SMART PARKING

IOT BASED SMART PARKING SYSTEM

- The project is a real world's application which can be incorporated to any parking management system. First of all, we will fix the proximity sensors as per the number of slots for parking. These sensors will tell whether any area is empty or not. This will be know to the authority in charge. It can have the report of it's complete management on tips. On the other hand, if the user wants to avail the service at XYZ parking area. Then he will first book his slot at the parking. However, this depends on the availability as well.
- Once the user books the slot with the date and time, he will be the owner of that area for the set period of time. To authenticate whether the user who is entering with the car has his pre booked slot or not, we will use RFID tags. These contain the information of the type of permit that the particular vehicle has been granted. The person enforcing it is equipped with a hand reader that captures the information of the vehicle, such as expiration time and corresponding spot location.

PROJECT OBJECTIVE:

- ❑ Real-Time Parking Space Management
- ❑ Mobile App Integration
- ❑ Efficient Parking Guidance

REAL TIME PARKING SPACE MANAGEMENT

Real-time parking space management for smart parking is a crucial component of modern urban planning and transportation systems. It involves the use of technology to efficiently manage and optimize the utilization of parking spaces in urban areas. Here's an overview of how real-time parking space management for smart parking works:

Sensor Technology: To monitor parking space availability in real-time, various types of sensors can be used. These sensors can include:

1. **Infrared Sensors**: These sensors detect the presence of a vehicle in a parking space by measuring changes in infrared radiation.
2. **Ultrasonic Sensors**: Ultrasonic sensors use sound waves to detect the presence or absence of vehicles in parking spaces.
3. **Magnetic Sensors**: Magnetic sensors can detect the presence of a vehicle by measuring disturbances in the Earth's magnetic field caused by the metal in the vehicle.
4. **Camera-Based Systems**: High-resolution cameras can be used to capture images of parking spaces and analyze them to determine occupancy.

- **Data Communication**: Data from these sensors is transmitted in real-time to a centralized parking management system. This can be achieved through wired or wireless communication networks, such as Wi-Fi, cellular networks, or LoRaWAN

- **Data Processing and Analysis**: The data collected from the sensors is processed and analyzed by the parking management system. Advanced algorithms can determine the occupancy status of each parking space, and this information is updated in real-time.

- **Interface**: A user-friendly interface is essential for both parking operators and drivers. Users can access real-time parking availability information through mobile apps, websites, or electronic signage located at key points in the city.

- **Reservation Systems**: In addition to real-time information, smart parking systems often provide the option for drivers to reserve parking spaces in advance, reducing the uncertainty of finding parking.

- **Navigation and Guidance**: Real-time parking systems can provide drivers with turn-by-turn navigation to available parking spaces. This not only saves time but also reduces traffic congestion as drivers no longer need to circle in search of parking.

MOBILE APP INTEGRATION

Integrating a mobile app into a smart parking system is crucial for providing a seamless and user-friendly experience for drivers. Below are the key steps and features involved in integrating a mobile app for smart parking:

- ✓ **Real-Time Parking Availability**: The core feature of the mobile app should be to provide real-time information on parking space availability. This information is typically obtained through sensors installed in parking spaces and transmitted to the app.
- ✓ **User Registration and Authentication**: Users should be able to create accounts and log in securely to the app. This allows for personalized features such as reservation history, payment methods, and preferences.

- **Search and Navigation**: The app should include a search functionality that allows users to find nearby parking spaces. Navigation features can guide drivers to the chosen parking spot using GPS and mapping services like Google Maps or Apple Maps.
- **Reservation and Booking**: Users should have the option to reserve parking spaces in advance through the app. This feature can include selecting a specific parking location, date, time, and duration of stay. Integration with payment gateways for booking fees is essential.
- **Payment Integration**: Secure payment options should be integrated into the app to enable users to pay for their parking reservations. This can include credit card, mobile wallets, or even integration with parking passes and subscriptions.
- **Notifications and Alerts**: The app can send push notifications to users regarding their reservations, upcoming expirations, and important information related to the parking facility.

