Problem definition:

The problem we want to address is the inefficiency and congestion in urban traffic management systems.The traffic jams,accidents and delay not only lead to frustration amoung commuters but also have environmental and economic consequences.To tackle this problem, we aim to design an IOT-based traffic management system.

Design thinking process:

1.EMPATHIZES:

- Conduct survey,interviews and observations to understand the pain points of commuters,traffic authorities and the environmental impact of traffic congestions.
- Identify common issues such as traffic jams,accidents lack of real-time informations and inefficient signal timings.

2.DEFINE:

- Clearly define the problem,taking into account the insights gained during the empathizes stage.
- The problem definition may be: “How might we create a traffic management system that reduces congestion,improves the safety,and enhance the overall commuting experience in urban areas?”

3.IDEATE:

- Brainstorm solution and ideas to address the defined problem.consider IOT technologies that can provide real-time data and control mechanisms.
- Ideas could include smart traffic signals,vehicle-to-infrastructures communication,predictive analytics for traffic patterns and real-time navigation apps.

4.PROTOTYPE:

- Create a prototype or mockup of the proposed IOT based traffic management system.
- Test the prototype in a controlled environment to ensure that it functions as intended.

5.IMPLEMENT:

- Scale up the IOT based traffic management system to cover larger urban areas.
- Collaborate with local government authorities and transportation agencies for a city-wide implementation.

6.EVALUATE:

- Continuously monitor and evaluate the system’s performance.
- Collect data on traffic flow,congestion reduction,accident prevention and commuter satisfaction.

RASPBERRY PI CODING:

```
import time
import RPi.GPIO as GPIO

# Initialize GPIO pins for sensors and actuators
sensor_pin = 17  # Example GPIO pin for a vehicle presence sensor
LED_pin = 18     # Example GPIO pin for a traffic signal LED

GPIO.setmode(GPIO.BCM)
GPIO.setup(sensor_pin, GPIO.IN)
GPIO.setup(LED_pin, GPIO.OUT)

def detect_vehicle():
    return GPIO.input(sensor_pin) == GPIO.HIGH

def control_traffic_signal(traffic_signal_state):
    GPIO.output(LED_pin, traffic_signal_state)

try:
    while True:
        if detect_vehicle():
            print("Vehicle detected!")
            control_traffic_signal(GPIO.HIGH)  # Turn on the traffic signal
        else:
            print("No vehicle detected.")
            control_traffic_signal(GPIO.LOW)   # Turn off the traffic signal
        time.sleep(1)  # Check the sensor every 1 second

except KeyboardInterrupt:
    GPIO.cleanup()  # Clean up GPIO pins on program exit
```