AI ENABLED CAR PARKING

SYSTEM

WITH OPENCV

Presented by

ArulKumaran.M

Karthik.S

SenthiLNathan.S Vijayaragavan.D

Team id:NM2023TMID05926

# ABSTRACT

Car parking with OpenCV is a computer vision-based system that helps in automating and managing the parking process. OpenCV (Open Source Computer Vision) is a popular opensource library that provides tools and functions for image and video analysis. By utilizing OpenCV, we can develop a system that can detect and track vehicles, identify parking spots, and manage the overall parking process.

The aim of this project is to develop a car parking system using OpenCV, which will utilize computer vision techniques to automate and streamline the parking process. The system will leverage image and video analysis capabilities offered by OpenCV to detect and track vehicles, identify available parking spots, and assist drivers in finding suitable parking spaces.

To achieve this, the proposed methodology involves several stages. First, the system will employ image processing techniques to extract relevant features from video streams or camera input, such as vehicle contours, motion analysis, and object recognition. Next, the extracted information will be used to detect and track vehicles within the parking area, enabling realtime monitoring of available parking spots.

Furthermore, the system will incorporate machine learning algorithms to train a model for vehicle classification, enabling the identification of different types of vehicles and distinguishing between occupied and vacant parking spaces. The model will be

trained on a labeled dataset of vehicle images, encompassing various angles, lighting conditions, and vehicle types.

The expected outcomes of this project include a fully functional car parking system that can accurately detect and track vehicles, accurately classify parking spaces, and provide real-time parking occupancy information to drivers. This will contribute to reducing the time and effort spent by drivers in finding parking spaces and improve the overall efficiency of parking management.

In conclusion, the proposed car parking system utilizing OpenCV and computer vision techniques aims to enhance the parking process by automating and optimizing parking spot detection, vehicle tracking, and occupancy management. The application of OpenCV in this context demonstrates the potential of computer vision technology to revolutionize the way we approach parking systems and improve the overall user experience.

# TABLE OF CONTENT

## INTRODUCTION

* 1. Project Overview
  2. Purpose

## IDEATION & PROPOSED SOLUTION

* 1. Problem Statement Definition
  2. Empathy Map Canvas
  3. Ideation & Brainstorming
  4. Proposed Solution

## REQUIREMENT ANALYSIS

* 1. Functional requirement
  2. Non-Functional requirements

## PROJECT DESIGN

* 1. Data Flow Diagrams
  2. Solution & Technical Architecture
  3. User Stories

## CODING & SOLUTIONING (Explain the features added in the project along with code)

* 1. Feature 1
  2. Feature 2
  3. Database Schema (if Applicable)

## RESULTS

* 1. Performance Metrics

## ADVANTAGES & DISADVANTAGES

1. **CONCLUSION**
2. **FUTURE SCOPE 10. APPENDIX** Source Code

# 1.INTRODUCTION

## Project Overview

AI car parking using OpenCV combines the power of artificial intelligence and computer vision techniques to automate and optimize the parking process. OpenCV (Open Source Computer Vision) is a popular open-source library that provides tools and functions for image and video analysis. By utilizing OpenCV, the system can detect and track vehicles, identify parking spots, and manage the parking process effectively.

The AI car parking system using OpenCV consists of several key components and stages:

1. Image/Video Input: The system takes input from cameras or video streams installed in the parking area. These cameras capture real-time images or videos of the parking lot.
2. Vehicle Detection: OpenCV is used to process the input images or video frames and detect vehicles present in the scene. Techniques such as object detection and feature extraction are applied to identify and locate vehicles accurately.
3. Vehicle Tracking: Once the vehicles are detected, OpenCVbased tracking algorithms are employed to track the vehicles' movement within the parking area. Tracking algorithms can utilize methods like optical flow, Kalman filters, or deep learningbased tracking.
4. Parking Space Detection: The system analyses the video frames to identify vacant parking spaces. This involves techniques like image segmentation, contour detection, or background subtraction to distinguish between occupied and vacant parking spots.
5. Parking Space Classification: OpenCV, along with machine learning algorithms, can be used to classify different types of parking spaces based on vehicle size, such as small, medium, or large. This information helps in guiding drivers to suitable parking spots.
6. Parking Management and Guidance: The system utilizes the collected data about vehicle occupancy and parking space availability to manage the overall parking process. This includes providing real-time information to drivers about available parking spots through digital signage, mobile apps, or integrated navigation systems.
7. User Interface and Interaction: A user-friendly interface is developed to facilitate driver interaction with the parking system. This can include features like parking reservation, payment integration, and guidance to the nearest available parking spot.

By combining AI algorithms and OpenCV techniques, the AI car parking system offers numerous benefits. It reduces the time and effort spent by drivers in finding parking spaces, improves parking space utilization, and enhances overall parking

efficiency. Additionally, the system provides real-time parking information to drivers, reducing traffic congestion and contributing to a more seamless parking experience.

In conclusion, the AI car parking system using OpenCV leverages computer vision techniques to detect vehicles, identify parking spaces, and optimize the parking process. It enhances the overall parking experience, improves operational efficiency, and contributes to better traffic management in parking areas.

## Purpose

The purpose of AI car parking using OpenCV is to automate and optimize the parking process by leveraging artificial intelligence and computer vision techniques. The system aims to address the challenges and inefficiencies associated with traditional parking methods, offering numerous benefits to both drivers and parking operators.

