# 

# Section 1 - Project Description

## 1.1 Project

The project Smart Parking System is a constructive approach for decent parking space and vehicle safety using the Internet of Things (**IoT**) as both of them are of crucial importance in current scenario. **Method Analysis:** Internet of Things play an important role where everything is connected to/with everything else. Ultrasonic sensors, RFID, switch, have been widely used to overcome problems faced with vehicle parking in order to meet up with the 4IR standard and provide safety.

Already there have been studies which have came up with prototypes which shall aid user or user in finding the available parking space, reserve a parking lot, report illegal behaviour within or bad service, and make efficient use of a mobile application.

## 1.2 Description

For the purpose of parking facility, specialized sensor is used to check whether the slot is available, if so then the Boom gate (entrance) opens or else not. Fee is charged on the time basis. The automobile industry and ICT industry amalgamate, the smart vehicle is an issue. To make vehicles safe, we aim at using ultrasonic sensors and a mobile app that has details about a particular driver which are unique (like drivers’ licence) thus the identification of an authorized user becomes more and more reliable. To improve more on safety, we also use Google Map features to make it easier to locate the nearest parking lots. The above prototype is to address traffic congestion, time wastage, vehicle safety, and its current location. **Applications/Improvements**: The best part about this prototype is that it has been developed using ultrasonic sensors, servo motors, mobile app, Arduino boards, NodeMCU WIFI modules. It not only enhances communication between the two or more connected devices but also maintains the transparency between them. All these is achieved at cheaper rates.

## 1.3 Revision History

|  |  |  |
| --- | --- | --- |
| **Date** | **Comment** | **Author** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Contents**

[Section 1 - Project Description](#_yr1n1w74g294)

[1.1 Project](#_mpniepv3sji5)

[1.2 Description](#_kxtdscgtj3ad)

[1.3 Revision History](#_x1jfzxbgcdpj)

[Section 2 - Overview](#_v0z2li3lbhgp)

[2.1 Purpose](#_wkcniuopwsk)

[2.2 Scope](#_40qz41btxs84)

[2.3 Requirements](#_b5joi8kutkvr)

[2.3.1 Estimates](#_p6lz0c1jpkfa)

[2.3.2 Traceability Matrix](#_khayjwa8jo2y)

[Section 3 - System Architecture](#_yeyfpufiww8s)

[Section 4 - Data Dictionary](#_drq5ibko1acc)

[Section 5 - Software Domain Design](#_mc3s4e33quku)

[5.1 Software Application Domain Chart](#_hzjgbuitbc87)

[5.2 Software Application Domain](#_mb7tg0hm62qt)

[5.2.1 Domain X](#_1bzzrr7eu9b7)

[5.2.1.1 Component Y of Domain X](#_qt8nkuvo35yn)

[5.2.1.1.1 Task Z of Component Y1 of Domain X](#_av8bqurqa2qq)

[Section 6 – Data Design](#_wk232hcifrl1)

[6.1 Persistent/Static Data](#_quga8kcr4qx9)

[6.1.1 Dataset](#_2rfx518fnjz5)

[6.1.2 Static Data](#_9as0jpuz4i6o)

[6.1.3 Persisted data](#_ktmhbqm32m9h)

[6.2 Transient/Dynamic Data](#_kfqngvi55nuv)

[6.3 External Interface Data](#_s8ifm2hzhj88)

[6.4 Transformation of Data](#_39glqs2tdqgy)

[Section 7 - User Interface Design](#_q2p4i71pnnm7)

[7.1 User Interface Design Overview](#_92yhlpkxkggg)

[7.2 User Interface Navigation Flow](#_v5yi3wwpwf5c)

[7.3 Use Cases / User Function Description](#_547gxdax3hhf)

[Section 8 - Other Interfaces](#_t2xxswkwylyn)

[8.1 Interface X](#_3cjmeucsrpv6)

[Section 9 - Extra Design Features / Outstanding Issues](#_fv06k0tsz5p6)

[Section 10 – References](#_zigwkt1lv1o6)

[Section 11 – Glossary](#_k7zsnsc4e2v7)

# Section 2 - Overview

## 2.1 Purpose

Due to the increase in traffic, many drivers roam around in search of decent safe parking space. This not only creates traffic congestion but also time wasting. What if a driver has advance information of available parking spaces in the required destination spot.

Studies have shown illegal encroachments by vehicles parking on the roads have a massive effect of the flow in traffic [1]. The purpose of this paper is to remove the all too familiar sight of cars parked on the roads by introducing smart parking systems.

## 2.2 Scope

In this Smart Parking System, I set the ultrasonic sensor in such a way that when a car in parked in a particular slot, the ultrasonic sensors will sense the presence of a vehicle in that particular slot and update the database.

The user can view how many parking lots are available while still at home. Make reservation. The user gets real time updates on available and occupied parking spaces on the mobile applications. When the user has reached the desired parking lot, the system makes necessary updates and notifies the database. The user vehicle will remain protected there until departure time has been reached thus payment will be made by leaving the premise or by extending the stay. Upon leaving the system will also get notified and make necessary updates on the database.

## 2.3 Requirements

### The design requirements are about how the system will be built, not how it will work. Below is the required requirements in order to fully build the Smart Parking System.