EFFICIENT PARKING GUIDANCE

1. **Communication Infrastructure**:
    1. Establish a robust communication infrastructure to transmit real-time parking data to a centralized system.
    2. Utilize wireless technologies like Wi-Fi, Bluetooth, or cellular networks for data transmission.
    3. Ensure redundancy and reliability in data communication to prevent system failures.
2. **User-Friendly Mobile Apps and Signage**:
    1. Develop mobile apps that allow users to access real-time parking availability information, make reservations, and navigate to available parking spaces.
    2. Install electronic signage at key points in the city to display real-time parking information and directions to available parking facilities.
3. **Navigation and Routing**:
    1. Provide turn-by-turn navigation within the mobile app to guide drivers to available parking spaces.
    2. Consider integration with popular navigation apps (e.g., Google Maps, Apple Maps) for seamless routing.

4.Dynamic Pricing and Payment Integration:

1. Implement dynamic pricing strategies that adjust parking fees based on demand, time of day, location, and other factors.
2. Integrate secure payment gateways within the app to facilitate cashless payments.

5.Reservation System:

1. Offer users the option to reserve parking spaces in advance through the mobile app.
2. Ensure that reserved spaces are clearly marked and accessible upon arrival.

6.Parking Garage Guidance:

1. Implement guidance systems within multi-level parking garages using LED lights, digital signage, or floor markings to indicate available spaces.
2. Ensure clear and intuitive directional signage to guide drivers efficiently.

IOT SENSOR DESIGN

- IoT (Internet of Things) sensors play a vital role in creating smart parking solutions by providing real-time data on parking space occupancy and availability. These sensors are designed to detect the presence or absence of vehicles in parking spaces and communicate this information to a central system or mobile app. Here are some common types of IoT sensors used for smart parking:

1. **Ultrasonic Sensors**:
    1. Ultrasonic sensors use sound waves to detect the presence of vehicles. They emit high-frequency sound waves and measure the time it takes for the waves to bounce back after hitting an object (a vehicle in this case).
    2. These sensors are typically installed above each parking space or on light poles above a group of spaces.
2. **Infrared Sensors**:
    1. Infrared sensors detect the heat emitted by vehicles. When a vehicle enters a parking space, it disrupts the infrared radiation pattern, triggering the sensor.
    2. These sensors are often installed on the pavement or mounted on nearby structures.

3. Magnetic Sensors:

1. Magnetic sensors use changes in the Earth's magnetic field caused by the presence of a vehicle to detect occupancy.
2. They are installed in or under the pavement of each parking space.

4. Surface-Installed Sensors:

1. Some sensors are designed to be mounted on the surface of the pavement and use various technologies, such as pressure-sensitive mats, to detect vehicle presence.
2. They are relatively easy to install and maintain.

5. Camera-Based Systems:

1. Camera-based systems use image recognition technology to analyze camera feeds and identify open parking spaces.
2. These systems are often used in conjunction with machine learning algorithms to improve accuracy.

flush mount & surface mount

Designed to fit every parking detection environment

Nwave wireless vehicle sensors streamline parking detection to fully utilize parking spaces and drive profit. Nwave is the smart parking sensor trusted by leading enterprises, municipalities and university campuses worldwide.

- **High Accuracy**
- Advanced filtering and noise reduction techniques allow differentiating parking events from electromagnetic interference or false events such as an underground train passing.
- **Easy Installation**
- Each parking sensor takes only 30 seconds to affix, with no extra wires or other street furniture required. The network is easily connected to Nwave's cloud-based app or end-user software of choice.
- **Long Life**
- Nwave sensors boast especial ruggedness, high load and damage resistance, an unbeatably high battery life of up to 10 years, with no need for service until then battery change is required. These features lead to quick parking detection.
- **LoRaWAN Support**
- Allows your project to leverage existing LoRaWAN infrastructure. Nwave sensors successfully tested in all major LoRaWAN regions (US, EU, Australia, Asia) and with all major gateway and LNS providers (TTN, Kerling, Actility, et.al.)

REAL-TIME TRANSIT INFORMATION PLATFORM

Designing a mobile app interface to display real-time parking availability to users requires careful consideration of user experience, readability, and ease of use. Here's a basic outline of how the interface might look:

1. Landing Page:

- The landing page should be clean and visually appealing, with a user-friendly interface.
- Include the app's logo and a simple, welcoming message.
- Provide the option for users to log in or continue as a guest.

