

Building Internet of Things, (IoT) based Smart Parking System for Addis Ababa City

Name of Candidate: **Anteneh Tiruneh**



A Thesis Submitted to the
Department of Computing
School of Electrical Engineering and Computing
Presented in Partial Fulfillment for the Degree of Master
of Science in Software Engineering

Office of Graduate Studies
Adama Science and Technology University

Adama, Ethiopia

January, 2018

Building Internet of Things, (IoT) based Smart Parking System for Addis Ababa City

Name of Candidate: **Anteneh Tiruneh**

Name of Advisor: **Dr. Mesfin Abebe**



A Thesis Submitted to the
Department of Computing
School of Electrical Engineering and Computing
Presented in Partial Fulfillment for the Degree of Master
of Science in Software Engineering

Office of Graduate Studies
Adama Science and Technology University

Adama, Ethiopia

January, 2018

Declaration

I hereby declare that this MSc Thesis is my original work and has not been presented for a degree in any other university, and all sources of material used for this thesis have been duly acknowledged.

Name: _____

Signature: _____

This MSc Thesis has been submitted for examination with my approval as thesis advisor.

Name: _____

Signature: _____

Date of submission: _____

Approval of Board of Examiners

We, the undersigned, members of the Board of Examiners of the final open defense by _____ have read and evaluated his/her thesis entitled “ _____ ” and examined the candidate. This is, therefore, to certify that the thesis has been accepted in partial fulfillment of the requirement of the Degree of

Supervisor /Advisor	Signature	Date
Chairperson	Signature	Date
Internal Examiner	Signature	Date
External Examiner	Signature	Date

Acknowledgement

“The fear of the LORD is the beginning of Wisdom” so I give my praise to the Almighty God, who made everything possible.

First, I would like to express my particular appreciation and great thank to Dr. Mesfin Abebe, for his indispensable effort in advising, checking and providing knowledgeable, constructive and valuable comments by the virtue of his professional experience.

I would like to give my heartfelt gratitude to Ministry of Transport workers and managers, Traffic polices, Banks and ICT professionals, Ministry of Tourism workers, managers, ministry of communication, and Information Technology interviewed individuals.

In addition, I would like to express my deepest thank to my dad, mom, sisters, brothers and friends specially Demeke Getaneh for their material and moral support!

Lastly, I would like to thank Adama Science and Technology University for education and material support.

Abstract

Getting free parking area in Addis Ababa city in the whole day especially in the morning and in the afternoon, is difficult. In the city drivers were searching parking stations without any aid; this led to wastage of fuel and time, traffic congestion, air pollution, accident and driver frustration. So, Building IoT based smart parking system architecture for real time application has significant roles to solve the above problems because IoT technology uses set of devices that have identities and virtual personalities operating in smart spaces using smart interfaces to connect and communicate with the physical environments. Interconnected devices to communicate with each other or to another machine on the network have increased the number of applications. To investigate the existing parking problems observation and interview was conducted in different areas and in different organizations in the city. After analyzing the problems smart parking system in this study was developed using ultrasonic sensor, Arduino Uno, 16× 2 LCD, GPS and EPS8266 Wi-Fi module and the hardware was simulated by proteus professional software simulator with Arduino IDE using C++programming language. The architecture of the smart parking system was developed using reference based architecture by considering hardware device and software tools. The smart parking system was implemented on Apache Cordova framework for hybrid mobile application. The web application of smart parking system was implemented and connected with mobile applications and local server implementation of hardware in each parking stations. To get the smart parking system services each local server and the central server communicated continuously about the status of parking, location of parking and users or drivers request. The driver can access parking services by installing mobile applications on their mobile phon. The smart parking system architecture, the free slot availability, the location and the speed of the system was evaluated and analyzed. The smart parking system was evaluated to check the working functionality of the system using different test cases especially in free slot identification and location identification in different parking stations and also the speed of the system was evaluated by software professionals. Based on evaluation test cases results was analyzed to achieve the goal and the objective of the smart parking system.

Keywords: internet of things, cloud computing, smart parking system, smart parking system architecture.

Table of Contents

Acknowledgement	i
Abstract	ii
List of Figures	vii
List of Tables	ix
List of Acronyms	x
CHAPTER ONE: INTRODUCTION.....	1
1.1. Background of the Study.....	1
1.2. Motivation	2
1.3. Statement of the Problems.....	2
1.4. Research Questions	3
1.4.1. General objective	3
1.5. Objective of the study	3
1.5.1. Specific objective.....	3
1.6. Methodology of the Study.....	4
1.6.1. Literature review	4
1.6.2. Methods of Data Collection	4
1.7. Scope	5
1.8. Limitations	5
1.9. Significance of the Prototype	6
1.10. Organization of the Thesis	6
CHAPTER TWO: LITERATURE REVIEW AND RELATED WORK.....	7
2.1. Literature Review	7
2.2. Overview of Internet of Things.....	7
2.3. Challenges in Developing IOT.....	8
2.4. Parking Technology of Internet of Things	10
2.4.1. Radio-Frequency Identification (RFID)	10
2.4.2. Wireless Sensor Network (WSN).....	10

2.4.3.	Global Positioning Service (GPS)	11
2.4.4.	Low Rate Wireless Personal Area Network (LWPAN)	12
2.4.5.	Near Field Communication (NFC)	12
2.5.	Internet of Things Communications Models.....	13
2.5.1.	Device-to-Device Communications.....	13
2.5.2.	Device-to-Cloud Communications	13
2.5.3.	Device-to-Gateway Model.....	14
2.5.4.	Back-End Data-Sharing Model.....	14
2.6.	Application of IOT	14
2.6.1.	Smart Environment Applications.....	15
2.6.2.	Healthcare Applications.....	16
2.6.3.	Transport Applications.....	16
2.6.4.	Smart parking System Applications.....	17
2.7.	Related Work.....	18
CHAPTER THREE: METHODOLOGY		21
3.1.	Design Methodology	21
3.2.	Hardware Tools	24
3.2.1.	Mobile Phone	25
3.2.2.	Arduino	25
3.2.3.	Sensors	25
3.2.4.	LCD display	25
3.2.5.	Global Positioning System.....	26
3.2.6.	ESP8266.....	26
3.3.	Software Tools	26
3.4.	Design Method	27
3.4.1.	Qualitative Research	28
3.4.2.	Sampling Techniques.....	28
3.4.3.	Data Collection	29
CHAPTER FOUR: HARDWARE TOOLS SELECTION FOR SPS		36

4.1.	Overview	36
4.2.	Breadboard Comparison.....	36
4.3.	Board Selection	37
4.3.1.	Common Arduino Uno pins and functions	38
4.4.	Sensor selection.....	41
4.4.1.	Characteristics of Sensors	41
4.4.2.	Choosing of Sensor to BSPS.....	42
4.5.	Tracking Device	45
4.5.1.	Passive GPS Tracking System	46
4.5.2.	Active GPS Tracking System	46
4.5.3.	Types of GPS Tracking Unit.....	46
4.5.4.	GPS Position Determination	47
4.6.	Display Device Selection	48
4.6.1.	Light-Emitting Diode.....	48
4.6.2.	Seven-Segment Display	48
4.6.3.	Liquid Crystal Display (LCD)	49
4.6.4.	Criteria to Selected Display Devices	49
4.7.	Arduino simulator selection	51
CHAPTER FIVE: BUILDING SMART PARKING SYSTEM ARCHITECTURE		52
5.1.	Architecture of Smart Parking System.....	52
5.2.	Architectural views and perspectives	53
5.2.1.	Architectural views	53
5.2.2.	<i>Architectural perspective</i>	54
5.3.	Architecture Guideline Principles	54
5.4.	Reference Architectures	55
CHAPTER SIX: IMPLEMENTATION OF SMART PARKING SYSTEM		63
6.1	Implementation of Smart Parking System	63
6.2	Simulation of Smart Parking System	63

6.3	Free Slot Identification of Smart Parking System.....	64
6.3.1.	Circuit Design of Slots Identification in Smart Parking System	66
6.4.	Location Identification of Smart Parking System	69
6.4.1.	Circuit design for Location identification of smart parking system	69
6.5.	Send Data from Arduino Uno to Database	70
6.6	Mobile Application Implementation of Smart Parking System.....	71
6.6.1.	Hybrid Application	71
6.7.	Web application Implementation of Smart Parking System	74
6.7.1.	Web Service Applications.....	78
6.8.	IOT Platforms of Smart Parking System	79
6.8.1.	Thing Speak	80
6.8.2.	Wovyn.....	80
6.8.3.	IBM IoT	81
6.8.4.	AWS IoT.....	81
CHAPTER SEVEN: EVALUATION AND RESULT ANALYSIS OF SMART PARKING SYSTEM.....		83
7.1.	Architecture Evaluation of Smart Parking System (AESPS).....	83
7.2.	Slot Identification Evaluation of Smart Parking System (SIESPS)	83
7.3.	Location identification Evaluation of Smart Parking System (LIESPS)	84
7.4.	Speed Evaluation of Smart Parking System (SESPS).....	85
7.5.	Test Design.....	86
7.6.	Result Analysis.....	87
CHAPTER EIGHT: CONCLUSIONS, RECOMMENDATION AND FUTURE WORKS		88
8.1.	Conclusions	88
8.2.	Recommendation.....	89
8.3.	Future work	89
References.....		90
Appendixes		94

Appendix 1. Interview questions.....	94
Appendix 1.1. Interview question for Ministry of Transport	94
Appendix 1.3. Interview Question for Traffic police	94
Appendix 1.4. Interview question for Drivers	94
Appendix 1.5. Interview Question for Students	94
Appendix 2. Location Parking Station Sample Code.....	95
Appendix 3. Arduino and MYSQL connection sample code	98
Appendix 4. Arduino Sample Code to Select Data from Database	100
Appendix 5. MYSQL and Arduino sample update slot status web client code	101
Appendix 5.1. Sample php code to update slot status.....	103
Appendix 6. Parking station java script sample code.....	103
Appendix 7. Admin login and signup page sample code	105

List of Figures

Figure 1.1 IoT communication	1
Figure 2.1. IOT interaction	7
Figure 2. 2 Classification of IOT application	15
Figure 2. 3. Transport Applications [2]	17
Figure 2. 4. Smart parking system concept [16]	17
Figure 3. 1. Aspects of design in IoT (Rowland et al. 2015).....	21
Figure 3. 2. 10 Step methodologies for IoT development (Bahga and Madiseti, 2014).....	22
Figure 3. 3. Design methodology for IoT development (Parker, 2015)	23
Figure 3. 4. Arduino IDE interface	27
Figure 4. 1. Arduino UNO pins [26].....	38
Figure 4. 2. Ultrasonic Sensor.....	44
Figure 4. 3. Working principle of Ultrasonic sensor [32].....	44
Figure 4. 4. Distance between the earth and the satellite [source Google]	48
Figure 5.1. General overview of SPS architecture (my own work).....	58
Figure 5. 2. Architecture of Smart Parking System (my own work)	60
Figure 6. 1. Proteus Simulator	64
Figure 6. 2. free slot identification.....	65

Figure 6. 3. Ultrasonic sensor and Arduino Uno circuit design.....	66
Figure 6. 4. Location tracking circuit design	69
Figure 6. 5. Mobile application home screen.....	73
Figure 6. 6. Sample snapshot JS code to get parking stations	74
Figure 6. 7. Status of parking Slots.....	75
Figure 6. 8. Sample parking reservation in Megenagna parking	76
Figure 6. 9. Database Design of smart parking system.....	77
Figure 6.10. web service application	79
Figure 6. 11. ThingSpeak platform local system and mobile application communication.....	82

List of Tables

Table 3. 1. Ministry of transport Interview question results.....	30
Table 3. 2. Interview question results from Traffic Police	30
Table 3. 3. Interview question results from drivers	31
Table 3. 4. Road Areas Observation	32
Table 3. 5. Cinemas and Theaters place observation.....	33
Table 3. 6. Observation of parking places	34
Table 4. 1. Comparison of breadboard.....	36
Table4. 2. Sensor Characteristics [27]	41
Table 4. 3. Comparison of Sensors	42
Table 4. 4. Speed of Sound in different Medium.....	43
Table 4. 5. Display Device Comparison	49
Table 4. 6. LCD pins and its purpose.....	50
Table 6. 1. Ultrasonic sensor and Arduino Uno circuit design pins connection.....	67
Table 6. 2. Location tracking circuit design pins connection	70
Table 7. 1. Slot Identification Evaluation result of Smart Parking System	83
Table 7. 2. Location identification Evaluation result of Smart Parking System.....	85
Table 7. 3. Speed Evaluation result of Smart Parking System	86
Table 7. 4. Whole system users' evaluation result.....	87

List of Acronyms

2D	Second Generation of Mobile Telecommunication Technology
3D	Third generation of Mobile Telecommunication Technology
4G	Fourth Generation of Mobile Telecommunication Technology
5D	Fifth Generation of Mobile Telecommunication Technology
ADC	Analog-to-Digital Converter
ADC	Analog to Digital Converter
ALG	Application-Layer Gateway
ALG	Application-Layer Gateway
ALG	Application-Layer Gateway
API	Application Programming Interface
APP	Application
AREF	Analog Read Reference
ARM	Adjustable Rate Mortgage
AT	Attension
ATMEL	American-based Designer and Manufacturer of Semiconductor
AVR	Advanced Virtual Risc
BSPS	Building Smart Parking System
CPU	Central Processing Unit
CSS	Cascading Style Sheet
EEPROM	Electrically Erasable Programmable Read Only Memory
EPRAM	Erasable Programmable Random-Access Memory
ESP8266	Espressi
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communication
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol
IAB	Internet Architecture Board

IAB	Internet Architecture Board
ICSP	In Circuit Serial Programming
ICSPC	In circuit Serial Programming
IDE	Integrated Development Environment
IOT	Internet of Things
IP	Internet Protocol
IPV4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
JSON	JavaScript Object Notation
LCD	Liquid crystal display
LED	Light emitting diode
LWPAN	Low Rate Wireless Personal Area Network
M2M	Machine to Machine
MHZ	Megahertz
MQTT	Message Queue Telemetry Transport
NFC	Near Field Communication
NFC	Near Field Communication
OSI	Open System Interconnection
OSI	Open System Interconnection
PHP	Hypertext Preprocessor
PWM	Pulse Width Modulation
REST	Representational State Transfer
RFID	Radio Frequency Identification
RTC	Real Time Counter
RX	Receive
SD	Secure Digital
SD	Secure Digital
SIM	Subscriber Identity Module
SMS	Short Message Service
SMS	Send Message Service
SPI	Serial Peripheral Interface

SRAM	Static random-Access Memory
TCP	Transport Control Protocol
TCP	Transport Control Protocol
TLS	Security Link Encryption
TX	Transmitter
UART	Universal Asynchronous Receiver-Transmitter
UMTS	Universal Mobile Telecommunication System
URL	Uniform Resource Locator
USART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
USB	Universal Serial Bus
VIN	Voltage
VLR	Visitor Location Register
Wi-Fi	Wireless Fidelity
WPAN	Wireless Personal Area Networks
WSN	Wireless Sensor Network

CHAPTER ONE: INTRODUCTION

1.1. Background of the Study

Technologies are growing dramatically in day to day to give smart services for the society. IoT is one of these technologies that give smart services by interconnecting the physical and virtual world. IoT also increases the interconnection of living and non-living things for unified communications infrastructure supporting a wide range of remote access, content streaming, interactive connectivity and information sharing services. By the Internet of Things, objects recognize themselves and obtain intelligence behavior by making or enabling related decisions thinks to the fact that they can communicate information about themselves [1]. These objects can access information that was collected by other things, or they can add to other services [2]. As shown in Figure1 the internet of things, anything's can able to communicate to the internet at any time from any place to provide any services by any network to anyone [3]. This concept will create a new type of applications can involve such as smart transportation, the smart home, smart parking, smart health, smart city and so on to provide many services such as notifications, security, energy saving, automation, communication, and entertainment [3] [4].

