

**KGiSL INSTITUTE OF TECHNOLOGY**

(Approved By AICTE, New Delhi, Affiliate to Anna University

Recognized by UGC, Accredited by NBA(IT)

265, KGISL Campus, Thudiyalur Road, Saravanampatti, Coimbatore-641035**.)**

**DEPARTMENT OF**

**ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

**NAAN MUDHALVAN - INTERNET OF THINGS**

**Smart Public Restroom**

**NAME:** HARIPRAKASH.M

**REG NO:** 711721243306

**NM ID:** au21aia72

**TEAM MENTOR:** Mr**.** Mohankumar M

**TEAM EVALUATOR:** Ms. Akilandeeshwari M

**Phase 5: Project Documentation & Submission**

**DESIGN THE PLATFORM TO RECEIVE AND DISPLAY REAL-TIME RESTROOM AVAILABILITY AND CLEANLINESS DATA INVOLVES BOTH HARDWARE AND SOFTWARE COMPONENTS.**

**Project Objectives:**

Begin by describing the overall goals and objectives of your Smart Public Restrooms project.

What are you trying to achieve with this project?

**IoT Sensor Interaction**:

The code interacts with the IoT sensors to collect data from the restrooms. This interaction involves the following steps:

**Data Collection**:

The code running on the Raspberry Pi communicates with the IoT sensors to collect data. For example, occupancy sensors may send signals to indicate restroom occupancy, air quality sensors may provide data on the air conditions, and water usage sensors might report water consumption.

**Data Processing:**

The collected data from the sensors is processed by the code. This processing may involve filtering, aggregating, or transforming the raw sensor data into a format suitable for analysis and presentation.

**Data Transmission:**

Once processed, the code can transmit this data to a central data storage or a cloud platform. The data can be sent using various communication protocols (e.g., MQTT, HTTP, or custom protocols) to ensure it reaches the designated data storage securely.

**Real-Time Updates**:

The code ensures that data from the sensors is continuously updated in real-time. Users and facility managers can access the latest information on restroom conditions and occupancy through the mobile app.

**2. Raspberry Pi Interaction**:

The Raspberry Pi serves as a central hub in your system, and the code running on it interacts with both the sensors and the mobile app:

**Sensor Data Handling:** The Raspberry Pi receives data from the IoT sensors and processes it locally. It may apply logic to ensure data quality, handle edge cases, and filter out noise or incorrect readings.

**Data Integration**: The code on the Raspberry Pi integrates data from multiple sensors, creating a cohesive dataset that represents the overall status of the restroom. This integrated data is then sent to the cloud or central server.

**Communication with Mobile App**: The code on the Raspberry Pi provides an API (Application Programming Interface) for the mobile app to access the data. The mobile app communicates with the Raspberry Pi, sending requests for data and receiving real-time updates on restroom availability, cleanliness, and other relevant information.

**Control Logic**: The Raspberry Pi may also implement control logic for certain functions, such as turning on automated air fresheners or alerting maintenance staff when a restroom issue is detected.

**3. Mobile App Interaction:**

The mobile app interacts with both the Raspberry Pi and the cloud-based system where data is stored. Here's how the code in the mobile app interacts with the other components:

**User Interface**: The app provides a user-friendly interface that allows users to search for nearby restrooms, view real-time availability, and receive alerts and updates.

**Requesting Data**: The code in the app sends requests to the Raspberry Pi for real-time data, such as restroom occupancy, cleanliness ratings, and wait times.

**Data Display**: When the app receives data from the Raspberry Pi, it formats and displays this information to the user. This can include graphical representations, text-based information, and other visual cues.

**User Feedback**: The app allows users to provide feedback on restroom conditions, cleanliness, and other factors. The feedback is sent to the central system for analysis.

**Notifications:** The app can push notifications to users based on the real-time data it receives, such as alerts about restroom availability or notifications about maintenance issues.

