JAI SHRIRAM ENGINEERING COLLEGE
TIRUPPUR – 638 660
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai
Recognized by UGC & Accredited by NAACandNBA (CSE and ECE)

DEPARTMENT OF

ELECTRONICS AND COMMUNICATION ENGINEERING

IBM - Naan Mudhalvan

Internet of Things-Group 3

Phase 3- Project Submission Part 3

NAME : R.SURYA

NM ID :AU711221106303

YEAR : III

Smart parking

Development of my project with Requirements technology wise:

AI&DS:

AI and Data Science (DS) play a crucial role in smart parking systems. These technologies enable the efficient management of parking spaces by collecting, analyzing, and utilizing data

Real-time Monitoring: AI and Data Science are used to monitor parking spaces in real-time. Sensors and cameras collect data on occupancy, enabling the system to provide accurate information to drivers.

Predictive Analytics: AI algorithms can predict parking space availability based on historical data, day of the week, and other factors. This helps drivers plan their parking in advance.

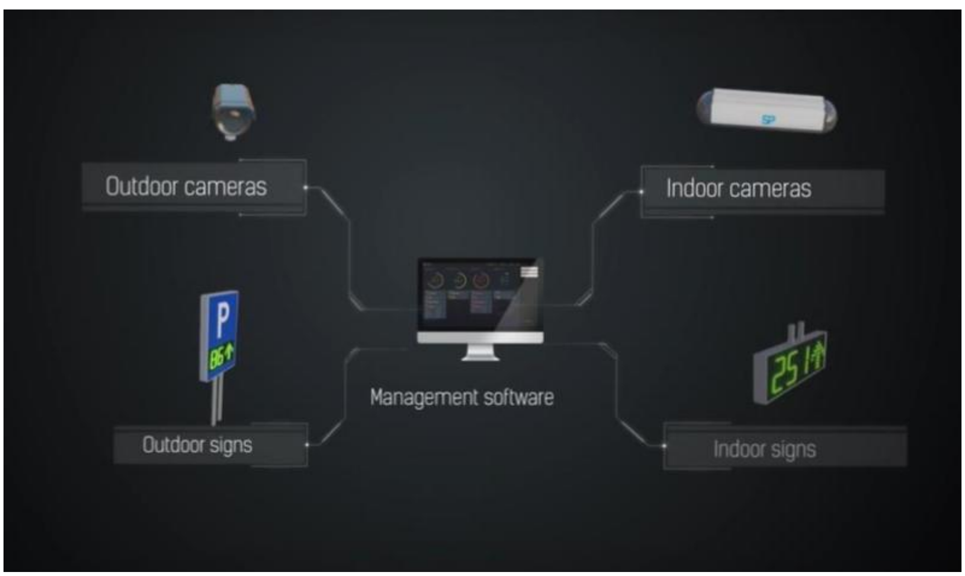
Traffic Optimization: AI can optimize traffic flow by guiding drivers to available parking spaces, reducing congestion and emissions in urban areas.

User Convenience: Smart parking apps use AI to provide users with information on nearby parking options, rates, and directions, making the parking experience more convenient.

Revenue Generation: Data Science is used to analyze historical parking data, helping cities and businesses optimize pricing and maximize revenue from parking facilities.

Security: AI can enhance security by monitoring parking areas and detecting suspicious activities through video analysis.

Sustainability: Smart parking systems can reduce fuel consumption and emissions by minimizing the time spent searching for parking spots.



In summary, AI and Data Science are integral to the functionality of smart parking systems, improving convenience for drivers, optimizing traffic, and helping businesses and cities manage parking resources efficiently

DAC:

A DAC (Data Access Control) system in smart parking refers to the management of access to data related to parking facilities and operations. It involves controlling who can access and modify data within the smart parking system. This can have various applications, such as:

User Access Control: DAC can regulate who can access information about available parking spaces, pricing, and reservations. Users may include drivers searching for parking spots or operators of parking facilities.

Security: It can enhance security by ensuring that only authorized personnel can access data related to parking operations. For example, only authorized administrators should be able to change pricing information or access security footage.

Payment and Billing: DAC can manage access to payment and billing data, ensuring that only authorized personnel can handle financial transactions and monitor revenue.

Operational Control: It can restrict access to data related to parking facility operations, like sensor data, maintenance logs, and occupancy rates.

In a smart parking context, a well-implemented DAC system is crucial to maintain data integrity, security, and overall system functionality. It typically involves user authentication, authorization, and auditing to ensure that data is handled responsibly and in compliance with privacy and security regulations.

