1 Retrieves the list of cleaning staff assigned to the location</div> <div>2.2 Displays cleaning schedule management page with the list of cleaning staff assigned to the location</div>									
3. Click on retrieve cleaning staff's available working days	<div>3.1 Retrieves all of the staff's approved off, leaves and MC</div> <div>3.2 Displays a variety of cleaning staff's available working days for the next month in a calendar table view (grey out working days which the staff has applied for off, leaves, and MC)</div>									
Exception conditions:	Nil									

Use case:	Retrieve rostering									
Triggering case:	Supervisor wants to retrieve the rostering									
Brief description:	Supervisor access system to retrieve the rostering									
Actors:	Supervisor									
Pre-conditions:	1. Supervisor must be logged in to the system 2. Roster must exist in the system									
Post-conditions:	Displays the rostering from the system									
Flow of events:	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include retrieve list of venues</td><td></td></tr><tr><td>2. Click on “cleaning schedule management”</td><td>2.1 Retrieves the list of cleaning staff assigned to the venue  2.2 Displays cleaning schedule management page with the list of cleaning staff assigned to the venue</td></tr><tr><td>3. Click on retrieve rostering</td><td>3.1 Retrieves rostering record from the database  3.2 Displays existing rostering record</td></tr></table>		Supervisor	System	1. Include retrieve list of venues		2. Click on “cleaning schedule management”	2.1 Retrieves the list of cleaning staff assigned to the venue  2.2 Displays cleaning schedule management page with the list of cleaning staff assigned to the venue	3. Click on retrieve rostering	3.1 Retrieves rostering record from the database  3.2 Displays existing rostering record
Supervisor	System									
1. Include retrieve list of venues										
2. Click on “cleaning schedule management”	2.1 Retrieves the list of cleaning staff assigned to the venue  2.2 Displays cleaning schedule management page with the list of cleaning staff assigned to the venue									
3. Click on retrieve rostering	3.1 Retrieves rostering record from the database  3.2 Displays existing rostering record									
Exception conditions:	Nil									

Use case:	Automate rostering											
Triggering case:	Supervisor wants to automate the rostering											
Brief description:	Supervisor access system to automate the rostering											
Actors:	Supervisor											
Pre-conditions:	1. Supervisor must be logged in to the system 2. Cleaning staff have indicated their available working days in the system 3. Cleaning staff details must exist in the system											
Post-conditions:	Create a new roster record in the system database											
Flow of events:	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include retrieve list of venues</td><td></td></tr><tr><td>2. Click on “cleaning schedule management”</td><td>2.1 Retrieves the list of cleaning staff assigned to the location  2.2 Displays cleaning schedule management page with the list of cleaning staff assigned to the location</td></tr><tr><td>3. Click on “add rostering” button</td><td>3.1 Retrieves list of venues records with gender specific requirement  3.2 Retrieves existing roster for the location  3.3 Displays add rostering page with the list of venues</td></tr><tr><td>4. If there is no existing roster, click “automate rostering”</td><td>4.1 Retrieves cleaning staff records under the location  4.2 Retrieves the gender requirement for the venue</td></tr></table>		Supervisor	System	1. Include retrieve list of venues		2. Click on “cleaning schedule management”	2.1 Retrieves the list of cleaning staff assigned to the location  2.2 Displays cleaning schedule management page with the list of cleaning staff assigned to the location	3. Click on “add rostering” button	3.1 Retrieves list of venues records with gender specific requirement  3.2 Retrieves existing roster for the location  3.3 Displays add rostering page with the list of venues	4. If there is no existing roster, click “automate rostering”	4.1 Retrieves cleaning staff records under the location  4.2 Retrieves the gender requirement for the venue
Supervisor	System											
1. Include retrieve list of venues												
2. Click on “cleaning schedule management”	2.1 Retrieves the list of cleaning staff assigned to the location  2.2 Displays cleaning schedule management page with the list of cleaning staff assigned to the location											
3. Click on “add rostering” button	3.1 Retrieves list of venues records with gender specific requirement  3.2 Retrieves existing roster for the location  3.3 Displays add rostering page with the list of venues											
4. If there is no existing roster, click “automate rostering”	4.1 Retrieves cleaning staff records under the location  4.2 Retrieves the gender requirement for the venue											

	<p>4.3 Filter the cleaning staff records according to the gender requirement</p> <p>4.4 Retrieves cleaning staff's addresses and categorize them into the individual town councils (West Coast TC, Chua Chu Kang TC, Jurong-Clementi TC, Marsiling-Yew Tee TC, Holland-Bukit Panjang TC, Tanjong Pagar TC, Bishan-Toa Payoh TC, Nee Soon TC, Sembawang TC, Ang Mo Kio TC, Pasir Ris-Punggol TC, Aljunied-Hougang TC, Jalan Besar TC, Marine Parade TC, Tampines TC, East Coast-Fengshan TC). (Refer to figure 1.0)</p> <p>4.5 Based on the adjacency of the cleaning staff's town council away from the town council which oversees the designated building, assign a score to the cleaning staff. (max score*: 11, the closer the cleaning staff's town council the higher the score)</p> <p>4.6 Retrieves cleaning staff's number of years of past experiences for that location and assign a score accordingly*</p> <p>4.7 Accumulate the scores for each cleaning staff</p> <p>4.8 Retrieves cleaning staff's available working days for the next month</p> <p>4.9 Retrieves the number of staff required for the venue</p> <p>4.10. Sorts the cleaning staff according to their accumulated score (from the highest score to the lowest in descending order)</p>
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<p>4.11 Randomly assigns the cleaning staff to the time slots according to the cleaning staff's available working day until the number of required staff has been fulfilled.</p> <p>4.12 If all the working days and time slots are occupied, displays message "successfully filled up the rostering"</p> <p>4.12a If there are time slots still not occupied, displays message "System was unable to fully fill the rostering", highlight the working days and time slots that are still vacant.</p> <p>* Supervisor will be able to adjust the allocation of scores to place more priority on either distance away or number of years of experience</p>
Exception conditions:	Nil



Figure 1.0 Singapore Town Councils

Use case:	Archive roster							
Triggering case:	Supervisor wants to archive roster							
Brief description:	Supervisor access the system to archive roster							
Actors:	Supervisor							
Pre-conditions:	1. Supervisor must be logged in to the system 2. Roster must exist in the system’s database							
Post-conditions:	Archive the roster record in the system’s database							
Flow of events:	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include retrieve rostering</td><td></td></tr><tr><td>2. Click on archive button next to the roster</td><td>2.1 Archive the selected roster record to the list of archived rosters in the database.  2.2 Displays message “successfully archived roster”</td></tr></table>		Supervisor	System	1. Include retrieve rostering		2. Click on archive button next to the roster	2.1 Archive the selected roster record to the list of archived rosters in the database.  2.2 Displays message “successfully archived roster”
Supervisor	System							
1. Include retrieve rostering								
2. Click on archive button next to the roster	2.1 Archive the selected roster record to the list of archived rosters in the database.  2.2 Displays message “successfully archived roster”							
Exception conditions:	Nil							