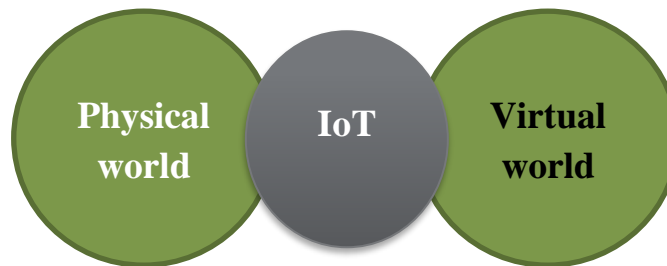


Figure 1. 1 IoT communication

So, choosing IoT technology to develop smart parking system in real-time application and machine to machine communication (mobile application of smart parking users and parking station servers) was a main objective. These communications were needed in the pick time especially in urban area drivers; since drivers are wasted fuel and time for searching parking stations. This leads traffic congestion and air pollution, so that building smart parking system (SPS) is the best solution to reduce the problems. Anyone can access information about the availability of free space, payment, and location of parking area within Addis Ababa city using mobile phone. The Smart parking

system can provide real-time information to drivers or users for satisfaction and safety for them and their cars.

1.2. Motivation

In Addis Ababa, the number of vehicles increases day to day, leads denser urban mobility. In the city especially, every day in the morning and every day in the afternoon the road is crowded by peoples and cars and many cars are parked in the roadside. The numbers of cars increment in the city leads to shortage of parking space in government, private and public areas. As a result, drivers are forced to park in road side. The roadside parking then leads traffic congestions, air pollution and wastage of time and resource and accident. To minimize this problem one solution was avoiding roadside parking and building smart parking system for private organization. The second motivation is no new technology related to parking is adapted in the city. Still, the drivers in Addis Ababa are manually searching or looking for available parking area in the city for many hours. For example, if strangers who are unfamiliar to the city may need parking, he/she may have got limited or misinformation or they watched parking sign so, they lost their time and fuel and it was a way stolen by theft. To minimize traffic congestions, air pollution and wastage of time and resource and accident smart parking system is one solution because drivers were not search parking stations they can easily access available parking services using mobile phone.

1.3. Statement of the Problems

According to the vision of the IoT, devices on the network are quickly growing to trillions of small-embedded devices in the physical world, and that increases challenges associated with building smart parking system. IoT has ability to identifying, analyzing, sense and communicating with the physical device and the environments. The communication ability of IoT device with each other improves the new services and information delivery. IoT is one of the prominent technologies in vehicle-vehicle, machine- machine, and machine-environment communications especially in smart parking system to deliver real-time information to users or drivers. Building smart parking system in Addis Ababa city using IoT was to minimize traffic congestion, pollution problems, accident problems, resource and time wastage, driver's frustrations, parking stations environmental behaviors, instrumental behaviors like sensors. After analyzing the behavior of the environment and materials the main concern of this study will be to find solutions for the above-

mentioned problems using IoT technology because IoT technology has an ability to identify an object by giving virtual personality and identities so any parking infrastructures, any parking environments and any parking services is analyzed easily in efficient and convenient manner

1.4. Research Questions

Aforementioned challenges and problems occurred in Addis Ababa with respect to smart parking system needed analytically assessed to provide efficient and affordable solutions. Therefore, the research is intended to get answers for the following research questions.

- What are the major challenges and problems of parking in Addis Ababa city?
- What is the main benefit of building smart parking system in Addis Ababa city?
- Who will be beneficiary in the smart parking system?
- Is it possible to solve parking problems using IOT technology?
- Is it possible to deliver service in real-time for parking user?

1.4.1. General objective

The general objective of the research is to develop a framework and building a prototype internet of things based smart parking system in Addis Ababa city to provide real time information to users/ drivers about the status and the location of parking.

1.5. Objective of the study

1.5.1. Specific objective

To accomplish the above stated general objective, the following specific objectives were carried out.

- To survey literature on IoT, reference based architectures, IoT and cloud computing integration, smart parking system and synthesize for each paradigm according to the purpose of this work. Different applications of parking been critically and comparatively evaluated using the IoT system consideration from a theoretical point of view.
- To allow the user to find the exact location of the vacant parking lot and mark their parked car using the mobile phone.

- Analyzing and modeling the SPS used in the real life especially in Addis Ababa city.
- Be able to provide the user with all the necessary information about the status of any changes in the parking area that might happen in real time
- To develop prototype and architecture for smart parking system for Addis Ababa city
- Evaluating and testing system applications, effectiveness and performance.

1.6. Methodology of the Study

In order to achieve the objectives of the study and to answer the research questions, the following research methods are used.

1.6.1. Literature review

Relevant literature such as: books, journal articles, conference papers and resources from internet that are been related to car parking system and manuals of development tools and techniques were reviewed. In addition to this, documents related to parking area in Addis Ababa city have been reviewed. Moreover, to assess what others have done in the area and to understand the problem better, related research works and projects assessed. In general, comprehensive investigations of available literature on different techniques of parking system been made.

1.6.2. Methods of Data Collection

To acquire the needed knowledge, both secondary and primary (documented and undocumented) sources of knowledge are used. Primary knowledge is gathered from Ministry of Transport, Traffic Police, Ministry of Tourism, drivers and other users in Addis Ababa by using interviewing and analyzing knowledge elicitation methods. The secondary sources of knowledge are collected by using document analysis.

Interview (both structured and unstructured) is used to collect tacit knowledge from the domain experts. In addition, analyzing elicitation methods are be used to purify the collected knowledge. The acquired knowledge is refined with the consultation of the experts. Moreover, secondary sources of knowledge are gathered from the Internet, documents from parking area.

1.7. Scope

The smart parking system (SPS) was only limited on private parking not include governmental and public parking areas and SPS was developed for small cars only. The smart parking system analysis the parking stations of the environmental factors, the quality of devices used in each parking stations and also, we were analysis the behaviors of parking environments for the suitability of sensors and other hardware's devices. We were conducted experiments to select hardware tools to build smart parking system. To get smart service from each parking stations machine to environment, machine to machine and machine to people communications were created between parking stations and users. Based on the objectives and goals reference architecture of smart parking system was built by decomposing services into components. The smart parking system was simulated, implemented, evaluated and tested before deployed in the private cloud. After deployed the smart parking system users or drivers can access parking services using mobile phone. The main services of smart parking system were identifying parking location, parking distance and checking status of parking lots in each parking stations in the city. The performance of smart parking system was evaluated using request and response time in the real-time application.

1.8. Limitations

The smart parking system was not considering types of parking like parallel, perpendicular and soon which may have an effect on degree of Ultrasonic sensor installation because all the parking stations were not build based on international parking rules. The most services in the SPS were only location services, parking free slots in each area in the city and parking payment services but not include path navigation so the driver may waste time if the road he/she took was crowded and loss of revenue for parking owners because the payment system was worked pay- as- use services. But the smart parking system can compare the longest distance from the current location of users to the location of smart parking system using smart mobile phone on Google Map. The smart parking system support smart phones only; so that if users or drivers have non-smart phone he/she did not get the smart parking services. In case of slot identification if an object was present in front of the Ultrasonic sensor, the object can be detected like a car and the system displays to local server as the slot is taken.

1.9. Significance of the Prototype

Building smart parking system is beneficial for drivers, tourist, park owners, park operators, Ministry of transport and other citizens. Drivers can save fuel consumption and wastage of time for seeking vacant park. The park operator can easily manage and get full information from SPS. Park owners can increase profits, Ministry of transport can access free flow of traffics, due to parking reduces traffic congestions also other citizens can walk freely in the side of street. This system also reduces air pollution for all Addis Ababa city populations and nearby.

1.10. Organization of the Thesis

Chapter 1 introduces the concept and the focus of this work. *Chapter 2* presents the overviews of IOT, challenges, communication models and applications of society, environments and industries. *Chapter 3* presents the methodologies, method, tools used to build reference based architecture of smart parking systems. *Chapter 4* presents hardware tools selections and compilations to develop the smart parking system. *Chapter 5* presents the architectures of smart parking system using reference based considering layer division. *Chapter 6* presents implementations of smart parking simulation, slot identification, location identifications, mobile application, local server and central server implementations of the smart parking system. *Chapter 7* presents evaluation of smart parking system architecture, slot identifications, location identifications and speeds of the whole system. *Chapter 8* presents the conclusion, recommendation and future works on the smart parking system.

CHAPTER TWO: LITERATURE REVIEW AND RELATED WORK

2.1. Literature Review

Literature in the area of parking system-using IOT was been reviewed and discussed. The review covers overview of internet of things, introductions, application areas, challenges in developing IOT, classification of parking, IOT communication models and discussion on major approaches that have been employed for smart parking system research with special focus on internet of things. Review facilitates the comparison of existing parking approaches to the specific solutions that are been employed in this study.

2.2. Overview of Internet of Things

In the digital world, especially the computer communication starts with sharing data between machine to machine, and it moves to machine to infrastructure, then machine to environment, and machine to people but now internet is everything [5]. The people also want to communicate with all non-living things through internet. IOT is a solution to solve people to non-living thing communications. The people already have many technologies to interact with living things but IoT enables to communicate with non-living things with comfort manner. “IoT is a merging of several technologies like cloud computing, ubiquitous, pervasive computing, Ambient Intelligence, Sensors, Actuators, Communications technologies, Internet Technologies, Embedded systems” [6][37]. In the smart parking concept all things can easily communicate using IoT technologies.

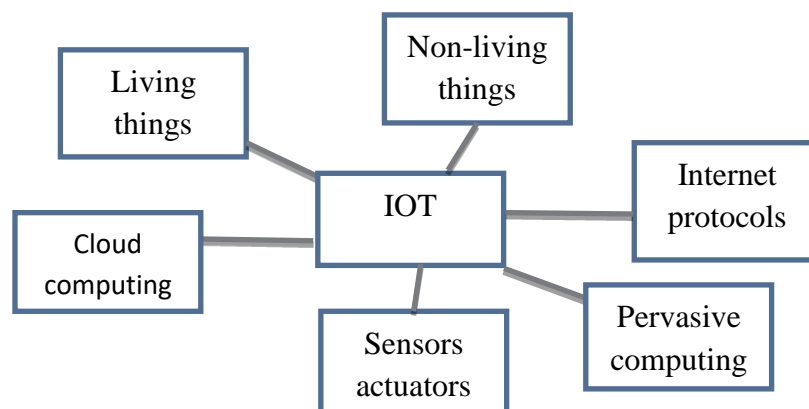


Figure 2.1. IOT interaction

Living and non-living things was communicated using sensors and actuators. People's cloud interacts with the physical world using IOT devices. The users manipulate the data through the physical component. Cloud, ubiquitous / pervasive computing, and Internet protocols are used to create communication among the devices and to manage the high-end user interactions [5].

2.3. Challenges in Developing IOT

Before building IOT based smart parking system first we were going to discuss some of the challenges of IOT. Most challenges come from technological, social, legal, financial, and business backgrounds in order to receive wide acceptance by the IoT community [5]. There are some challenges to the application of the Internet of Things concept in cost of implementation. IoT are also faced with many other challenges [6], such as:

- **Scalability:** Internet of Things has a big concept than the conventional Internet of computers, because things are cooperated within an open environment. Basic functionality such as communication and service discovery [6], therefore need to function equally efficiently in both small scale and large-scale environments. The IoT requires a new functions and methods in order to gain an efficient operation for scalability.
- **Complexity:** Multiple platforms, numerous protocols and large numbers of applications, IoT systems integration and testing will be a challenge to say the least [2]. The confusion around evolving standards is almost sure to slow adoption. Slower adoption and unanticipated development resource requirements will likely slip schedules and slow time to revenues, which will require additional funding for IoT research and longer runways for startups [8, 9].
- **Standards and interoperability:** Standards and interoperability Standards are important in creating markets for new technologies [8]. If devices from different manufacturers do not use the same standards, interoperability will be more difficult, requiring extra gateways to translate from one standard to another.
- **Security and privacy:** In addition to the security and protection aspects of the Internet such in communications confidentiality, the authenticity and trustworthiness of communication partners, and message integrity, other requirements would also be important in an Internet

of Things [7] [8]. There is a need to access certain services or prevent from communicating with other things in IoT and also business transactions involving smart objects would need to be protected from competitors' prying eyes. IoT connects more devices together; it provides more decentralized entry points for malware. Less expensive devices that are in physically compromised locales are more subject to tampering. More layers of software, integration middleware, APIs, machine-to machine communication, etc. create more complexity and new security risks. Expect to see many different techniques and vendors addressing these issues with policy-driven approaches to security and provisioning [8]. Information security is regarded as preserving:

- ***Trust and Privacy***: With remote sensors and monitoring a core use case for the IoT, there will be heightened sensitivity to controlling access and ownership of data [2]. New compliance frameworks to address the IoT 's unique issues will evolve. Social and political concerns in this area may also hinder IoT adoption. Privacy & data and information security are complementary requirements for IoT services.
- ***Automatic Discovery***: In dynamic environments, suitable services for things must be automatically identified, which requires appropriate semantic means of describing their functionality [6].
- ***Software complexity***: A more extensive software infrastructure would have needed on the network and to manage the smart objects and provide services to support them [6]. That because the software systems in smart objects will have to function with minimal resources, as in conventional embedded systems.
- ***Fault tolerance***: Objects in internet of things is much more dynamic and mobile than the internet computers, and they are in changing rapidly in unexpected ways. Structuring an Internet of Things in a robust and trustworthy manner would require redundancy on several levels and an ability to automatically adapt to changed conditions [6].
- ***Addressing schemes***: The ability to uniquely identify things is critical for the success of IoT. This will not only allow us to uniquely identify billions of devices but also to control remote devices through the Internet [2]. The few most critical features of creating a unique address are uniqueness, reliability, persistence and scalability. Every element that is

already connected and those that are going to be connected must be identified by their unique identification, location and functionalities [2]. The current IPv4 may support to an extent where a group of cohabiting sensor devices can be identified geographically, but not individually.

- **Wireless communications:** From an energy point of view, established wireless technologies such as GSM, UMTS, Wi-Fi and Bluetooth are far less suitable; more recent WPAN standards such as ZigBee and others still under development may have a narrower bandwidth, but they do use significantly less power [6].