1. Efficient Parking Space Management: The primary purpose is to effectively manage parking spaces by accurately detecting and tracking vehicles. OpenCV's image processing and computer vision capabilities enable the system to identify vacant parking spots and monitor parking occupancy in real-time. This ensures optimal utilization of available parking spaces and reduces instances of overbooking or underutilization.
2. Enhanced Driver Experience: The system aims to provide a seamless and convenient parking experience for drivers. By utilizing OpenCV to detect and classify parking spaces, the system can guide drivers to available spots quickly. Real-time information about parking availability, communicated through digital signage, mobile apps, or integrated navigation systems, saves drivers time and reduces the frustration associated with searching for parking.
3. Reduction in Traffic Congestion: AI car parking using OpenCV contributes to reducing traffic congestion in parking

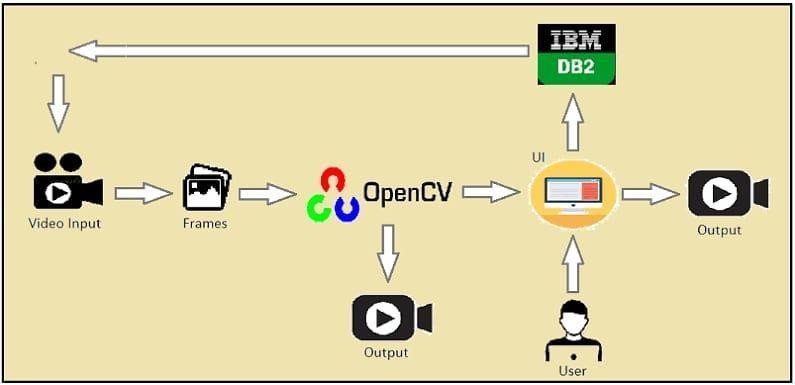
areas. By efficiently guiding drivers to available parking spaces, it minimizes the time spent circling around in search of a spot, which in turn decreases traffic congestion and the associated environmental impact. This promotes smoother traffic flow and reduces emissions.

1. Optimization of Parking Infrastructure: The system assists parking operators in making data-driven decisions for infrastructure optimization. By analysing parking occupancy patterns, operators can identify areas of high demand or low utilization, enabling them to optimize parking infrastructure allocation, adjust pricing strategies, and plan for future expansion or renovations.
2. Integration with Smart City Initiatives: AI car parking systems using OpenCV can be integrated into broader smart city initiatives. By providing real-time parking occupancy data, city authorities can analyse parking patterns and make informed decisions regarding urban planning, traffic management, and transportation infrastructure development. This integration contributes to the overall efficiency and sustainability of smart city ecosystems.
3. Improved Revenue Generation: Parking operators can benefit from increased revenue generation through better parking space utilization and efficient management. By leveraging AI algorithms and OpenCV, operators can implement dynamic pricing strategies based on demand, optimize parking

duration, and minimize revenue losses due to unauthorized or improper parking.

In summary, the purpose of AI car parking using OpenCV is to automate and optimize the parking process, providing efficient parking space management, enhanced driver experience, reduced traffic congestion, optimized infrastructure utilization, integration with smart city initiatives, and improved revenue generation. The system leverages the capabilities of artificial intelligence and computer vision to revolutionize traditional parking methods and create a more efficient and user-friendly parking environment.

**Technical Architecture:**



# IDEATION & PROPOSED SOLUTION

## Problem Statement Definition

The problem addressed by the car parking system using OpenCV is the inefficiency and inconvenience associated with traditional parking methods. Traditional parking systems often suffer from limited parking space availability, inefficient allocation of parking spots, and time-consuming search for vacant spaces. This leads to increased traffic congestion, driver frustration, and decreased

overall parking efficiency. The objective is to develop an automated parking system using OpenCV that can accurately detect and track vehicles, identify available parking spots, and streamline the parking process. The system aims to address these challenges by utilizing computer vision techniques to optimize parking space management, improve driver experience, reduce traffic congestion, and enhance overall operational efficiency for parking operators.