Use case:	Manual update to roster									
Triggering case:	Supervisor wants to manually update roster									
Brief description:	Supervisor access system to manually update roster									
Actors:	Supervisor									
Pre-conditions:	1. Supervisor must be logged in to the system 2. Roster must exist in the system									
Post-conditions:	Update roster record in the system's database									
Flow of events:	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include retrieve rostering</td><td></td></tr><tr><td>2. Click on "manual update" button next to the roster</td><td>2.1 Displays roster pre-filled with the existing roster record from the database.</td></tr><tr><td>3. Manually update the roster by either shifting the employees' field (drag and drop), adding new employee field, deleting existing employee field in the calendar view</td><td>3.1 Updates roster record in the database  3.2 Displays message "successfully updated roster"</td></tr></table>		Supervisor	System	1. Include retrieve rostering		2. Click on "manual update" button next to the roster	2.1 Displays roster pre-filled with the existing roster record from the database.	3. Manually update the roster by either shifting the employees' field (drag and drop), adding new employee field, deleting existing employee field in the calendar view	3.1 Updates roster record in the database  3.2 Displays message "successfully updated roster"
Supervisor	System									
1. Include retrieve rostering										
2. Click on "manual update" button next to the roster	2.1 Displays roster pre-filled with the existing roster record from the database.									
3. Manually update the roster by either shifting the employees' field (drag and drop), adding new employee field, deleting existing employee field in the calendar view	3.1 Updates roster record in the database  3.2 Displays message "successfully updated roster"									
Exception conditions:	Nil									

Use case:	Send cleaning requests
Triggering case:	Supervisor sends cleaning request to cleaning staff
Brief description:	Supervisor access system to send cleaning request to cleaning staff
Actors:	Supervisor
Pre-conditions:	<ol style="list-style-type: none"><li>1. Supervisor must be logged in to the system</li><li>2. Cleaning staff details must exist in the system</li></ol>
Post-conditions:	Sends SMS messages to the cleaning staff regarding the cleaning request

Flow of events:		
	Supervisor	System
	1. Include retrieve rostering	
	2. Click on “message” button	2.1 Retrieve message template and displays messaging page
	3. If the supervisor wishes to make changes, click on modify message  3a. If the supervisor does not wish to make changes, go to step 5	3.1 Displays a form pre-populated with the existing message template
	4. Edits the message template, and click save	4.1 Create new message template record to the database
	5. Click on “send cleaning requests” button	5.1 Retrieves all staff records assigned to the location  5.2 Displays prompt message to the user: “You are about to send cleaning requests to the cleaning staff, do you wish to proceed?”
	6. Click on “confirm and send”	6.1 Sends pre-populated SMS messages (cleaning requests) about the current rostering to the respective cleaning staff’s contact number
Exception conditions:	Nil	

Simplified entity class diagram

*Refer to attachment 4A*

Non-entity class diagram

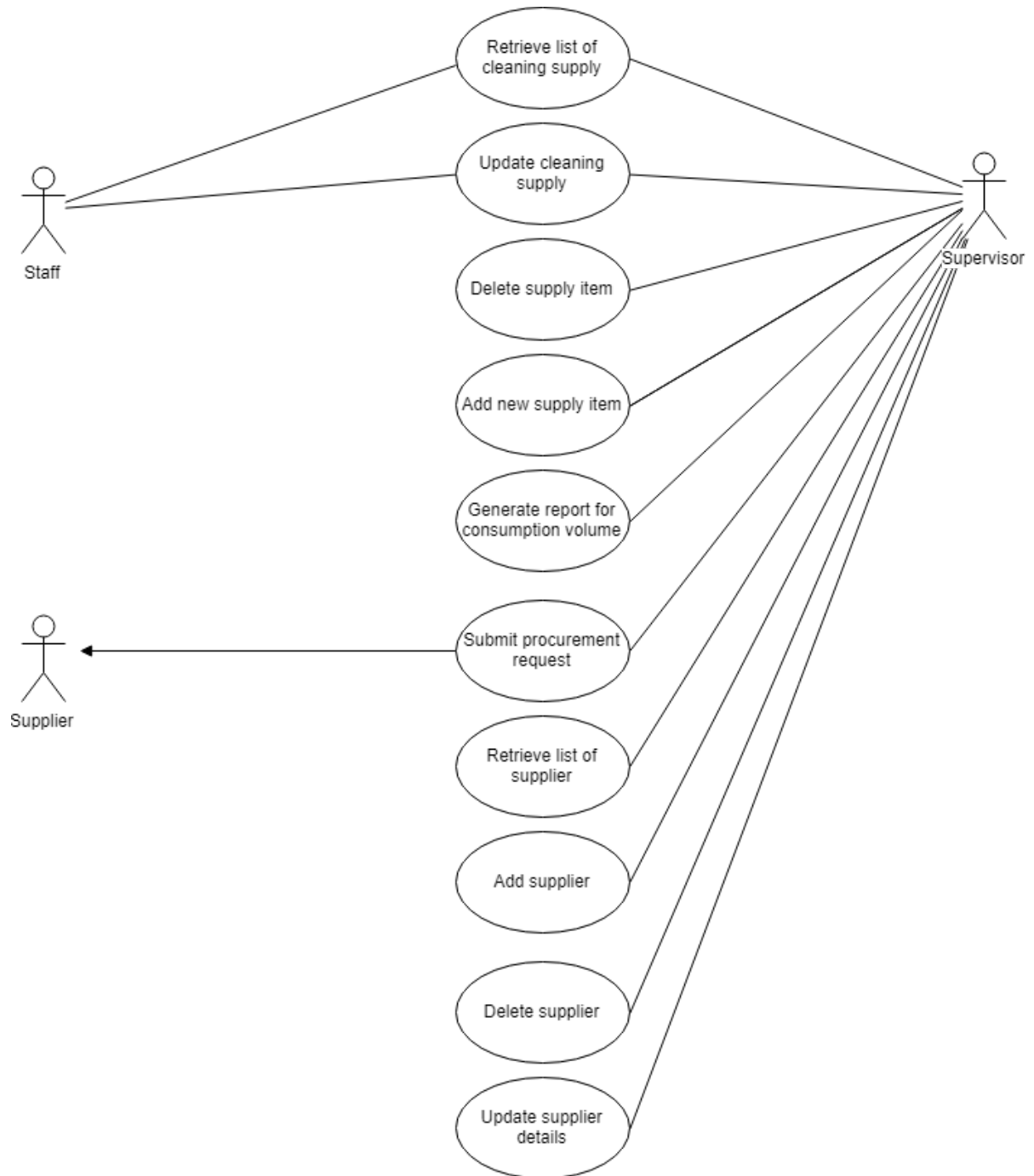
*Refer to attachment 4B*

Sequence diagrams

*Refer to attachment 4C, 4D and 4E*

# Asset Management Subsystem

Use case diagram



*Use case diagram for asset management subsystem*

## Use case descriptions for asset management subsystem

Use case:	Retrieve list of cleaning supply					
Triggering case:	User wants to check current stock of supplies.					
Brief description:	Allows user to view current inventory of cleaning supplies.					
Actors:	User					
Pre-conditions:	1. User must have already logged in to the system.					
Post-conditions:	Nil					
Flow of events:	<table><tr><th>User</th><th>System</th></tr><tr><td>1. Select “retrieve item list” on the logged in page.</td><td>1.1 System displays all types of supplies with their respective quantity.</td></tr></table>		User	System	1. Select “retrieve item list” on the logged in page.	1.1 System displays all types of supplies with their respective quantity.
User	System					
1. Select “retrieve item list” on the logged in page.	1.1 System displays all types of supplies with their respective quantity.					
Exception conditions:	Nil					