2.4. Parking Technology of Internet of Things

Different researchers used different technologies to develop parking system. Before building SPS, it is better to see technologies that support IOT.

2.4.1. Radio-Frequency Identification (RFID)

RFID including readers and tags has a vital task in the framework of the IoT [8]. Employing the technologies on each related thing, accomplishing their automatic identification and dedicating the single digital identity to any of the things will be possible, to include the network associated with the digital information and services. RFID provides some applications in smart grids, including tracking and localization of objects, healthcare applications, parking lots and asset management. Each tag can be as a sensor because they have not only data which is written manually but also capture data like environmental information [8].

2.4.2. Wireless Sensor Network (WSN)

Wireless sensor network makes various proper data available and might been applied in many uses like transportation (smart parking, traffic congestions, lighting, etc.), healthcare, as well as government and environmental services [9]. Moreover, WSNs can be collected with RFIDs to obtain several targets such as gaining data related to the location of people and objects, movement, temperatures, etc. A Wireless sensor network contains of wireless sensor nodes which embrace a radio interface, an analog-to-digital converter (ADC), multiple sensors, memory and a power supply [10]. WSNs are regarded as an innovative information gathering method to build the information and communication system which will significantly progress the reliability and

efficiency of infrastructure systems. Compared with the wired solution, WSNs feature easier deployment and better flexibility of devices. With the rapid technological development of sensors, WSNs will become the key technology for IoT. WSN is a collection of distributed sensors that monitor physical or environmental conditions, such as temperature, sound, and pressure. Data from each sensor passes through the network node-to-node. [2]. Nowadays vehicles are using modern electronic technologies such as GPS, wireless sensors and accumulators, that help a lot in improving the driver parking skills and reduce traffic congestions, air pollutions, resource, and time and car risk.

2.4.3. Global Positioning Service (GPS)

The growing of popularity of mobile devices and global positioning service (GPS) need for location-based applications have increased [11]. GPS is available to all anywhere and any weather condition without any controls. If the driver is having GPS receiver in his phone, then he/she can easily determine his/her current position on earth. GPS receiver told the exact position of the car and parking area. The GPS represented the position in terms of latitude and longitude values. The GPS tracker depends on the location of a mobile phone. It is used as an IP service that uses geographic information in order to track the location of the park. The park is tracked and sent to the server and the server then forwards this tracked park information to the client device which makes the user keep track of the park location and get an estimate remaining time for the user to reach his/her parking. It is a widely used application in mobile data services which has led to the rapid development in wireless communication strategies as well as location positioning technologies. The user having the location-aware mobile phones can find out the nearest and the lowest cost parking place in Addis Ababa city. Drivers can use a smartphone application to view a real-time map of parking spaces, updated every minute, showing where there are empty parks, and directing to the nearest one. This smart phone application will reduce the time spent driving around looking for a parking space and the congestion on in Addis Ababa city roads. A GPS tracking system can work in various ways i.e. Active and passive tracking. In Passive tracking, the position is usually stored in internal memory or on a memory card along the ride, while in the active tracking, also refers to a real-time tracker; data was transmitted to central database via a modem within the GPS unit [12].

2.4.4. Low Rate Wireless Personal Area Network (LWPAN)

LWPAN is amongst short-range radio technology that covered large distances of up to 10–15 km. The energy consumption of this technology is extremely low and battery lifetime is about 10 years [10, 12]. According to the IEEE 802.15.4 standard, it provides low cost and low-rate communication for sensor networks. It has the lowest two layers of protocols including physical and medium access level, besides upper layers' protocols including 6LoWPAN and ZigBee [13].

2.4.4.1. ZigBee

ZigBee was applied as a low-power and low-cost wireless communication technology [13]. It is based on the IEEE 802.15.4 standard and it was suitable for creating wireless personal area networks (WPAN) such as home automation, medical device collection and other low-power, low-bandwidth. Some of its applications include wireless light switches, electrical meters, and traffic management systems. ZigBee is suitable for limited ranges, coverage of city region and supporting billions of devices. With the ZigBee-based network, a mechanism for transmission of IPv6 packets was been specified. To apply ZigBee, additional equipment usually is required involving a coordinator, router and ZigBee end-devices [8].

2.4.4.2. 6LoWPAN

The 6LoWPAN standard was specified to adapt IPv6 communication [8]. IPv4 was been a leading addressing technology. which replaced by IPv6 due to the exhaustion of its address blocks and the inability to separately address billions of nodes which is a characteristic of IoT networks, by providing 128-bit addresses solves the lack of enough nodes for IoT networks, but it creates another problem however, which is compatibility with constrained nodes. This problem was addressed by 6LoWPAN, which is the compression format for IPv6.

2.4.5. Near Field Communication (NFC)

Near Field Communication (NFC) is bidirectional short distance communication, especially in smart-phones [8]. This range usually involves a centimeter range. The application of NFC in smartphones enables us to use it in smart cities and smart transportation especially smart parking. One of its applications includes using smartphones with NFC as a wallet, which enables us to use smartphones as our personal cards such as bankcard, identification card, public transportation card,

access control cards. Moreover, since NFC is bi-directional, it can be used to share data between devices, multimedia, and documents [13].

2.5. Internet of Things Communications Models

Before building SPS it was necessary, differentiate different IOT communication models to build the architecture and prototype of smart parking system in Addis Ababa city. From an operational perspective, it is useful to think about how IoT devices connect and communicate in terms of their technical communication models [14]. In March 2015, the Internet Architecture Board (IAB) released a guiding architectural document for networking of smart objects [14] which outlines a framework of four common communication models used by IoT devices.

2.5.1. Device-to-Device Communications

The device-to-device communication model represents two or more devices that directly connect and communicate between one another, rather than through an intermediary application server. These devices communicate over many types of networks, including IP networks or the Internet. These device-to-device networks allow devices that adhere to a particular communication protocol to communicate and exchange messages to achieve their function. This communication model is commonly used in applications like home automation systems [14], which typically use small data packets of information to communicate between devices with relatively low data rate requirements. Residential IoT devices like light bulbs, light switches, thermostats, and door locks normally send small amounts of information to each other.

2.5.2. Device-to-Cloud Communications

The IoT device connects directly to an Internet cloud service like an application service provider to exchange data and control message traffic. This approach frequently takes advantage of existing communications mechanisms like traditional wired Ethernet or Wi-Fi connections to establish a connection between the device and the IP network, which ultimately connects to the cloud service. Device-to-cloud model adds value to the end user by extending the capabilities of the device beyond its native features.

2.5.3. Device-to-Gateway Model

The device-to-application-layer gateway (ALG) model, the IoT device connects through an ALG service as a conduit to reach a cloud service. In simpler terms, this means that there is application software operating on a local gateway device, which acts as an intermediary between the device and the cloud service and provides security and other functionality such as data or protocol translation [15]. In many cases, the local gateway device is a smartphone running an app to communicate with a device and relay data to a cloud service. Model employed with popular consumer items like personal fitness trackers. These devices do not have the native ability to connect directly to a cloud service, so they frequently rely on smartphone app software to serve as an intermediary gateway to connect the fitness device to the cloud [15].

2.5.4. Back-End Data-Sharing Model

The back-end data-sharing model refers to a communication architecture that enables users to export and analyze smart object data from a cloud service in combination with data from other sources. This architecture supports desire for granting access to the uploaded sensor data to third parties. IoT devices upload data only to a single application service provider [15] a back-end sharing architecture allows the data collected from single IoT device data streams to aggregate and analyze.

2.6. Application of IOT

Internet of Things was shifting individuals' quality of life and increase productivity by a step using widely distributed, locally intelligent network of smart devices, the IoT has the possible to enable payments and improvements to vital services in transportation, parking, logistics, security, environment, education, healthcare and other areas, while providing a new environment for application development. However, IOT application in the smart parking system was been divided in two three by considering the effects of roadside parking and searching free parking area for a long time.

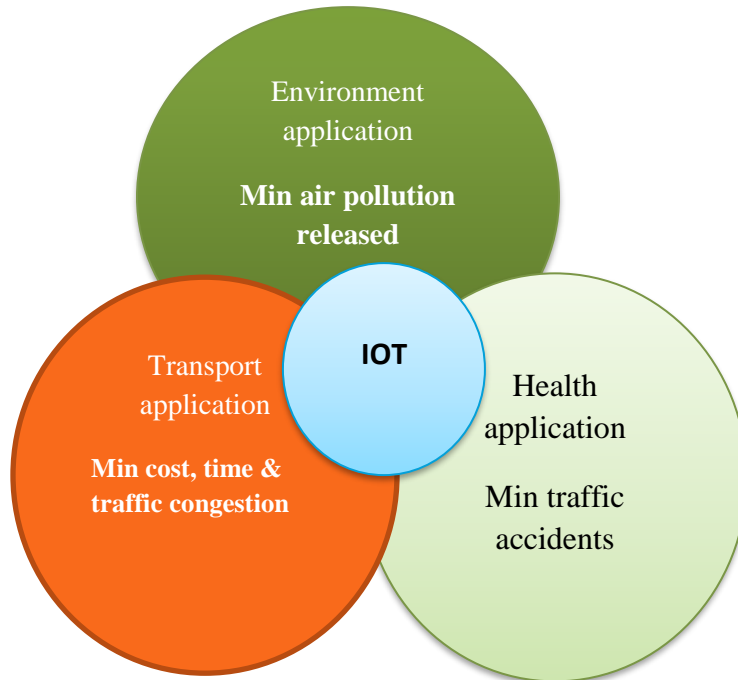


Figure2. 2 Classification of IOT application

2.6.1. Smart Environment Applications

Environment plays a major effect in human life. People, even animals, and plants may be affected in unhealthy environment [2]. There were many researches efforts has been paid to solve the problems of environmental pollution and waste resources [2, 4]. Creating of a healthy environment is not easy because of industries and transportations wastes, with irresponsible human activities are daily factors that make the environment damaged [2, 5]. The environment needs a smart ways and new technologies for monitoring and management. Monitoring the environment is important in order to assess the current condition of the environment to takes correct life decision according to collected data from monitoring systems, and management was needed to have an efficient resource consuming and use in addition to decrease the factories and vehicles wastes. Smart environment is an important technology in our everyday life which provides many facilities and solutions for many environmental applications such as water and air pollution, weather and radiation monitoring, waste management, natural disaster and all may have connected to each person through home area network. Smart environment devices integration with Internet of Things (IoT) technology was been developed for tracking, sensing and monitoring objects of environment, which provide potential benefits to achieve a green world and sustainable life [2]. The IoT for

environmental monitoring can aid in measuring emissions from factories, cars and others industry [2]. [15] Environmental application in case of smart parking system drivers can easily accessing parking area, so no more CO₂ and CO release to air this is the way of to minimize environment and health risk. IOT smart device connectivity brings consumers' physical and digital life closer together. Connected smart parking bring the benefits drivers by driving dramatic improvements in (services and resource) efficiency. Smart parking ensures a networked urban society shares in the benefits of smart parking management, smart services usage, security, smart traffic congestion control (using cameras at every signal, increasing average speed in the city). At every layer of SPS, IoT provides improved communication, control, and data distribution to drivers and parking managers to use parking services easily.

2.6.2. Healthcare Applications

In the healthcare domain, IoT technologies have many benefits in smart cities and in rulers' area. Especially people and objective tracking, the status of patients in a clinic or hospital is monitoring in order to provide better and faster workflow in the hospital. IoT health application is to tracking of people and objects including patients, staff and ambulance, identification of people, and automatic data gathering and sensing [8]. Building smart parking system using IOT technology has a boundless application by minimizing traffic accidents and air pollution that cause diseases for human.

2.6.3. Transport Applications

The development in transportation was one of the factors to indicate the wellbeing of the country. A road condition monitoring and prepared application was one of the most important of IoT transformation application [2]. The transportation analytic characterized the examination of demand prediction and anomaly detection. The routing of vehicles and speed control in addition to traffic management are all known as transportation control which they actually firmly related to the way of the vehicles connectivity communication, and overall governed by multi-technology dissemination [2]. Smart transport improves efficiency and safety. Parking is one part of transport system that may reduce many problems, like traffic congestion accidents air pollutions problems of so building smart parking system was more safety for drivers/users and beneficiary for all users directly or indirectly.

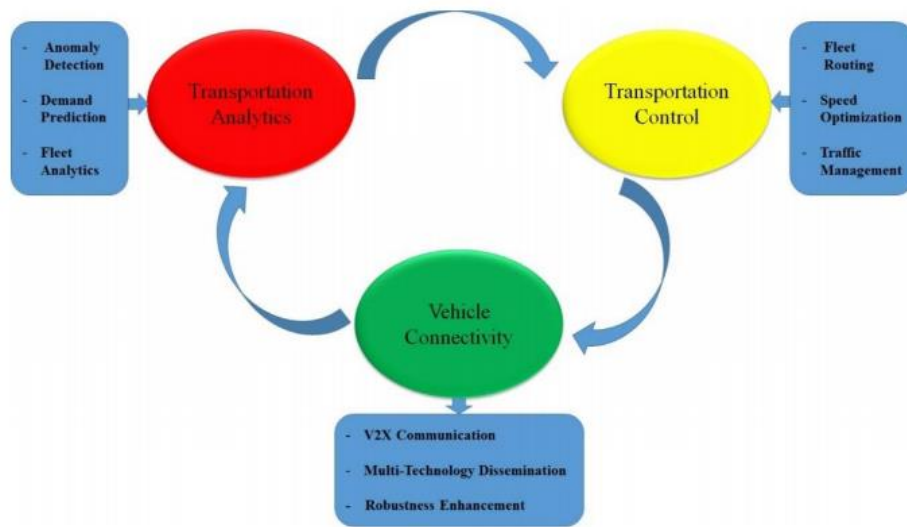


Figure 2. 3. Transport Applications [2]

2.6.4. Smart parking System Applications

Smart parking system was integrated form of hardware software application to help drivers to find the empty spot in parking lot more easily with less time. The Smart Parking System (SPS) was been based on several innovative technologies and can automatically monitor and manage car parks [4]. SPS, which perform simultaneously in one platform for unique task within a time [2, 10]. Real-time SPS systems have recently attracted significant interest in both industry and academic due to its safety, efficiency and customer satisfaction.

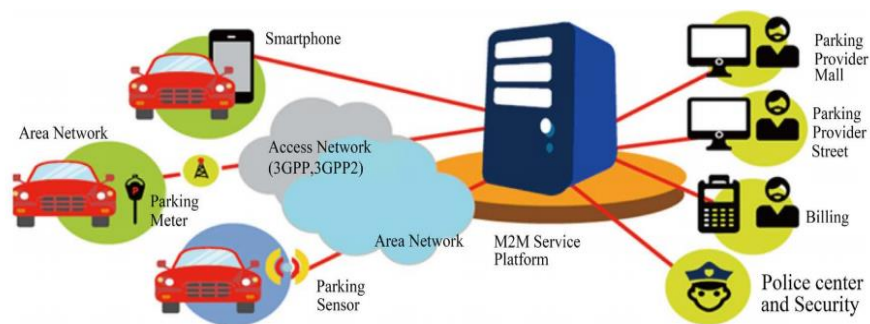


Figure 2. 4. Smart parking system concept [16]

2.7. Related Work

In Addis Ababa city, the number of cars increases day to day and traffic problems always exist. This problem arises due to unawareness of residence about traffic, driver's carelessness, roadside parking, and searching park for a long period of time. To minimize traffic problems BPS in the city is one solution. In the previous study, different authors develop SPS using IOT technologies to check free parking slots.

