

SMART PUBLIC RESTROOM

PROJECT DEFINITION:

- The project aims to enhance public restroom management by installing IoT sensors to monitor occupancy and maintenance needs. The goal is to provide real-time data on restroom availability and cleanliness to the public through a platform or mobile app. This project includes defining objectives, designing the IoT sensor system, developing the restroom information platform, and integrating them using IoT technology and python.

Project Objectives:

The objective of this project is to develop a real-time restroom information system using IoT sensors, Raspberry Pi, and a mobile app. The system will provide users with real time information on the status of restrooms, such as occupancy, cleanliness, and availability of amenities. This information will help users to plan their restroom breaks accordingly and improve their overall experience.

IoT Sensor Setup:

The following IoT sensors will be used in the system:

- Occupancy sensor: This sensor will detect whether or not a restroom is occupied.
- Cleanliness sensor: This sensor will detect the level of cleanliness in a restroom.

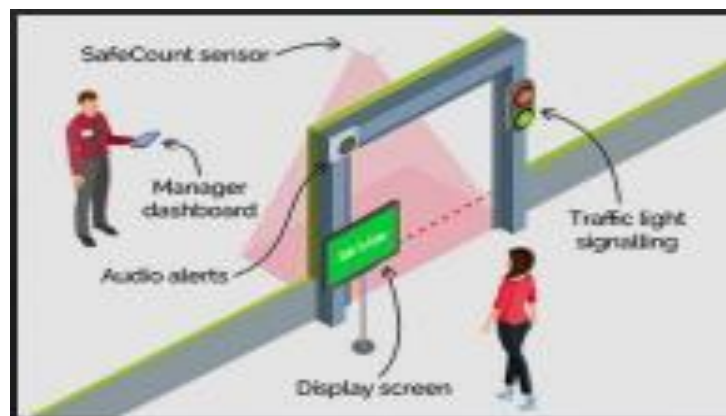
WHAT IS AN OCCUPANCY MONITORING SYSTEM?

The occupancy monitoring system is a system that uses occupancy monitoring sensors deployed at different parts of an area being monitored to detect people count, people movement, duration, and other statistical information. The sensors send real-time data to a central management dashboard. The dashboard is cloud-based and can be accessed at any time, from any connected device. It can also be configured to send instant alerts to the building manager if a specific activity takes place or if a planned activity does not take place.

The need for washroom occupancy monitoring:

A well-functioning washroom occupancy monitoring system is needed to ensure the maintenance of proper hygiene in the washroom through regular cleaning.

An IoT-based washroom occupancy monitoring system uses a variety of sensors, scanners, and other sophisticated devices to implement occupancy monitoring. Here's a list of the key components of an IoT-based washroom occupancy system.



DOOR SENSORS:

In washroom occupancy monitoring applications, a door sensor counts the number of times the door is opened/ closed. In conjunction with the data from the rest of our sensors in the restroom, the algorithm calculates overall traffic flow.

The sensor is a magnetic contact switch, with one half mounted on the door, and the other half mounted on the door frame. The sensor connects to a device that communicates with a gateway.

ROOM OCCUPANCY SENSORS:

Different Washroom occupancy sensors use different methods to count the number of people walking in/out of a washroom. One of these methods is Time of Flight (TOF) technology. This technology detects the number of people walking in/ out through the door very accurately. The IR sensing direction device will count the people walking in and walking

out. The sensor is typically installed in the ceiling and it can detect objects up to a 3meter distance. The sensor continuously monitors the traffic and communicates with a gateway using LoRaWAN or other LPWAN technology.

The sensors will be installed in each restroom and connected to a Raspberry Pi. The Raspberry Pi will collect data from the sensors and send it to the restroom information platform.