Use case:	Add new supply item											
Triggering case:	User brings in a new type of supply											
Brief description:	Add new type of supply into current inventory of cleaning supplies.											
Actors:	User											
Pre-conditions:	1. User must have already logged in to the system. 2. Supply type must be non-existent in the system yet.											
Post-conditions:	New item added to the inventory.											
Flow of events:	<table><tr><th>User</th><th>System</th></tr><tr><td>1. Include “retrieve list of inventory”.</td><td>1.1 System displays list of existing item types and quantities.</td></tr><tr><td>2. Select “add new supply item” on the logged in page.</td><td>2.1 Display field to key in name of new supply item.  2.2 System verifies that the name keyed in must not already exist.  2.2a If verification fails, display error message “Name of supply item already exists!” and return to step 2.1.  2.2b If verification succeeds, prompt user to key in quantity for the new item.</td></tr><tr><td>3. Key in quantity for the new item.</td><td>3.1 Systems displays the name and quantity for the new item, prompts user to check and save if information is correct.</td></tr><tr><td>4. Select “save” option.</td><td>4. System saves the new item with its quantity.</td></tr></table>		User	System	1. Include “retrieve list of inventory”.	1.1 System displays list of existing item types and quantities.	2. Select “add new supply item” on the logged in page.	2.1 Display field to key in name of new supply item.  2.2 System verifies that the name keyed in must not already exist.  2.2a If verification fails, display error message “Name of supply item already exists!” and return to step 2.1.  2.2b If verification succeeds, prompt user to key in quantity for the new item.	3. Key in quantity for the new item.	3.1 Systems displays the name and quantity for the new item, prompts user to check and save if information is correct.	4. Select “save” option.	4. System saves the new item with its quantity.
User	System											
1. Include “retrieve list of inventory”.	1.1 System displays list of existing item types and quantities.											
2. Select “add new supply item” on the logged in page.	2.1 Display field to key in name of new supply item.  2.2 System verifies that the name keyed in must not already exist.  2.2a If verification fails, display error message “Name of supply item already exists!” and return to step 2.1.  2.2b If verification succeeds, prompt user to key in quantity for the new item.											
3. Key in quantity for the new item.	3.1 Systems displays the name and quantity for the new item, prompts user to check and save if information is correct.											
4. Select “save” option.	4. System saves the new item with its quantity.											
Exception conditions:	Nil											

Use case:	Delete supply item									
Triggering case:	Type of supply is discontinued and will not be replenished.									
Brief description:	Removes type of supply and quantity from current inventory of cleaning supplies.									
Actors:	User									
Pre-conditions:	1. User must have already logged in to the system. 2. Supply type must be existent in the system.									
Post-conditions:	Supply type removed from inventory.									
Flow of events:	<table><tr><th>User</th><th>System</th></tr><tr><td>1. Include “retrieve list of inventory”.</td><td>1.1 System displays list of existing item types and quantities.</td></tr><tr><td>2. Select type of item to be deleted on the inventory page.</td><td>2.1 System checks the selected item's quantity.  2.1a If the item has 0 remaining quantity, proceed to step 2.2  2.1b If the item has &gt;0 quantity left, system prompt the user “There is existing quantity of the selected item!” with the error message and return to step 2.1.  2.2 Prompts to confirm delete the selected type of item.</td></tr><tr><td>3. User selects “Confirm”</td><td>3.1 Systems removes the item type for the selected item.</td></tr></table>		User	System	1. Include “retrieve list of inventory”.	1.1 System displays list of existing item types and quantities.	2. Select type of item to be deleted on the inventory page.	2.1 System checks the selected item's quantity.  2.1a If the item has 0 remaining quantity, proceed to step 2.2  2.1b If the item has >0 quantity left, system prompt the user “There is existing quantity of the selected item!” with the error message and return to step 2.1.  2.2 Prompts to confirm delete the selected type of item.	3. User selects “Confirm”	3.1 Systems removes the item type for the selected item.
User	System									
1. Include “retrieve list of inventory”.	1.1 System displays list of existing item types and quantities.									
2. Select type of item to be deleted on the inventory page.	2.1 System checks the selected item's quantity.  2.1a If the item has 0 remaining quantity, proceed to step 2.2  2.1b If the item has >0 quantity left, system prompt the user “There is existing quantity of the selected item!” with the error message and return to step 2.1.  2.2 Prompts to confirm delete the selected type of item.									
3. User selects “Confirm”	3.1 Systems removes the item type for the selected item.									
Exception conditions:	Nil									



Use case:	Update cleaning supply											
Triggering case:	User select update cleaning supplies option											
Brief description:	User select the option to update status or quantity of item in the system after they are being taken from the central storeroom to the facility, or after restocking.											
Actors:	User											
Pre-conditions:	1. User must have already logged in to the system.											
Post-conditions:	Cleaning supply quantity or status updated to the correct levels.											
Flow of events:	<table><tr><th>User</th><th>System</th></tr><tr><td>1. Include “retrieve list of cleaning supply”.</td><td>1.1 System displays list of existing item types and quantities.</td></tr><tr><td>1. Select item to be updated.</td><td>1.1 System displays current quantity for the selected item.</td></tr><tr><td>2. Key in new quantity for the selected item.</td><td>2.1 Systems saves the new quantity for the selected item.</td></tr><tr><td></td><td>2.2a. If selected item’s new quantity falls below certain stipulated value, display “low stock” and prompt staff to select “notify supervisor” option.</td></tr></table>		User	System	1. Include “retrieve list of cleaning supply”.	1.1 System displays list of existing item types and quantities.	1. Select item to be updated.	1.1 System displays current quantity for the selected item.	2. Key in new quantity for the selected item.	2.1 Systems saves the new quantity for the selected item.		2.2a. If selected item’s new quantity falls below certain stipulated value, display “low stock” and prompt staff to select “notify supervisor” option.
User	System											
1. Include “retrieve list of cleaning supply”.	1.1 System displays list of existing item types and quantities.											
1. Select item to be updated.	1.1 System displays current quantity for the selected item.											
2. Key in new quantity for the selected item.	2.1 Systems saves the new quantity for the selected item.											
	2.2a. If selected item’s new quantity falls below certain stipulated value, display “low stock” and prompt staff to select “notify supervisor” option.											
Exception conditions:	Nil											

