AI ENABLED CAR PARKING USING OPENCV

COLLEGE NAME: VIT-AP

TEAM MEMBERS:

Team ID: Team-592063

Team Size: 4

Team Leader: Goluguri Surya Chandra Sekhar Reddy

Team member: Jasti Hari Surya

Team member: Akshitha Karanam

Team member: Shaik Arshad

Mentors Name: Saumya Mohandas

1. INTRODUCTION

To finding parking availability for a specific time period is a very tedious job in urban areas. The Indian government now focusing on the smart city project, already they published city name for an upcoming smart city project. In smart city application , intelligent transportation system (ITS) plays an important role- in that finding parking place, specifically for thecar owner to avoid time computation, as well as congestion in traffic isgoing to be very important.

In this article, we propose an intelligent car parking system for the smart city using Circle Hough Transform (CHT).Keywords- Intelligent transportation system (ITS), Circle HoughTransform (CHT), Circle detection, Video-Image processing, smart city, parking system , OpenCV. For today's traffic monitoring and itsmanagement is a recent trend in research development. In this paper, weare focusing on parking component of the traffic parameter.

Traffic very congested from last decade due to the increasing rise of automobilecompanies offers to a customer, privatization of that- mainly more andmore used in present day compared to last decade and it's also increasing inthe future may be with same or more speed. So now government thinkinghow to solve this problem in real time? Within specified time duration.

1.1 Project Overview

S.NO	TOPICS
1	INTRODUCTION
2	LITERATURE SURVEY

1.2 PURPOSE:

Itallows car park operators and companies to track their facilities, vehicleentry, and real-time reporting of the availability of parking spots. Thishelps companies manage their parks in a central digital hub offered with parking software.

1.Superior Technology

Parking management systems are known for their integration withtechnology. Most of these systems are based on improved models andtechnological innovations, due to which they are suited to be used invarious car parks.

2.Better parking experience

Better car park management means happier customers. A

parkingmanagement system enhances the customer journey by providing themwith a unified procedure.

3.Increased Protection

Parking management systems have technologically advanced securityfeatures that enable you to prevent parking misuse and suspicious activityin your parking facility

4.Reduced traffic and pollution

Vehicles that keep circling an area in search of an empty parking spacecause most of the city traffic. Moreover, significantly driving around or waiting for a parking space to be vacant burns through a lot of fuel andreleases emissions daily.

5.Easy implementation and management

Another of the benefit of a parking management system is that it canefficiently be designed and implemented. These systems have a well-organised structure

6.Cost-effective

Another advantage that you obtain from installing a smart parkingmanagement system is the cost. It runs on a low workforce, so you cansave money and time

7.

Uses integrated software and applications

Parking management solutions use software and applications that can becombined with another. Depending on your car park's requirements, thereare lots of customisations available.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

1. **Detection Accuracy:** Ensuring precise identification of parking spaces under various conditions.
2. **Real-Time Processing:** Minimizing latency for providing instant information to drivers.
3. **Scalability:** Designing systems capable of handling large parking lots

and multiple facilities.

4. **Integration Challenges:** Seamless compatibility with existing infrastructure and sensor technologies.
5. **Privacy and Security:** Addressing concerns about data privacy, unauthorized access, and system security.
6. **Cost Considerations:** Balancing affordability with the benefits of implementation.
7. **Adaptability to Environment:** Adapting to dynamic conditions with moving obstacles and changing traffic.
8. **User Adoption:** Designing user-friendly interfaces and educating users about AI-based systems.
9. **Maintenance:** Ensuring ongoing reliability through regular maintenance and updates.
10. **Regulatory Compliance:** Adhering to local regulations and standards.
11. **Environmental Impact:** Considering energy efficiency and environmental implications.

2.2 REFERENCES

1. **Online Databases:**

Explore IEEE Xplore, PubMed, and Google Scholar for relevant research.

2. **Conferences and Journals:**

Look into ICCV, CVPR, and journals like IEEE Transactions on Intelligent Transportation Systems.

3. **University Repositories:**

Check university repositories for theses on computer vision and parking systems.

4. **Books:**

Review "Computer Vision: Algorithms and Applications" and "Learning OpenCV 4."

5. **OpenCV Documentation:**

Refer to the official OpenCV documentation for research paper references.

6. **Industry Reports:**

Find reports and whitepapers from industry organizations and smart city tech companies.

7. **Research Centers:**

Investigate AI and transportation research centers like MIT's Intelligent Transportation Research

Center.

2.3 PROBLEM STATEMENT DEFINITION

Design an AI-driven car parking system using OpenCV for real-time detection and monitoring of parking spaces, aiming to enhance efficiency, provide accurate information to drivers, and seamlessly integrate with existing infrastructure while addressing challenges such as accuracy, scalability, and user adoption.