IOT:

IoT (Internet of Things) in smart parking refers to the use of connected devices and technology to improve the management and efficiency of parking spaces. Here are some key aspects of IoT in smart parking:

Sensors: IoT-enabled parking systems often use sensors embedded in parking spaces to detect the presence or absence of vehicles. These sensors can be ultrasonic, infrared, or magnetic, and they provide real-time data about parking space occupancy.

Data Collection: Data from these sensors is collected and transmitted to a central server or cloud-based platform, where it can be analyzed and processed. This data includes information about available parking spaces, occupancy levels, and historical usage patterns.

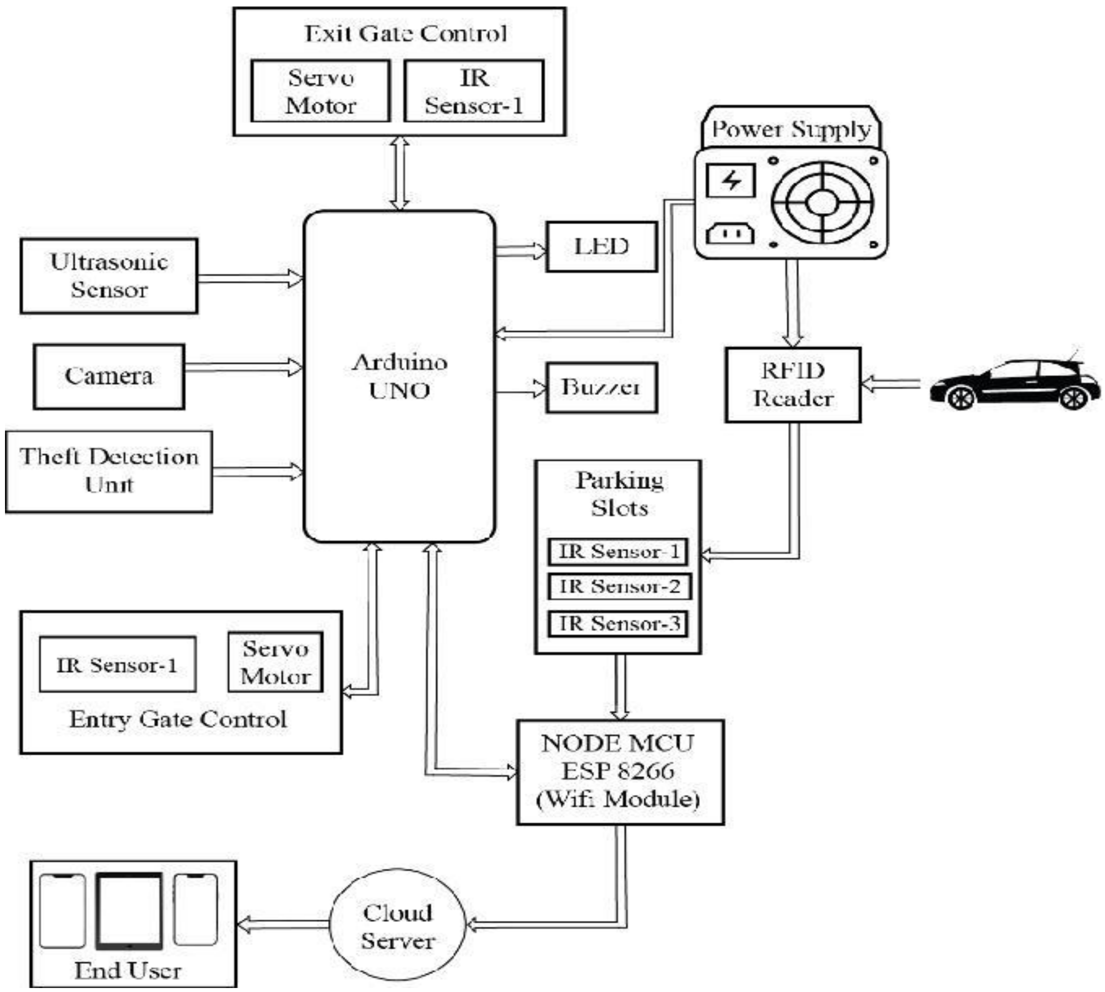
Smartphone Apps: Many smart parking solutions offer smartphone apps that allow users to find available parking spaces, reserve spots in advance, and make payments. This enhances the user experience and reduces the time and stress of finding parking.

Sustainability: Smart parking systems can contribute to sustainability efforts by reducing unnecessary circling for parking, which can lower emissions and fuel consumption.

Security: IoT technology can enhance the security of parking areas through surveillance cameras and automated alerts for suspicious activities.

Maintenance: Sensors can also monitor the condition of parking infrastructure, such as parking meters or barriers, and alert maintenance teams when repairs or maintenance are needed.

