SOFTWARE DESIGN SPECIFICATION (SDS)

for

Smart Toilet Cleaning Management System

GROUP MEMBERS:

- Neel Patel (IIT2020106)
- Agrim Verma (IIT2020136)
- Shantanu (IIT2020166)
- Jiniya Singal (IIT2020181)

Table of Contents :-

1. Introduction:

‘TOMS’, a smart toilet cleaning management system ANDROID APP that monitors the hygiene within the toilet and generates alerts or alarms for the staff members if the toilet requires any attention. The toilet will be equipped with turbidity and gas sensors that monitor water quality and smell respectively.

1.1. Purpose:

A smart toilet cleaning management system that monitors the hygiene within the toilet and generates alerts or alarms for the staff members if the toilet requires any attention. And allow building owners and Facility Management(FM) and cleaning companies to manage usage across multiple restrooms.

1.2. Scope

a. Mobile Application for interaction between Admin and Facility Management
b. Staff members can receive alert messages from Admin

1.3. Definitions, Acronyms, and Abbreviations:

1.3.1. Acronyms and abbreviations:

a. TOMA: Smart toilet cleaning management system : app name
b. SRS: Software Requirements Specification
c. NTU: Nephelometric Turbidity unit

1.3.2. Definitions

a. "Smart toilet cleaning management system”: An Application for IIIT ALLAHABAD admin building which helps staff workers for smart management of the toilet system.

1.4. Overview:

Smart toilet cleaning management system that monitors the hygiene within the toilet and generates alerts or alarms for the staff members if the toilet requires

any attention. The toilet will be equipped with turbidity and gas sensors that monitor water quality and smell respectively.
The turbidity sensor is measured in Nephelometric Turbidity unit (NTU) where high NTU value indicates the water is dirty, whereas the gas sensor (MQ2) detects the foul smell with high output voltage, and later sends a signal to the system to compare with the hygiene threshold and accordingly send the message/ email to the staff regarding the toilet being unhygienic and require immediate cleaning.

2. Conceptual Architecture

We have made a rough model of the smart toilet cleaning management system. Its conceptual architecture deals with admin, staff members and system. First the admin and staff member register themselves by a username and password.

Admin's role is to set the threshold value of the gas sensor and turbidity sensor and he also has the power to change it according to the situation. Admin also gets to know all about the updates at various places at different instances of time.

Staff members are allowed to update the required information about their work. Whenever an alarm is detected by any staff member then he can switch off it and then clean the toilet and update the log about the work done by him by mentioning the date and time whenever he does the cleaning of the toilet.

The work of the system would be to compare the values of NTU and MQ2 with the threshold value which is already set by the admin. If any of the values is higher than the threshold value then it will give an alarm indicating that this particular toilet is unhygienic and needs to be cleaned. It will also be keeping the record of the work done by staff members in log.

3. Logical Architecture (Class Diagram, Sequence Diagram, State Diagram)

3.1 Class Diagram:

Class diagram, one of the most commonly used diagrams in object-oriented systems, models the static design view for a system. The static view mainly supports the functional requirements of a system – the services the system should provide to the end users. A class diagram shows a set of classes, interfaces, and collaborations and their relationships. Class diagrams involve global system description, such as the system architecture, and detail aspects such as the attributes and operations within a class as well.

We have built this class diagram from the view of object composition of the system –

1. Login – There are two actors which will be using this application so there will be two buttons one will be Admin login and the other one is for staff members.