The researcher of [17] [18] put forward a swarm intelligence-based vehicle parking system employing context awareness and wireless communications. In this system, parking areas were instrumented with web-based tools and with a wireless sensor infrastructure. Parking information is collected and visualized through suitable Internet-based dashboards and was been communicated to the vehicles searching for parking spaces. The route to the nearest available parking lots is computed through particle swarm optimization algorithms and is sent to the drivers. Data mining for vehicular maintenance and insurance data are investigated in [19] [18], experimenting with Bayes and logistic regression models. A recent paper [20] employs data from on-street parking sensors from the city of Santander, in Spain, to design and validate a framework for parking availability prediction (either by area or assessing the future state of specific parking spaces). Statistical properties of parking signals were explored in [21] [16], where the authors propose the concept of lean sensing. Specifically, they trade some sensing accuracy at the benefit of improved operational costs: temporal and spatial correlation been then utilized to infer the system-state from incomplete parking availability information in an attempt at reducing the power consumption associated with sensing and reporting.

Paper [22] deals IoT Based Smart Parking System Using RFID in order to reduce the parking problem and to do secured parking using the smart parking system. The researcher proposes web based mobile applications. The various steps involved in this operation are vehicle identification using RFID tags, free slot detection using IR sensors and payment calculation is done on the basis of period of parking and this is done with the help of real time clock. Raspberry PI3 is a processor which performs all of the above functions through the use of Internet. To provide location and time information anywhere on earth, Global Positioning System (GPS) is commonly used as a space-based global navigation satellite system. This paper minimizes the parking waiting time in a large- sized parking facility. It also helps in maximizing their venue generation for the parking

facility owners also reduce the need for man-power in the parking facility which would greatly reduce the cost and errors in the process.

As the author stated the disadvantage of the system: no driver guidance systems to guide towards the parking lot, Automated parking system uses Bluetooth device to find a vacant space in the parking area and the information about the vacant parking space can be obtained only within the range of the Bluetooth. The system has also the following problems but in the researcher, is not aware the cost of each park, the distance of each park, and the security issue of the car (deals only car identification using RFID), Sensor scalability (because IR sensor is affected by rain) and the weather condition. The new proposed system was more related to this system in case of mobile-based web application, tools like GPS and WSN. However, our system, which implements a reservation service to reduce the traffic volume caused parking search. Navigation facility could have added to guide the user at each step to make sure that the user reaches the destination slot in the minimum possible time.

In paper [23] minimize much problems by automatically find free car parking zone at the smallest amount of time also maximize revenue for parking owners. The researcher used RFID (to manage parking system) IR sensor and MAX 232 controller (to handle the entry point and point of park) Dijkstra's algorithm (for checking the routs) Sha1 algorithm (for password security). These papers been more related with the current study, but both the methodologies, tools and objectives were different. The current study considered mobile-based location, free slot identification and payment system or manually payment after using services and the payment system was been connected with the banking system. The current study also analysis the number of parking slots and the location and distance of parking by calculating the distance from the current location of drivers or users to each parking area in Addis Ababa city. In this study, drivers or users can easily access all available parking information from local or central server based on their requests. In each time, the available information was updated automatically in both database of local server and the central server of smart parking system and aware the drivers continuously if they request information's. in the current study Ultrasonic sensor, Arduino UNO ,16*2 LCD and GPS was used to identify the slot, analyze available information from each device, display the information and location identification and path navigation respectively. The hardware device was simulated in proteus software using Arduino IDE in C++ programming language and connected MySQL database using XAMPP local

server the all the SPS information was been stored in ThingSpeak IOT platforms.in case of user/driver side hybrid mobile application was developed using Cordova, html5, java script and JQuery and installed in the user mobile phones. PHP language was selected due to the following reason:

- ***Open source***: It is completely free and it is also supported by a large and active community to enhance PHP functionality.
- **is easy to use**: It is easily even for beginners and cost-effective method of creating robust applications.
- ***PHP Accepts Platform Diversity***: PHP can run on many platforms like Windows, Unix, and Linux.
- ***PHP is Powerful Flexible and Scalable***: compare to other language level of flexibility, robustness, and scalability that is built into PHP and it has an advantage for updating the systems code without requiring a server reboot.
- **PHP has high level of flexibility, robustness, and scalability.**
- ***PHP has a Huge Standard Library***: because a library plays a crucial role in simplifying and speeding the data processing ability of the language.
- ***PHP Delivers Exceptional Performance***: PHP improves speed of development and speed of execution of systems.
- ***PHP offers Diverse Frameworks***: PHP is able to deliver exceptionally fast turnaround times because it has a sufficiently large family of PHP-frameworks.

Therefore, based on the above reasons, PHP is the best choose for smart parking system to access parking services by users with minimum turnaround times.

CHAPTER THREE: METHODOLOGY

3.1. Design Methodology

The proposed research involves building IOT based smart parking system in Addis Ababa city for real-time application. The IoT Methodology aims to provide a roughly organized environment of common value for all who participate, driven by sharing, collaboration, community and learning. An environment made up of tools, design patterns, architecture references and guidelines to build IoT based smart parking systems. A well-defined methodology will be an important factor to develop the architecture and prototype of the smart parking system in Addis Ababa city in real-time. Most research methodologies do not cover all aspects of an IoT applications development process. Therefore, a design methodology for the development of an IoT based smart parking systems is required. The IoT introduces a unique combination of physical and virtual development in which the development of hardware systems and software systems are strongly interconnected. IOT has no agreed design methodology for its applications. Different researchers supposed IOT design aspects (Borowicz, 2014; Pradhan & Upale, 2015; Rowland et al. 2015; McEwen & Cassimally, 2014). They consider in design aspects; cost, size, power, networking, communication range and security. Rowland et al. (2015) propose the following IOT design model for different applications as shown in figure 3.1 below.

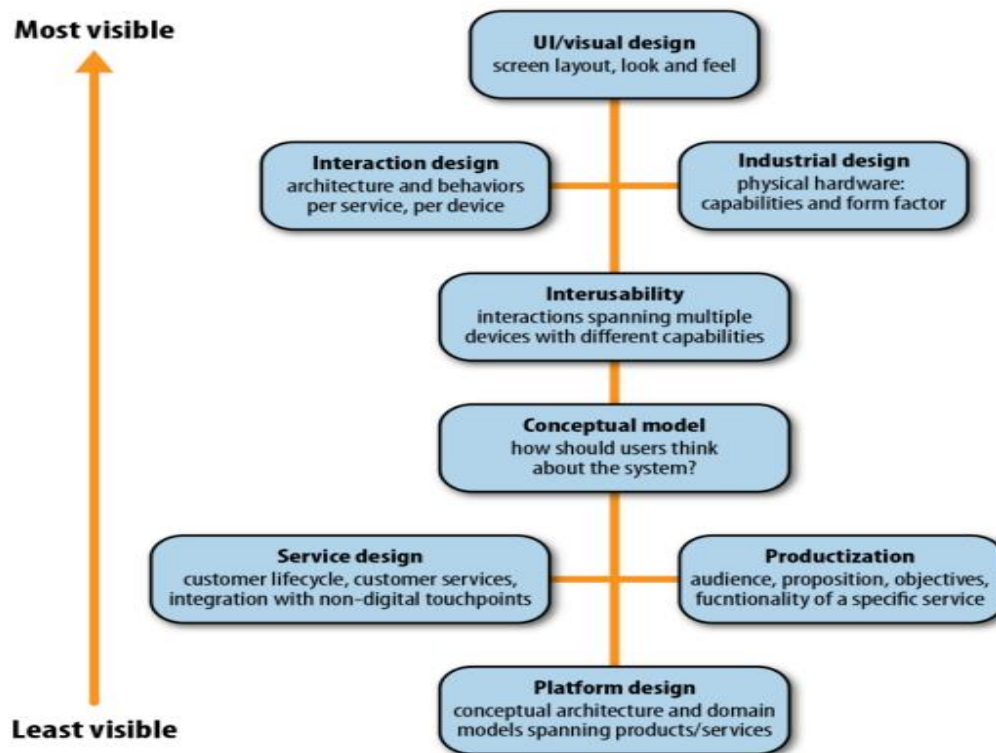


Figure 3. 1. Aspects of design in IoT (Rowland et al. 2015)

Building an IoT based smart parking system in Addis Ababa city was been used iterative methodology for end user visibility, but aspect of design in IOT model is strong focus on user interactions and design aspects of IOT development. Bahga and Madisetti (2014) proposed a ten-step methodology for the development of an IoT application. The ten-step methodology covers many important aspects for forming a solid understanding of the IoT application development process. The proposed design methodology is fail in user and system interactions covered by Rowland et al.'s methodology. The ten-step methodology considers tasks at a high level and therefore risks explaining over vital technical considerations. The methodology does not include a hardware selection phase based on the use cases and requirements for the IoT system.

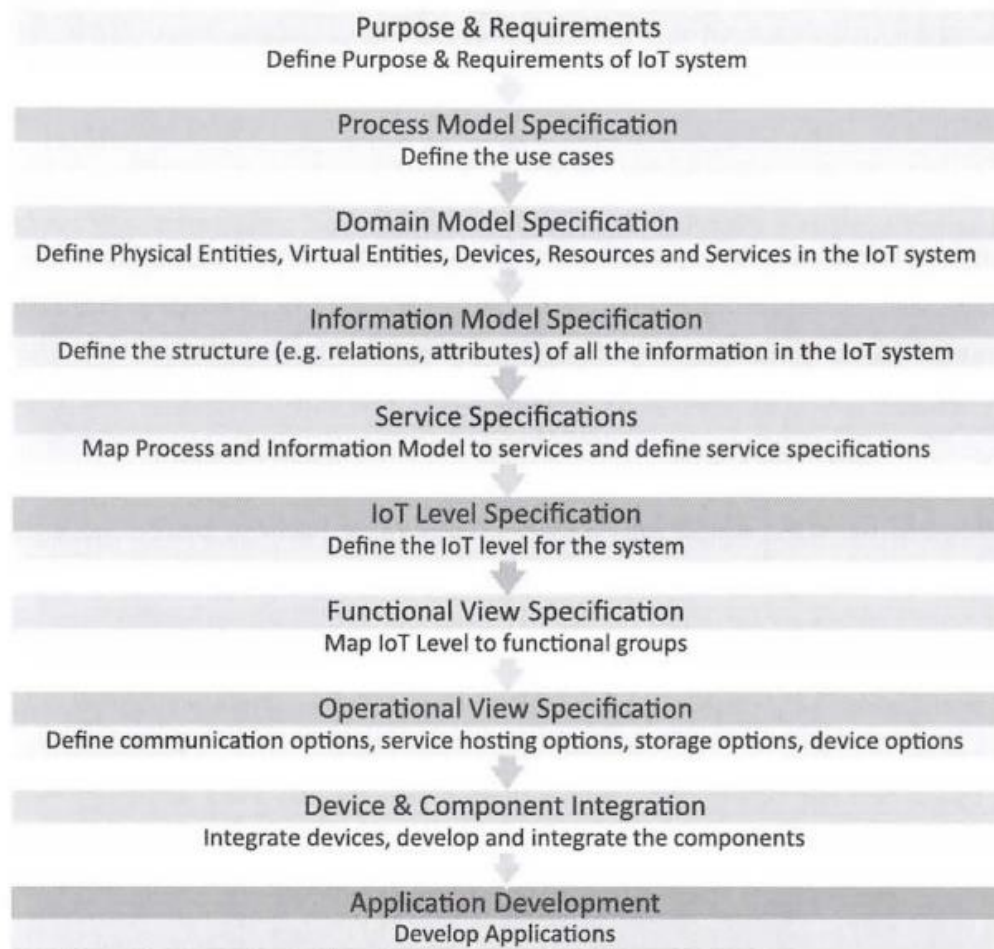


Figure 3. 2. 10 Step methodologies for IoT development (Bahga and Madisetti, 2014)

The base model consists of five stages. Each stage consists of a worksheet with multiple steps. Parker’s (2015) model and worksheets break the process down into the following steps: context

statement, problem statement, mission statement, justification statement, business rules, physical and virtual requirements, constraints, course of action development, solution analysis and technical analysis. However, this methodology, similar to the ten-step methodology, overlooks all design aspects of building an effective IoT application.

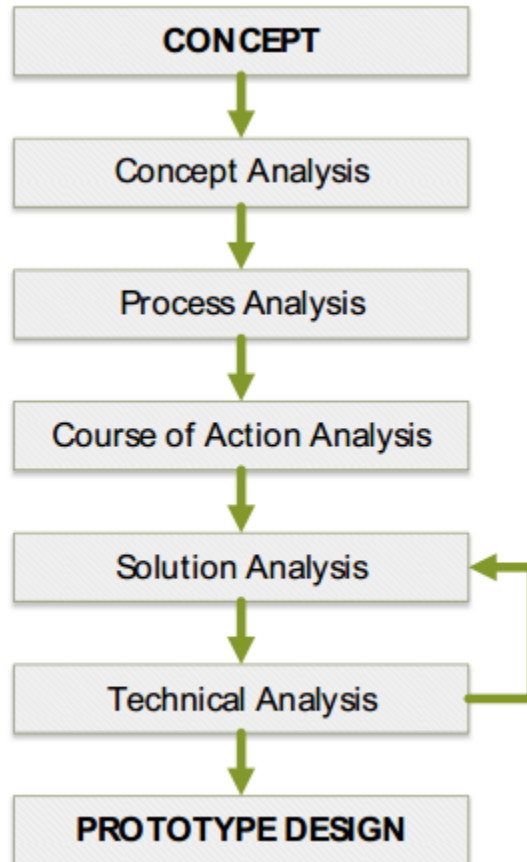


Figure 3. 3. Design methodology for IoT development (Parker, 2015)

Since IoT is a reasonably new concept, it is not unexpected that a complete methodology is not available for developing IoT applications. In the above proposed design methodologies are not cover all aspects of IoT application especially in hardware and software prototype and architecture design and development. Thus, more complete methodology architecture will be created in the methodology section of this paper. The proposed methodology architecture will draw inspiration from the above methodologies presented in this section. Ultimately, the architecture will aim to afford a fuller view of the IoT development process by including user interaction and design aspects as well as implementation, deployment and maintenance aspects. Furthermore, the

architecture will move away from design aspects, 10-step design is an iterative approach, which is developing iteratively. Because IOT needs hardware and software communications and end user interactions over IOT communication models, protocols and sensors. So, the current paper was used the integration of the ten-step design and Iterative design approach. Iterative design is a design methodology based on a cyclic process of prototyping, testing, analyzing, and refining a process. Based on the results of testing the most recent iteration of a design, changes and refinements are made. This process was intended ultimately improve the quality and functionality of a design [24]. The iterative design process may apply throughout the new product development process [24]. However, changes are easiest and less expensive to implement in the earliest stages of development. The first step in the iterative design process is to develop a prototype. The prototype should be evaluated by a focus group or a group not associated with the product in order to deliver non-biased opinions. Information from the focus group should be synthesized and incorporated into the next iteration of the design. The process should be repeated until user issues have been reduced to an acceptable level. Iterative design has the following steps:

- ***Co-create***: to communicate with end-users and stakeholders to identify pain problem areas in a non-technical way.
- ***Ideate***: to simplify discussions to communicate requirements with designers, implementers and project managers.
- ***Question & Answer***: To translate soft concepts into hard requirements, analyze solutions, and brainstorm options.
- ***Map*** IoT OSI (Open System Interconnection): map requirements to a valid architecture, infrastructure, and business frameworks, similar to the layered approach used in ISO-OSI model.
- ***Prototype***: to use standardized toolkits to build prototypes and iterate towards minimal viable products
- ***Deploy***: to deploy continuously to close the feedback loop and improve the products.