Use case:	Generate report for consumption volume					
Triggering case:	Supervisor wants to monitor consumption volume.					
Brief description:	Monitor consumption volume of cleaning supplies.					
Actors:	Supervisor					
Pre-conditions:	<div>1. Supervisor must have already logged in to the system.</div> <div>2. Time period selected is valid</div> <div>3. The computer accessing the system must be connected to an active printer(if selecting “print report”)</div>					
Post-conditions:	Report is generated and saved.					
Flow of events:	<table><thead><tr><th>Supervisor</th><th>System</th></tr></thead><tbody><tr><td><div>1. Supervisor selects a start and end date in the system and selects “generate report” option.</div><div>2. Supervisor selects “sort by ascending order” or sort by “descending order”</div><div>3. Supervisor clicks the save button for logging and/or printing purposes.</div></td><td><div>1.1 System retrieves the list of supplies consumed within the selected time period and generates a report of the consumption levels sorted in alphabetical order by default.</div><div>2.1 System sorts the consumption levels based on the selected option.</div><div>3.1 System saves the generated report on the local computer.</div></td></tr></tbody></table> <div>* Refer to diagram 5F for a sample of generated report</div>		Supervisor	System	<div>1. Supervisor selects a start and end date in the system and selects “generate report” option.</div> <div>2. Supervisor selects “sort by ascending order” or sort by “descending order”</div> <div>3. Supervisor clicks the save button for logging and/or printing purposes.</div>	<div>1.1 System retrieves the list of supplies consumed within the selected time period and generates a report of the consumption levels sorted in alphabetical order by default.</div> <div>2.1 System sorts the consumption levels based on the selected option.</div> <div>3.1 System saves the generated report on the local computer.</div>
Supervisor	System					
<div>1. Supervisor selects a start and end date in the system and selects “generate report” option.</div> <div>2. Supervisor selects “sort by ascending order” or sort by “descending order”</div> <div>3. Supervisor clicks the save button for logging and/or printing purposes.</div>	<div>1.1 System retrieves the list of supplies consumed within the selected time period and generates a report of the consumption levels sorted in alphabetical order by default.</div> <div>2.1 System sorts the consumption levels based on the selected option.</div> <div>3.1 System saves the generated report on the local computer.</div>					
Exception conditions:	Nil					

Use case:	Submit procurement request									
Triggering case:	Supervisor receives notification from cleaning staff that at least 1 type of supply is low.									
Brief description:	Submits procurement request for suppliers to send new cleaning supplies.									
Actors:	Supervisor									
Pre-conditions:	1. Supervisor must have already logged in to the system.									
Post-conditions:	Request sent for supplies low in stock to be delivered to central storeroom.									
Flow of events:	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Supervisor selects all the items to be added into the procurement request.</td><td>1.1 System retrieves list of suppliers and displays the names of the supplier(s) which supply the selected item types and their respective prices.</td></tr><tr><td>2. Supervisor selects the supplier(s) to order new supplies from.</td><td>2.1 System prompts for amount of each type of supply to order from each supplier(s).</td></tr><tr><td>3. Supervisor keys in amount(s).</td><td>3.1 System sends the procurement request to the supplier(s).</td></tr></table>		Supervisor	System	1. Supervisor selects all the items to be added into the procurement request.	1.1 System retrieves list of suppliers and displays the names of the supplier(s) which supply the selected item types and their respective prices.	2. Supervisor selects the supplier(s) to order new supplies from.	2.1 System prompts for amount of each type of supply to order from each supplier(s).	3. Supervisor keys in amount(s).	3.1 System sends the procurement request to the supplier(s).
Supervisor	System									
1. Supervisor selects all the items to be added into the procurement request.	1.1 System retrieves list of suppliers and displays the names of the supplier(s) which supply the selected item types and their respective prices.									
2. Supervisor selects the supplier(s) to order new supplies from.	2.1 System prompts for amount of each type of supply to order from each supplier(s).									
3. Supervisor keys in amount(s).	3.1 System sends the procurement request to the supplier(s).									
Exception conditions:	Nil									

Use case:	Retrieve list of suppliers					
Triggering case:	Supervisor wants to check the list of saved suppliers.					
Brief description:	Allows supervisor to view current saved list of suppliers and their details					
Actors:	Supervisor					
Pre-conditions:	1. Supervisor must have already logged in to the system.					
Post-conditions:	-					
Flow of events:	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Select “retrieve list of suppliers” option</td><td>1.1 System displays all suppliers details in each row, with their company name, contact person name, number, address, type of supplies in separate columns.</td></tr></table>		Supervisor	System	1. Select “retrieve list of suppliers” option	1.1 System displays all suppliers details in each row, with their company name, contact person name, number, address, type of supplies in separate columns.
Supervisor	System					
1. Select “retrieve list of suppliers” option	1.1 System displays all suppliers details in each row, with their company name, contact person name, number, address, type of supplies in separate columns.					
Exception conditions:	Nil					

Use case:	Add supplier					
Triggering case:	Supervisor wants to add a regular supplier into the list of suppliers.					
Brief description:	Adds a new supplier to a list of maintained suppliers for easy reference.					
Actors:	Supervisor					
Pre-conditions:	1. Supervisor must have already logged in to the system. 2. Supplier must not already exist in the list.					
Post-conditions:	Supplier added into the list.					
Flow of events:	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include “retrieve list of suppliers” .  2. Select “add new supplier” option.  3. Supervisor keys in key information of supplier, including company name, contact person name, number, company address, email address and what type(s) of supplies the company is responsible for delivering.</td><td>1.1 System displays list of existing suppliers.  2.1 Systems displays blanks to key in the key information of the new supplier.  3.1 System verifies that there is no existing record with the same company name, company address or email address.  3.1a If verification fails, prompts user “Record with the same (variable) already exists!” and return to step 2.1.  3.2b If verification succeeds, system saves the new entry into a list of maintained and suppliers and popup to let user know that “New record saved!”.</td></tr></table>		Supervisor	System	1. Include “retrieve list of suppliers” .  2. Select “add new supplier” option.  3. Supervisor keys in key information of supplier, including company name, contact person name, number, company address, email address and what type(s) of supplies the company is responsible for delivering.	1.1 System displays list of existing suppliers.  2.1 Systems displays blanks to key in the key information of the new supplier.  3.1 System verifies that there is no existing record with the same company name, company address or email address.  3.1a If verification fails, prompts user “Record with the same (variable) already exists!” and return to step 2.1.  3.2b If verification succeeds, system saves the new entry into a list of maintained and suppliers and popup to let user know that “New record saved!”.
Supervisor	System					
1. Include “retrieve list of suppliers” .  2. Select “add new supplier” option.  3. Supervisor keys in key information of supplier, including company name, contact person name, number, company address, email address and what type(s) of supplies the company is responsible for delivering.	1.1 System displays list of existing suppliers.  2.1 Systems displays blanks to key in the key information of the new supplier.  3.1 System verifies that there is no existing record with the same company name, company address or email address.  3.1a If verification fails, prompts user “Record with the same (variable) already exists!” and return to step 2.1.  3.2b If verification succeeds, system saves the new entry into a list of maintained and suppliers and popup to let user know that “New record saved!”.					
Exception conditions:	Nil					