IoT in smart parking is a practical application of technology that can significantly improve urban mobility, reduce congestion, and enhance the overall parking experience for users.



CAD :

Computer-Aided Design (CAD) can be used in the development of smart parking systems in several ways:

Site Planning: CAD software can be employed to design the layout of parking areas, including the arrangement of parking spaces, entry and exit points, and traffic flow within the parking facility. This ensures efficient space utilization and user-friendly design.

Sensor Placement: CAD can help in planning the placement of sensors such as ultrasonic sensors, cameras, or electromagnetic sensors within the parking area to monitor parking space occupancy in real-time.

Structural Design: For multi-story parking structures, CAD can be used to design the architectural and structural aspects, optimizing the use of space and ensuring safe and efficient construction.

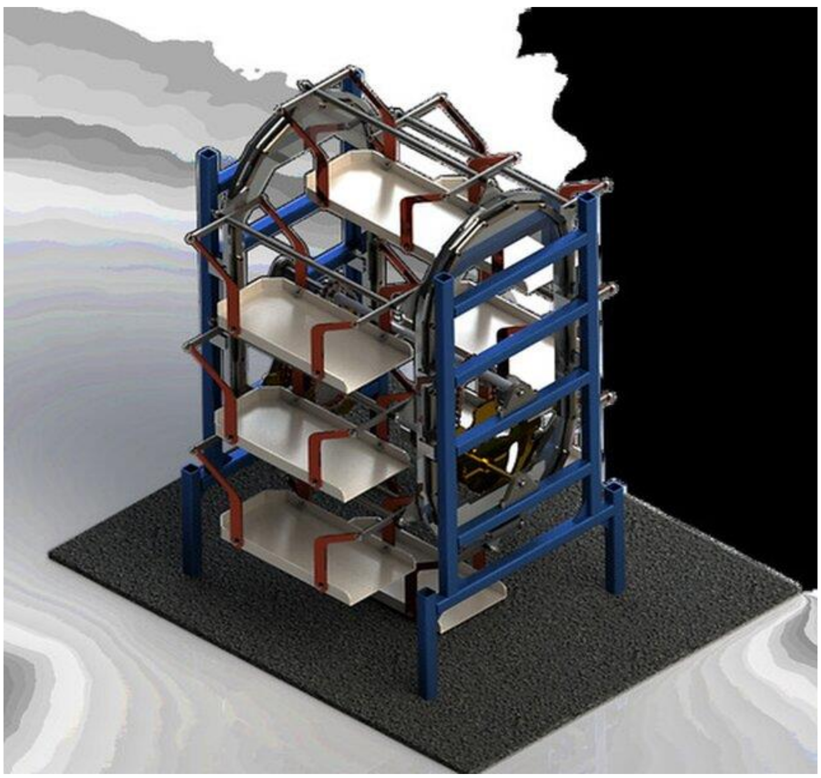
Wiring and Connectivity: CAD can be used to plan the wiring and connectivity infrastructure required for smart parking systems, including the installation of network cables and power supply for sensors, access control, and payment terminals.

User Interface Design: CAD can aid in the design of user interfaces for payment kiosks, mobile apps, and digital signage systems used to guide drivers to available parking spaces.

Data Visualization: CAD can be integrated with data visualization tools to create real-time dashboards and maps showing parking space availability and occupancy, helping drivers locate open spots.

Maintenance Planning: CAD can assist in planning maintenance routes and schedules for servicing and repairing the sensors and other components of the smart parking system.