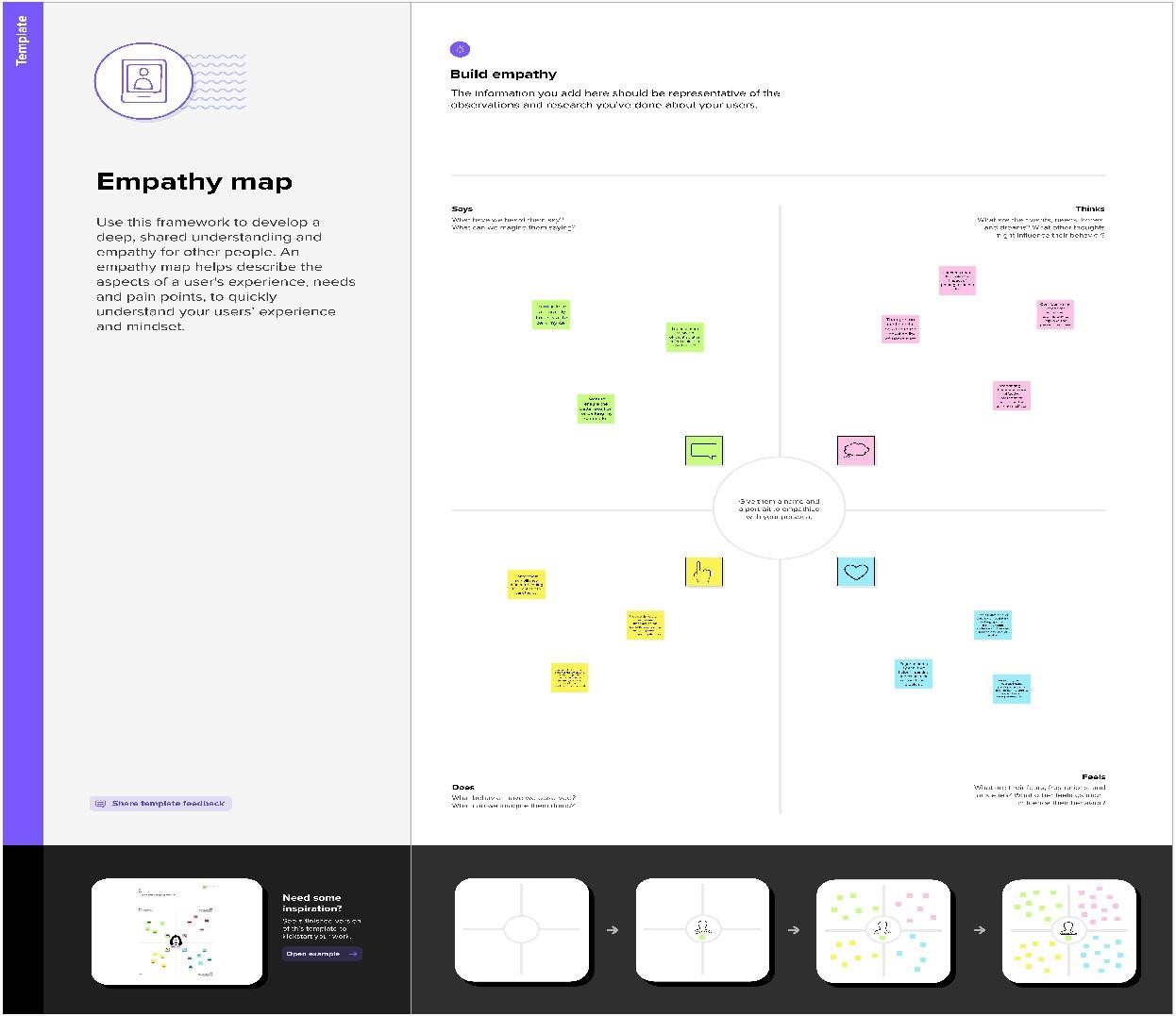
## Empathy Map Canvas

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.