3.2. Hardware Tools

Design tools were used to produce a drawing of the parking model, the parking lot map information, the circuit design and the code programming for the research.

3.2.1. Mobile Phone

Mobile phone has a great benefit especially in real time applications. It receives parking lot information from central web server and display least cost and nearest parking place to users.

3.2.2. Arduino

Arduino is an open-source electronic prototyping of Micro-controller embedded system. Arduino has hardware and software components to design various electronics systems like home automation, smart parking system and smart health. The hardware component is a family of ATMEL Arduino development board and the software component contains an open-source IDE (Integrated Development Environment) to design, run, sketch and upload the code. The circuit board of Arduino is programmed using C or C++ programming language. Arduino software is compatible for Windows, Linux, and Mac operating software. Arduino board introduces in 2005, after that students and professionals create advice that interact with the environments using sensors and actuators.

3.2.3. Sensors

Sensors are an electrical/mechanical/chemical device that maps an environmental attribute to qualitative measurement. It provides a usable output in response to a specified measuring input. A sensor fetches the signal of physical quantities and it convert to the output signal in suitable format for processing. At the current time most, sensors convert measurement of physical phenomena into an electrical signal using the active transducer, which converts one form of energy to another form of energy. The sensor was used to identify the slot of smart parking system.

3.2.4. LCD display

Liquid crystal display (LCD) is a flat panel display technology commonly used in TVs and computer monitors. It was used in screens for mobile devices, such as laptops, tablets, and smartphones. Nevertheless, in this work LCD display was been used to display parking status for parking users.

3.2.5. Global Positioning System

GPS receiver receives the data or information mainly latitude and longitude of the particular parking areas all over the city by calculate the exact location; distance the smart parking via Internet.

3.2.6. ESP8266

ESP8266 is an UART-Wi-Fi transparent transmission module with ultralow power consumption, specially designed for the needs of a new connected world. ESP8266 has powerful on-board processing and storage capabilities that allow it to integrate with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime.

3.3. Software Tools

Software tools were used to implement the hardware and the software of smart parking systems. To monitor free parking slot in real time software tools must be communicate continuously to the hardware tools. To transmit information from sensor and Arduino to web server ESP8266 Wi-Fi module was been used and this information transferred to a central server over Hypertext Transfer Protocol (HTTP) or RESTful connection by storing the required data in the database. The mobile application was developed using Cordova, Bootstrap, HTML5, CSS and java script. A web application was built using JavaScript or jQuery, which is a library of JavaScript, CSS, HTML5, PHP and MYSQL for database connection in XAMPP local server and Arduino IDE was used in hardware implementations. Arduino IDE software is an open source software use for compiling the program into the microcontroller. It uses C/C++ programming language. IDE contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common function

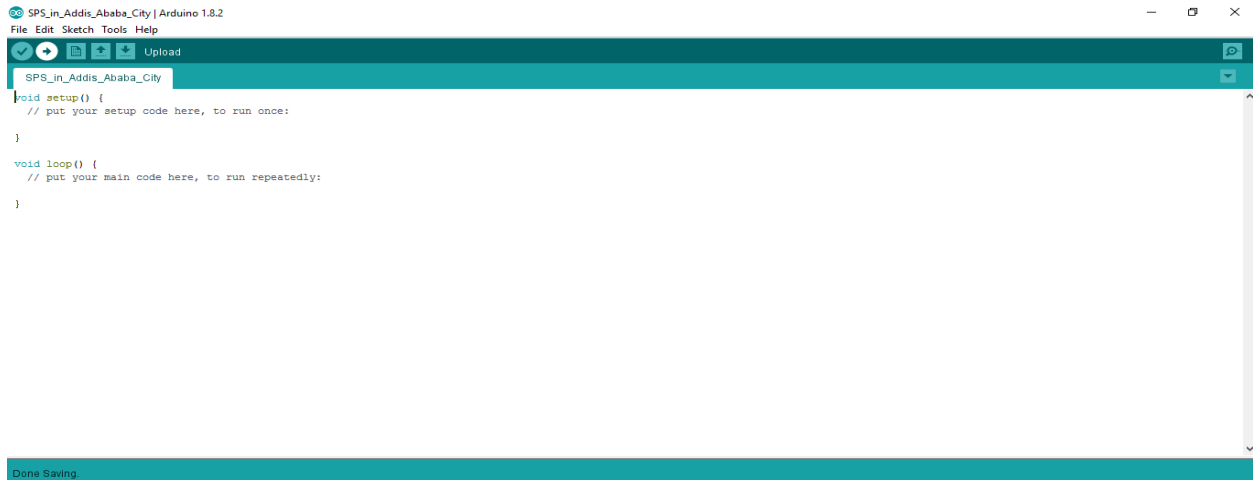


Figure 3. 4. Arduino IDE interface

A Programming language knowledge is needed to building smart parking system and to ensure the communication among ultrasonic sensor, Arduino UNO to store and retrieve data of parks location into the web application. Arduino IDE is an application, which can have installed in Windows, Linux and Macintosh and gives to the user a development environment for coding the programs that are uploading to the board.

3.4. Design Method

Selecting appropriate method to build real-time smart parking system architecture will be the crucial point. The parametric study method was used to identify the appropriate sensor, actuators, GPS trackers, suitable microcontroller and proper software to be used throughout the process. The literature review was made on previous chapter to facilitate smart parking system by comparing to the standard of a parking type, parking tools, parking services and the parking communication models.

Qualitative paradigm of research was used, due to its suitability for studying and understanding complex phenomenon of IOT devices (hardware and software), protocols, communication models and architectures by analyzing data gathered via interviews, and observations. By using the qualitative paradigm of research, it would be possible to uncover various issues concerned with perceptions and experiences about the use of IoT in the future [25]. An exploratory research design

was also used for the current research study, mainly because there are few studies dealing with features, weaknesses, and challenges smart parking system applications of IoT.

3.4.1. Qualitative Research

This research has one of its aims, to study the problems drivers in Addis Ababa city and building solutions, through gathering information from real world environment by conducting interview and observation in different transport organizations (private and public), parking owners and on different users (drivers, tourists, students, teachers, and other residences) in Addis Ababa city.

3.4.2. Sampling Techniques

purposive sampling was used to choose specific people within the population to use for a parking study. In the smart parking system purposive sampling technique was used to select the participants for interview and to select observation places to get parking information. The selection criteria of participants for interview was based on the professions/expertise, educational qualification level, year of experience and their immediate position in the Ministry of transport, in the traffic police and drivers and the selection criteria of places was based on high traffic follows. Even if purposive sampling is one of the most common sampling techniques in qualitative research it has the following advantages in which participants group are decided to pre-selected criteria relevant to a particular research question.

- ***Wide range of techniques:*** Since there are several different types of purposive sampling like homogenous sampling, expert sampling, critical case sampling, and so on. One of the main benefits of this sampling method is the ability to gather large amounts of information by using a range of different techniques.
- ***Stage building blocks:*** Qualitative research usually involves a number of different phases, with each phase building progressively onwards from the original. This being the case, purposive sampling is useful to a researcher because they can use the variety of methods available to build and increase their research data.
- ***Economical in terms of time and resource:*** The sampling technique was performed within limited resource and a short period of time.

3.4.3. Data Collection

To acquire information, both secondary and primary (documented and undocumented) sources of data are used. Primary data is gathered from Ministry of transport, driver, tourists, students and parking owners, in Addis Ababa city using interview and observation. In the same way, secondary sources of data were collected by using document analysis in Ministry of transport and parking owners.

3.4.3.1. Interview

Structured and unstructured technique is used to get information from the concerned body. In addition, to analyzing the data, elicitation methods are used to purify the collected data. The data is refined with the consultation of the experts. Moreover, secondary sources of knowledge are gathered from the Internet, newspapers, research papers and articles by using document analysis technique. The interviews were conducted to find answers to problems associated with parking in Addis Ababa city and to check that is really a smart Parking in Addis Ababa city is necessary or not. In the Structured interview some pre-determined questions were prepared to all interviewees and the data was analysis by comparing and contrasting different answers given to the same questions. Even if unstructured interviews are sometimes least consistent because no questions are prepared prior to the interview and data collection was conducted in an informal method. Unstructured interviews mostly associated with a high level of unfairness and comparison of answers given by different respondents tends to be difficult due to the differences in design of questions. The interview was conducted in many public and private organizations and to analyze parking problems in Addis Ababa city. All the interview questions (see in appendixes 1.1 to 1.5) was answered by the selected participants. The interviews time was carried out by the choice of interviewees. The interview took 6 to 8 minutes. At the start of the interview, the aim of the interview was explained and participants were informed about the intentions. The intentions were not to examine them but to understand the challenges associated with parking in Addis Ababa city. They were also informed that there was no correct or wrong answer but explain what you know about parking problems, parking problem cases and parking problem results. Then the result of interview question was summarized in the table 3.1.

Table 3. 1. Ministry of transport Interview question results

Questions	Interviewees			
	Interviewee1	Interviewee2	Interviewee3	Interviewee4
	Build additional road	Adapting-new technology	Build additional road	Minimizing small cars
Question1	Shortage of roads	Road-side parking	Increase of cars	Increase of cars
Question2	Yes	Yes	Yes	Yes
Question3	No	No	No	Yes
Question4	No	I do not know	No	No
Question6	May be	Yes	Yes	Yes
Question6	In our organization	I have no car	I have no car	I have no car

Ministry of transport have a to vision to give sustainable service using new technology and also have plane to minimize traffic accident and traffic congestions by avoiding road-side parking and building alternative roads. And most employees do not have sufficient knowledge about parking.

Table 3. 2. Interview question results from Traffic Police

Questions	Interviewees			
	Interviewee1	Interviewee2	Interviewee3	Interviewee4
Questions1	Yes, it is high, morning & Afternoon	Yes, it is high, morning afternoon	Yes, it is high, morning Afternoon	Yes, it is high, morning& afternoon
Questions2	Yes	Yes	Yes	Yes
Questions3	Yes	Yes	Yes	Yes
Questions4	Yes, Yes	Yes, Yes	Yes, Yes	Yes, No

From the interview result of traffic police most of the traffic congestions and accidents occurs in the morning and in the afternoon, this is the reason of shortage of roads, increasing of cars, peoples and some big small cars parked sometimes in the road-side.

Table 3. 3.Interview question results from drivers

Questions	Interviewees			
	Interviewee1	Interviewee2	Interviewee3	Interviewee4
Questions1	Road-side	Road-side	Parking place	Parking place
Questions2	Yes	Yes	Yes	Yes
Questions3	By searching	By asking people	By asking people	By asking people
Questions4	Direct pay	Direct pay	Direct pay	Direct pay

From the result of drivers’ interview accessing parking stations in Addis Ababa is more difficult, because all the parking stations did not use new technologies that assist drivers to get parking locations and free parking stations, even the drivers said after searching the parking stations, it may be full, then they will again be searching another parking stations this may lead traffic congestions in addition to wastage of resource and fuel.

Even if interview requires longer time in additions to difficulties associated with arranging an appropriate time with perspective Interviewees group members to conduct interviews it has options of collecting detailed information about research questions. So, it is possible to control flow of process and to clarify certain issues during the process if needed.

3.4.3.2. Observation

Observation was conducted for two weeks in 18 road areas, 3 shopping centers, 6 cinemas and theaters places of Addis Ababa city to get answer the question road side parking is a cause for traffic problems and to check is there any parking available in market place, shopping center, theater and cinema place. The selection of the areas was based on high frequency of traffic flows, problems and crowdedness.

Table 3. 4. Road Areas Observation

<i>Road areas</i>	<i>Traffic flow / problems</i>			<i>Road side parking</i>	
	<i>Morning</i>	<i>Mid-day</i>	<i>After noon</i>	<i>Yes</i>	<i>No</i>
Mexico	very high	very high	very high	✓	
Legehar	very high	very high	very high	✓	
Paisa	very high	very high	very high	✓	
Megenagna	very high	High	very high	✓	
Lamberet	very high	Low	very high	✓	
Bole	very high	High	very high	✓	
Kality	very high	High	very high	✓	
Saris	very high	Low	very high	✓	
Merkato	very high	very high	very high	✓	
Jemo	very high	Low	very high	✓	
Autobus tera	very high	very high	very high	✓	
Alemgena	very high	Low	very high	✓	
Alembank	High	Low	High	✓	
Kera	High	Low	High	✓	
Ayertena	High	Low	High	✓	
Torahayloch	very high	Low	very high	✓	
Aratkillo	very high	High	very high	✓	
Tikuranbessa	very high	High	very high	✓	

The list of road areas in the table almost all has very high traffic problems especially in the morning and in the afternoon. Roadside parking has great impact for this problem in addition to road roots problems (the free road route of the city is not available).

Table 3. 5. Cinemas and Theaters place observation

<i>Cinemas and theaters place</i>	<i>Location</i>	<i>Traffic crowdedness</i>			<i>Parking availability</i>	
		<i>Weekend</i>	<i>other day</i>		<i>Yes</i>	<i>No</i>
			<i>Morning</i>	<i>Afternoon</i>		
Lafto mall cinema	Bisrate Gebral	very high	Low	High		✓
Adotcinema and theater	Bisrate Gebral	very high	Low	High		✓
Ednamall cinema	Bole medhanialem	very high	Low	High		✓
Abel cinema	Agona	High	Low	High		✓
National theater	Legehar	very high	Low	High		✓
Ambassador theater and cinema	Legehar	very high	Low	High		✓

The above table 3.5 indicates no sufficient parking area even in the cinemas and theater places. They are very crowded especially in the weekend because many people need parking in recreational area. Shopping and marketplaces have not any parking area; due to this customer stopped their car in the roadside until they finish the tasks. This led traffic congestions, so BSPS in the city has a potential to minimalize these problems and to get valid data interview and observation was conducted. And the reset parking information was from documents from transport Ministry of Addis Ababa and city administration of Addis Ababa. The City Administration of Addis Ababa in 2011 created partnership among private sector, government and public private in the city center and in the areas with a high volume of traffic congestions to facilitate parking and in 2012 Mekuria Haile, former general manager of Addis Ababa City Administration, who is now minister of Construction and Urban Development, said the parking facility will construct within a short period of time because parking is a serious problem affecting both drivers and pedestrians. Also in the Addis Ababa that is celebrating its 125th anniversary various social and economic

problems are being faced the first one is lack of sufficient transportation is a major difficulty although the city administration is said to be working on it with huge budget allocation to construct new roads. The transportation fluidity problem is not expected to be solved with just the construction of new roads but the city administration needs to see related aspects to develop the sector, an expert on urban planning said. According to studies, sufficient parking facilities are one of the major focus to develop Addis Ababa’s transportation structure. According to the city administration plan, the road from Meskel Square to Megenagna and from Meskel Square to Bole International Airport are priority areas that need immediate parking lots as there is a large concentration of vehicles in these areas that lack sufficient parking facilities compared with the traffic flow. Due to this parking stations was built in Addis Ababa city in different areas to solve parking problems starting from 2012 G.C. We were observed five parking stations to check the infrastructure and environmental factors of parking stations to build smart parking system for each parking stations. The infrastructures and the environmental factors of the five parking stations were summarized in the table 3.6.