Use case:	Delete supplier									
Triggering case:	Supervisor wants to delete a regular supplier from the list of suppliers.									
Brief description:	Deletes a supplier from the list of maintained suppliers.									
Actors:	Supervisor									
Pre-conditions:	1. Supervisor must have already logged in to the system. 2. Supplier must already exist in the list.									
Post-conditions:	Supplier deleted from the list.									
Flow of events:	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include “retrieve list of suppliers”.</td><td>1.1 System displays list of existing suppliers.</td></tr><tr><td>2. Supervisor searches and selects name of company to be deleted</td><td>2.1 System checks if the selected supplier has any items that are in transit, ie ordered but not yet delivered to central storage.  2.1a If there is no item in transit, proceed to step 2.2.  2.1b If there is item in transit, popup error message that tells user “Unable to delete supplier: this supplier still has undelivered items. Try again after items have been received”. Return to step 1.1.  2.2 System prompts for confirmation for all details related to the supplier to be deleted.</td></tr><tr><td>3. User confirms delete.</td><td>3.1 Delete supplier entry from system.</td></tr></table>		Supervisor	System	1. Include “retrieve list of suppliers”.	1.1 System displays list of existing suppliers.	2. Supervisor searches and selects name of company to be deleted	2.1 System checks if the selected supplier has any items that are in transit, ie ordered but not yet delivered to central storage.  2.1a If there is no item in transit, proceed to step 2.2.  2.1b If there is item in transit, popup error message that tells user “Unable to delete supplier: this supplier still has undelivered items. Try again after items have been received”. Return to step 1.1.  2.2 System prompts for confirmation for all details related to the supplier to be deleted.	3. User confirms delete.	3.1 Delete supplier entry from system.
Supervisor	System									
1. Include “retrieve list of suppliers”.	1.1 System displays list of existing suppliers.									
2. Supervisor searches and selects name of company to be deleted	2.1 System checks if the selected supplier has any items that are in transit, ie ordered but not yet delivered to central storage.  2.1a If there is no item in transit, proceed to step 2.2.  2.1b If there is item in transit, popup error message that tells user “Unable to delete supplier: this supplier still has undelivered items. Try again after items have been received”. Return to step 1.1.  2.2 System prompts for confirmation for all details related to the supplier to be deleted.									
3. User confirms delete.	3.1 Delete supplier entry from system.									
Exception conditions:	Nil									

Use case:	Update supplier details									
Triggering case:	Supervisor wants to update details of a supplier.									
Brief description:	Allows supervisor to update details of a supplier.									
Actors:	Supervisor									
Pre-conditions:	1. Supervisor must have already logged in to the system. 2. Supplier must already exist in the list.									
Post-conditions:	-									
Flow of events:	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include “retrieve list of suppliers”.</td><td>1.1 System displays list of existing suppliers.</td></tr><tr><td>2. Supervisor searches for and selects the entry of the supplier to be updated.</td><td>2.1 System displays all details of the selected supplier.</td></tr><tr><td>3. Keys in the details to be updated in respective column(s) and selects “update”</td><td>3.1 Save the new details of the supplier. Popup to let user know that “Changes saved!”</td></tr></table>		Supervisor	System	1. Include “retrieve list of suppliers”.	1.1 System displays list of existing suppliers.	2. Supervisor searches for and selects the entry of the supplier to be updated.	2.1 System displays all details of the selected supplier.	3. Keys in the details to be updated in respective column(s) and selects “update”	3.1 Save the new details of the supplier. Popup to let user know that “Changes saved!”
Supervisor	System									
1. Include “retrieve list of suppliers”.	1.1 System displays list of existing suppliers.									
2. Supervisor searches for and selects the entry of the supplier to be updated.	2.1 System displays all details of the selected supplier.									
3. Keys in the details to be updated in respective column(s) and selects “update”	3.1 Save the new details of the supplier. Popup to let user know that “Changes saved!”									
Exception conditions:	Nil									