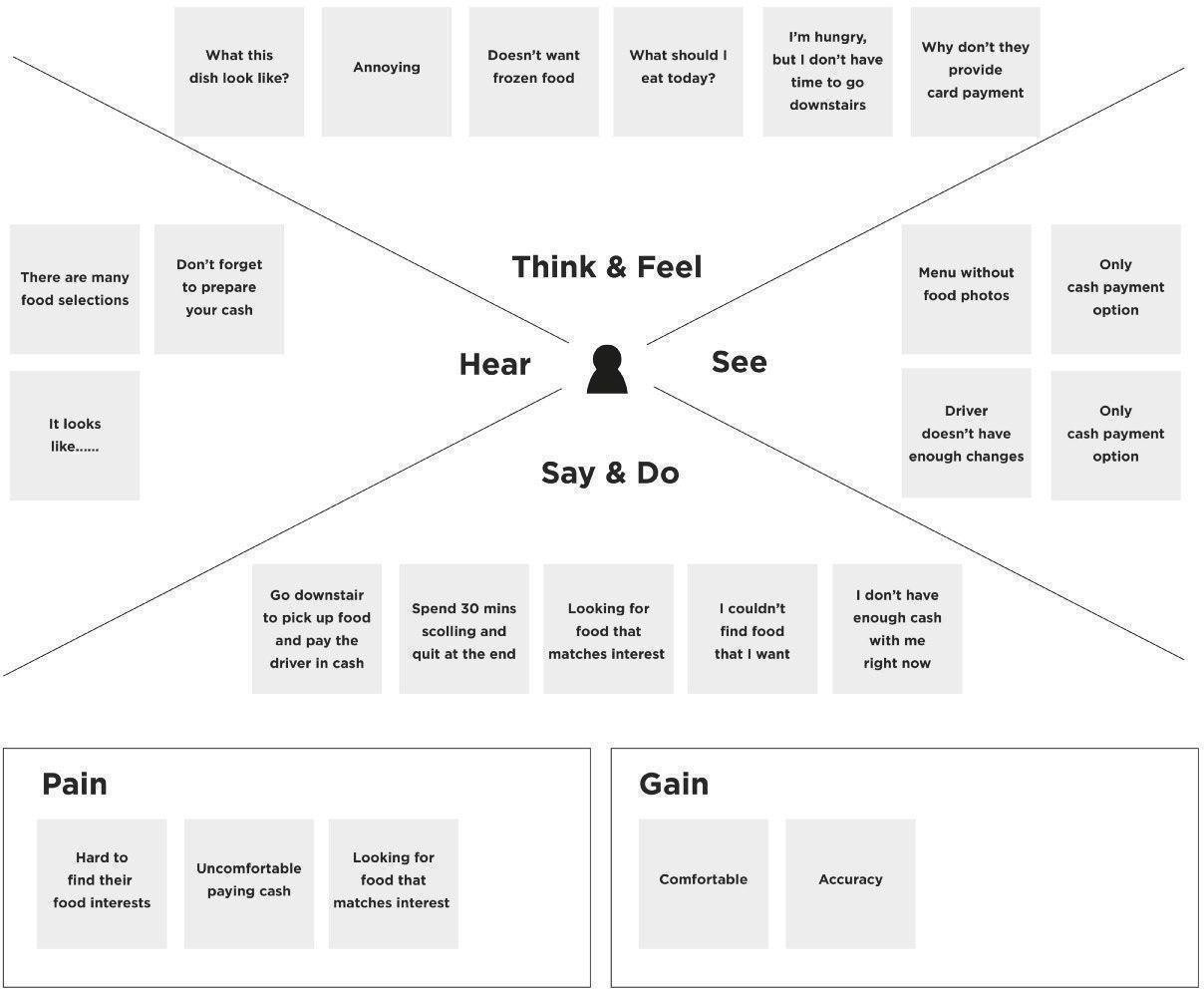
## Example



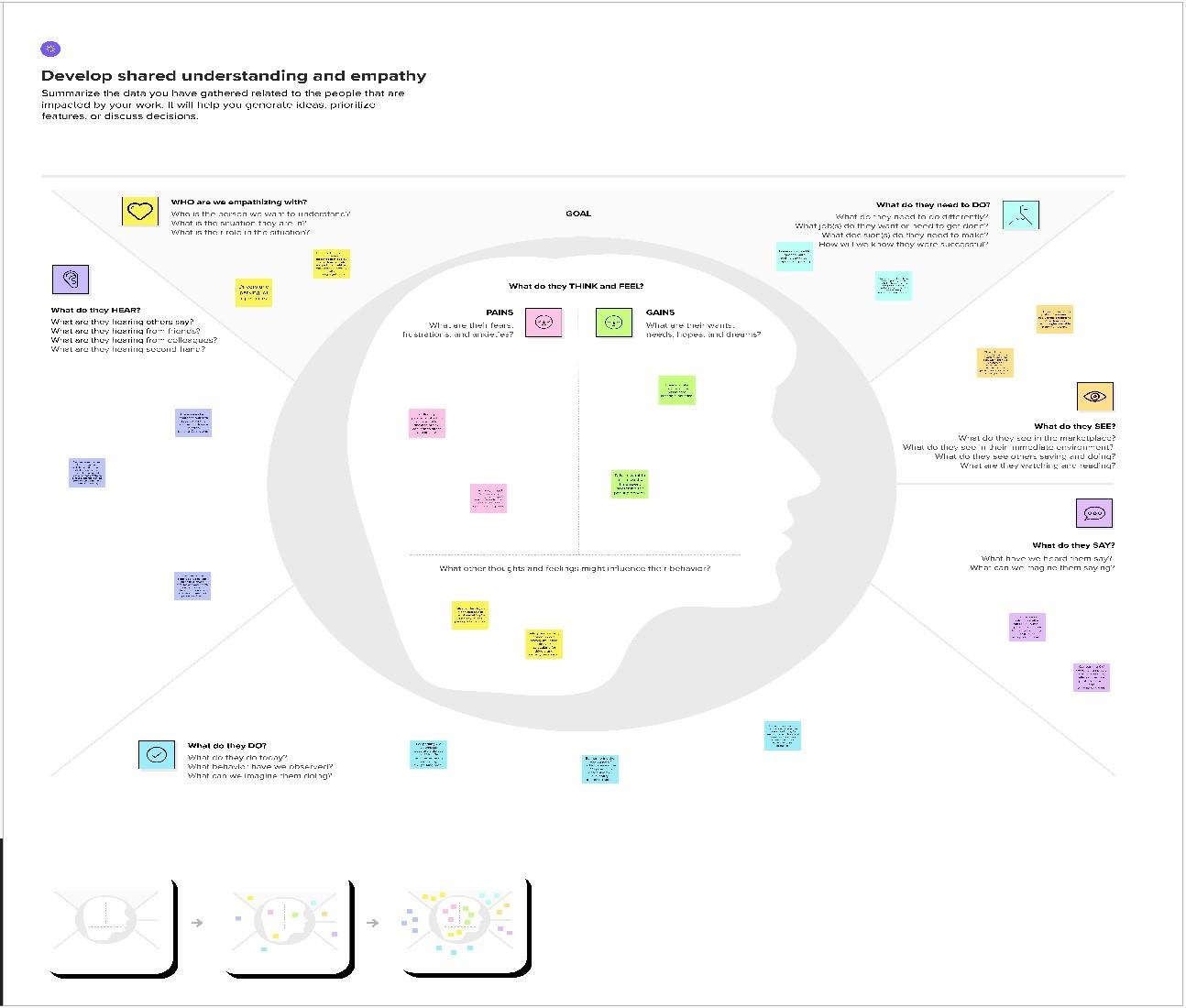
**Reference:**

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

**Example: Food Ordering & Delivery Application**



## Eg:



* 1. **Ideation & Brainstorming**

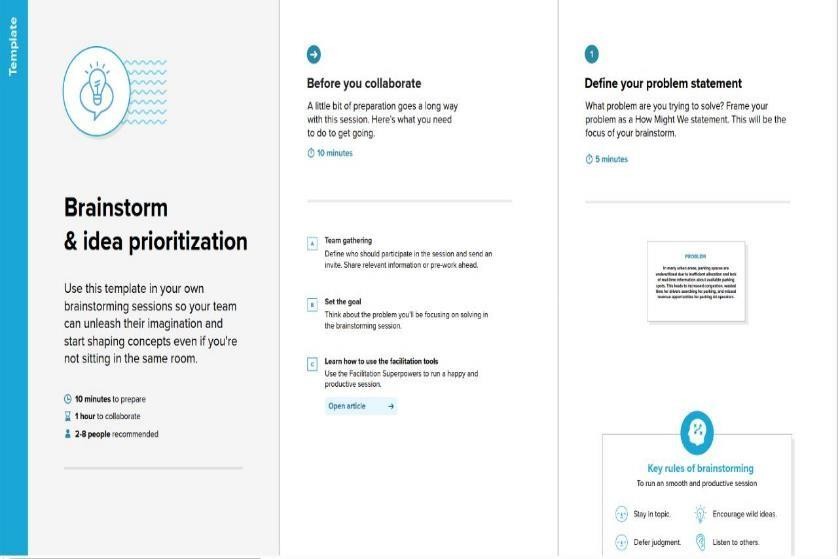
Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate,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