Table 3. 6.Observation of parking places

Parking Name	Exposed by environmental factors			Infrastructures in each parking stations
	Air pressure	Rain	Temperature	
Megenagna parking	Not exposed	Not exposed	Not exposed	Have only lifting device
Zefmesh parking	Not exposed	Not exposed	Not exposed	Have no any device
Friendship parking	Exposed	Exposed	Exposed	Have no any device
Gotera parking	Not exposed	Not exposed	Not exposed	Have only lifting device
Merkato parking	Not exposed	Not exposed	Not exposed	Have only lifting device

As we were observed the parking stations in the city, no sufficient parking infrastructure facilities and most of the parking stations is exposed by environmental factors like rain, air pressure and temperature. This factor will influence the Ultrasonic sensor that will install in each parking stations, Ultrasonic sensor was greatly affected by temperature rather than other environmental

factors. But Ultrasonic sensor is working between -10°C to $+50^{\circ}\text{C}$ and the temperature of the parking stations is 18°C to 27°C so even if the parking stations is exposed by temperature the Ultrasonic sensors were work efficiently. From the parking stations in the city Megenagna, Gotera and Merkato parking stations were built using modern instruments specially they use parking lifts. As the managers of each parking stations told us the performance and the capacity of the lift and other mechanical instruments were tested before built by material Engineer professionals and the working principle of the system is when the lift button was pressed, the lift takes the car to the free parking slot. The three parking stations have security camera to control the car. But even if they built in a modern way the parking lots were not built efficiently because the size of each slot in the stations were not divided equally by international way. For more utilization of parking slot environments, the slot must be built based on the international size by considering the type of car that may be park. For example, Megenagna parking accommodate 140 cars at the same time ground floor as the parking manager told us. But when the we were measure the flor and when we calculate by international way the ground floor can accommodate up to 160 cars at the same time. As the manager of the parking told us the customers get parking services by phone call rather than any systems and the free parking slots also checked by watching manually.

Generally, the automated versions of parking space, room height, time, and fuel must be conserved for each parking stations and almost no space for circulation for each parking stations. In addition to this, each parking stations requires high human resources, low securities and maintenances. So, after analyzing environmental of parking and parking infrastructures, smart parking system was built for each parking stations to solve parking problems in Addis Ababa city.

CHAPTER FOUR: HARDWARE TOOLS SELECTION FOR SPS

4.1. Overview

To select the hardware laboratory experiments are carried out in 123D circuit online lab to perform all the required investigations and to complete the desired work correctly. Some laboratory experiments are carried out for examining the designed circuit by using Proteus Virtual System Modeling as well as for testing the prototype monitoring system. Concerning this research, a study is carried out to evaluate the economic feasibility of hardware's, the environmental factors that affect the hardware's, the performance of the hardware's. The results are recorded and associated with many recommendations that may be valuable to the parking owners.

4.2. Breadboard Comparison

There were different types of electronic prototyping board like, Arduino, Raspberry Pi, and Beagle Board. The selection of each board based on many criteria depending on its application and feature. Arduino easily connected to LED, LCD, sensors and actuators. Table 4.1. shows the comparisons of each board.

Table 4. 1. Comparison of breadboard

<i>Feature</i>	<i>Arduino</i>	<i>Raspberry pi</i>	<i>Beagle Board</i>
Processor	ARM 11	ATMEL AVR	ARM CORTEX
Clock speed	16MHZ	700MHZ	1GHZ
Hardware open-source	Full open-source	Partially open-source	Full open-source
Cost (US \$)	2-29.95 based on model	25-42 based on model	2-199.5 based on model
Operating system	No	Raspbian	Linux kernel
Flash memory	8GB-32GB		
Compatibility platform	Microsoft, Linux and Mac OS X	Linux/window10 Pi 2B and above	Linux
USB	1	2	1
Ethernet	No	Yes	Yes

4.3. Board Selection

Based on the above table 4.1. Arduino is multiplatform, fully open-source and inexpensive. Even if Arduino has not Ethernet and flash memory, it uses different types of shields to connect the various modules to Arduino boards. Arduino Ethernet shield to connect Arduino board to connect to the internet using the Ethernet library and to read and write an SD card using the SD library, Arduino Wireless shield used to connect Arduino board to communicate wirelessly using ZigBee and EEPROM shield was been connected to Arduino board to store the data. The communication of shields with Arduino board can be done either directly over various pins or in directly via serial bus. There are different types of Arduino boards (i.e. Arduino Uno, Galileo, Arduino Yun, Arduino BT, Arduino Mega, Arduino Due, Arduino Nano, Arduino Esplora, Arduino Leonardo, Arduino Pro, Arduino Micro, Arduino Mini), but the most popular official versions available are the Arduino Uno R3 and the Arduino Nano V3. Both of these runs 16 MHZ Atmel ATmega 328p, 8-bit microcontroller with 32KB of flash RAM, 14 digital input/output pins (of which 6 can be used as PWM outputs) and 6 analog input. Arduino Nano V3 lacks DC power jack instead use Mini-B USB cable to work but Arduino UNO R3 has a USB connection, a power jack, an ICSP header and a reset button Due to this and the following criteria Arduino Uno was selected for smart parking system.

- *It has an easy USB interface:* The chip on the board plugs straight into your USB port and supports on your computer as a virtual serial port. The benefit of this setup is that serial communication is an extremely easy protocol, which is time-tested, and USB makes connection with modern computers and makes it comfortable.
- It is easy-to-find the microcontroller brain which is the ATmega328 chip. It has more number of hardware features like timers, external and internal interrupts, PWM pins and multiple sleep modes.
- It is an open source design and there is an advantage of being open source is that it has a large community of people using and troubleshooting it.
- It is a 16 MHz clock, which is fast enough for most applications.
- It is very convenient to manage power inside it and it had a feature of built-in voltage regulation and powered directly off a USB port without any external power. Also, can connect an external power source of up to 12v and this regulates it to both 5v and 3.3v.

- Digital pins of 13 and analog pins of 6 pins allow connecting hardware to Arduino Uno board externally. These pins used as a key for extending the computing capability of the Arduino Uno into the real world. Simply plug your electronic devices and sensors into the sockets that correspond to each of these pins and you are good to go.
- Has an ICSP connector for bypassing the USB port and interfacing the Arduino directly as a serial device. This port is necessary to re-boot load chip if it corrupts and can no longer used to computer.
- It has a 32 KB of flash memory for storing code.
- An on-board LED is attached to digital pin 13 to make fast the debugging of code and to make the debug process easy.

4.3.1. Common Arduino Uno pins and functions

The following figure 13 shows the pins of the Arduino UNO board.

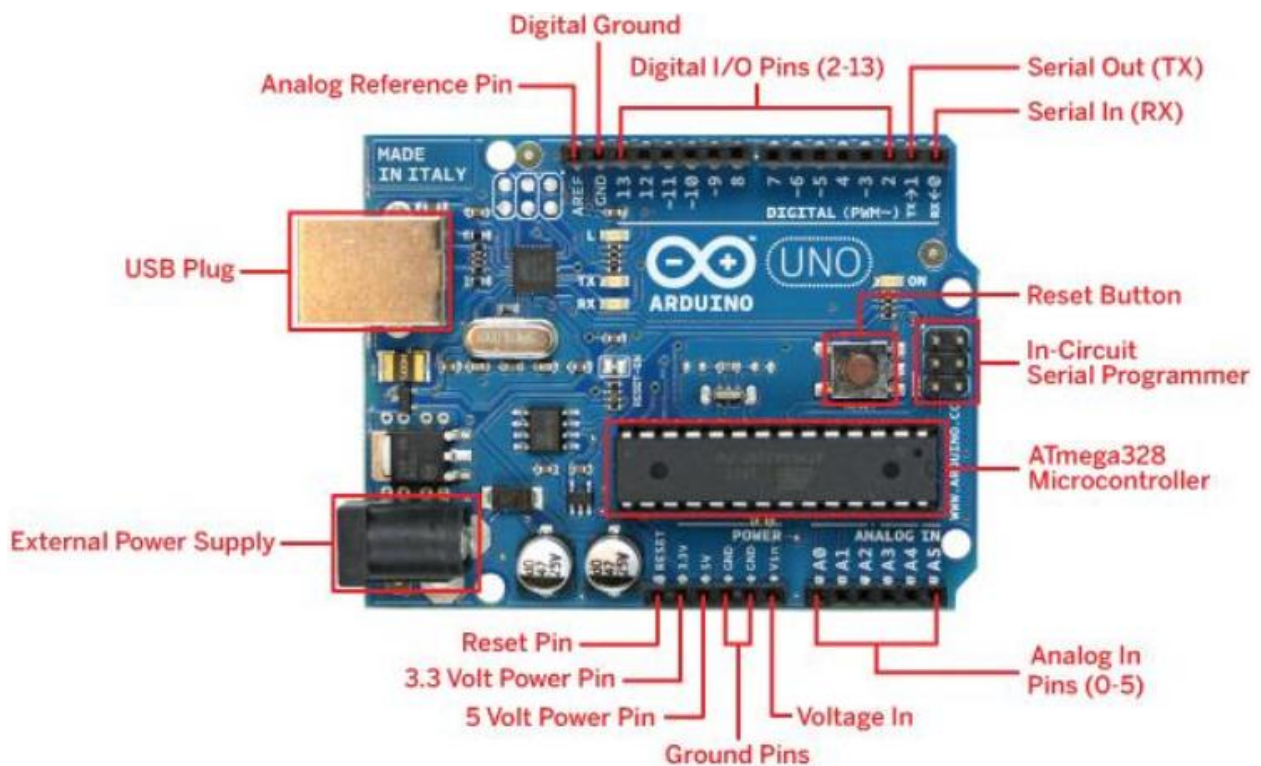


Figure 4. 1. Arduino UNO pins [26]

4.3.1.1. Power

The Arduino Uno board could be power via the USB connection or with an external power supply. The power source was selected automatically. External (non-USB) power can come from either an AC-to-DC adapter or battery. The adapter connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and VIN pin headers of the POWER connector. The board can operate on an external supply from 6 volts to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The power pins are:

- VIN: the input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin. 5V: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board.
- 3V3: A 3.3-volt supply generated by the on-board regulator and maximum current draw is 50 mA.
- GND. Ground pins.
- IOREF: this pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.

4.3.1.2. Memory

The ATmega328 has 32 KB (with 0.5 KB occupied by the boot loader). It also has 2 KB of SRAM and 1 KB of EEPROM, which can be used to read and to written with the EEPROM library.

Input and Output: The mapping for the Atmega8, 168, and 328 is identical. Each of the 14 digital pins on the Uno can be used as an input or output, using `pinMode ()`, `digitalWrite ()`, and `digitalRead ()` functions. They operate at 5 volts and each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor by default of 20-50k ohm.

A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. Input output pins are:

- Serial: 0 (RX) and 1 (TX) is used to receive (RX) and transmit (TX) TTL serial data. These pins connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: 2 and 3 these pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analog Write () function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK) these pins are supporting SPI communication using the SPI library.
- LED: there is a built-in LED driven by digital pin 13
- TWI: A4 or SDA pin and A5 or SCL pin is support TWI communication using the Wire library.
- The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution.
- AREF: is a reference voltage for the analog inputs and used with analog Reference ().
- Reset. Bring this line LOW to reset the microcontroller and typically used to add a reset button to shields, which block the one on the board.

4.3.1.3. Communication

Arduino Uno has a number of facilities for communicating with a computer, another Arduino board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver was needed. However, on Windows, an .inf file is required. The Arduino Software (IDE) includes a serial monitor, which allows simple textual data was been sent to and from the board. A Software Serial library allows serial communication on any of the Uno's digital pins. Sensor selection

4.4. Sensor selection

4.4.1. Characteristics of Sensors

Why need to study the characteristics of sensor is to know the factors that influence on each sensor to use in smart parking system in addition to selection criteria of each sensor.

Table 4. 2. Sensor Characteristics [27]

<i>Sensors</i>	<i>DC supply (volts)</i>	<i>Current mA</i>	<i>Power mW</i>	<i>Output type</i>	<i>Accuracy</i>	<i>Cost (\$)</i>	<i>Operating Temp (^oC)</i>	<i>Range (m)</i>	<i>Area (m²)</i>
Active Infrared Sensor	4.5 to 5.5	33	165	Analog	10 cm to 30 cm	14	-10 ⁰ to +60 ⁰	0.2 to 1.5	-
Passive Infrared Sensor (Standard Detection Type)	4.5 to 5.5	0.1	2	Digital and analog	Accurate within a range	15	-20 ⁰ to +60 ⁰	0 to 5	41.9 m ²
Passive Infrared Sensor (spot detection type)	3 To 6	0.17	0.57 to 0.85	Digital and Analog	Accurate within range	15	-20 ⁰ to +60 ⁰	0 to 5	2.8 m ²
Passive Infrared Sensor (slight detection type)	5	0.17 to 0.3	0.85 to 1.5	Digital and Analog	Accurate within range	15	-20 to +60	0 to 2	19.6 m ²
Passive Infrared Sensor (10m detection type)	5	0.17 to 0.3	15 to 100	Digital and Analog	Accurate within range	15	-20 to +60	0 to 10	196 m ²
Micro Wave Sensor	4.75 To 5.25	30 to 40	100	Digital and Analog	Accurate within Set of Frequency	10	-15 to +55	Varies to Set Freq.	-

Ultrasonic Sensor	5	20		Digital	High Accurate within 3mm	2	-10 to +50	0.02 to 4.5	1.1 m ²
-------------------	---	----	--	---------	--------------------------	---	------------	-------------	--------------------

4.4.2. Choosing of Sensor to BPS

Sensor selection is a crucial activity in any system design, as it will make a great impact on the process of the system performance during its entire lifetime and could even has consequences related to the quality of the product [28].

The criteria of sensor selection were, based on environmental factor, economical factor, range of sensing, installation ways, power usage and their characteristics.