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

Ideation Phase
Empathize & Discover

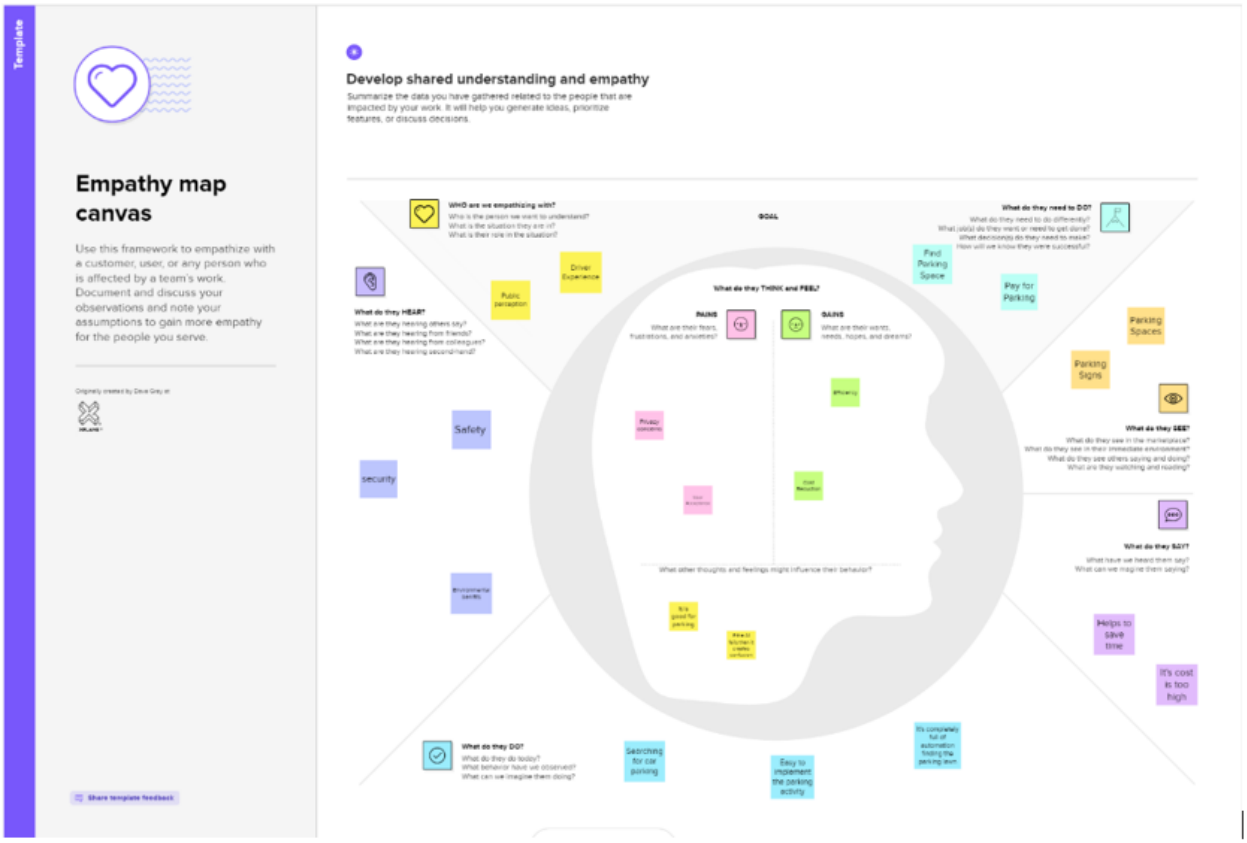
Date	30 October 2023
Team ID	Team-592063
Project Name	AI enabled car parking using open CV
Maximum Marks	4 Marks

Empathy Map Canvas:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

It is a useful tool to helps teams better understand their users.
Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

Reference: https://www.mural.co/templates/empathy-map-canvas



3.2 Ideation & Brainstorming

Ideation Phase

Brainstorm & Idea Prioritization Template

Date	30 October 2023
Team ID	Team-592063
Project Name	AI enabled car parking using open CV
Maximum Marks	4 Marks

Brainstorm & Idea Prioritization Template:

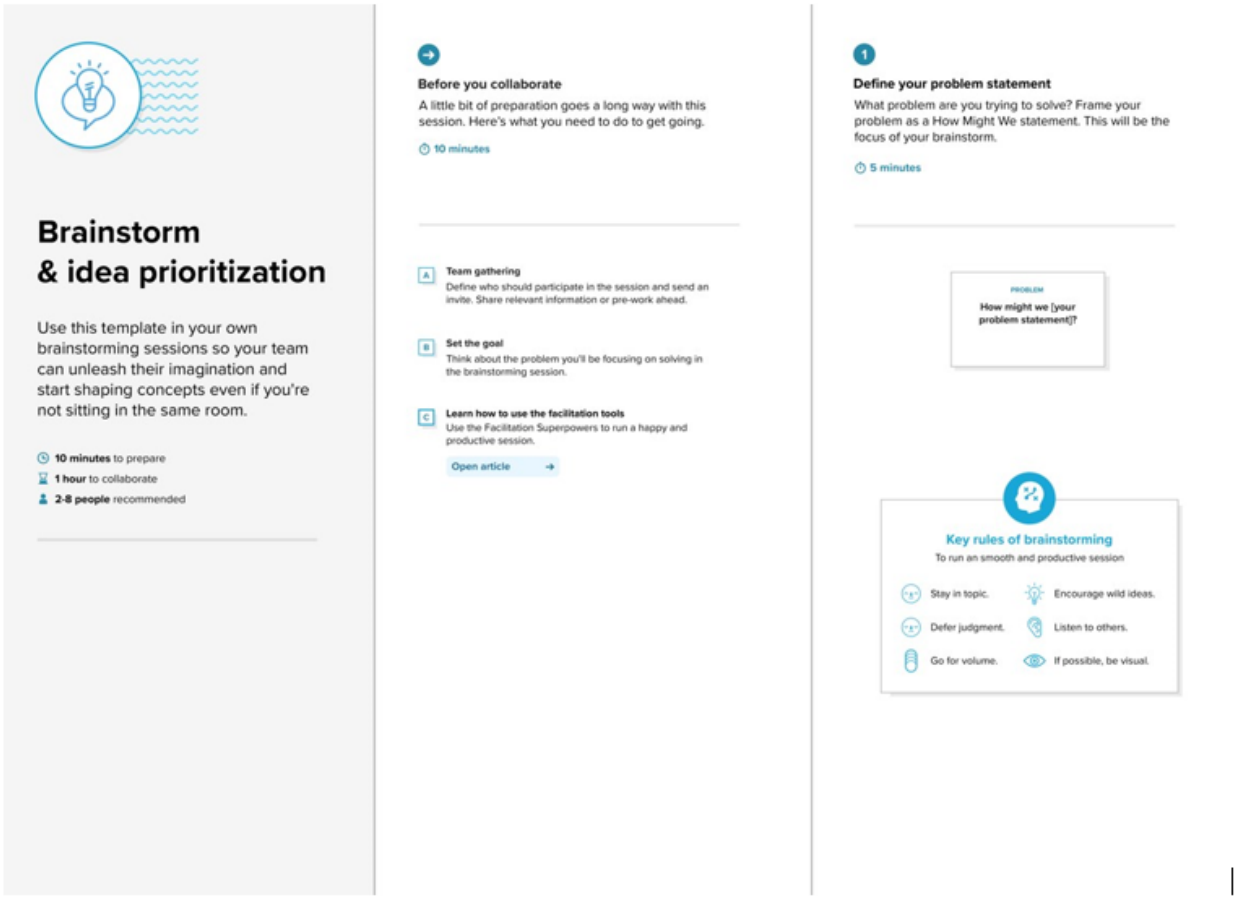
Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate,

helping each other develop a rich amount of creative solutions.