- **2. Map View:**
- The main screen of the app should display a map of the user's current location.
- Mark nearby parking facilities with pins or icons.
- Use color-coding to indicate parking availability (e.g., green for available, red for full).
- Include an option to search for parking at a specific location (e.g., by address or landmark).
- **3. Filters and Sorting:**
- Allow users to filter parking options by criteria such as price, distance, and type (e.g., street parking, garages, lots).
- Provide sorting options (e.g., by distance, price, availability) to help users find the most suitable parking.
- **4. Parking Details:**
- When a user taps on a parking facility on the map, a pop-up or a new screen should display more information about that facility.
- Include the facility name, address, pricing details, and the number of available spaces.
- Add a "Get Directions" button that integrates with navigation apps

- **5. Search Bar:**
- Place a search bar at the top of the map view for users to manually search for parking at specific locations.
- Include auto-suggestions as users type.
- **6. Menu and Navigation:**
- Use a navigation drawer or tabs at the bottom for easy access to various app sections, such as Home, Favorites, Reservations, and Account.
- Include a "Nearby Events" or "Local Attractions" section that users can explore while searching for parking.
- **7. User Account and Settings:**
- Allow users to create accounts or log in to access additional features like booking reservations and saving favorite parking spots.
- Provide options to customize user profiles and set notification preferences.

- **8. Real-Time Updates:**
- Ensure that parking availability updates in real-time as users interact with the map.
- Display real-time notifications or alerts for significant changes in parking availability or important updates.
- **9. Favorites and Reservations:**
- Allow users to save their favorite parking spots and view them easily on a dedicated screen.
- Provide the ability to reserve parking spots directly within the app.
- **10. Feedback and Help:** - Include a section for users to provide feedback, report issues, or seek assistance. - Provide contact information for customer support.
- **11. Accessibility and Multilingual Support:** - Ensure that the app is accessible to users with disabilities. - Offer content and support in multiple languages if applicable.

- **12. Security and Privacy:** - Display clear privacy and data usa policies. - Implement security measures to protect user data transactions.
- **13. App Logo and Branding:** - Use a memorable app logo and consistent branding throughout the interface.
- **14. App Ratings and Social Sharing:** - Encourage users to rate and share their experiences on social media platforms.
- **15. Logout Option:** - Provide a clear option for users to log o accounts if they wish.

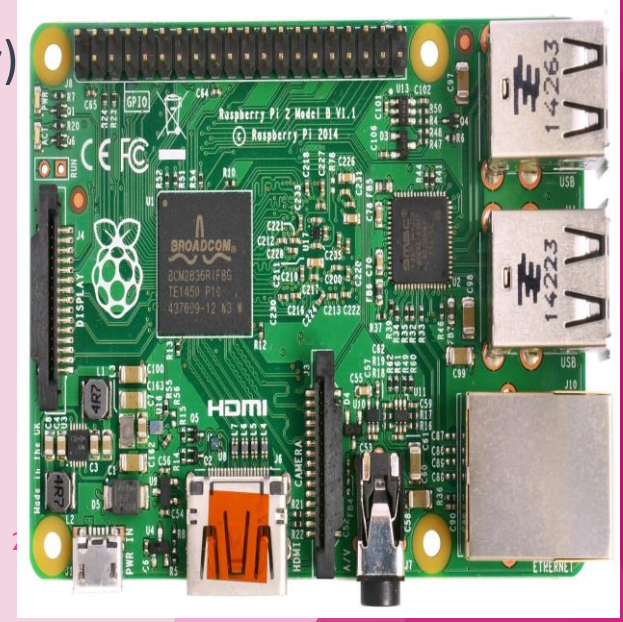
INTEGRATED APPROACH

To collect data from sensors using a Raspberry Pi and update a mobile app, you'll need to set up the Raspberry Pi as a sensor data collector and implement a communication mechanism between the Raspberry Pi and the mobile app. Here's a step-by-step guide on how to achieve this:

Hardware and Software Requirements:

1. **Raspberry Pi**: You'll need a Raspberry Pi board (e.g., Raspberry Pi 3 or 4) with an operating system (e.g., Raspbian) installed.

2. Sensors: Connect the sensors (e.g., ultrasonic, infrared, temperature, humidity) to the Raspberry Pi using appropriate GPIO pins or interfaces. Ensure that you have the necessary libraries or drivers to interact with the sensors.



3. **Mobile App**: Develop a mobile app for iOS or Android using a suitable framework (e.g., React Native, Flutter, or native development tools). Implement the app's user interface and functionality, including data visualization.