Table 4. 3. Comparison of Sensors

<i>Sensor type</i>	<i>Power usage</i>	<i>Cost</i>	<i>Installation</i>	<i>Sensing area</i>	<i>Environmental Factors</i>
Ultrasonic sensor	Medium	Cheap	Easy to install	Small	Affected
Active infrared sensor	High	Cheap	Easy to install	Small	Unaffected
Passive infrared sensor	Low	Expensive	Easy to install	Large	Unaffected
Micro sensor	Medium	Expensive	Require antenna to be install	Large	Unaffected

Ultrasonic sensors are increasingly popular compared to other devices in the field of distance measurement up to a few meters, because they have advantages of low cost, easy implementation, anti-electromagnetic interference, and strong ability to adapt all kinds of targets and light intensities [29, 25]. Due to this reason, ultrasonic sensors are widely used in distance measurement tasks ultrasonic sensor is also suitable in mapping and localization in case of object detections. It can detect all types of obstacle (like, metal, wooden based object, concrete wall, plastics, rubber based product, transparent object, etc.) and it is not affected by poor lighting condition [30].

However, the velocity of ultrasonic wave travels in air affected by environmental parameters such as temperature, humidity and wind and rain. To check how much ultrasonic sensor will influence by the above factors it is necessary to know the speed of sound in many mediums.

Table 4. 4. Speed of Sound in different Medium

<i>Medium</i>	<i>Velocity in M/S</i>
Air	331.6
Iron	5130
Rock	6000
Water	1440
Wood	3320

The performance of ultrasonic wave was mostly affected by temperature from the above listed mediums in the table 4.4 ultrasonic sensor can work between -10°C to $+50^{\circ}\text{C}$ so, before BSPS it is better to calculate velocity of sound in air.

The velocity of sound in the air will calculated in the following formula.

$$V=331.6 \times 0.6 \times T \text{ (m/s)}$$

Where V is the velocity of sound and T is the amount of temperature in degree centigrade. Using this formula, we are going to check the amount of temperature in the environment before building smart parking system.

Humidity have even smaller effect but if it increases in 10%, the transmission of ultrasonic velocity increase by 0.035% so, in 0°C the air speed also increases from 331.6m/s to 331.7m/s.

Air pressure has no effect in ultrasound wave but in case of wind, ultrasound is affected by 4mm in a distance of 0.5m in the airflow with the velocity of 5km/h. So before building SPS it is better to check such like environmental factors for the easily, efficiently and fast identifying and checking free parking slot using ultrasonic sensor.

4.4.2.1. Pins of Ultrasonic Sensor

Ultrasonic sensor has four pins and each pin has its own functions in the following fig shows its interface and pins.



Figure 4. 2. Ultrasonic Sensor

Each pin in the fig above has its own functions. A trigger signal provided to TRIG input, a HIGH signal of at least $10\mu\text{s}$ duration. This enables the module to transmit eight 40 KHz ultrasonic burst [31]. If there is an obstacle in front of the module, it will reflect those ultrasonic waves. If the signal comes back, the ECHO output of the module will be HIGH for duration of time taken for sending and receiving ultrasonic signals. The pulse width ranges from $150\mu\text{s}$ to 25ms depending upon the distance of the obstacle from the sensor and it will be about 38ms if there is no obstacle.

4.4.2.2. Working Principle of Ultrasonic Sensor

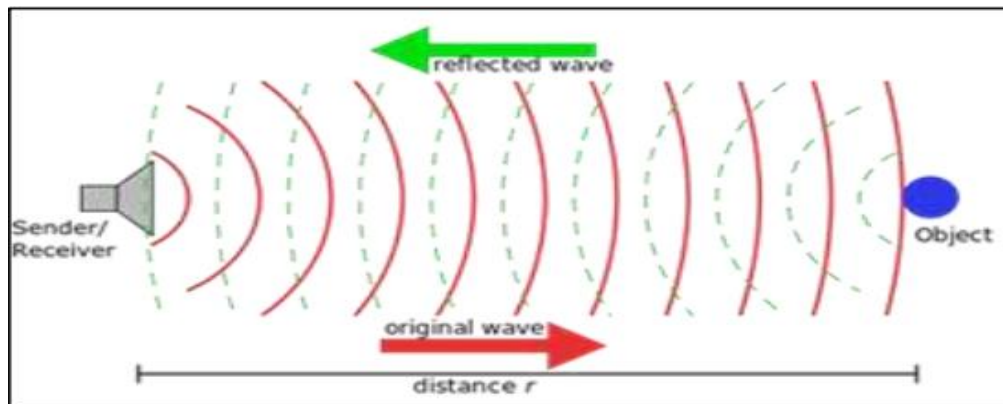


Figure 4. 3. Working principle of Ultrasonic sensor [32]

The ultrasonic sensor transmits sound waves and receives sound reflected from an object. When ultrasonic waves are incident on an object, diffused reflection of the energy takes place over a wide solid angle, which might be as high as 180 degrees [31]. Some fraction of the incident energy reflected back to the transducer in the form of echoes. The speed of sound wave depends on the distance between the objects and the sensor. If the object is very close to the sensor, the sound waves returns quickly, but if the object is far away from the sensor, the sound waves takes longer

to return. But if objects are too far away from the sensor, the signal takes so long to come back (or is very weak when it comes back) that the receiver cannot detect it [29].

Ultrasonic sensor is similar to sonar or radar in which interpretation of echoes from radio or sound waves to measure the attributes of a target by generating the high-frequency-sound waves up to 40 kHz. It uses transducer for converting energy into ultrasound or sound waves and measuring the properties of sound waves with frequency above the human audible range. Transducer has a transmitter to transmit signal to the object and receiver receives bounce back signal from object to the sensor. In parking ultrasonic sensor is used to detect whether the vehicles are present or not within the set distance by the way of ultrasonic reflection. The ultrasonic waves from transmitter head to the parking lot and again receives the ultrasonic waves reflected from the parking lot.

The receiver receives and evaluates the echo pulse to check whether the free space is available or not. The distance measured by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back from parking lot by recording the elapsed time between the sound wave being generated and the sound wave bouncing back, it is possible to calculate the distance between the ultrasonic sensor and the parking lot.

$$Distance = Speed\ of\ sound \times Time\ taken / 2$$

Where, time is the time duration for which the ultrasonic waves have travelled and Speed of sound in air is 340m/s. We divide the product of time and speed by 2 because the time is the total time it took to reach the obstacle and return.

Why this calculation is needed ultrasonic sensor has an ability to check the parking lot is free or not up to three meters so, before building parking area and installing ultrasonic sensor it is important to measure the distance between ultrasonic sensor and parking lot.

4.5. Tracking Device

Tracking device is an electronic tag that can be used to monitor the location of an object by using the radio signal or satellite signal. Tracking device was used for indoor like ultrasonic, mechanical, infrared, and inertial or radio signal and outdoor like GPS for high accuracy. GPS [33] is a space based navigation system which is being developed, operated and maintained by the Defense

Department of US. Although the US government is controlling it, it is freely accessible by anyone with a GPS receiver. GPS has passive and active tracking system.

4.5.1. Passive GPS Tracking System

Passive GPS tracking system monitor location and store its data based on certain types of events. This GPS system may log data such as turning the ignition on or off or opening and closing doors. The data stored on this kind of GPS tracking system is usually stored in internal memory or on a memory card, which can have downloaded by computer later for analysis. In some cases, the data can have sent automatically for wireless download at predetermined points/times or can be requested specific points.

4.5.2. Active GPS Tracking System

Active GPS tracking system is real-time system applications that automatically send the information on the GPS system to a central computer or system in real-time as it happens. This kind of system is usually a better option for commercial purposes such as fleet tracking and park tracking as it allows the driver to know exactly where the smart parking system located. GPS devices used to record the position of smart parking system as they make their drives. Some systems will store the data within the GPS tracking system itself (known as passive tracking) and some send the information to a centralized database or system via a modem within the GPS system unit on a regular basis (known as active tracking).

4.5.3. Types of GPS Tracking Unit

There are currently three categories of GPS tracking units. The categories were how GPS data is logged and retrieved.

- Data Loggers are usually the most basic type of GPS tracking; a GPS data logger simply logs the position of the object at regular intervals and retains it in an internal memory. Usually, GPS loggers have flash memory on board to record data that was been logged. The flash memory transferred and accessed using USB or accessed on the device itself. They might include devices that help log location for hikers, bikers and joggers.
- Data Pushers are GPS tracking units mainly used for security purposes. A data pusher sends data from the device to a central database at regular intervals, updating location,

direction, speed and distance. Data pushers are common in fleet control to manage trucks. For instance, delivery smart parking location can be located instantly and their progress tracked if owners of the parking system change the parking location. Data pushers are also common for espionage type tasks. It is extremely easy to watch the movements of an individual or valuable asset. This particular use of GPS tracking has become an important issue in the field of GPS tracking, because of its potential for abuse.

- Data Pullers are push data or send data when the unit reach a specific location or at specific intervals. These GPS units are usually always on and constantly monitoring their location. Most, if not all data puller unit also allow data pushing (the ability to query a location and other data from a GPS

4.5.4. GPS Position Determination

The satellite of GPS will transmit the one-way signals to the GPS receiver equipment on the earth. Every satellite will transmit the data that indicates its location and the time they sent out the signal. The timing information plays an important role in determine the user's location on the earth, thus, GPS satellites are equipped with atomic clock on board to provide an accurate time reference [34]. The distance between the particular satellite and the GPS receiver as shown in Figure 4.4 can be determined by calculating the travel time of a signal from the satellite to the receiver, where:

$$\text{Travel time} = \text{signal reception time} - \text{signal transmission time}$$

$$\text{Distance} = \text{travel time} \times \text{speed of light}$$

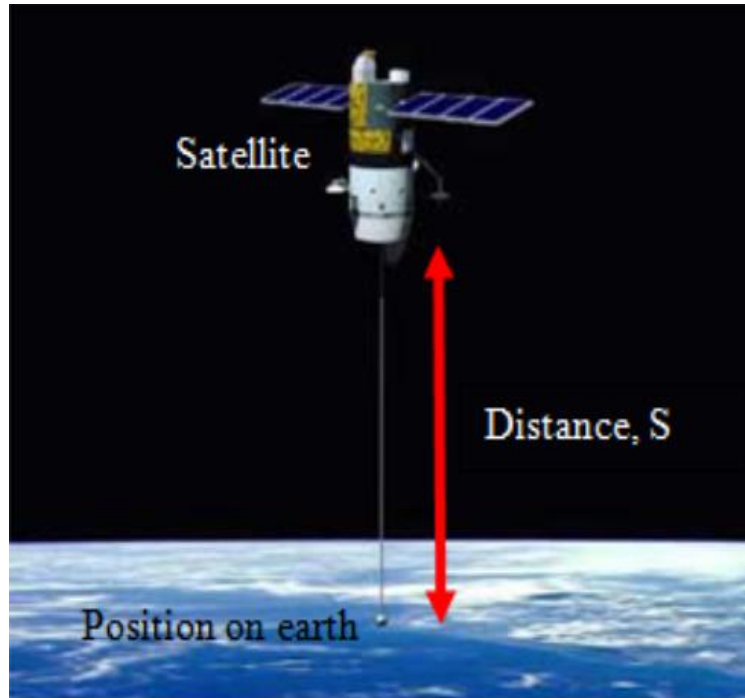


Figure 4. 4. Distance between the earth and the satellite [source Google]

4.6. Display Device Selection

Display means presenting information in a visual way. To establish a good communication between human world and machine world, display units play an important role. There are different types of display device, which display information/data to users. However, the three display devices have used for comparison for smart parking system. These are light-emitted diode, liquid crystal and seven-segment display.

4.6.1. Light-Emitting Diode

Light-emitting diode (LED) is a semiconductor light source LEDs are used as indicator lamps in many devices and are increasingly used for other lighting. Light emitting diodes have used in applications as diverse as aviation lighting, automotive lighting, advertising, and general lighting, parking light and traffic signals.

4.6.2. Seven-Segment Display

Seven-segment display (SSD) is an electronic display device for displaying decimal numerals that is an alternative to the more complex dot-matrix displays. Seven-segment displays are widely used

in digital clocks, electronic meters, and other electronic devices for displaying numerical information.

4.6.3. Liquid Crystal Display (LCD)

Liquid Crystal Display technology has originated in military displays in the 1960s. Commercially LCD displays became available in watches and calculators in the early 1970s. However, it became in 1971 as a display medium, LCDs have moved into a variety of fields, including miniature televisions, video cameras and monitors. A liquid-crystal display is a flat panel display, electronic visual display, or video display that uses the light modulating properties of liquid crystals. A liquid-crystal display device is visible in all light conditions, can display changes instantly, and consumes less power than other display devices. Liquid-crystal display produces no illumination of its own, but depends entirely on illumination falling on it from an external source for its visual effect.

4.6.4. Criteria to Selected Display Devices

To select the display device from the above listed, we are going to check the resolution, viewable size, dot pitch and contrast ratio.

Table 4. 5. Display Device Comparison

<i>Display devices</i>	<i>Resolution</i>	<i>Viewable</i>	<i>Dot pitch</i>	<i>Contrast ratio</i>
LED	Good	Very good	Excellent	Good
LCD	Good	Excellent	Very good	Excellent
SSD	Good	Good	Excellent	Good

To select display units the Since resolution the best indicator of picture quality in this case both the three display devices have good quality of resolutions but in case of viewable and contra ratio LCD was excellent to measure the sound and brightest bright and the darkest dark. Due to this LCD was selected for smart parking system to display the number of free parking lots to the owner of parks.

4.6.4.1. LCD Pins and Its Purposes

The need to know LCD pins and its purposes was to avoid the complexity of local system implementation in case of Arduino Uno interfacing to display free parking slot in chapter six. So, if the purposes each pin was understand it will be easy for implementation.

Table 4. 6. LCD pins and its purpose

Pin number	Symbol	Purpose
1	VSS	Ground connection
2	VDD	Power supply
3	VO	Contrast adjustments
4	RS	Register selection
5	RW	Read/Write
6	EN	Enable
7	D0	Data line0
8	D1	Data line1
9	D2	Data line2
10	D3	Data line3
11	D4	Data line4
12	D5	Data line5
13	D6	Data line6
14	D7	Data line7
15	A	Backlight anode
16	K	Backlight cathode

In table 4.6, the contrast adjustment pin changes the dark is display and it connects to the center pin of a potentiometer. The register selection pin sets the LCD to command or character mode and know and interpret the next set of data that was been transmitted via the data lines. RW pin is used write to, read from the display, and tied to the ground. EN pin tells the LCD when data is ready. Data pins 4-7 are using for actually transmitting data and data pins zero-three are not using in the SPS left unconnected. The backlight is connecting the anode pin to 5V and the cathode pin to ground with using resistors that limit the current.

4.7. Arduino simulator selection

Simulation is the execution of a model that represented by a computer programs that given information about the system for investigation.