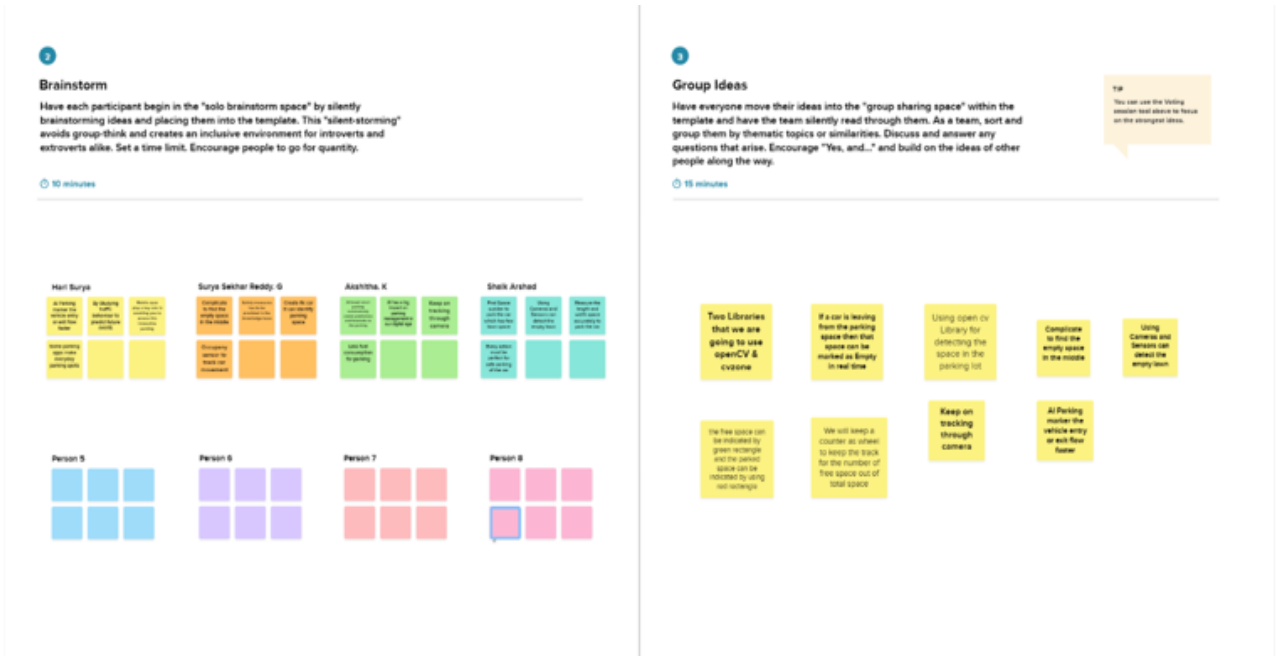
Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Reference: https://www.mural.co/templates/empathy-map-canvas

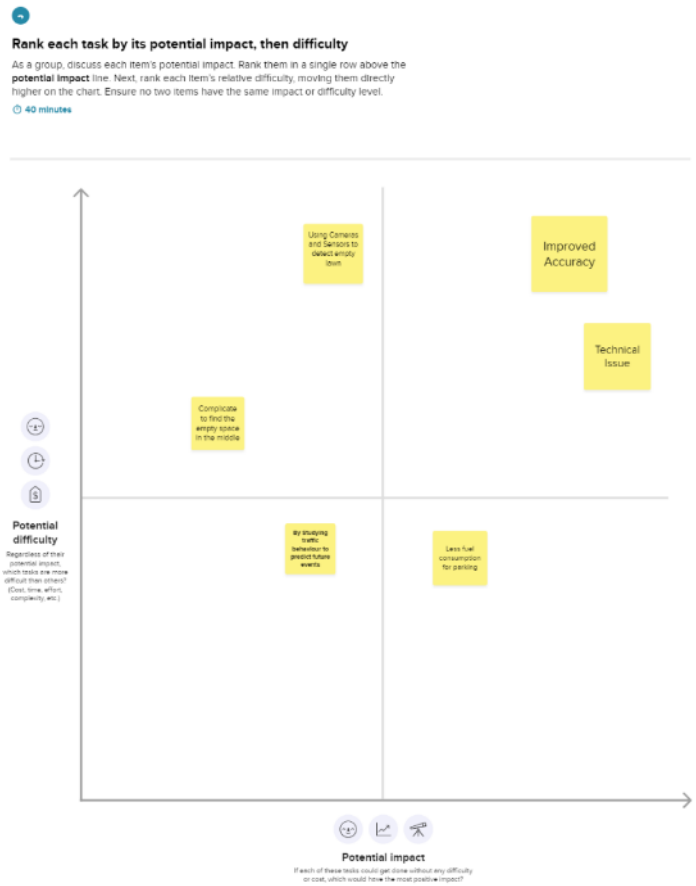
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

FR.NO	Functional Requirement	Sub Requirement
1	User Registration	Registration through form Registration through familiarize with the system Registration through mobile app
2	User Confirmation	Confirmation Via Email Confirmation Via approval of parking spaces
3	Object Detection	The System should be able to detect the presence of a car in a parking spot
4	Parking Monitoring	The system should be able to monitor the parked cars and detect any illegal activities, such as double parking or parking in a handicap spot
5	Real time updates	The system should provide real time updates on parking availability and other relevant information to drivers and parking lot staff
6	User friendly interface	The system should have a user-friendly interface that is easy to use and understand, to ensure a smooth and hassle-free parking experience for drivers

4.2 Non-Functional requirements

Following are the non-functional requirements of the proposed solution.

FR.no	Non-Functional Requirement	Description
NFR-1	Usability	The system should be a user-friendly and intuitive, with a simply and ease to use interface that is accesible to all users
NFR-2	Security	The system should be designed with robust security features, to ensure the privacy and safety of drivers

		and their vehicles, and to prevent unauthorized access and data breaches
NFR-3	Reliability	The system should be reliable and stable, with high availability and minimal downtime
NFR-4	Performance	The system should be able to process data quickly and accurately, with minimal delay and high efficiency.
NFR-5	Availability	The system should be highly available, with minimal downtime and interruption to the parking service
NFR-6	Scalability	The system should be able to handle a large number of parking spots adn users, and be easily scalable as the demand increases

5. PROJECT DESIGN

Project Design Phase-I
Solution Architecture

Date	1 November 2023
Team ID	Team-592063
Project Name	AI enabled car parking using open CV
Maximum Marks	4 Marks

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

1. Find the best tech solution to solve existing business problems.
2. Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
3. Define features, development phases, and solution requirements.
4. Provide specifications according to which the solution is defined, managed, and

delivered.