4. **Communication Protocol**: Decide on a communication protocol to transmit sensor data from the Raspberry Pi to the mobile app. MQTT (Message Queuing Telemetry Transport) and RESTful APIs are common choices.

- **Steps to Collect Data and Update the Mobile App:**

1. **Sensor Data Collection**:
    1. Write Python scripts on the Raspberry Pi to collect data from the connected sensors. Depending on the sensors used, you'll need to read values periodically (e.g., temperature and humidity) or in response to events (e.g., motion detection).
    2. Store the collected data in variables or data structures within your Python script.
2. **Data Processing**:
    1. Process and format the collected sensor data as needed. You may need to convert sensor readings into meaningful values (e.g., temperature in degrees Celsius) and prepare the data for transmission.

3. Communication with the Mobile App:

1. Decide on the communication method between the Raspberry Pi and the mobile app. MQTT is a lightweight messaging protocol suitable for real-time data updates. Alternatively, you can expose a RESTful API on the Raspberry Pi to receive HTTP requests from the mobile app.

4. MQTT Communication (Option 1):

1. If using MQTT, install an MQTT broker (e.g., Mosquitto) on the Raspberry Pi.
2. In your Python script, use an MQTT client library (e.g., Paho MQTT) to publish sensor data to MQTT topics.
3. In the mobile app, implement an MQTT client to subscribe to the same MQTT topics and receive real-time updates.

5. RESTful API Communication (Option 2):

1. If using a RESTful API, create an API on the Raspberry Pi using a web framework like Flask or Django.
2. Implement endpoints on the Raspberry Pi API to receive sensor data updates from the Python script and provide access to that data.
3. In the mobile app, use HTTP requests (e.g., GET or POST) to send data to and retrieve data from the Raspberry Pi API.

6. **Mobile App Integration**:
   - In the mobile app, implement logic to make requests (MQTT or HTTP) to the Raspberry Pi to retrieve sensor data.
   - Update the app's user interface to display the real-time sensor data using appropriate widgets (e.g., charts, text fields).
   - Implement error handling and data parsing in the app to ensure smooth data retrieval and display.
7. **Testing and Deployment**:
   - Test the entire system by running the Raspberry Pi script and launching the mobile app on a test device.
   - Debug and refine the system as needed.
   - Once tested successfully, deploy the Raspberry Pi and mobile app in your target environment.
8. **Security Considerations**:
   - Implement security measures, such as encryption and authentication, to protect data transmission between the Raspberry Pi and the mobile app.

By following these steps, you can collect data from sensors using a Raspberry Pi and update a mobile app with real-time sensor data. The choice of communication method (MQTT or RESTful API) depends on your specific project requirements and constraints.

BENEFITS OF SMART PARKING

- Smart parking systems offer numerous benefits for both city planners and citizens. These benefits contribute to improved traffic management, reduced environmental impact, and enhanced overall quality of life. Here are some of the key advantages of smart parking:

1. **Reduced Traffic Congestion**:
    1. Smart parking systems provide real-time information on available parking spaces, helping drivers find parking quickly. This reduces the need to circle in search of parking, which, in turn, lowers traffic congestion and reduces fuel consumption.
2. **Time Savings**:
    1. Drivers spend less time searching for parking, resulting in shorter commutes and reduced stress levels. This can improve overall quality of life and productivity.
3. **Lower Emissions**:
    1. Reduced congestion and idling result in lower greenhouse gas emissions. Smart parking contributes to a more environmentally friendly urban environment.

4. Revenue Generation:

1. Cities and parking operators can generate revenue through dynamic pricing models, where parking rates are adjusted based on demand. Additionally, data collected from smart parking systems can be monetized through partnerships or advertising.

5. Enhanced Safety:

1. Smart parking systems often include security features such as video surveillance, well-lit areas, and emergency call buttons, which improve the safety of parking facilities.

6. Improved Accessibility:

1. Accessible parking spaces for individuals with disabilities can be better managed, ensuring that they are available when needed.

7. Enhanced User Experience:

1. Users can access real-time parking information via mobile apps or electronic signs, making their parking experience more convenient and user-friendly.**Efficient Enforcement**:
2. Parking enforcement becomes more efficient with real-time data. Authorities can enforce parking regulations more effectively, reducing illegal parking.

THANK YOU