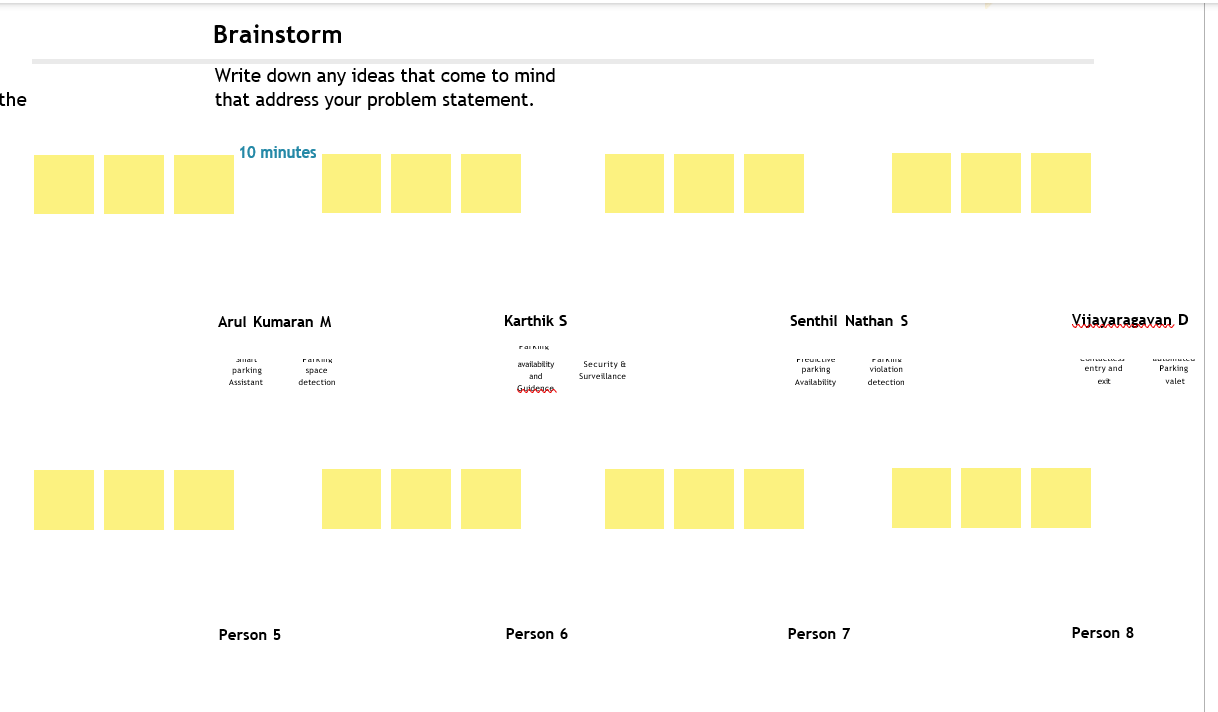
Reference:

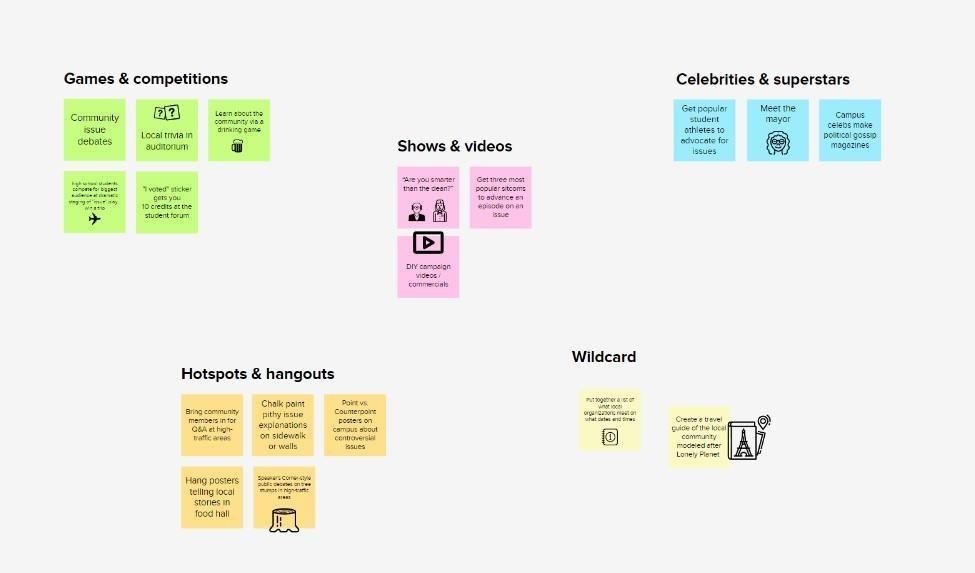
<https://www.mural.co/templates/empathy-map-canvas>