Simplified entity class diagram

*Refer to attachment 5A*

Non-entity class diagram

*Refer to attachment 5B*

Sequence diagrams

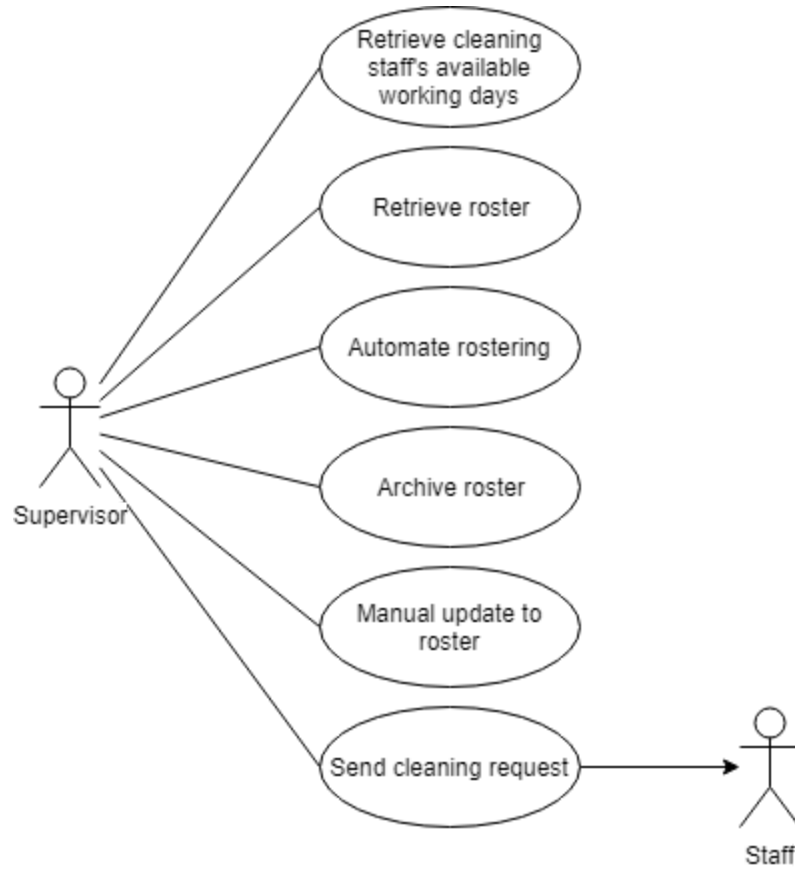
*Refer to attachment 5C, 5D and 5E*

Sample report

*Refer to attachment 5F*

# Cleaning Schedule Management Subsystem

Use case diagram



*Use case diagram for cleaning schedule management subsystem*



## Use case description for cleaning schedule management subsystem

Use case:	Retrieve cleaning staff's available working days									
Triggering case:	Supervisor wants to retrieve cleaning staff's available working days									
Brief description:	Supervisor access system to retrieve cleaning staff's available working days									
Actors:	Supervisor									
Pre-conditions:	<div>1. Supervisor must be logged in to the system</div> <div>2. Offs, Leaves and Medical Certificates have been approved and recorded in the system</div>									
Post-conditions:	Displays the list of cleaning staff's available working days									
Flow of events:	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include retrieve list of venues</td><td></td></tr><tr><td>2. Click on "cleaning schedule management"</td><td><div>2.1 Retrieves the list of cleaning staff assigned to the location</div><div>2.2 Displays cleaning schedule management page with the list of cleaning staff assigned to the location</div></td></tr><tr><td>3. Click on retrieve cleaning staff's available working days</td><td><div>3.1 Retrieves all of the staff's approved off, leaves and MC</div><div>3.2 Displays a variety of cleaning staff's available working days for the next month in a calendar table view (grey out working days which the</div></td></tr></table>		Supervisor	System	1. Include retrieve list of venues		2. Click on "cleaning schedule management"	<div>2.1 Retrieves the list of cleaning staff assigned to the location</div> <div>2.2 Displays cleaning schedule management page with the list of cleaning staff assigned to the location</div>	3. Click on retrieve cleaning staff's available working days	<div>3.1 Retrieves all of the staff's approved off, leaves and MC</div> <div>3.2 Displays a variety of cleaning staff's available working days for the next month in a calendar table view (grey out working days which the</div>
Supervisor	System									
1. Include retrieve list of venues										
2. Click on "cleaning schedule management"	<div>2.1 Retrieves the list of cleaning staff assigned to the location</div> <div>2.2 Displays cleaning schedule management page with the list of cleaning staff assigned to the location</div>									
3. Click on retrieve cleaning staff's available working days	<div>3.1 Retrieves all of the staff's approved off, leaves and MC</div> <div>3.2 Displays a variety of cleaning staff's available working days for the next month in a calendar table view (grey out working days which the</div>									

	<table border="1"> <tr> <td></td><td>staff has applied for off, leaves, and MC)</td></tr> </table>		staff has applied for off, leaves, and MC)
	staff has applied for off, leaves, and MC)		
Exception conditions:	Nil		

Use case:	Retrieve rostering
Triggering case:	Supervisor wants to retrieve the rostering
Brief description:	Supervisor access system to retrieve the rostering
Actors:	Supervisor
Pre-conditions:	<ol style="list-style-type: none"> <li>1. Supervisor must be logged in to the system</li> <li>2. Roster must exist in the system</li> </ol>
Post-conditions:	Displays the rostering from the system

Flow of events:		
	Supervisor	System
	1. Include retrieve list of venues	
	2. Click on “cleaning schedule management”	2.1 Retrieves the list of cleaning staff assigned to the venue  2.2 Displays cleaning schedule management page with the list of cleaning staff assigned to the venue
Exception conditions:	3. Click on retrieve rostering	3.1 Retrieves rostering record from the database  3.2 Displays existing rostering record
Exception conditions:	Nil	

Use case:	Automate rostering
Triggering case:	Supervisor wants to automate the rostering
Brief description:	Supervisor access system to automate the rostering
Actors:	Supervisor
Pre-conditions:	1. Supervisor must be logged in to the system 2. Cleaning staff have indicated their available working days in the system 3. Cleaning staff details must exist in the system

Post-conditions:	Create a new roster record in the system database	
Flow of events:		
	Supervisor	System
	1. Include retrieve list of venues	
	2. Click on “cleaning schedule management”	2.1 Retrieves the list of cleaning staff assigned to the location  2.2 Displays cleaning schedule management page with the list of cleaning staff assigned to the location
	3. Click on “add rostering” button	3.1 Retrieves list of venues records with gender specific requirement  3.2 Retrieves existing roster for the location  3.3 Displays add rostering page with the list of venues
4. If there is no existing roster, click “automate rostering”	4.1 Retrieves cleaning staff records and the gender requirement for each venue  4.2 Filter the cleaning staff records according to the gender requirement  4.3 Retrieves cleaning staff’s addresses and categorize them into the individual town councils (West Coast TC, Chua Chu Kang TC, Jurong-Clementi TC, Marsiling-Yew Tee TC, Holland-Bukit Panjang TC, Tanjong Pagar TC, Bishan-Toa Payoh TC, Nee Soon TC,	

	<p>Sembawang TC, Ang Mo Kio TC, Pasir Ris-Punggol TC, Aljunied-Hougang TC, Jalan Besar TC, Marine Parade TC, Tampines TC, East Coast-Fengshan TC).</p> <p>4.4 Based on the adjacency of the cleaning staff's town council away from the town council which oversees the designated building, assign a score to the cleaning staff. (max score*: 11, the closer the cleaning staff's town council the higher the score)</p> <p>4.5 Retrieves cleaning staff's number of years of past experiences for that building and assign a score accordingly*</p> <p>4.6 Accumulate the scores for each cleaning staff</p> <p>4.7 Retrieves cleaning staff's available working days for the next month</p> <p>4.8 Retrieves the number of staff needed for the venue</p> <p>4.9. Based on the accumulated scores (higher scores having more priority), assign each cleaning staff starting from the highest score to the lowest to a random time slot for each available working day until the number of required staff has been fulfilled.</p> <p>4.10 If all the working days and time slots are occupied, displays</p>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<div data-bbox="456 205 1411 556"> <div data-bbox="922 220 1380 296">message “successfully filled up the rostering”</div> <div data-bbox="922 342 1398 533">4.10a If there are time slots still not occupied, displays message “unable to fully fill the rostering”, highlight the working days and time slots that are still vacant.</div> </div> <div data-bbox="456 600 1373 682">* Supervisor will be able to adjust the allocation of scores to place more priority on either distance away or number of years of experience</div>
Exception conditions:	Nil



*Distribution of town council in Singapore*

Use case:	Archive roster							
Triggering case:	Supervisor wants to archive roster							
Brief description:	Supervisor access the system to archive roster							
Actors:	Supervisor							
Pre-conditions:	1. Supervisor must be logged in to the system 2. Roster must exist in the system’s database							
Post-conditions:	Archive the roster record in the system’s database							
Flow of events:	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include retrieve rostering</td><td></td></tr><tr><td>2. Click on archive button next to the roster</td><td>2.1 Archive the selected roster record to the list of archived rosters in the database.  2.2 Displays message “successfully archived roster”</td></tr></table>		Supervisor	System	1. Include retrieve rostering		2. Click on archive button next to the roster	2.1 Archive the selected roster record to the list of archived rosters in the database.  2.2 Displays message “successfully archived roster”
Supervisor	System							
1. Include retrieve rostering								
2. Click on archive button next to the roster	2.1 Archive the selected roster record to the list of archived rosters in the database.  2.2 Displays message “successfully archived roster”							
Exception conditions:	Nil							

