Start building the IoT sensor system and Raspberry Pi integration.

Configure IoT sensors (e.g., ultrasonic sensors) to detect parking space occupancy.

Write Python scripts on Raspberry Pi to collect data from sensors and send it to the cloud or mobile app server.

Building an IoT sensor system with Raspberry Pi integration to detect parking space occupancy involves several steps. Here's a high-level overview:

1. \*\*Hardware Setup:\*\*

- Acquire the required hardware, including Raspberry Pi, ultrasonic sensors, wires, and power sources.

- Connect the ultrasonic sensors to the Raspberry Pi. Typically, ultrasonic sensors have trigger and echo pins that need to be wired to GPIO pins on the Raspberry Pi.

2. \*\*Install Raspberry Pi OS:\*\*

- Install a suitable Raspberry Pi OS (e.g., Raspberry Pi OS, Raspbian) on your Raspberry Pi if not already installed.

3. \*\*Python Installation:\*\*

- Ensure Python is installed on your Raspberry Pi. Most Raspberry Pi OS distributions come with Python pre-installed.

4. \*\*Sensor Data Collection:\*\*

- Write Python scripts to interface with the ultrasonic sensors. You can use libraries like RPi.GPIO to interact with GPIO pins.

- Implement code to trigger the ultrasonic sensor and measure the time it takes for the signal to bounce back. This time can be used to calculate the distance to the nearest object.

5. \*\*Data Processing:\*\*

- Process the sensor data as needed. For parking space occupancy detection, you might need to set a threshold distance value that indicates whether a space is occupied or vacant.

6. \*\*Cloud Integration:\*\*

- Choose a cloud platform (e.g., AWS, Azure, Google Cloud) or a mobile app server to send your data.

- Write Python code to send data from your Raspberry Pi to the cloud. This may involve using MQTT, HTTP requests, or other communication protocols.

7. \*\*Mobile App Integration (Optional):\*\*

- If you want to display the occupancy data on a mobile app, create a mobile app or integrate with an existing one. Use relevant APIs or protocols for communication between the app and the Raspberry Pi.

8. \*\*Data Visualization:\*\*

- On the cloud or mobile app server, set up data storage and visualization tools to display parking space occupancy data in a user-friendly format.

9. \*\*Testing and Debugging:\*\*

- Thoroughly test the system to ensure it accurately detects parking space occupancy and successfully sends data to the chosen platform.

10. \*\*Monitoring and Maintenance:\*\*

- Implement monitoring and error-handling mechanisms to ensure the system's reliability.

- Regularly maintain and update the system as needed.

Remember to consult the datasheets and documentation for your ultrasonic sensors, as they may have specific requirements and usage instructions. Additionally, make sure to handle data privacy and security considerations, especially if you're collecting and transmitting data to the cloud or a mobile app.