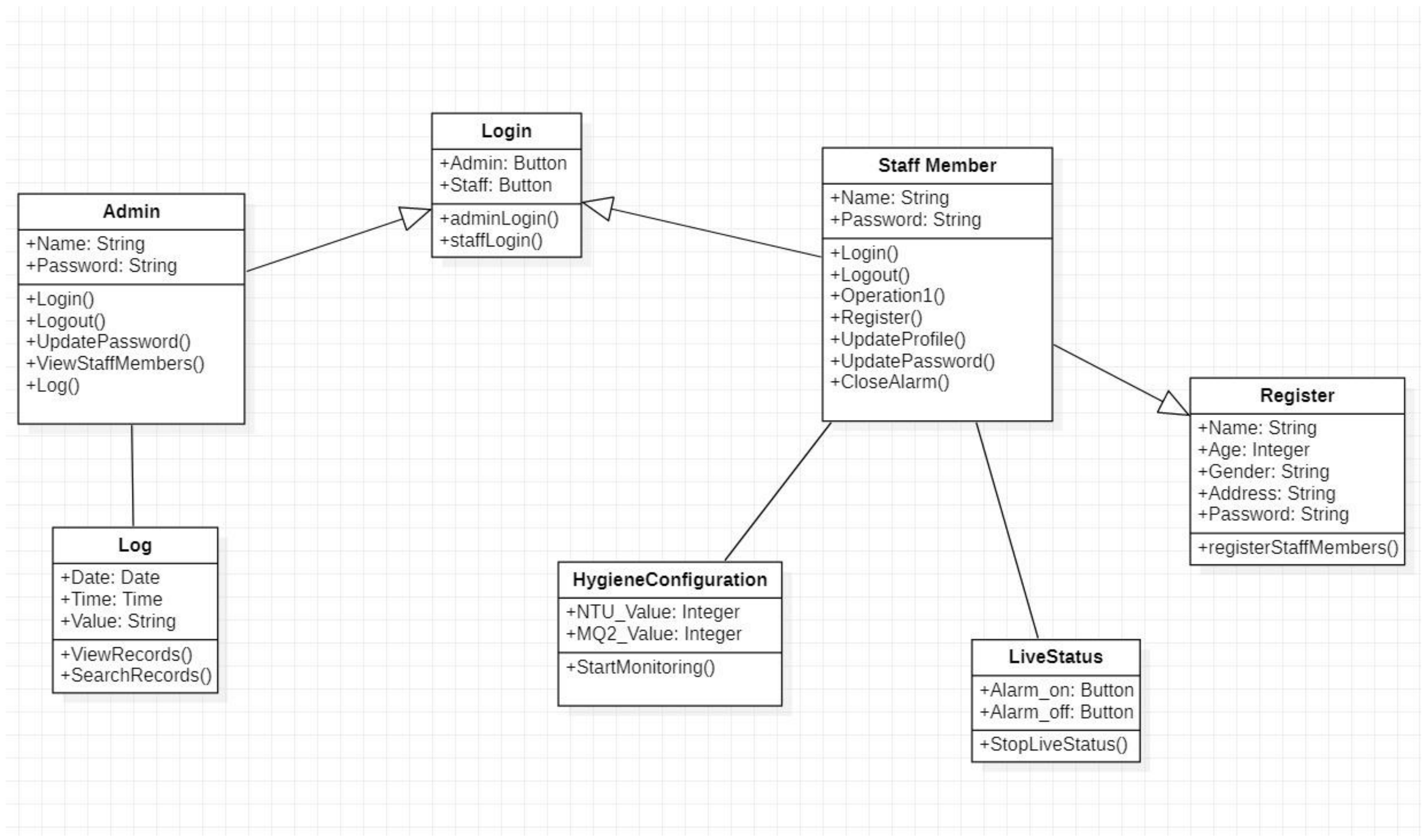
2. Admin login – In this page admin will login by putting their name password and id. And after that admin will set threshold parameters for sensors in the system. They can update their information and can also check the log.

3. Staff member login – In this page staff members will login by putting their name, password and admin id. They too can update their information and can also view the threshold parameters. And they will keep updating logs whenever they clean the toilet.

4. Hygiene System – Whenever the system gets the value above threshold then they will send an alarm to staff members and they will keep checking the status of the toilet and the sensor values.

5. Alarm – This alarm will ring directly to a staff member and then the staff member will turn it off and clean the toilet.

6. Log – In this after cleaning the toilet the staff member will update that activity here and it can be seen by the admin.



3.2 Sequence Diagram

Sequence diagram is one kind of interaction diagram, which shows an interaction among a set of objects and their relationships. The purpose of the Sequence diagram is to document the sequence of messages among objects in a time-based view. The scope of a typical sequence diagram includes all the message interactions for (part-of) a single use case. There may be multiple sequence diagrams per use case, one per use case scenario.

The objects in sequence diagrams are based on the class diagram from the software architecture view. The reason for doing that is we want to neither stay in the object construction view, in which the functions of objects are obscure and inadequate, nor go too further in the system architecture view, where many technical details obstruct a quick understanding of interaction among objects.