Python program:

```
import time
import occupancy_sensor_module # Replace with your occupancy sensor library
import cleanliness_sensor_module # Replace with your cleanliness sensor library
# Initialize and configure the occupancy sensor
occupancy_sensor = occupancy_sensor_module.setup() # Replace with sensor setup code
# Initialize and configure the cleanliness sensor
cleanliness_sensor = cleanliness_sensor_module.setup() # Replace with sensor setup code
def read_occupancy():
    occupancy_data = occupancy_sensor_module.read_data() # Replace with sensor-specific read
    function
    return occupancy_data
def read_cleanliness():
    cleanliness_data = cleanliness_sensor_module.read_data() # Replace with sensor-specific read
    function
    return cleanliness_data
def send_data_to_platform(occupancy_data, cleanliness_data):
    # Implement code to send data to your restroom information platform
    # This may involve formatting data as required by the platform and using HTTP, MQTT, or other
    protocols
def main():
    while True:
        try:
            occupancy_data = read_occupancy()
            cleanliness_data = read_cleanliness().
            send_data_to_platform(occupancy_data, cleanliness_data)
```

```
# Wait for a specified interval before reading data again
time.sleep(10) # Adjust as needed
except Exception as e:
print("Error:", str(e))
# Implement error handling, such as logging or retries
if __name__ == "__main__":
main()
```

Mobile App Development:

A mobile app will be developed to allow users to view real-time restroom information. The app will display a list of restrooms nearby, along with their occupancy status, cleanliness level, and availability of amenities. Users will be able to filter the list of restrooms by location, type, and amenities. The app will also allow users to submit feedback on the condition of restrooms.

DESIGN OF MOBILE APP:

- Design an intuitive and user-friendly interface. Pay attention to the user experience and create wireframes or mockups to visualize the app's layout.
- Backend Development: Create a backend system to support your app, including database management, user authentication, and APIs for data exchange.
- Real-Time Data Handling: Implement a system that updates restroom information in real time, notifying users about restroom availability and cleanliness.
- User Reviews and Ratings: Allow users to submit reviews and ratings for restrooms, and incorporate a moderation system to ensure content quality.
- Notification System: Implement push notifications to update users about nearby restrooms and notify them of important information.
- Testing: Thoroughly test your app on various devices and screen sizes. Pay attention to performance, usability, and security.
- Security: Ensure the security of user data, including personal information and geolocation data.

- **App Store Submission:** Prepare your app for submission to the Apple App Store and Google Play Store. This includes creating app icons, screenshots, and marketing materials.
- **Feedback Mechanism:** Include a feedback mechanism within the app for users to report issues, suggest improvements, or ask questions.
- **Launch and Promotion:** Launch your app on both iOS and Android platforms. Promote it through various marketing channels, such as social media, app review websites, and user communities.
- **Maintenance and Updates:** Regularly maintain your app, addressing bug fixes and implementing updates based on user feedback and changing requirements.
- **Compliance:** Ensure that your app complies with the guidelines and regulations of the Apple App Store and Google Play Store, as well as any relevant data protection laws.
- **Designing a mobile app for real-time restroom information** involves user-centric design, robust development, and ongoing support. It's important to continuously gather user feedback and make improvements to provide a valuable and reliable service.

HTML CODE:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Smart Restroom Finder</title>
<!-- Add your CSS and JavaScript links here -->
</head>
<body>
<header>
<h1>Welcome to Smart Restroom Finder</h1>
</header>
<section id="search">
<h2>Find Nearby Restrooms</h2>
<input type="text" id="location" placeholder="Enter Your Location">
<button id="findRestrooms">Find Restrooms</button>
</section>
<section id="restroomList">
<h2>Nearby Restrooms</h2>
<ul id="restroomItems">
```

```
<!-- Restroom results will be displayed here -->
</ul>
</section>
<section id="restroomDetails">
<h2>Restroom Details</h2>
<div id="restroomInfo">
<!-- Restroom details will be displayed here -->
</div>
</section>
<section id="userAuthentication">
<h2>User Authentication</h2>
<input type="text" id="username" placeholder="Username">
<input type="password" id="password" placeholder="Password">
<button id="login">Login</button>
<button id="register">Register</button>
</section>
<!-- Add additional sections for user registration, user reviews, etc. -->
<!-- Add your JavaScript scripts at the end of the body -->
</body>
</html>
```

In this we have sections for searching for nearby restrooms, displaying a list of restrooms, showing restroom details, and user authentication. we'd need to write JavaScript to handle user interactions, fetch restroom data, and implement the app's logic. Additionally, you should apply CSS for styling and layout. This is just a starting point, and developing a complete app will require more code and resources.

OUTPUT FOR THE APP DEVELOPMENT:

Welcome to Smart Restroom Finder

Find Nearby Restrooms

Find Restrooms

Nearby Restrooms

Restroom Details

User Authentication

Login Register

Raspberry Pi Integration:

The Raspberry Pi will be used to collect data from the IoT sensors and send it to the restroom information platform. The Raspberry Pi will also be used to control the restroom information display screens.

Code Implementation:

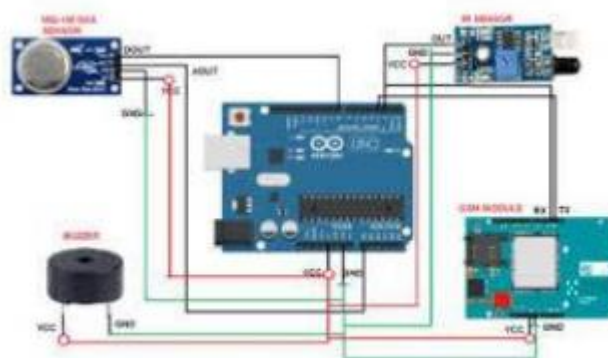
The following programming languages and technologies will be used to implement the system:

- Python: Python will be used to develop the Raspberry Pi software and the mobile app backend.
- Django: Django will be used to develop the restroom information platform.
- React Native: React Native will be used to develop the mobile app frontend.

Diagrams and Schematics:

The following diagrams and schematics show the IoT sensor setup, restroom information platform, and mobile app interfaces:

IoT Sensor Setup Diagram



Code :