Example - Solution Architecture Diagram:

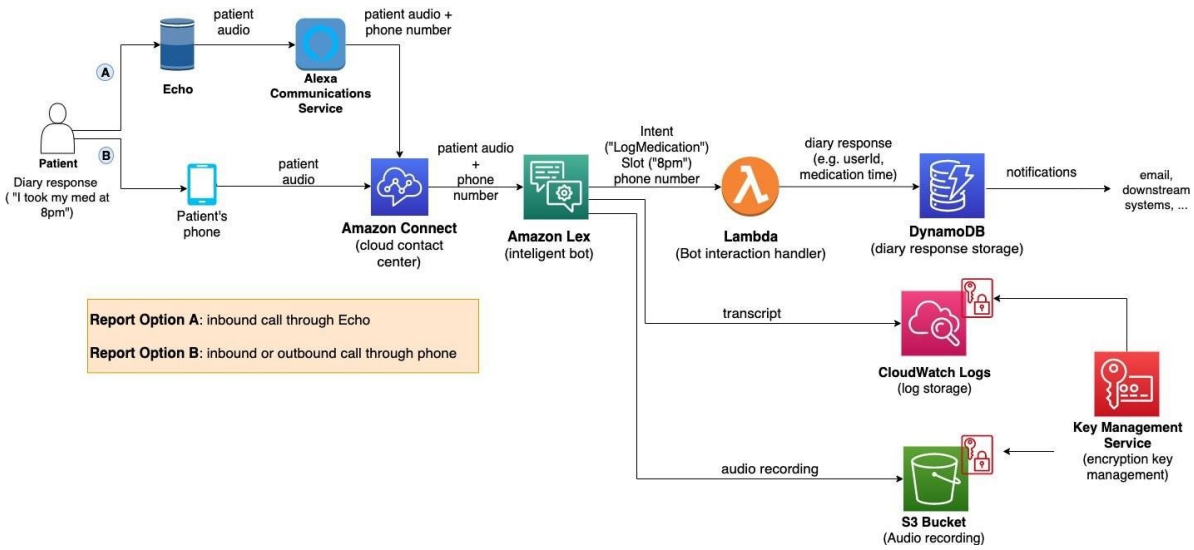


Figure 1: Architecture and data flow of the voice patient diary sample application

Reference: https://aws.amazon.com/blogs/industries/voice-applications-in-clinical-researchpowered-by-ai-on-aws-part-1-architecture-and-design-considerations/

Project Design Phase-I
Proposed Solution Template

Date	1 November 2023
Team ID	Team-592063
Project Name	AI enabled car parking using open CV
Maximum Marks	2 Marks

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To find the free parking slot in a minimum distance from a starting point.

2.	Idea / Solution description	The idea of this project to find the difference between empty slot and occupied slot and given numbers for each slot in ascending order to find minimum distance unoccupied slot.
3.	Novelty / Uniqueness	By above approach the parking slot is segregated as occupied and unoccupied slots.
4.	Social Impact / Customer Satisfaction	This technique will reduce the time taken park their car and thereby improving the customer satisfaction.
5.	Business Model (Revenue Model)	The result of this project could be implemented in public places and they will be able to achieve the accuracy
6.	Scalability of the Solution	The outcome of this project will be very helpful in parking management system

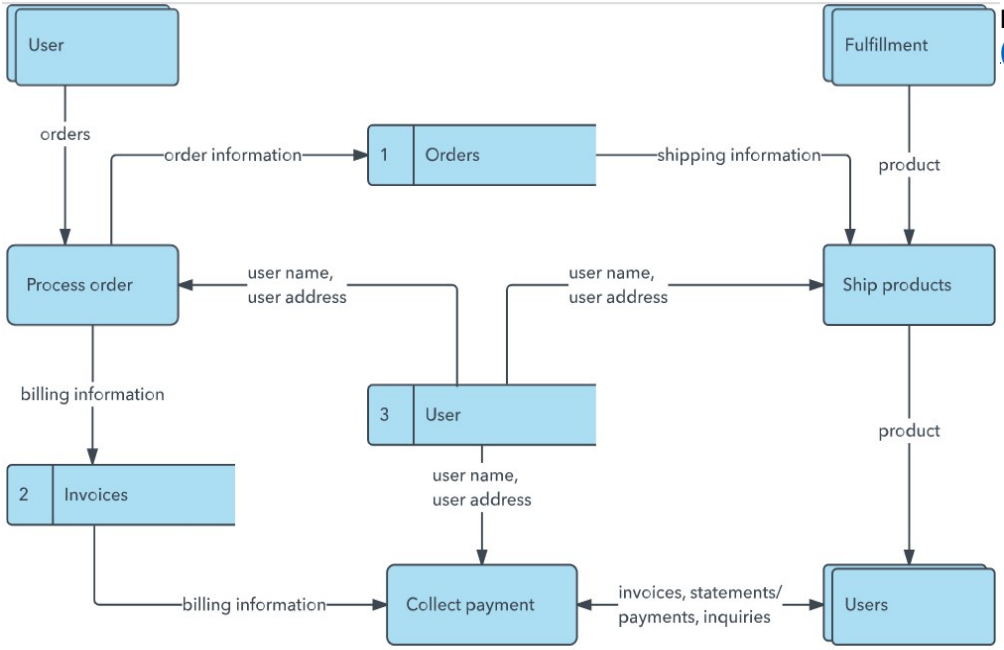
Project Design Phase-II
Data Flow Diagram & User Stories

Date	2 November 2023
Team ID	Team-592063
Project Name	AI enabled car parking using open CV
Maximum Marks	4 Marks

Data Flow Diagrams:

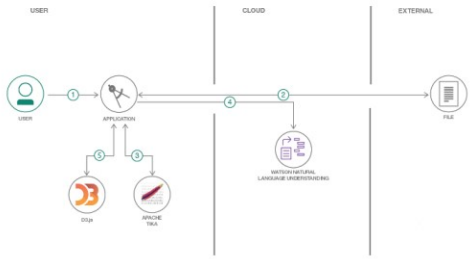
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Example: DFD Level 0 (Industry Standard)