Use case:	Manual update to roster									
Triggering case:	Supervisor wants to manually update roster									
Brief description:	Supervisor access system to manually update roster									
Actors:	Supervisor									
Pre-conditions:	1. Supervisor must be logged in to the system 2. Roster must exist in the system									
Post-conditions:	Update roster record in the system's database									
Flow of events:	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include retrieve rostering</td><td></td></tr><tr><td>2. Click on “manual update” button next to the roster</td><td>2.1 Displays roster pre-filled with the existing roster record from the database.</td></tr><tr><td>3. Manually update the roster by either shifting the employees’ field (drag and drop), adding new employee field, deleting existing employee field in the calendar view</td><td>3.1 Updates roster record in the database  3.2 Displays message “successfully updated roster”</td></tr></table>		Supervisor	System	1. Include retrieve rostering		2. Click on “manual update” button next to the roster	2.1 Displays roster pre-filled with the existing roster record from the database.	3. Manually update the roster by either shifting the employees’ field (drag and drop), adding new employee field, deleting existing employee field in the calendar view	3.1 Updates roster record in the database  3.2 Displays message “successfully updated roster”
Supervisor	System									
1. Include retrieve rostering										
2. Click on “manual update” button next to the roster	2.1 Displays roster pre-filled with the existing roster record from the database.									
3. Manually update the roster by either shifting the employees’ field (drag and drop), adding new employee field, deleting existing employee field in the calendar view	3.1 Updates roster record in the database  3.2 Displays message “successfully updated roster”									
Exception conditions:	Nil									



Use case:	Send cleaning requests													
Triggering case:	Supervisor sends cleaning request to cleaning staff													
Brief description:	Supervisor access system to send cleaning request to cleaning staff													
Actors:	Supervisor													
Pre-conditions:	1. Supervisor must be logged in to the system 2. Cleaning staff details must exist in the system													
Post-conditions:	Sends SMS messages to the cleaning staff regarding the cleaning request													
Flow of events:	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Include retrieve rostering</td><td></td></tr><tr><td>2. Click on “message” button</td><td>2.1 Retrieve message template and displays messaging page</td></tr><tr><td>3. If the supervisor wishes to make changes, click on modify message  3a. If the supervisor does not wish to make changes, go to step 5</td><td>3.1 Displays a form pre-populated with the existing message template</td></tr><tr><td>4. Edits the message template, and click save</td><td>4.1 Create new message template record to the database</td></tr><tr><td>5. Click on “send cleaning requests” button</td><td>5.1 Retrieves all staff records assigned to the location  5.2 Displays prompt message to the user: “You are about to send cleaning</td></tr></table>		Supervisor	System	1. Include retrieve rostering		2. Click on “message” button	2.1 Retrieve message template and displays messaging page	3. If the supervisor wishes to make changes, click on modify message  3a. If the supervisor does not wish to make changes, go to step 5	3.1 Displays a form pre-populated with the existing message template	4. Edits the message template, and click save	4.1 Create new message template record to the database	5. Click on “send cleaning requests” button	5.1 Retrieves all staff records assigned to the location  5.2 Displays prompt message to the user: “You are about to send cleaning
Supervisor	System													
1. Include retrieve rostering														
2. Click on “message” button	2.1 Retrieve message template and displays messaging page													
3. If the supervisor wishes to make changes, click on modify message  3a. If the supervisor does not wish to make changes, go to step 5	3.1 Displays a form pre-populated with the existing message template													
4. Edits the message template, and click save	4.1 Create new message template record to the database													
5. Click on “send cleaning requests” button	5.1 Retrieves all staff records assigned to the location  5.2 Displays prompt message to the user: “You are about to send cleaning													

		requests to the cleaning staff, do you wish to proceed?"
	6. Click on "confirm and send"	6.1 Sends pre-populated SMS messages (cleaning requests) about the current rostering to the respective cleaning staff's contact number
Exception conditions:	Nil	

Simplified entity class diagram

*Refer to attachment 6A*

Non-entity class diagram

*Refer to attachment 6B*

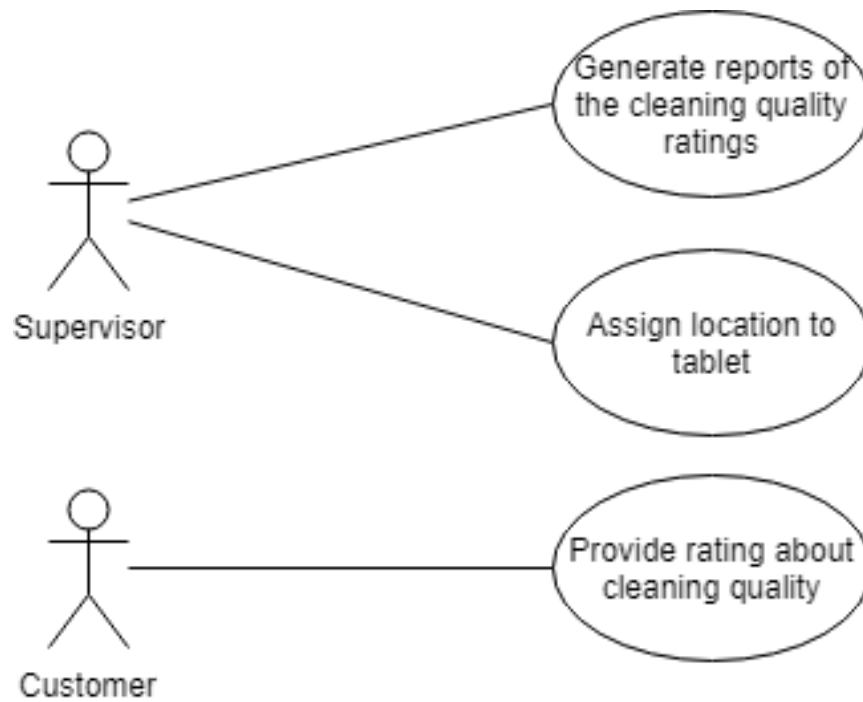
Sequence diagrams

*Refer to attachment 6C, 6D and 6E*

# Feedback Management Subsystem

## Rating Management Module

Use case diagram



*Use case diagram for rating management module*

## Use case description for rating management module

Use case:	Provide rating about cleaning quality					
Triggering case:	Customer presses the “give rating” button on the screen provided in the toilet					
Brief description:	Customer has finished patronizing the toilet and wishes to provide feedback					
Actors:	Customer					
Pre-conditions:	Tablet in toilet has been assigned to a specific toilet					
Post-conditions:	System registers the response and returns to the welcome screen					
Flow of events:	<table><tr><th>Customer</th><th>System</th></tr><tr><td>1. Customer presses the “give rating” button on the screen 2. Customer chooses a rating from 1 to 5 stars for separate categories consisting of Cleanliness, Odour, Amenities, and Overall Rating 3. Customer presses the “submit rating” button on the screen</td><td>1.1. System redirects to the rating screen  3.1. System registers the ratings given and records it in the database 3.2. System redirects to the welcome screen</td></tr></table>		Customer	System	1. Customer presses the “give rating” button on the screen 2. Customer chooses a rating from 1 to 5 stars for separate categories consisting of Cleanliness, Odour, Amenities, and Overall Rating 3. Customer presses the “submit rating” button on the screen	1.1. System redirects to the rating screen  3.1. System registers the ratings given and records it in the database 3.2. System redirects to the welcome screen
Customer	System					
1. Customer presses the “give rating” button on the screen 2. Customer chooses a rating from 1 to 5 stars for separate categories consisting of Cleanliness, Odour, Amenities, and Overall Rating 3. Customer presses the “submit rating” button on the screen	1.1. System redirects to the rating screen  3.1. System registers the ratings given and records it in the database 3.2. System redirects to the welcome screen					
Exception conditions:	Customer leaves halfway after triggering the system					