The state diagrams commonly contain:

● Objects

● Links

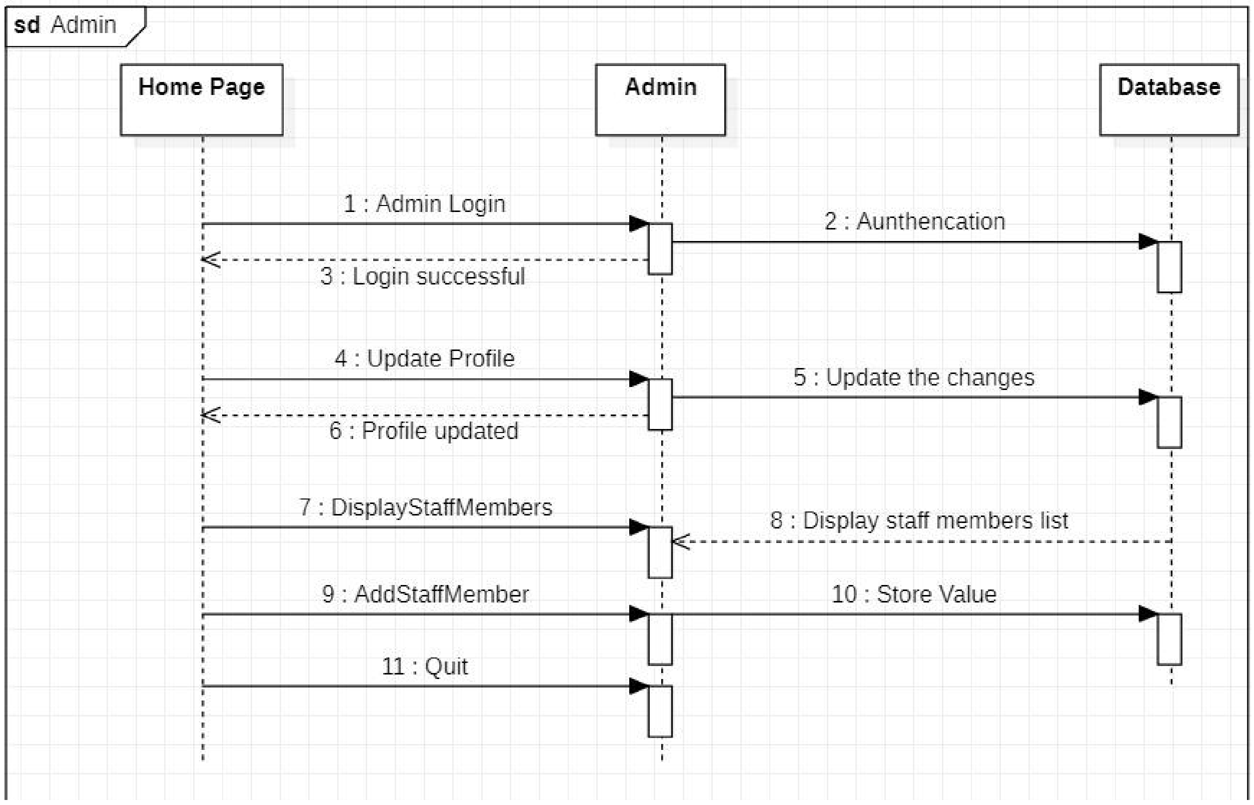
● Messages

● Response Time (especially useful in real-time systems)

Arrow line signifies there is a send message taking place. Response is being shown by dotted arrows.

