

Project Planning Phase

Technology Stack (Architecture & Stack)

| | |
|---------------|------------------------------------|
| Date | 17 November 2023 |
| Team Id | Team-592467 |
| Project Name | AI Enable car parking using OpenCV |
| Maximum Marks | 4 Marks |

Technology Architecture:

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

Guidelines:

1. Include all the processes (As an application logic / Technology Block)
2. Provide infrastructural demarcation (Local / Cloud)
3. Indicate external interfaces (third party API's etc.)
4. Indicate Data Storage components / services
5. Indicate interface to machine learning models (if applicable)

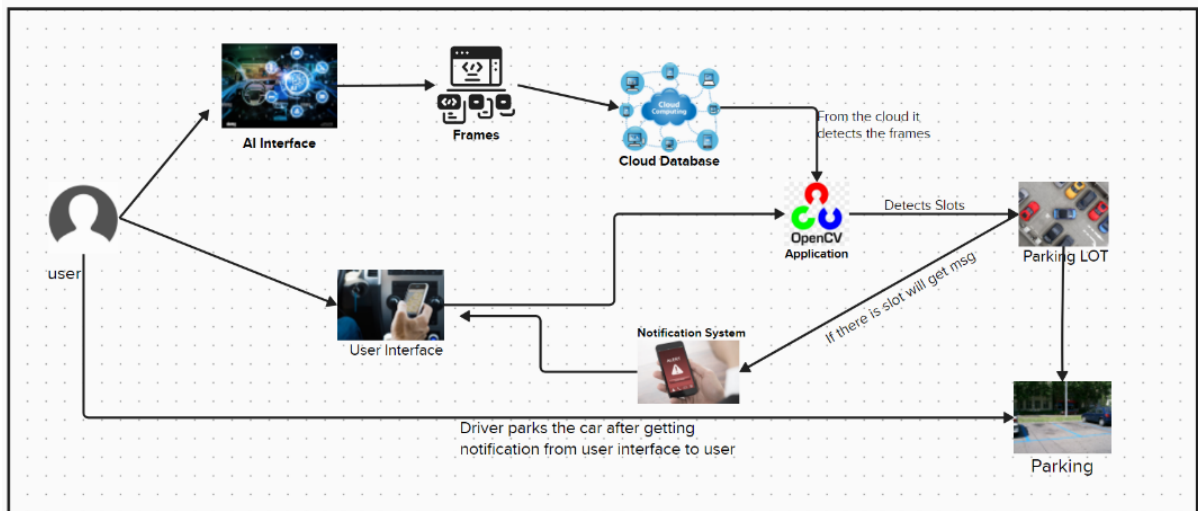


Table-1: Components & Technologies:

| S. No. | Component | Description | Technology |
|--------|-------------------------------|--|--|
| 1 | User Interface | User Interface is used by user in mobile application or In Build in car display itself | HTML, CSS, JavaScript / Angular JS / React JS etc. |
| 2 | User Logic-1 | Framework used for design the software | Python, python-flask |
| 3 | User Logic-2 | Access the software in the car by the driver to detect spot | Python, Open CV |
| 4 | Application Logic-1 | Open CV is an opensource platform for providing real time computer vision technology | Open CV |
| 8 | External API-1 | They make it easy for developers to store manage and deploy container images | Container registry |
| 9 | Machine Learning Model | Uses test and trained data images and video to learn the environment | Object recognition models, etc. |
| 10 | Infrastructure (Server/Cloud) | Application Development on Local system / cloud | Local, cloud Foundry, python-flask, etc. |

Table-2: Application Characteristics:

| S. No. | Component | Description | Technology |
|--------|-------------------------|--|--|
| 1 | Open-Source Frameworks | The utilization of open-source frameworks to build core components of the system. | Open CV, Tensor Flow, Django, MQTT, SQLite, React |
| 2 | Security Implementation | Measures implemented to ensure the security of the application and user data. | Encryption, Secure Storage, User Authentication, API Key, Firewalls, etc. |
| 3 | Scalable Architecture | Design considerations and implementations for a scalable and flexible architecture | Microservices Architecture, Load Balancing, Auto-Scaling, Elastic Scaling |
| 4 | Availability | Measures taken to ensure continuous system operation and high availability. | Redundancy, Content Delivery networks (CDN), Backup Systems and Data. |
| 5 | Performance | Techniques and strategies employed to optimize system performance. | Caching Mechanisms, Database Indexing, Content Compression, Regular Monitoring, Performance Tuning |