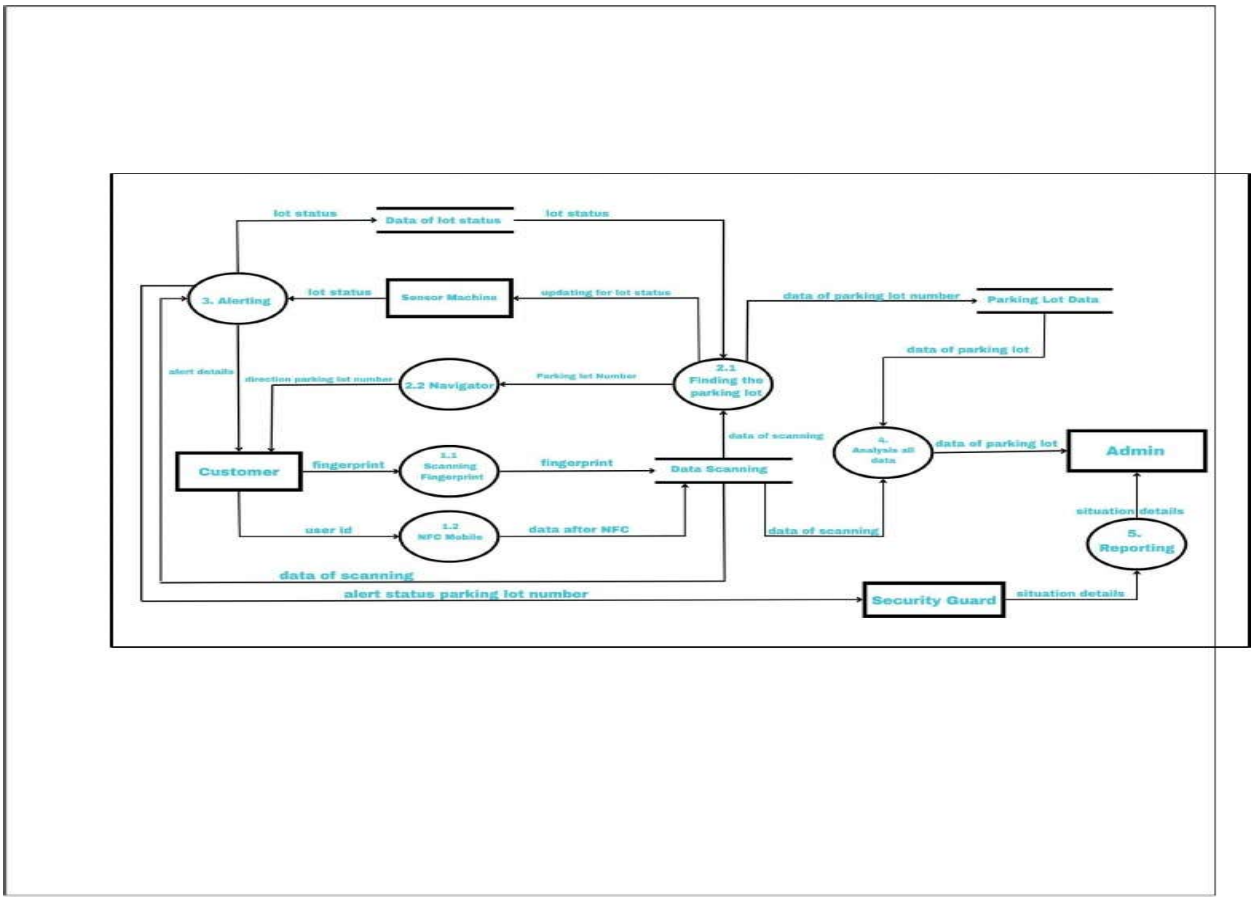


Example: [(Simplified)]

Flow



1. User configures credentials for the Watson Natural Language Understanding service and starts the app.
2. User selects data file to process and load.
3. Apache Tika extracts text from the data file.
4. Extracted text is passed to Watson NLU for enrichment.
5. Enriched data is visualized in the UI using the D3.js library.



User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1

		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register the app with email account	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can register & access user profile with Gmail Account	High	Sprint-1
	Dashboard	USN-6	As a conferrer I can request vacant parking space to park my car and see the number of slots available	I can get information about parking rates	High	Sprint-2
Customer (Web user)	Profile	USN-7	As a user I can see registration page, login page and request page to see available slots and camera footage screen where I can check availability of parking spots in real time	I can login through email and social media account for registration	Medium	Sprint-2
Customer Care Executive	Help desk / user support	USN-8	As a customer care executive, I can solve the queries of the users	I can reply to their queries and solve their related problems	High	Sprint-3

Administrator	Registration	USN-9	As a administration or I can view the database of registered users	I can check and verify the persons who are the registered their mail id's and information	Medium	Sprint-4
	Dashboard	USN-10	As an administration, I can view how many members requested for what trouble occurs in parking vehicle	I can check the numbers of requirements and monitor the availability	Low	Sprint-4
Camera Screen	User Interface	USN-11	Car parking spots available number displayed, Spots available highlighted with green colour and car number changing continuously, screen shows the CCTV footage coverage of parking area	I can get information of number of slots available and area in which slots available on the screen	High	Sprint-5

6. PROJECT PLANNING & SCHEDULING

Project Design Phase-II
Technology Stack (Architecture & Stack)

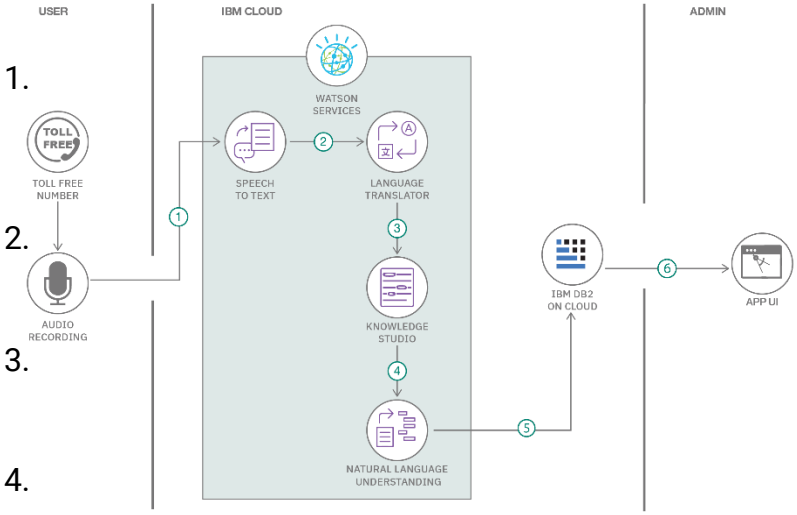
Date	03 November 2023
Team ID	Team-592063
Project Name	AI enabled car parking using open CV
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Example: Order processing during pandemics for offline mode

Reference: https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/



1.
2.
3.
4.
5. Indicate interface to machine learning models (if applicable)

Guidelines:

Include all the processes (As an application logic / Technology Block)

Provide infrastructural demarcation (Local / Cloud)

Indicate external interfaces (third party API's etc.)