**Location Services**: The mobile app may use location services to help users find the nearest restrooms within their vicinity.

The code in the mobile app, the Raspberry Pi, and the central data storage work together to provide a seamless and real-time experience for users and an efficient management system for facility operators. This interaction ensures that the data is collected, processed, and presented in a way that enhances the user experience and optimizes restroom management. cleanliness.

Discuss how the system aids restroom management by providing real-time data on restroom usage, maintenance needs, and resource optimization.

**Enhancing User Experience:**

Restroom Availability: Users can easily find and access available restrooms in their vicinity through the mobile app. This information is updated in real-time, ensuring that users don't waste time searching for restrooms that are already occupied**.**

**Wait Time Estimation:**

The system provides estimated wait times for each restroom. Users can make informed decisions about which restroom to use based on these estimates. This feature is particularly valuable during peak hours or in busy public places.

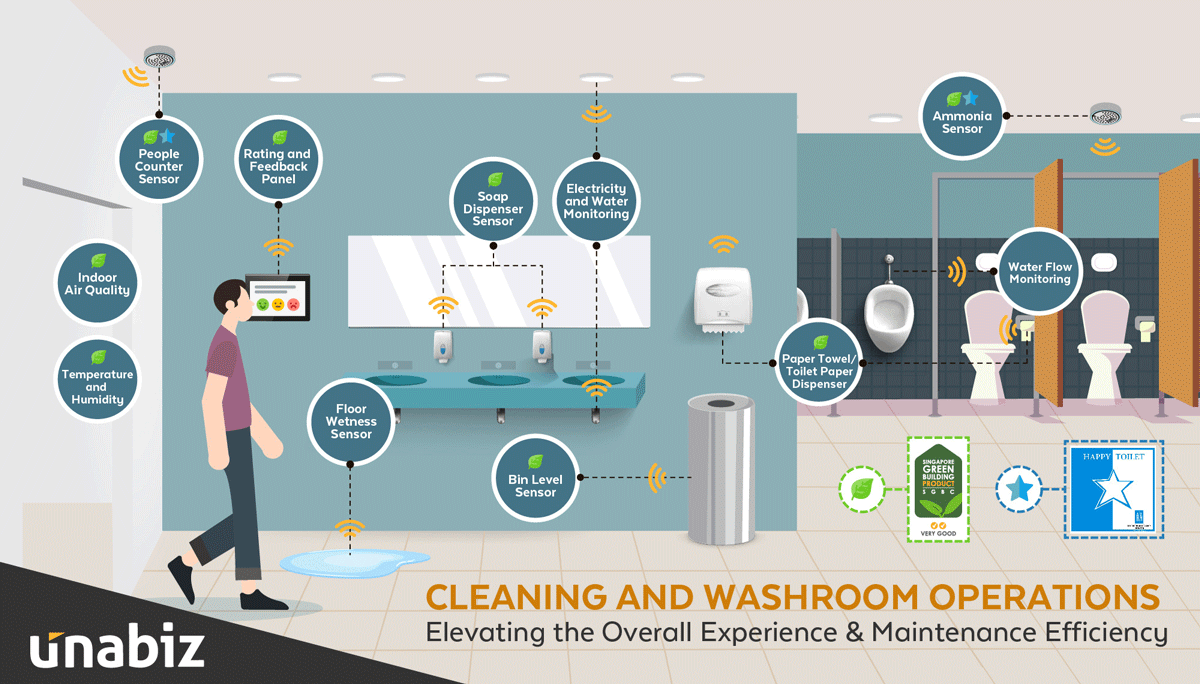
**Cleanliness Ratings**:

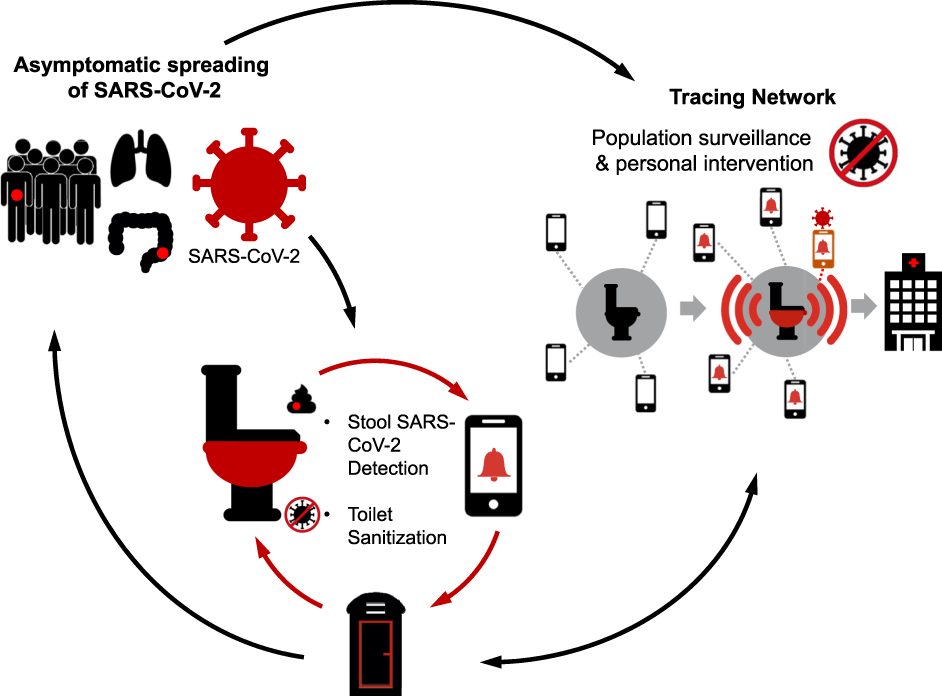
Users can rate and provide feedback on the cleanliness of restrooms they visit through the app. This crowdsourced data helps other users select cleaner and more hygienic restrooms, contributing to a better overall restroom experience.

**Accessibility Information**: The system can provide details on the accessibility features of restrooms, such as wheelchair-accessible stalls, baby changing stations, and gender-neutral options. This ensures that users with specific needs can find suitable facilities easily.

**Navigation Assistance:**

For large venues like shopping malls or airports, the app can provide navigation assistance, guiding users to the nearest available restroom. This feature reduces user frustration and ensures they reach the restroom quickly.



****

**Restroom Management:**

**Resource Optimization:** Real-time data on restroom occupancy and usage patterns enable facility managers to optimize their resources efficiently. Cleaning schedules, restocking of supplies, and maintenance activities can be aligned with actual usage, reducing operational costs.

**Maintenance Alerts**: IoT sensors can detect issues like clogged toilets, broken fixtures, or low supplies, and automatically alert maintenance staff. This leads to faster response times and ensures that restrooms are maintained in good condition**.**

**Usage Patterns Analysis**: The data collected from sensors and user interactions provide insights into restroom usage patterns. This information can help in staffing restrooms during peak hours and optimizing facilities based on real data.

**Customer Feedback:** The real-time restroom information system allows users to provide feedback on their restroom experience. Facility managers can use this feedback to address recurring issues, improve overall cleanliness, and adapt services based on user preferences.

**Data-Driven Decision-Making:** Facility managers can make data-driven decisions about restroom layouts, supply management, and facility design based on actual usage data. This ensures that resources are allocated efficiently to meet user demand.

**Environmental Sustainability:** The system can contribute to sustainability by triggering automated water-saving features and adjustments to HVAC systems when restrooms are not in use. This helps conserve resources and reduce the facility's environmental footprint.