3.2.1 Sequence Diagram on Admin

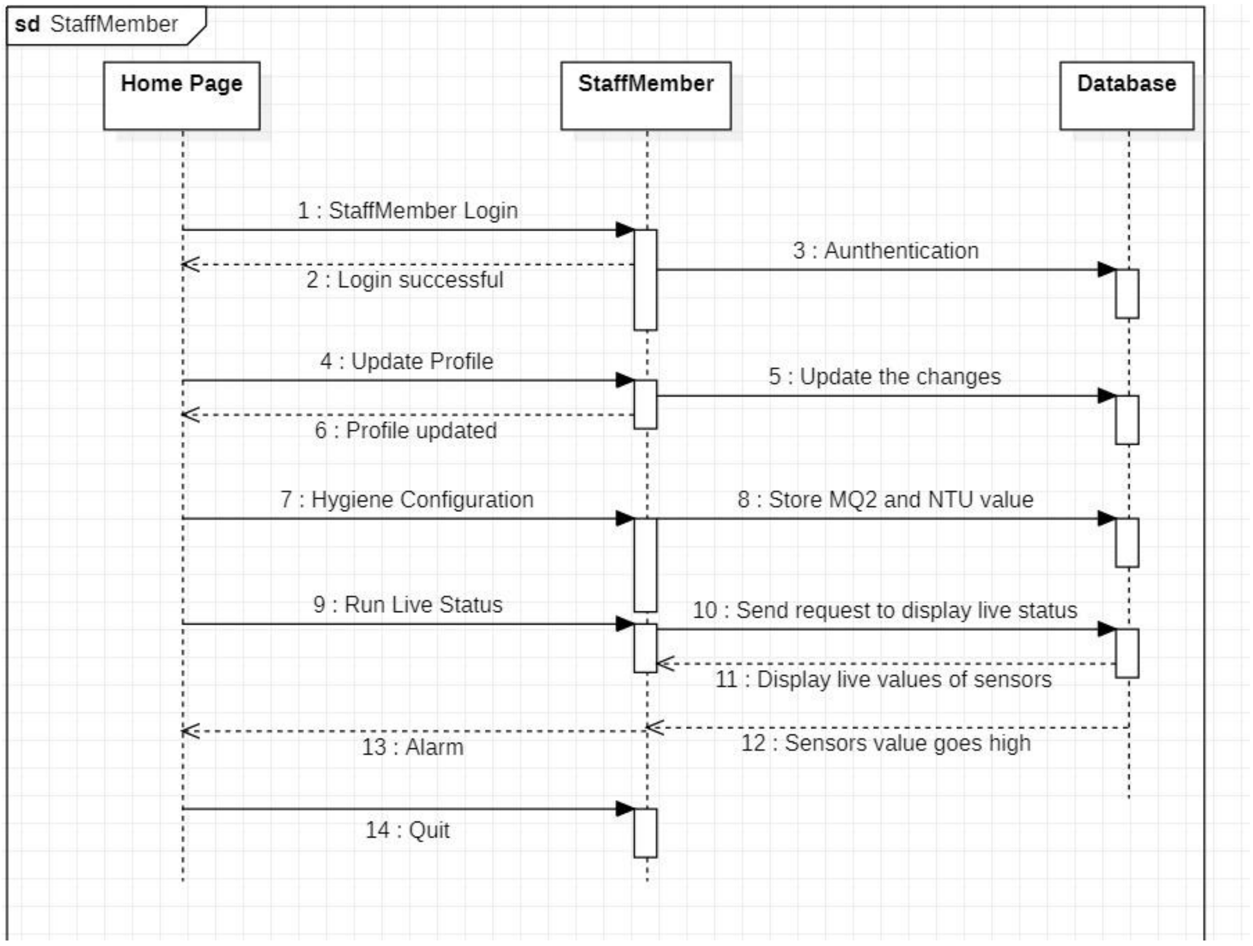
Admin logins to the system via password and details and is able to monitor the history of all processes and work of all staff members at various time stamps at various places and sets the threshold as per the requirement.