Use case:	Generate reports of the cleaning quality ratings											
Triggering case:	Supervisor selects the time period of interest and presses the “Generate Cleanliness Feedback Report” button											
Brief description:	Supervisor generates the report of toilet cleanliness ratings and feedback in a selected time period and rearranges it for easier browsing											
Actors:	Supervisor											
Pre-conditions:	Supervisor is logged into the system (The time period selected is valid and ratings have been given in that time period)											
Post-conditions:	Report is saved and system returns to default state											
Flow of events:	<table><thead><tr><th>Supervisor</th><th>System</th></tr></thead><tbody><tr><td>1. Supervisor selects a start and end date in the system and presses the “generate cleaning quality report” button</td><td>1.1. System retrieves the records of ratings given in that time period and generates a report of the ratings given for all facilities sorted by location by default</td></tr><tr><td>2. Supervisor selects a category such as cleanliness or odour for ease of viewing</td><td>2.1 System groups the ratings based on the chosen method</td></tr><tr><td>3. Supervisor selects “sort by Ascending order” or “sort by Descending order”</td><td>3.1. System sorts the ratings based on the selected option</td></tr><tr><td>4. Supervisor clicks the save button for logging and/or printing purposes</td><td>4.1 System saves the generated report to a folder on the computer and returns to the default page</td></tr></tbody></table> <p>* Refer to diagram 7F for a sample of generated report</p>		Supervisor	System	1. Supervisor selects a start and end date in the system and presses the “generate cleaning quality report” button	1.1. System retrieves the records of ratings given in that time period and generates a report of the ratings given for all facilities sorted by location by default	2. Supervisor selects a category such as cleanliness or odour for ease of viewing	2.1 System groups the ratings based on the chosen method	3. Supervisor selects “sort by Ascending order” or “sort by Descending order”	3.1. System sorts the ratings based on the selected option	4. Supervisor clicks the save button for logging and/or printing purposes	4.1 System saves the generated report to a folder on the computer and returns to the default page
Supervisor	System											
1. Supervisor selects a start and end date in the system and presses the “generate cleaning quality report” button	1.1. System retrieves the records of ratings given in that time period and generates a report of the ratings given for all facilities sorted by location by default											
2. Supervisor selects a category such as cleanliness or odour for ease of viewing	2.1 System groups the ratings based on the chosen method											
3. Supervisor selects “sort by Ascending order” or “sort by Descending order”	3.1. System sorts the ratings based on the selected option											
4. Supervisor clicks the save button for logging and/or printing purposes	4.1 System saves the generated report to a folder on the computer and returns to the default page											
Exception conditions:	Nil											

Use case:	Assign location to the tablet					
Triggering case:	Supervisor presses the “register location” button on the tablet					
Brief description:	Supervisor registers a location for the tablet in the cleaning office so that the incident management reports can be tied to a location					
Actors:	Supervisor					
Pre-conditions:	Supervisor has logged into the tablet with their admin credentials					
Post-conditions:	Location of the tablet updated					
Flow of events:	<table><tr><th>Supervisor</th><th>System</th></tr><tr><td>1. Supervisor presses the “register location” button on the tablet 2. If desired location is present in the list, supervisor selects it and presses the “update” button 2a. If the desired location is not present in the list, supervisor selects the “add a new location” button 2a.1 Supervisor enters the new location and presses the “update” button</td><td>1.1. System brings up a dropdown list of locations 2.1 System updates the location tied to that tablet in the database  2a.1.1 System updates the location tied to that tablet in the database and adds the new location to the overall list of locations</td></tr></table>		Supervisor	System	1. Supervisor presses the “register location” button on the tablet 2. If desired location is present in the list, supervisor selects it and presses the “update” button 2a. If the desired location is not present in the list, supervisor selects the “add a new location” button 2a.1 Supervisor enters the new location and presses the “update” button	1.1. System brings up a dropdown list of locations 2.1 System updates the location tied to that tablet in the database  2a.1.1 System updates the location tied to that tablet in the database and adds the new location to the overall list of locations
Supervisor	System					
1. Supervisor presses the “register location” button on the tablet 2. If desired location is present in the list, supervisor selects it and presses the “update” button 2a. If the desired location is not present in the list, supervisor selects the “add a new location” button 2a.1 Supervisor enters the new location and presses the “update” button	1.1. System brings up a dropdown list of locations 2.1 System updates the location tied to that tablet in the database  2a.1.1 System updates the location tied to that tablet in the database and adds the new location to the overall list of locations					
Exception conditions:	Nil					

Simplified entity class diagram

*Refer to attachment 7A*

Non-entity class diagram

*Refer to attachment 7B*

Sequence diagrams

*Refer to attachment 7C, 7D and 7E*

Sample report

*Refer to attachment 7F*

## Conclusion

The business requirements document includes the various functionalities implemented in the cleaning management system that is customised for SG Cleaning Solutions to help streamline the working process and assist in operations smoothly. The system manages the basic access rights management while managing the cleaning staff of the company. Cleaning staff attendance taking is captured via tapping of card due to the likelihood that some of the cleaning staff might not be as tech savvy, thus we adopted the simplest implementation method for doing so. Supervisors are granted access to multiple platforms to perform operations such as leave application on behalf of the cleaning staff for the same reason.

The utilisation of technology also allows the company to automate and better manage the allocation of manpower across different locations. In addition, it increases the company's ability to handle more projects as the management has been greatly improved. Asset management allows the inventory of cleaning supplies to be tracked real time and ensure cleaning supplies are always well stocked such that it does not affect the cleaning process. Rostering of cleaning schedule has also been automated to better allocate manpower according to several criteria, for example, location, which results in greater convenience for both cleaning staff and supervisor in the case when there is lack of manpower. Automated rostering also significantly reduces the workload of supervisors and allow them to spend the time to work on other more important tasks.

Finally, the system will also be able to take in and organize feedback given by the users of the facilities and venues covered by SG Cleaning Services, allowing supervisors to easily generate, sort and save reports. The functionality for employees to report equipment damages is also present and will allow for quick and seamless communication to maintain maximum efficiency.

## References:

Profile of Town Councils. (2018, September 15). Retrieved October 11, 2018, from <http://www.towncouncils.sg/about/profileoftowncouncils.html>