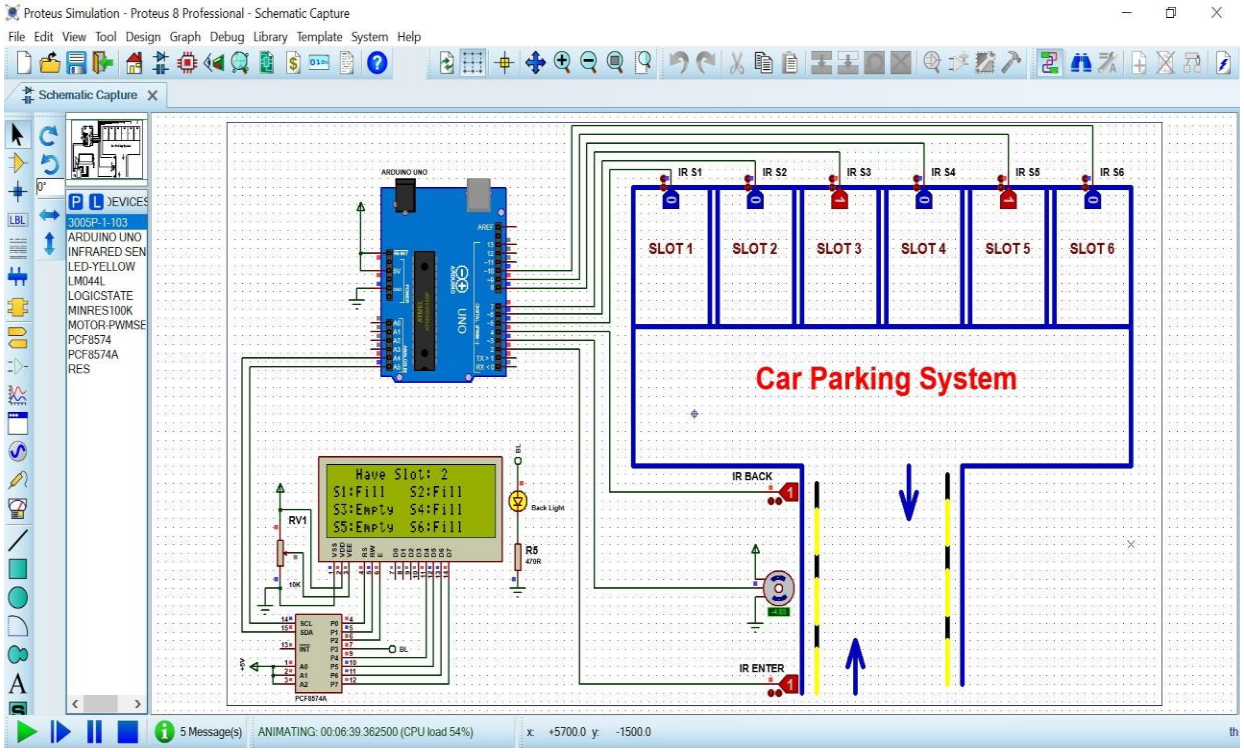
By using CAD in these aspects of smart parking system design and implementation, parking facilities can be more effectively organized, monitored, and managed, improving the overall experience for drivers and the efficiency of the parking operation.



Python coding for smart parking

```
Class ParkingLot:
    Def __init__(self, total_spaces):
        Self.total_spaces = total_spaces
        Self.available_spaces = total_spaces
        Self.spaces = [False] * total_spaces  # Initialize all spaces as vacant
    Def check_availability(self):
        Return self.available_spaces
    Def park(self):
        If self.available_spaces > 0:
            For I in range(self.total_spaces):
                If not self.spaces[i]:
                    Self.spaces[i] = True
                    Self.available_spaces -= 1
                    Return f"Parking space {I + 1} is now occupied."
        Else:
            Return "Sorry, the parking lot is full."
    Def leave(self, space_number):
        If 1 <= space_number <= self.total_spaces and self.spaces[space_number – 1]:
            Self.spaces[space_number – 1] = False
            Self.available_spaces += 1
            Return f"Parking space {space_number} is now vacant."
        Else:
            Return f"Invalid space number or space is already vacant."
```

SIMULATION:



Coding :

```
Import random
Class ParkingLot:
    Def __init__(self, capacity
         Self.capacity = capacity
        Self.available_spaces = capacity
    Def park_vehicle(self):
        If self.available_spaces > 0:
            Self.available_spaces -= 1
            Return True
        Else:
            Return False
```

```
    Def exit_vehicle(self):
        If self.available_spaces < self.capacity:
            Self.available_spaces += 1
            Return True
        Else:
            Return False
Class SmartParkingSimulation:
    Def __init__(self, capacity):
        Self.parking_lot = ParkingLot(capacity)
        Self.total_vehicles = 0
        Self.successful_parkings = 0
        Self.successful_exits = 0
    Def simulate(self, num_iterations):
        For _ in range(num_iterations):
            Action = random.choice([“arrive”, “depart”])
            If action == “arrive”:
                If self.parking_lot.park_vehicle():
                    Self.successful_parkings += 1
                Self.total_vehicles += 1
            Else:
                If self.parking_lot.exit_vehicle():
                    Self.successful_exits += 1
    Def display_results(self):
        Print(“Simulation Results:”)
        Print(f’Total Vehicles: {self.total_vehicles}”)
        Print(f’Successful Parkings: {self.successful_parkings}”)
        Print(f’Successful Exits: {self.successful_exits}”)
        Print(f’Available Parking Spaces: {self.parking_lot.available_spaces}”)
```

```
If __name__ == “__main__”:
    Capacity = 50
    Simulation = SmartParkingSimulation(capacity)
    Num_iterations = 1000
    Simulation.simulate(num_iterations)
    Simulation.display_results()
```