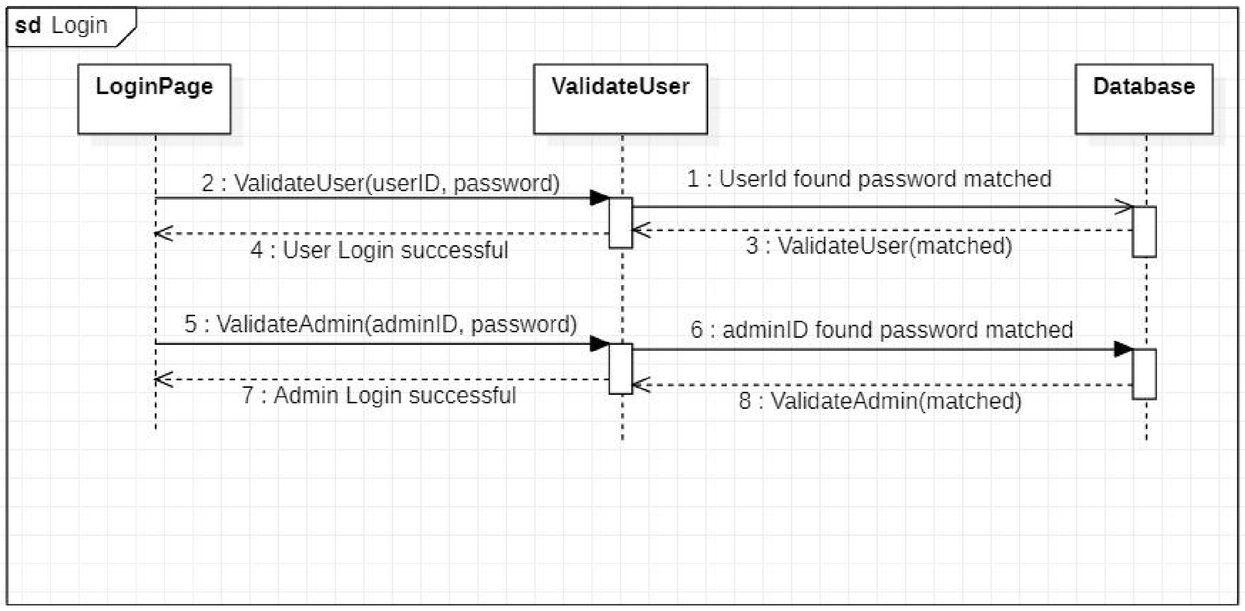
3.2.2 Sequence Diagram on Staff member

The Staff Member will first have to register himself on the system via login page.On passing the id and password verification,he will get the access to update his personal details,reset his password, view the threshold values set by admin.

Also, after cleaning the toilet unit,he will have to enter the time spent in cleaning in the log.After his service is over,he may log out of the system using quit option.



3.2.3 Sequence Diagram on Login Page


sd Login
LoginPage
ValidateUser
Database
2 : ValidateUser(userID, password)
1 : UserId found password matched
4 : User Login successful
3 : ValidateUser(matched)
5 : ValidateAdmin(adminID, password)
6 : adminID found password matched
7 : Admin Login successful
8 : ValidateAdmin(matched)

3.3 State Diagram

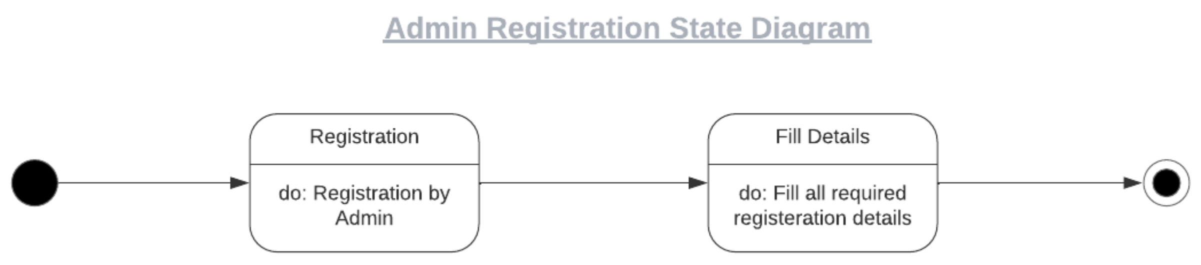
A State chart diagram shows a state machine. Usually the state machine in a state chart models the behavior of a reactive object, whose behavior is best characterized by its response to events dispatched from outside its context. The object has a clear lifetime whose current behavior is affected by its past. State chart diagrams are important for constructing executable systems through forward and reverse engineering.

State: A state represents the state of an object at a particular given point of time.

Transition: The transition from one state to another state of objects is represented by an arrow.