Indicate Data Storage components / services

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	User Interface is used by user in mobile application or in Build in car	HTML, CSS, JavaScript / Angular Js / React Js etc.

		display itself	
2.	Application Logic-1	Framework used for design the software	Python, python-flask
3.	Application Logic-2	Access the software in the car by the driver to detect the spot	Python, OpenCV
4.	Application Logic-3	Open CV is an open source platform for providing real time computer vision technology	Open CV
5.	Database	Contains images and video frames stores in data base	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	They make it easy for developers to store manage and deploy container images	Container registry
9.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
10.	Machine Learning Model	Uses test and trained data images and video to learn the environment	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description

1.	Real-time processing	The OpenCV library provides fast and efficient image processing functions, making it possible to detect and track vehicles in real-time. This is important for car parking systems as it allows parking availability information to be updated in real time
2.	Scalability	An AI-enabled car parking system using OpenCV can be easily scaled to accommodate a large number of parking spots.
3.	Scalable Architecture	Open CV provides accurate object detection and tracking algorithms. Making it possible to accurately detect and track vehicles within the parking lot.
S.No	**Characteristics**	**Description**
4.	Customizability	OpenCV provides a wide range of customizable image processing algorithms, allowing car parking systems to be tailored to specific requirements. For example, the system can be trained to detect and track specific types of vehicles, such as motorcycles or trucks.
5.	Performance	OpenCV can be easily integrated with other technologies, such as IOT sensors or payment systems. This allows car parking systems to be seamlessly integrated with other systems and services.

References:

https://c4model.com/

https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/

https://www.ibm.com/cloud/architecture

https://aws.amazon.com/architecture

https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-

diagrams-2d20c9fda90d

7. CODING & SOLUTIONING

7.1 FEATURES 1

```
#Import Time
import time
Vehicle_Number=['XXXX-XX-XXXX']
 Vehicle_Type=['Bike'
]vehicle_Name=['Intruder']
Owner_Name=['Unknown']
Date=['22-22-3636']
Time=['22:22:22']
```

In this code block, we are importing thetimemodule to implement its methodsand function in the project. We have initialized the variables vehicle number,vehicle type, vehicle name, owner name date, and time to some default value.As well as bikes, cars, and bicycles with some initial value.

2.Create a while loop block to display the options inVehicle Parking Management Project

```
def main():
global
 bikes,cars,bicycles
try:
while True:
print(" ")
print("\t\tParking print(" ")
print("1.Vehicle Entry")
print("2.Remove Entry" )
print("3.View Parked Vehicle")
print("4.View Left ParkingSpace ")
print("5.AmountDetails ")
print("6.Bill")
print("7.Close Programme ")
```

In this code block, we have initialized the bikes, cars, and bicycles as global variables. They are accessible through the entire main block. Here we are providing the options to choose the service options from the list, for the vehicle parking management system.

3.Code for vehicle number entry

Ch is for choice, Once we select the ch option as 1 which is for vehicle entrynumber, then we provide the while loop. while the number(no is True). We willstore the vehicle number in Vno. If the vno is empty i.e vno==“”.The user asksto enter the vehicle number, else If the vno entered is already present in thevehicle number then it prints the vehicle number already exists. Else iflen(vno)==12, It will ask to append the info to the vehicle number variable.

4.Code to enter the vehicle type

```
typee=True

while typee==True:

Vtype=str(input("\tEnter vehicletype(Bicycle=A/Bike=B/Car=C):")).lower()

if Vtype=="": print("###### Enter Vehicle Type######")

 elif Vtype=="a":Vehicle_Type.append("Bicycle")

bicycles-=1typee=notTrue

elif bikes-=1typee=notTrueelif Vtype=="c":Vehicle_Type.append("Car") cars-=1typee=not
```

Here we have to initialize the typee variable to true. While the condition is True,the system asks to enter the vehicle type i.e a,b, or c which will accept the inputin the lower case. Here A is for bicycle, B is for Bike and C is for Car. Anyvehicle type you enter is stored in the variable Vtype. If the Vtype==””(empty).It will ask to enter the vehicle type. According to the type of variable you enterthe vehicle type will be stored in the variable and typee variable is set to notTrue.

5.Code to enter the vehicle name

```
name=True

while name==True:

vname=input("\tEnter vehiclename - ")
```

```
if vname=="":

print("########Please Enter Vehicle Name########")

else:vehicle_Name.append(vname)
```

Here we have set the name== True. While the name == True i.e until we enterthe name.vname store the value i.e. vehicle name. if the vname is empty systemasks to enter the vehicle name, else it will store the name using the appendfunction to the vehicle name variable The name variable is initialized to notTrue.

7.2 Features 2

1.Code to enter the owners name

```
o=True

while o==True:

OName=input("\tEnter owner"  )

if OName==" ":
 print("###### Please Enter Owner Name######")

 else:

Owner_Name.append(ONa m e  o = n o t  T r u e
```

O is initialized to True. While the condition satisfies the Owner's name is storedin the OName variable. If the OName is empty it system asks to enter the ownername else it will store the Owner name in the owner name variable. O is nowinitialized to Not True

2.Code enter the date and time

```
d=True

while d==True:

date=input("\tEnter Date (DD-MM-YYYY) - ")

 if date=="":

print("###### Enter Date######")

elif len(date)!=10: print("###### Enter Valid Date######") else:Date.append(date) d=not Truet=Truewhile t==True:time=input("\tEnter Time(HH:MM:SS) - ") if t=="": print("###### Enter Time######") elif len(time)!=8: print("###### Please Enter Valid Time######") else:Time.append(time)
```