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

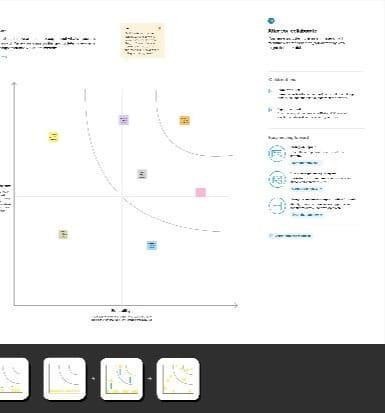


**+ Step-2: Brainstorm, Idea Listing and Grouping**





## Step-3: Idea Prioritization



* 1. **Proposed Solution**

|  |  |  |
| --- | --- | --- |
| **S.N**  **o.** | **Parameter** | **Description** |

|  |  |  |
| --- | --- | --- |
| 1. | Problem Statement (Problem to be solved) | The problem addressed by the car parking system using OpenCV is the inefficiency and inconvenience associated with traditional parking methods.  Traditional parking systems often suffer from limited parking space availability, inefficient allocation of parking spots, and time-consuming search for vacant spaces. This leads to increased traffic congestion, driver frustration, and decreased overall parking efficiency. The objective is to develop an automated parking system using OpenCV that can accurately detect and track vehicles, identify available parking spots, and streamline the parking process. The system aims to address these challenges by utilizing computer vision techniques to optimize parking space management, improve driver experience, reduce traffic congestion, and enhance overall operational efficiency for parking operators. |

|  |  |  |
| --- | --- | --- |
| 2. | Idea / Solution description | The car parking system using OpenCV is to utilize computer vision techniques to automate and optimize the parking process. The system will employ OpenCV's image processing and analysis capabilities to detect and track vehicles, identify vacant parking spaces, and manage parking occupancy. Using cameras or video streams, the system captures realtime images or video frames of the parking area. OpenCV algorithms are then applied to process the input and detect vehicles by analyzing features such as contours, motion, and object recognition. Once the vehicles are detected, OpenCVbased tracking algorithms are utilized to track their movement within the parking area. This allows for real-time monitoring of available parking spots and accurate assessment of parking occupancy.  To guide drivers, the system employs techniques like image segmentation and contour detection to identify vacant parking spaces. Additionally, machine learning algorithms can be incorporated |

|  |  |  |
| --- | --- | --- |
|  |  | to classify parking spaces based on vehicle size or type.  The system will provide a user-friendly interface, such as mobile apps or digital signage, to display real-time parking information to drivers, indicating available parking spots and guiding them to the nearest vacant space.  By automating the parking process and providing accurate parking occupancy information, the car parking system using OpenCV improves parking efficiency, reduces search time for drivers, minimizes traffic congestion, and enhances the overall parking experience. |

|  |  |  |
| --- | --- | --- |
| 3. | Novelty / Uniqueness | By utilizing OpenCV, car parking systems can employ advanced algorithms to detect, track, and analyze vehicles in realtime. This allows for efficient monitoring of  parking spaces, automated vehicle counting, and accurate identification of available spots. |

|  |  |  |
| --- | --- | --- |
| 4. | Social Impact / Customer Satisfaction | Car parking systems utilizing OpenCV have a significant social impact and contribute to increased customer satisfaction. The integration of OpenCV in car parking systems brings forth various social benefits and positively impacts customer satisfaction. By leveraging advanced computer vision algorithms, OpenCVbased parking solutions offer the following  advantages: |

|  |  |  |
| --- | --- | --- |
| 5. | Business Model (Revenue Model) | Car parking systems utilizing OpenCV can generate revenue through multiple streams.  These include parking fees charged to customers based on usage duration, subscription models offering recurring access to parking spaces, premium services for added convenience, advertising and sponsorships on digital displays, data monetization by providing valuable insights, and partnerships with ride- sharing or car-sharing companies. By employing these revenue models, OpenCV-based car parking systems can ensure a sustainable and profitable operation while delivering enhanced parking experiences to customers. |

|  |  |  |
| --- | --- | --- |
| 6. | Scalability of the Solution | The customer places the order on the client’s website ,the order is processed by the web application and analyses the amount of raw |

# REQUIREMENT ANALYSIS

## Functional Requirement

Following are the functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR**  **No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Registration | Registration through Form Registration through Gmail  Registration through LinkedIN |

|  |  |  |
| --- | --- | --- |
| FR-2 | User Confirmation | Confirmation via Email Confirmation via OTP |
| FR-3 | Directory access | Getting agree/not agree |
| FR-4 | Camera , mic access | Getting agree/not agree |

## Non Functional Requirements

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

|  |  |  |
| --- | --- | --- |
| **FR**  **No.** | **Non-Functional Requirement** | **Description** |
| NFR  -1 | **Usability** | Ease of login |
| NFR  -2 | **Security** | Password intimation |

|  |  |  |
| --- | --- | --- |
| NFR  -3 | **Reliability** | Transparency |
| NFR  -4 | **Performance** | Highly comfort and simplicity |
| NFR  -5 | **Availability** | Large medium availability |
| NFR  -6 | **Scalability** | Large no of acceptance |