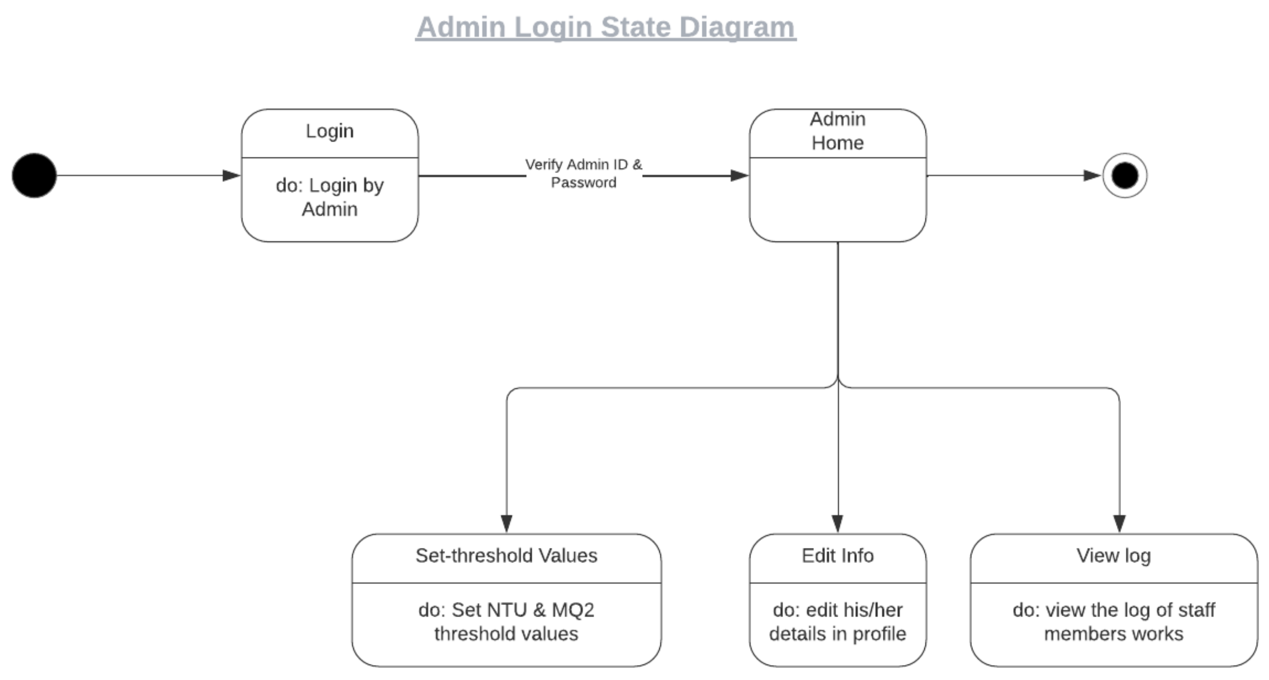
Event and Action*:* A trigger that causes a transition to occur.

3.3.1 State Diagram for Admin Registration


Admin Registration State Diagram
Registration
do: Registration by Admin
Fill Details
do: Fill all required registeration details

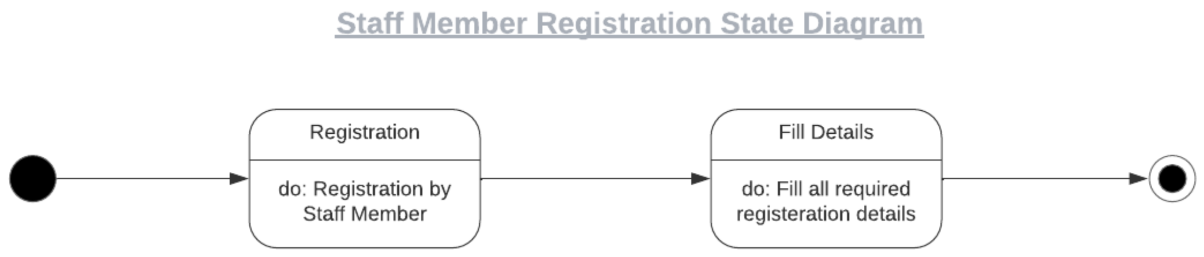
The admin will register himself/herself first and will provide all required following information.

3.3.2 State Diagram for Admin Login


Admin Login State Diagram
Login
do: Login by Admin
Verify Admin ID & Password
Admin Home
Set-threshold Values
do: Set NTU & MQ2 threshold values
Edit Info
do: edit his/her details in profile
View log
do: view the log of staff members works