3.Code to generate bills for different type of vehicles parked

```
elif ch==6:

 print("..........Generating
Bill..........")

no=True

while no==True:

Vno=input("\tEnter vehicle number to Delete(XXXX-XX-XXXX) -").upper()

if Vno=="": print("###### Enter Vehicle No. ######")

elif len(Vno)==12:

if Vno in Vehicle_Number:i=Vehicle_Number.index(Vno)no=not True

elif Vno not in Vehicle_Number:

print("###### No Such Entry ######")

else: print("Error")

else: print("###### Enter Valid Vehicle Number ######")

 print("\tVehicle Check in time - ",Time[i])

print("\tVehicle Check in Date - ",Date[i])

print("\tVehicle Type- ",Vehicle_Type[i])

inp=Trueamt=0while inp==True:

hr=input("\tEnter No. of Hours Vehicle Parked - ").lower()if hr=="":

print("###### Please Enter Hours ######")elif int(hr)==0 and Vehicle_Type[i]=="Bicycle":amt=20inp=not
Trueelif int(hr)==0 and Vehicle_Type[i]=="Bike":amt=40inp=not Trueelif int(hr)==0 and
Vehicle_Type[i]=="Car":amt=60inp=not Trueelif int(hr)>=1:if Vehicle_Type[i]=="Bicycle":


amt=int(hr)*int(20) inp=not True

elifVehicle_Type[i]=="Bike":amt=int(hr)*int(40)inp=not True

elifVehicle_Type[i]=="Car":amt=int(hr)*int(60)inp=not True

 print("\t Parking Charge -",amt)ac=18/100*int(amt)

print("\tAdd. charge 18 %- ",ac)

print("\tTotal Charge - ",int(amt)+int(ac))
```

The billing section generates the bill for the vehicle parked. We have to enterthe correct vehicle number and check the length of the vehicle number and thevehicle number present in the database. After this check the time and the type of vehicle. Depending upon the vehicle type

the system calculates the charges. Thelast choice ch==7 is to come out of the service options and quit the program.

8. PERFORMANCE TESTING

Performance metrics

Performance of the PMS, which employs resource allocation, real-time anddynamic path planning, and elevator scheduling algorithms are assessed andevaluated using the metrics

Average traveldistance for storage(ATDS)
This performance parameter measures theaverage travel distance in terms of cellmovements for all storage requests during themorning rush hours
Average travel distance for retreival(ATDR)
This performance parameter measures theaverage travel distance in terms of cellmovements for all retrieval requests during theevening rush hours
Range of utilization rate for elevators
This performance parameter measures therange of values for the ratio of time during whichan elevator is busy transporting a vehicle to the 2-hsimulation time, and separately for each of thetwo rush hours, namely, the morning rush hour and the afternoon rush hour
Customer waiting timefor storage (WTS)

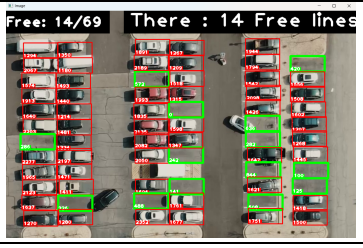
This performance metric measures the time that elapsesfrom the instant the customer arrives at the end of thefirst-in-first-out (FIFO) queue extending from the frontof delivery bays of the parking ground floor outwarduntil (s)he reaches the delivery/retrieval bay on theground floor This performance parameter measures the time thatelapses from the instant a customer arrives at the parkingstructure to retrieve his/her
Time to full capacity utilization
Customer waiting timefor retrieval (WTR):This performance parameter records the occurrence timefor the full space evaluation metrics andfor parking structure utilization for the parkingstructure during the morning rush hours their definitions

Project Development Phase
Model Performance Test

Date	15 November 2023
Team ID	Team-592063
Project Name	AI enabled car parking using open CV
Maximum Marks	10 Marks

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Model Summary	93.18%	
2.	Accuracy	Training Accuracy – 92.2% Validation Accuracy -93.50%	
3.	Confidence Score (Only Yolo Projects)	Class Detected - Car Confidence Score – 0.84	

9.RESULTS

9.1 Output Screenshots


Image
Free: 14/69
There : 14 Free lines


Smart Vehicle Parking System
Start_Parking_Camera
EXIT
EXIT

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

Recent increments in car ownership and urbanization, coupled with poor city planning, have contributed greatly to the rising parking problems across theUS[6]. Fortunately, the adoption of AI in smart parking has completelyrevolutionized how we look for parking spaces. Here are a few advantages of smart parking for drivers and business owners.

Less fuel consumption

Driving around looking for a parking space consumes a lot of fuel. Andconsidering the current fuel prices, that's a lot of money going down the drain.Smart parking solutions provide easy access to parking spots, thereby saving precious resources such as time, fuel, and space.

A reduction of search traffic on the streets

Nearly a third of traffic in urban areas is created by drivers looking for a parkingspot. By leveraging AI-based smart parking solutions, municipalities can manageand reduce search traffic on busy streets. Smart parking solutions not onlyminimize search traffic but also smoothen traffic flow. The result? People spendless time looking for a parking spot since they already know where to get one.

Reduced parking stress

Looking for a good parking spot in a congested part of the city is simplyoverwhelming. You might find yourself driving across the same street severaltimes, only to end up parking far away from your destination.AI-based smart parking solutions incorporate smart parking technologies withIoT devices so drivers can find parking spots easily using their smartphones and

Reduced parking stress

Looking for a good parking spot in a congested part of the city is simplyoverwhelming. You might find yourself driving across the same street severaltimes, only to end up parking far away from your destination.AI-based smart parking solutions incorporate smart parking technologies withIoT devices so drivers can find parking spots easily using their smartphones and computers. This way, they can see all parking spots available in the area they plan to travel to long before they get there. Some parking lots even allow you toreserve a space. This means you don't have to drive around looking for a spaceto park.

Benefits of AI-based smart parking for businesses:

Surface parking lots take up 5% of all urban land in the US [7]. This means morecompetition among parking lot businesses. You'd think parking lots around busystreets get a lot of business, but sadly, that's not the case. In most cases, driverscan't find these parking lots, and when they do, there are tons of irregularly parked cars, making the parking lot inefficient.The result is that some drivers opt to drive longer distances for safer and moreaccessible parking lots, which means less business for the better-situated but poorly managed parking lots. Fortunately, by incorporating smart parkingsolutions in their management process, parking lot managers can improve their lots' efficiency, boost customer satisfaction and ultimately boost profits. Here area few other benefits parking lot businesses stand to gain from leveraging smart parking solutions.

Improved parking experience:

AI-based smart parking solutions leverage collected data to provide specializedservices that ultimately lead to stress-free parking experiences. Besides lettingnearby drivers know if there are available spots in the parking lot, managers canalso install digital signs that receive real-time data from parking lot managementsoftware to direct drivers to their parking spots.This eliminates frustration among drivers trying to find a spot within the parkinglot, thus improving customer satisfaction.Also, the mere fact that drivers don't have to drive around the parking lookingfor a free space means fewer emissions, which could improve the air quality inindoor parking lots.