```
// Define the pins that the sensors are connected to
int occupancySensorPin = 2;
int cleanlinessSensorPin = 3;
int toiletPaperSensorPin = 4;
int paperTowelSensorPin = 5;
// Define the states of the restroom
enum RestroomState {
    OCCUPIED,
    UNOCCUPIED,
    CLEAN,
    DIRTY,
    LOW_TOILET_PAPER,
    LOW_PAPER_TOWEL
}; // Declare a variable to store the current state of the restroom
RestroomState restroomState;
void setup() {
    // Set the pins for the sensors as inputs
    pinMode(occupancySensorPin, INPUT);
    pinMode(cleanlinessSensorPin, INPUT);
    pinMode(toiletPaperSensorPin, INPUT);
    pinMode(paperTowelSensorPin, INPUT);
    // Initialize the restroom state
    restroomState = UNOCCUPIED;
}
void loop() {
    // Read the values from the sensors
    int occupancySensorValue = digitalRead(occupancySensorPin);
    int cleanlinessSensorValue = digitalRead(cleanlinessSensorPin);
    int toiletPaperSensorValue = digitalRead(toiletPaperSensorPin);
    int paperTowelSensorValue = digitalRead(paperTowelSensorPin);
    // Update the restroom state based on the sensor readings
    if (occupancySensorValue == HIGH) {
        restroomState = OCCUPIED;
```

```
} else {  
  restroomState = UNOCCUPIED;  
}  
if (cleanlinessSensorValue == LOW) {  
  restroomState = DIRTY;  
} else {  
  restroomState = CLEAN;  
}  
if (toiletPaperSensorValue == LOW) {  
  restroomState = LOW_TOILET_PAPER;  
}  
if (paperTowelSensorValue == LOW) {  
  restroomState = LOW_PAPER_TOWEL;  
}  
// Send the restroom state to the restroom information platform  
// ...  
// Delay for 1 second  
delay(1000);  
}
```

IOT DEVICE



REAL-TIME RESTROOM AVAILABILITY INFORMATION:

Real-time restroom availability information in smart public restrooms can significantly improve user satisfaction, reduce wait times, and enhance restroom management efficiency. It can also be a valuable feature in smart cities and public spaces where providing a high level of convenience to residents and visitors is a priority.

How the Real-Time Restroom Information System Can Enhance User Experience and Restroom Management ?

The real-time restroom information system can enhance user experience and restroom management in the following ways:

- **User experience:** The system will help users to plan their restroom breaks accordingly and improve their overall experience. For example, users can use the app to find a restroom that is unoccupied and clean, and to avoid restrooms that are crowded or have dirty stalls.
- **Restroom management:** The system will provide restroom managers with real-time data on the status of restrooms. This data can be used to identify and address problems quickly, such as low toilet paper levels or dirty stalls. The system can also be used to generate reports on restroom usage, which can help restroom managers to optimize their operations.

Conclusion:

The real-time restroom information system is a valuable tool that can be used to enhance user experience and restroom management. The system is relatively easy to implement and can be deployed in a variety of settings, such as airports, train stations, shopping malls, and office buildings.