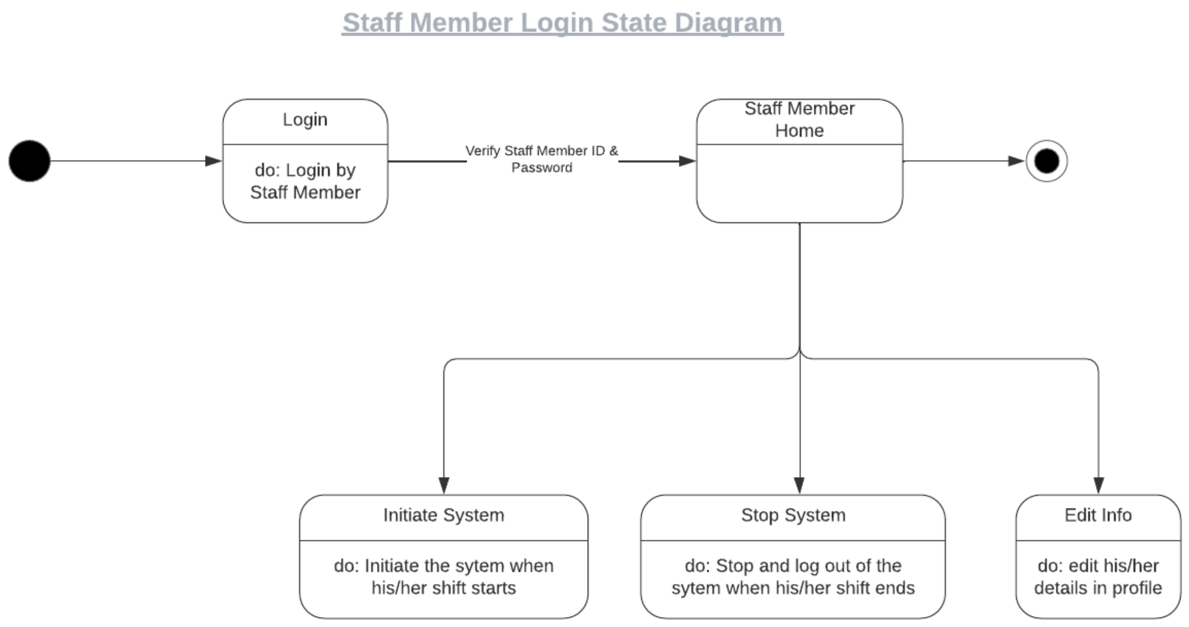
After the registration, the admin will login with required credentials, and can set the NTU & MQ2 threshold values. The admin can also check all the logs of staff workers working there.

3.3.3 State Diagram for Staff Member Registration


Staff Member Registration State Diagram
Registration
do: Registration by Staff Member
Fill Details
do: Fill all required registeration details

The staff members can also register in the same way as admin, by creating their profile and filling up their required information.

3.3.4 State Diagram for Staff Member Login


Staff Member Login State Diagram
Login
do: Login by Staff Member
Verify Staff Member ID & Password
Staff Member Home
Initiate System
do: Initiate the sytem when his/her shift starts
Stop System
do: Stop and log out of the sytem when his/her shift ends
Edit Info
do: edit his/her details in profile

The staff members' permissions include initiating the system when their shift starts and stopping the system when their shift is over. And all these actions will be recorded in a log file.

4. Execution Architecture

4.1 Topic

Any device running the Android operating system with the minimum version of Ice Cream Sandwich, as well as Android Studio as a development platform, are required for the runtime environment.

5. Design decisions and tradeoffs

5.1 Topic

To provide isolation, the design decision is to use two screens, one for admin and one for staff. It might have been possible to display all of the data on a single screen. The use of two screens, on the other hand, will provide a better interface and will be less crowded. The following will be the design:

- The system will compare the values of NTU and MQ2 after each use of the toilet by an external user with the threshold values set by system admin. In every case, depending on the level of odor and turbidity of the air inside the toilet, there will be two possible use cases.

- If the NTU or MQ2 values are within thresholds, then nothing happens and just levels get updated after the use of the toilet.

- If the NTU or MQ2 values go outside of the threshold values, then a notification will be sent to all active staff workers on their mails.

Using buttons instead of menu items as a tradeoff when considering links is a possibility. The use of buttons to navigate between screens is a design decision made to improve visibility. The text links in the menu bar at the bottom of the screen, on the other hand, are difficult to see. Using buttons with descriptive names instead of text links in the menu bar will make it easier to navigate from one screen to the next.

The user will be able to tell where he is navigating thanks to descriptive labels. The buttons on the PDA are larger than the text links in the menu bar. As a result, the user will have an easier time locating the mechanisms required to move from one screen to the next.