# PROJECT DESIGN

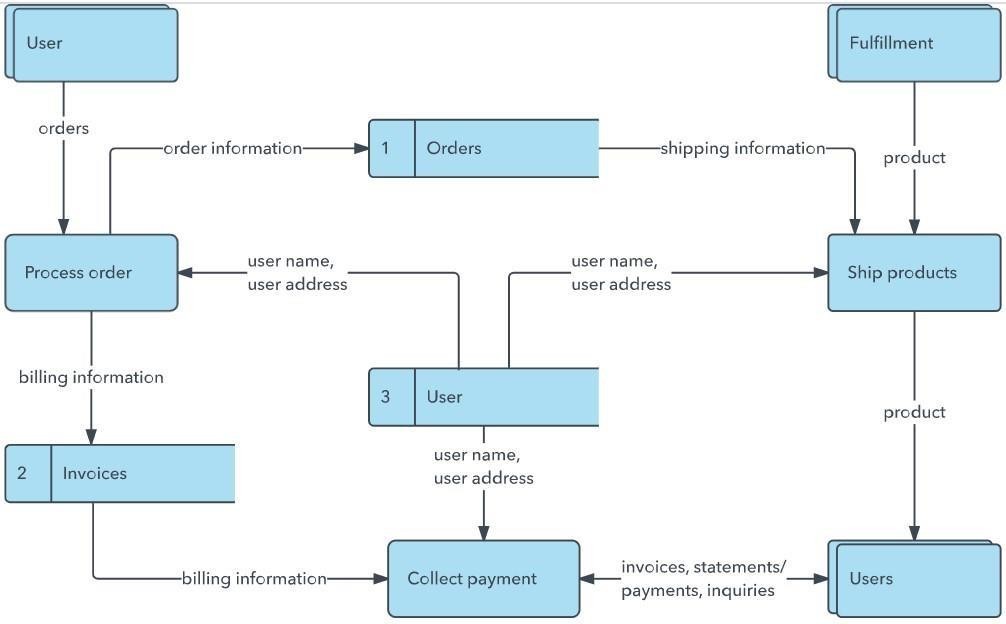
## Data Flow Diagrams

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(Simplified)

**Example: DFD Level 0 (Industry Standard)**

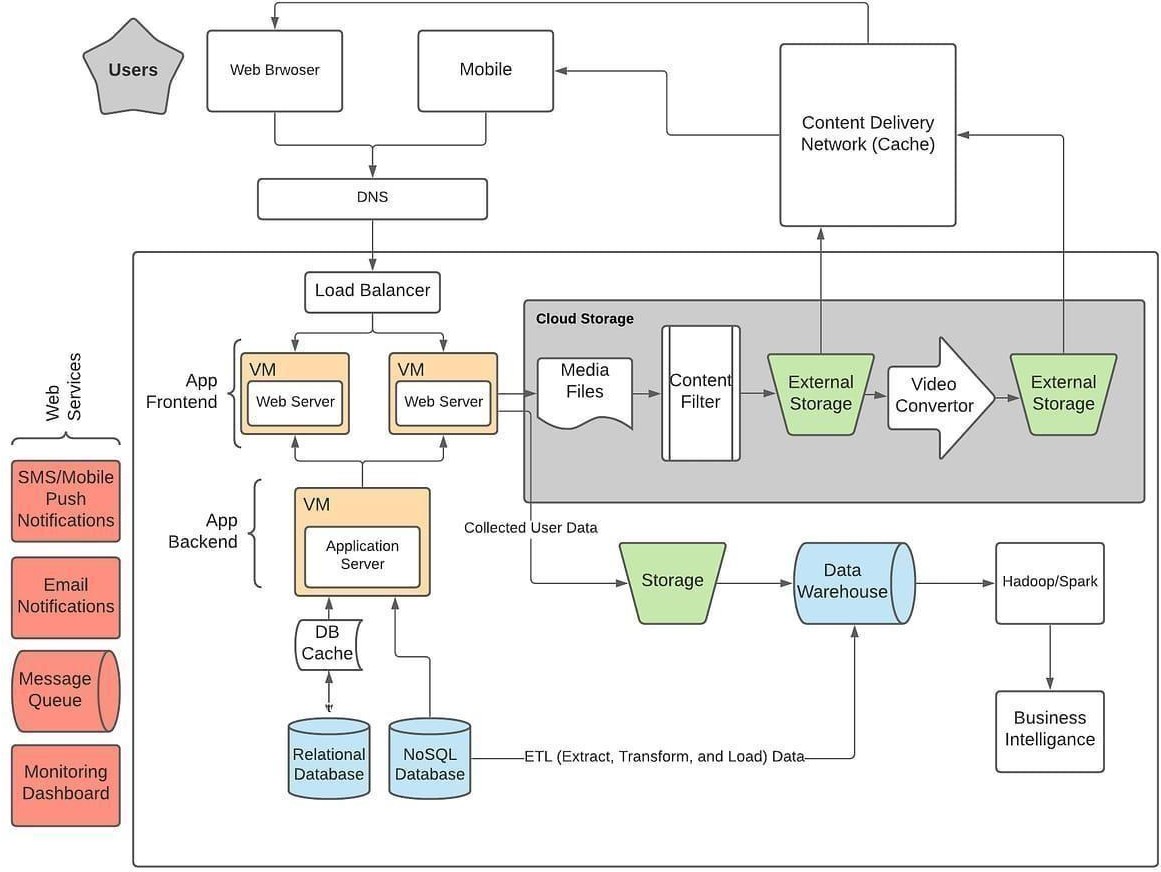


## Solution & Technical Architecture

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

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

## Example Solution Architectuire Diagram:



* 1. **User Stories**

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User Type** | **Function al Require ment (Epic)** | **User Story Numb er** | **User Story / Task** | **Acceptan criteria** |