Pinpoint inefficiencies in parking lot management:

Managing a parking lot takes a lot of work. For instance, parking lot managersoften have to check the parking duration of certain vehicles or deal withirregularly and illegally parked vehicles. By leveraging real-time informationfrom connected on-site devices, they can accurately determine the parkingduration of all vehicles.This comes in handy, especially in parking lots offering short-time parkingservices like supermarkets. With this data, the managers can monitor excessive parking durations, which in most cases imply unauthorized use of their parkingareas. They can also monitor spaces that stay empty for extended periods, whichcould imply an issue with the spot in question.

Optimized usage of the facility:

Through data collected from various sensors installed around the streets and parking lot, businesses can monitor which areas have the highest and lowest parking traffic. With this data, they can better determine where to expand or cut back operations. Sensor data also enables businesses to monitor misuse of emergency access roads and dedicated parking spots

DISADVANTAGES:

Smart parking solutions present a lot of advantages for both drivers and businesses. But they also present a few drawbacks that might cause some peopleto put back or even avoid incorporating them altogether. Here are some of thedrawbacks of incorporating smart parking solutions.

High cost of installation

Numerous systems and technologies go into building an effective smart parkinglot management system. Things like sensors, cameras, automated ticketingmachines, and software cost a lot of money to install. Unfortunately, some businesses can't afford these systems, making it nearly impossible to incorporatethe system.

Regular maintenance requirements

Despite being automated, smart parking management systems require regular maintenance to ensure smooth operation. The frequency of maintenance allcomes down to the system in question, but most systems require monthlymaintenance. This further racks up the running costs of the parking lot, whichmight have a significant impact on profits.

11. CONCLUSION:

As conclusion, the objectives of this project have been achieved. Thehassle in searching for available parking slots has been completelyeliminated. The designed system could be applied

everywhere due to itsease of usage and effectiveness. It facilitates the problems of urbanlivability , transportation mobility and environment sustainability. TheInternet of Things integrates the hardware, software and network connectivity that enable objects to be sensed and remotely controlledacross existing network. Such integration allows users to monitor availableand unavailable parking spots that lead to improved efficiency, accuracyand economic benefit.

12. FUTURE SCOPE:

The smart parking management system can be broadly applied for many futureapplications. Apart from its basic role of parking management of cars it can also be applied for plane and ship and fleet management. With the ever growing fieldof Internet of Things many concepts can be interfaced along with our system. For residential and domestic parking system the device can be interfaced with HomeAutomation system which can control the various home appliances by sensingwhether the user is arriving or departing from the parking space. For instance if the user has arrived then the module will sense the presence and will sendinformation about arrival to the Home automation system which can accordinglyswitch on the selected appliances like HVAC (Heating Ventilation and Air Conditioning) units, Coffee maker, toaster, Wi-Fi routers etc. For commercial parking system the device can be interfaced with a module which can sense thearrival of employee and can switch on his computer and HVAC systems andaccordingly switch off the appliances when the employee departs. The systemcan also be used to track the reporting and departing time of the employee for alldays with precision thus acting as an attendance system. Thus many suchmodules can be interfaced with our system to provide better facility, security,and optimization of electricity and resources with the principle idea of flawlessfleet management system.

13. APPENDIX

Source Code

```
import cv2
import pickle
import cvzone
import numpy as np
```

```
# Video feed
cap = cv2.VideoCapture('carPark.mp4')

with open('CarParkPos', 'rb') as f:
    posList = pickle.load(f)
```

```
width, height = 107, 48


def checkParkingSpace(imgPro):
    spaceCounter = 0

    for pos in posList:
        x, y = pos

        imgCrop = imgPro[y:y + height, x:x + width]
        # cv2.imshow(str(x * y), imgCrop)
        count = cv2.countNonZero(imgCrop)


        if count < 900:
            color = (0, 255, 0)
            thickness = 5
            spaceCounter += 1
        else:
            color = (0, 0, 255)
            thickness = 2

        cv2.rectangle(img, pos, (pos[0] + width, pos[1] + height), color,
thickness)
        cvzone.putTextRect(img, str(count), (x, y + height - 3), scale=1,
                           thickness=2, offset=0, colorR=color)

    cvzone.putTextRect(img, f'Free: {spaceCounter}/{len(posList)}', (0, 50),
scale=3,
                           thickness=5, offset=20, colorR=(0,0,0))
    if spaceCounter>0:
        cvzone.putTextRect(img, f'There : {spaceCounter} Free lines', (390,
50), scale=4,
                           thickness=5, offset=20, colorR=(0, 0, 0))


while True:

    if cap.get(cv2.CAP_PROP_POS_FRAMES) == cap.get(cv2.CAP_PROP_FRAME_COUNT):
        cap.set(cv2.CAP_PROP_POS_FRAMES, 0)
    success, img = cap.read()
    imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    imgBlur = cv2.GaussianBlur(imgGray, (3, 3), 1)
    imgThreshold = cv2.adaptiveThreshold(imgBlur, 255,
cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
```

```
                                            cv2.THRESH_BINARY_INV, 25, 16)
    imgMedian = cv2.medianBlur(imgThreshold, 5)
    kernel = np.ones((3, 3), np.uint8)
    imgDilate = cv2.dilate(imgMedian, kernel, iterations=1)

    checkParkingSpace(imgDilate)
    cv2.imshow("Image", img)
    cv2.waitKey(10)
    if cv2.waitKey(2) & 0xFF == ord('q'):
        break
```

Parking space

```
import cv2
import pickle

width, height = 107, 48

try:
    with open('CarParkPos', 'rb') as f:
        posList = pickle.load(f)
except:
    posList = []


def mouseClick(events, x, y, flags, params):
    if events == cv2.EVENT_LBUTTONDOWN:
        posList.append((x, y))
    if events == cv2.EVENT_RBUTTONDOWN:
        for i, pos in enumerate(posList):
            x1, y1 = pos
            if x1 < x < x1 + width and y1 < y < y1 + height:
                posList.pop(i)

    with open('CarParkPos', 'wb') as f:
        pickle.dump(posList, f)


while True:
    img = cv2.imread('carParkImg.png')
    for pos in posList:
        cv2.rectangle(img, pos, (pos[0] + width, pos[1] + height), (255, 0,
255), 2)

    cv2.imshow("Image", img)
```

```
    cv2.setMouseCallback("Image", mouseClick)
    cv2.waitKey(1)
    if cv2.waitKey(2) & 0xFF == ord('q'):
        break
```