**Compliance and Reporting:** The system can generate reports to monitor compliance with restroom regulations and accessibility requirements, ensuring that facilities are in accordance with relevant laws and guidelines**.**

In summary, the real-time restroom information system enhances the user experience by providing convenient and up-to-date information about restroom availability, cleanliness, and accessibility. Simultaneously, it improves restroom management by optimizing resources, streamlining maintenance, and enabling data-driven decision-making to create a more efficient and pleasant restroom experience for both users and facility operators.

**Raspberry Pi Data Transmission (Python):**

Here's a simple Python code snippet that simulates data transmission from a Raspberry Pi to a central server using HTTP requests. You'd need to adapt this to your specific IoT sensor setup and data handling.

import requests

import json

import time

# Simulated sensor data

sensor\_data = {

"restroom\_id": 1,

"occupancy": 3,

"air\_quality": "Good",

"water\_usage": 200,

}

# Central server URL

server\_url = "https://your-server.com/api/update"

# Periodically send sensor data

while True:

try:

# Convert sensor data to JSON

data = json.dumps(sensor\_data)

# Send data to the server

response = requests.post(server\_url, data)

if response.status\_code == 200:

print("Data sent successfully.")

else:

print("Failed to send data.")

# Adjust the data transmission interval

time.sleep(60) # Transmit data every 1 minute

except Exception as e:

print(f"An error occurred: {str(e)}")

**Example of Mobile App UI (simplified representation):**

Below is a simplified mockup of a mobile app interface for your "Smart Public Restrooms" project. Please note that this is a basic representation, and the actual UI design should consider user experience and aesthetics.

**Main Screen**:

Display a map showing nearby public restrooms.

List of nearby restrooms with their occupancy status and wait times.

Search bar to find restrooms in specific locations.

Filter options to sort restrooms by cleanliness or accessibility.

Main Screen

**Restroom Details:**

When a user selects a restroom, they can see detailed information.

Real-time occupancy status (e.g., 3 of 5 stalls occupied).

Estimated wait time (e.g., 5 minutes).

Cleanliness rating (e.g., 4.5 out of 5 stars).

Accessibility features (e.g., wheelchair-accessible, baby changing station).

Restroom Details

**User Feedback and Ratings**:

Users can provide feedback and rate the restroom.

Text input for feedback (e.g., "Restroom is clean and well-maintained.").

Star rating for cleanliness and accessibility.

User Feedback

**Maintenance Alerts**:

Maintenance staff receive alerts about restroom issues.

Notifications for issues like a clogged toilet, broken faucet, or low supplies.

Option to acknowledge and mark issues as resolved.

Maintenance Alerts

Remember, the actual mobile app's design and functionality can be much more sophisticated. These examples provide a basic idea of what your app's UI might look like and how users can interact with it to find and rate restrooms.

**Navigation Assistance:**

If your project includes navigation assistance, you can have a dedicated screen or feature that guides users to the selected restroom.

Show a map with directions from the user's current location to the chosen restroom.

Provide step-by-step directions and estimated arrival time.

Navigation Assistance

**User Profile**:

Allow users to create profiles and customize their settings.

Profile picture, name, and contact information.

Notification preferences for maintenance alerts or updates about their reviewed restrooms.

User Profile

**Search and Filters**:

A screen where users can search for restrooms by name, location, or specific features (e.g., baby changing stations).

Filters to sort restrooms by cleanliness, ratings, and accessibility features.

Search and Filters

**Accessibility Information**:

Provide a dedicated section with information about restroom accessibility features.

Details on wheelchair accessibility, braille signage, and more.

Icons and descriptions for each feature.

Accessibility Information

**Settings:**

A screen where users can customize app settings.

Options to set the preferred language, units (e.g., miles or kilometers), and notification preferences.

Settings

Keep in mind that the actual design and features of your mobile app will depend on your project's specific requirements, user needs, and the platforms you choose to develop for (iOS, Android, or both). User experience and interface design play a critical role in ensuring that your app is user-friendly and engaging.