* Software requirements:

1. **Arduino IDE** [2], which will be used to create commands/instructions for the microcontroller boards.
2. **Android Studio** [3], will be used to create the mobile application which is a useful tool when it comes to designing the layouts for the user interface.
3. **Firebase Database** [4], it’s a stable real-time database that will be important for making sure that the systems information get stored for better management.

* Hardware requirements:

1. **Servo motor** [5], will be used as boom gate at the entrance of SPS and the exit. They have high power and efficiency.
2. **Arduino Boards** [6], it is cheap. It comes under open source hardware, so experienced circuit designers can make their own version of the module, extending it and improving it. You can program it on Windows, Macintosh OSX, and Linux operating systems. Most micro-controller systems are limited to Windows only.
3. **NodeMCU** [7], it is also open source. Operates on low power which is good for demonstration of the project. Its able to connect to the database with ease.
4. **HC-SR04** [8], the ultrasonic sensor operates on signals which are usually in the 40- to 70-kHz range. These signals are used like radar—they’re radiated toward a target and reflected back to the source. They are low cost, flexible, and especially useful in shorter-range applications, especially automotive design.

### 2.3.1 Estimates

|  |  |  |
| --- | --- | --- |
| **#** | **Description** | **Wks. Est.** |
| 1 | Do a research on each electronic component and sensors | 1-2 weeks |
| 2 | Oder/purchase required components | 1-2 weeks |
| 3 | Designing the project | 3-4 weeks |
| 4 | Testing | 2 weeks |
|  | **TOTAL**: | 8-9 weeks |

### 2.3.2 Traceability Matrix

Cross reference this document with your requirements document and link where you satisfy each requirement

|  |  |
| --- | --- |
| **SRS Requirement** | **SDD Module** |
| Req 1 | 5.1.1 (link to module), 5.1.2 (link) |
|  |  |
|  |  |

# Section 3 - System Architecture

Describe/include a figure of the overall system architecture (and where this module fits in)

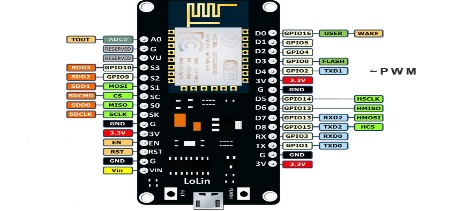


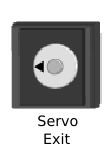
  

User



LAN



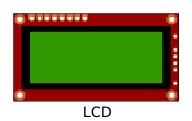


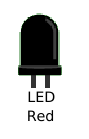
Parking Exit

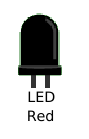


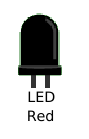
Parking Entrance

nodeMCU









Parking 3

Parking 2

Parking 1

# Section 4 - Data Dictionary

Brief description of each element in this module or a link to an actual data dictionary

(template of a database table description)

|  |
| --- |
| **Table** |

|  |  |  |
| --- | --- | --- |
| **Field** | **Notes** | **Type** |
| ID | Unique Identifier from TABLE\_SEQ | DECIMAL |
| NAME | The Name in Object.Name() | VARCHAR |
| VALUE | The Value output from somewhere | VARCHAR |

# Section 5 - Software Domain Design

## 5.1 Software Application Domain Chart

Describe / chart each major software application domain and the relationships between objects (UML, etc)

## 5.2 Software Application Domain

A Comprehensive high-level description of each domain (package/object wherever it is better to start) within the scope of this module (or within the greater scope of the project if applicable)

### 5.2.1 Domain X

A high level description of the family of components within this domain and their relationship. Include database domain, stored procedures, triggers, packages, objects, functions, etc.

#### 5.2.1.1 Component Y of Domain X

Define Component Y, describe data flow/control at component level

##### 5.2.1.1.1 Task Z of Component Y1 of Domain X

Define Task Z, describe data flow/control at task level

# Section 6 – Data Design

Describe the data contained in databases and other shared structures between domains or within the scope of the overall project architecture

## 6.1 Persistent/Static Data

Describe/illustrate the logical data model or entity relationship diagrams for the persistent data (or static data if static)

### 6.1.1 Dataset

Describe persisted object/dataset and its relationships to other entities/datasets

### 6.1.2 Static Data

Describe static data

### 6.1.3 Persisted data

Describe persisted data

## 6.2 Transient/Dynamic Data

Describe any transient data, include any necessary subsections

## 6.3 External Interface Data

Any external interfaces’ data goes here (this is for the data, section 8 is for the interface itself)

## 6.4 Transformation of Data

Describe any data transformation that goes on between design elements

# Section 7 - User Interface Design

## 7.1 User Interface Design Overview

Pictures, high level requirements, mockups, etc.

## 7.2 User Interface Navigation Flow

Diagram the flow from one screen to the next

## 7.3 Use Cases / User Function Description

Describe screen usage / function using use cases, or on a per function basis

# Section 8 - Other Interfaces

Identify any external interfaces used in the execution of this module, include technology and other pertinent data

## 8.1 Interface X

Describe interactions, protocols, message formats, failure conditions, handshaking, etc

# Section 9 - Extra Design Features / Outstanding Issues

Does not fit anywhere else above, but should be mentioned -- goes here

# Section 10 – References

Any documents which would be useful to understand this design document or which were used in drawing up this design.

# Section 11 – Glossary

Glossary of terms / acronyms