L

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Customer (Mobile user) | Registratio n | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password. | I can acces account dashboard |
|  |  | USN-2 | As a user, I will receive confirmation email once I have registered for the application | I can re confirmatio email & confirm |
|  |  | USN-3 | As a user, I can register for the application through Facebook | I can regis access dashboard Facebook |
|  |  | USN-4 | As a user, I can register for the application through Gmail |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User Type** | **Function al Require ment (Epic)** | **User Story Numb er** | **User Story / Task** | **Acceptan criteria** |
|  | Login | USN-5 | As a user, I can log into the application by entering email & password |  |
|  | Dashboard |  |  |  |
| Customer (Web user) |  |  |  |  |
| Customer Care Executive |  |  |  |  |
| Administr ator |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Coding:** login.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial- scale=1.0">

<title>Login</title>

<style> body {

background-image: url(static/1.jpg); background-size:1550px 780px; font- family: Arial, sans-serif; background-color: #f2f2f2;

margin: 0;

padding: 0;

}

.navbar { position: fixed;

top:0;

}

.navbar ul {

list-style-type: none; margin: 0;

padding: 0; overflow: hidden;

}

.navbar li { float: right;

}

.navbar li a { display: block; color: #fff; text- align: center; padding: 14px 16px;

text-decoration: none;

}

.navbar li a:hover { background-color: #111;

}

.container { display: flex;

justify-content: center;

align-items: center; height: 100vh;

}

.login {

background-color: #fff; padding: 20px; border- radius: 5px;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1); text- align: center;

}

h1 {

margin-top: 0; font- size: 24px;

color: #333;

}

label {

display: block; margin- top: 10px; font-size: 16px;

color: #333;

}

input[type="text"], input[type="password"] {

width: 100%; padding: 10px; margin-top: 5px;

border: 1px solid #ccc; border-radius: 4px;

box-sizing: border-box;

}

button {

display: inline-block; padding: 10px 20px; margin- top: 10px;

background-color: #4CAF50; color: #fff;

border: none; border-radius: 4px; cursor: pointer;

font-size: 16px;

}

button:hover {

background-color: #45a049;

}

p {

margin-top: 10px;

}

a {

color: #4CAF50; text- decoration: none;

}

a:hover {

text-decoration: underline;

}

</style>

</head>

<body>

<div class="navbar">

<ul>

<li><a href="{{url\_for('index')}}">Home</a></li>

<li><a href="{{url\_for('login')}}">Login</a></li>

<li><a href="{{url\_for('register')}}">Register</a></li>

<l

<l

{% endwith %}

<div class="container">

<div class="login">

<form action="" method="POST">

<h1>Login</h1>

<label>Email</label>

<input type="text" placehold[er="ab](mailto:abc@exampl.com)c[@exampl.c](mailto:abc@exampl.com)om" name="EMAIL" required="">

<label>Password</label>

<input type="password" placeholder="enter your password" name="PASSWORD" required="">

<button>Submit</button>

<p>

<a href="{{url\_for('register')}}">Create a new account?</a>

</p>

</form>

</div>

</div>

</body>

</html>

## RESULTS

In this chapter we will deal with the data that the application would handle and process. After the data collection, data presentation was a really crucial aspect for this project. The sorting, arranging, processing the data to make it meaningful was a lot was crucial. Then the data had to be modified to be compatible for the databases

**6.1 . Performance Metrics**

The data was collected from various vendors, the data included a lot of information which will be used by the application to produce meaningful methodologies and results. The data obtained was all raw data and was in unmodified form that humans can understand, but this cannot be understood by the application. This data had to be modified so that the application can use the data. All the raw data was analyzed, sorted out and put under different tables, which was used by the DBA team and was pushed on to the Databases. The data here was mapped on to the application using a lot of java rules and

the Hibernate framework was used for this mapping. Hibernate usage reduced the time taken for the data entry and reduced the chance for the human induced error. the usage of Oracle SQL Developer for the data entry in to the data bases.

## ADVANTAGES

* + Allows a business to implement end-to-end quality management
  + Increased efficiencies throughout the entire product life cycle. By improving the accuracy of demand forecasts and schedules, optimising production lines and reducing costs companies are more flexible and profitable.
  + streamlining the global supply chain provides increased efficiencies throughout the entire product life cycle

**DISADVANTAGES**

* + Global supply chains are complex and involve many different stakeholders.
  + Changing a supply chain management system takes financial investment, time, and human resources. If not implemented properly, there will be wasted labor,

service redundancy, and missed deadlines that result

in significant costs.

* + Among the most common supply chain challenges in 2021 and 2022 were things like port congestion, manufacturing delays, and extreme weather events.
  + An unfavorable, inferior, or prejudicial condition.

1. **CONCLUSION**

The organization now has a fancier (or smart), more reliable and strong and healthy system which helps the organization not only the billing department, the system also helps the other departments like (amount or quantity of items stored now) management by providing the current usage details and the described/ explained analysis of the data collected from all the meters. The employee survey that was taken a week ago mentioned that the employees are much happier with this new web application and fell that the application made their work a lot easier. This data can help the organization to maintain the proper amount of (amount or quantity of items stored now) to be maintained according to the usage. This way reports helps the (amount or quantity of items stored now) team to make key decisions. The lead times were reduced by a considerable amount, customer satisfaction has increased as they get their products on time,

and can keep track of their products and plan accordingly for their demand.