Arduino simulators are the perfect platforms for programmers and designers to build the basics of circuit schematics and designs. But in this study Arduino simulators was used to design the circuit of smart parking system and to test the working functionality of the system without warring about money and time. There are different types of Arduino simulators but the most common and the most suitable simulators are

- **Virtual Breadboard:** which is created by James Caska in 1999 and it is designed for Breadboard integrated circuits and is able to function as a microcontroller development environment. But it is not designed to resolve circuit related problems because it is not a circuit analyzer and special simulators.
- **123D Circuits Simulator:** 123D Circuits is an online electronics simulator and collaborative design platform, it is from the partnership between Autodesk and Circuits.io, since in Fall 2013. It is an excellent tool for beginners who want to quickly get a grip of creating with Arduino.
- **ArduinoSim:** it is a cross-platform Arduino simulator and it is a great platform to program and design the circuit. ArduinoSim is not open source but it runs on windows and Linux operating system. The ArduinoSim was created with Python and it was built to integrate itself seamlessly with the Arduino environment.
- **Systemvision Simulator:** Systemvision is a cloud-based simulator used for running the Arduino simulations and circuit designs. It is a free online tool to simulate the system in easily way.
- **Proteus simulator:** Proteus virtual system modelling(VSM), is introduced for industrial and educational use. Proteus VSM combines circuit simulation, animated components and microprocessor models to facilitate co-simulation of complete microcontroller based designs. Proteus VSM makes it possible to develop and test designs before a physical prototype is constructed. The operating procedure and the information flows of proteus is not disturbing the real system operations and it is easy to understand the operations of a system. Due this and other advantages proteus simulator was selected for simulating the hardware part of smart parking systems.

CHAPTER FIVE: BUILDING SMART PARKING SYSTEM ARCHITECTURE

5.1. Architecture of Smart Parking System

The current method for finding free parking area in Addis Ababa city is based on asking the people where the parking area is available or searching parking area for much time. This way led traffic congestions, road accidents, wastage of resource and time. To avoid these problems, we were built smart parking system to access parking stations and free parking slots with the nearest by comparing the drivers' current location easily anywhere in the city using mobile phone. so, to implement the smart parking system building smart parking system architecture was a backbone because architecture plays great role to established, designing, implementing, and testing the structure of the smart parking system, before implementing the actual smart parking system. The architecture of smart parking system was built using IOT devices and a technology so developing a single architecture that covers all the requirements of the connected ecosystem of smart parking system was a big challenge. Due to this challenges reference architecture was used because it consists of a set of components and layers can be understood by the usage of specific technologies [40].

Due to this challenge reference architecture was used because reference architecture is a vendor-neutral and not specific to any set of technologies and covers multiple aspects of the application layers including parking services applications using web application and mobile application services by integrating and accessing data from the IoT devices and sensing layer. The principles, standards, reference models, and best practices of the reference architecture ensure that the new smart parking system's ranged funds have the greatest possible drivers and parking owners of achievement, better was of SPS implementation and better way service delivery to drivers or users. From the previous researcher's point of view and the current textbooks of IOT reference architecture is more interactively in the IOT based hardware and software systems. Reference architecture is composed of core platform services and application-level components to facilitate the processing needs across three major areas of a typical IoT solution [Microsoft 2016] these are:

- Device connectivity
- Data processing, analytics, and management
- Presentation and business connectivity

These platform services and application level components was essential for building smart parking system architecture. In the smart parking system, the stakeholders, systems and subsystems were described using architectural views, viewpoints and perspectives.

5.2. Architectural views and perspectives

A viewpoint is a collection of patterns, templates, and conventions for constructing one type of view. It defines the stakeholders whose concerns are been reflected in the viewpoint and the guidelines, principles, and template models for constructing its views [IEEE Architecture].

5.2.1. Architectural views

Architectural views provide a standardized way for structuring architectural descriptions. To build reference-based architecture of smart parking system using IOT it is necessary to choose architectural views. The architectural views divided into five [Rozanski 2012] finally we chosen one or more views for the developments of the system.

Information view: The main purposes of connected and smart objects in the IoT was the exchange of information in the smart parking system and with external users and the information was stored, managed and distributed in the architecture. The information view helps to generate an overview about static information structure and dynamic information flow in the smart parking system. It focuses on the description, the handling and the life cycle of the information and the flow of information through the system and the components involved.

Physical view: is central to IoT, the interaction of variety of physical objects and their aspects so, it impossible to present a common model for all IoT use-cases. Therefore, the physical view was not covered in the IoT Reference Architecture especially in the smart parking system.

The context view: it was generated at the beginning of the architecture process. Context view describes the relationships, dependencies, and interactions between the system and its environment (the people, systems, and external entities with which it interacts) [Rozanski 2012]. The concerns addressed by the context view are [Rozanski 2013] System scope and responsibilities, identity of external entities and services and data used, nature and characteristics of external entities, identity and responsibilities of external interfaces, nature and characteristics of external interfaces, other external interdependencies, and impact of the system on its environment, overall completeness,

consistency, and coherence was described. But in case of smart parking system context view was used to identify the interactions and the relationship of systems and parking users and also to identify the environment factors especially in the Ultrasonic sensor in the parking areas.

Functional View: Describes the smart parking systems functional Components and its interactions with the other components. The functional view reflects the qualitative properties the systems like maintainability and performance of smart parking system.

Deployment view: A connected and smart object in the IoT was been realized in many different ways and communicated using many different technologies. Deployment and Operation view is very important to address how actual smart parking system realized by selecting technologies and making, them communicate and operate in a comprehensive way. The Deployment View aims at providing users of the IoT Reference Model with a set of guidelines to drive them through the different design choices that they have to face while designing the actual implementation of their services.

5.2.2. Architectural perspective

An architectural perspective is a collection of activities, tactics, and guidelines that are used to ensure that a system exhibits a particular set of related quality properties that require consideration across a number of the system 's architectural views [Rozanski 2005]. The perspective includes evolution and interoperability, performance, security, and privacy, availability using IoT reference model to provide the concepts and definitions on which IoT architectures can build. The reference model consists of several sub-models that set the scope for the IoT design space and that address the previously discussed architectural view and perspectives.

5.3. Architecture Guideline Principles

The reference architecture provides a degree of component commonality that allows assembling secure, complex solutions supporting extreme scale, and yet allowing for maximum flexibility with regard to possible solution of IOT smart parking systems. In the IOT architecture the following guiding principles was used.

Heterogeneity and security: many devices were used in the proposed reference model must accommodate for a vast variety of environments (parking stations), devices (Ultrasonic

sensors, Arduino Uno), information processing (from parking stations to parking users and vice versa), and standards. It also handles hardware and software heterogeneity especially in security Issue Because IoT solutions represented a powerful connection between the digital and physical worlds, building secure systems were a necessary foundation for building safe systems. This reference model contemplates security and privacy measures across all areas, including device and user identity, authentication and authorization, data protection for data at rest and data in motion, as well as strategies for data attestation.

Hyper-scale deployments: The proposed reference based architecture of smart parking system should support many connected devices in ThingSpeak IoT platform it should allow proof-of-concepts and pilot researches that start with a small number of devices to be scaled-out to hyper-scale dimensions.

Flexibility: The heterogeneous needs of the IoT smart parking services necessitate open-ended composition of various services and components to users or drivers easily. The smart parking system may need to change based on users, organizations need. It was a loosely coupled composition of services and processing modules in the users need.

5.4. Reference Architectures

Based on reference model, architectural guideline, architectural view and perspectives Identifying and structuring architecture of smart parking system was a tedious process with much negotiation to abstract from specific needs and technologies. Such a reference can serve as an overall, generic guideline; not all domain applications will require each detail for real-life smart parking system implementation. Building reference based architecture for smart parking system had the following benefits.

- ***Connectivity and communication:*** Data was collected from Ultrasonic sensor and disseminated using unicast, multicast and anycast connectivity to web based smart parking system and to drivers.
- ***Device management:*** a device was added or a device configuration changes and must be propagated to other devices in the smart parking system based on the requirement of users, drivers or parking owners or technologies.

- **Data collection, analysis:** the data was collected and analyzed to extract information's and offering parking services.
- **Scalability:** was used to handle increased processing volumes for different installation of smart parking system.
- **Security:** to provide trust, privacy, and are required for all aspects of the smart parking system especially in the IOT context.

In the smart parking system reference based architecture has different components and sub-systems and each components and sub-systems was interacted and communicated each other because the reference based architecture of smart parking system supports four communication models these are device-to-device, device-to-cloud, device-to-gateway and back-end data-sharing communication models. In the device-to-device communication models two or more sensor devices are directly connecting and communicating with device layer. Device-to-cloud communication model, the all the smart parking system device in the sensing, device, communication and application layer connects onto a web based parking service to exchange information and management message between users and park owners. This model was advantageous for the existing communications mechanisms like wired local area network or wireless network like Wi-Fi associations to ascertain a connection between the device and the information-processing network that ultimately connects to the cloud service. Device-to-gateway Communication Models The local gateway device is a smartphone running an app to communicate with parking device and transmit data to a cloud service. Back-end data-sharing model is an extension of device-cloud models. Permits users to export and analyze sensible object knowledge from a cloud service together with knowledge from alternative sources and upload the single sensor data in cloud web server for the third party. To build the architecture of smart parking system all the communication models needed, but the most appropriate is device-to-cloud and device-to-device communication models.

Reference architectures also support different types of protocols and deployment models. the protocols we were used in the current study IPV4 and IPV6 and so on but, in the current day IPV4 was not sufficient especially in IOT technology millions of device was connected. As CISCO states in 2025 billions of IOT device is connected to the Internet and as the IOT continues to grow, devices that require true end-to-end Internet connectivity will not be able to rely on IPv4, the

protocol most Internet services use today. So, we would use IPv6 protocol because the Internet is running out of original IPv4 addresses. While IPv4 can support 4.3 billion devices connected to the Internet, IPv6 with two to the 128th power addresses, is for all practical purposes inexhaustible. This represents about 340 trillion, trillion, trillion addresses, which more than satisfies the demand of the estimated 100 billion IoT devices going into service in the coming decades. Given the estimated permanence of some of the sensors and other devices imagined for the Internet of Things, design decisions will affect the utility of solutions periods from now. Due to this it was better to use IPV6 in smart parking system to access parking information over internet or Wi-Fi. The general overview of the reference based architecture of smart parking system was developed based on the objective of the system by considering architectural views, perspectives, guidelines, benefits, communication models, protocols and deployment models.

The general overview of the reference architecture of Smart parking system was described how the driver could get the parking area using smart phone in Addis Ababa city by considering the web application and mobile application of implementation and how the parking slots was identified and transferred to web applications and then drivers' mobile applications to fetch parking services specially parking locations and free slot availabilities in Addis Ababa city.

The slot was identified using Ultrasonic sensor and analyzed using Arduino Uno. After the free slot was been detected the data was send to local server over Wi-Fi. The data from each local sever was been sent to central server for more analysis. The driver or user was got parking services and locations. The GPS was installed in each parking area to get location identification and distance from each parking area. The path navigation was not considered to assist the driver which root was been free to gate the smart parking area.

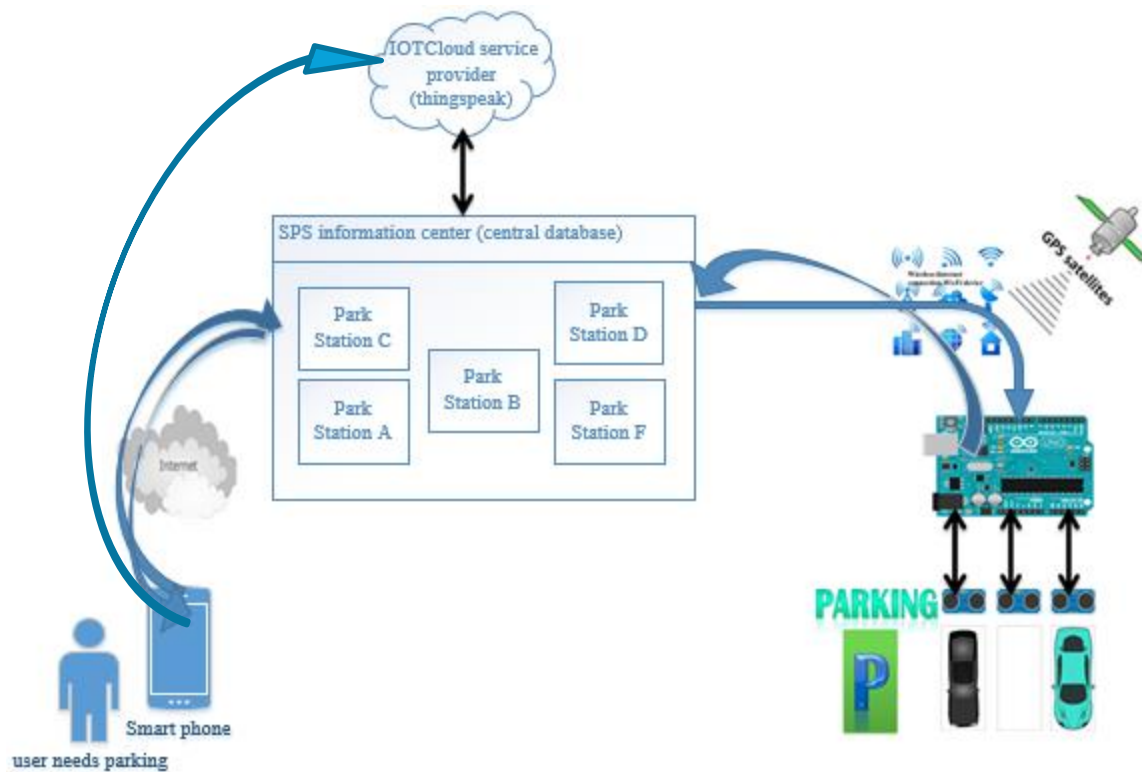


Figure 5.1. General overview of SPS architecture (my own work)

The availability of free parking slots was identified using Ultrasonic sensors and the distance was analyzed using Arduino Uno then the status of parking information was sent to SPS information center from each parking stations over Wi-Fi devices and the parking station locations was identified using GPS the information was stored in the central database of SPS or IoT Cloud service provider (ThingSpeak) then drivers can have accessed parking services easily using mobile phone. Based on the general overview of smart parking system architecture, detailed parking architecture was built by dividing into four layers these are from the bottom to top sensing layers, device layer, communication layer and application layer. Why we were divided smart parking system architecture into four layers was to achieved the goal and the objective of smart parking system specially to develop real-time application and also reference architecture supports many components, systems and sub-systems and also sensor to Arduino interactions, interfaces and hardware and software prototype implementations. IOT based reference architecture for smart parking system, which contains IoTCloud service provider (ThingSpeak), which provides storage of information about status about parking slots in a parking stations and parking station locations.

The centralized database which; manages to store entire smart parking systems information such as number of slots availability, payment information and distance from the current location of driver to availability parking area. Moreover, this information accessed through some secured gateway because Gateways support a parameter configuration of systems when the communication crosses different networking environments in Addis Ababa city. Central database stores information provided by local units located at each parking stations. The smart parking system allowed a driver to search and find information on parking spaces from central database without directly access the local server database. Local server database was located in each parking stations and stores the information of each parking space and locations co-ordinations. The Arduino module connects with the IoTCloud server through an Internet connection to transfer data from the local car park to the cloud server database specially to access park station locations. Also, mobile application was implemented and install in the users' mobile phone and users use this application to get parking services. The users access the system via 2G/3G/4G/5G mobile connections.

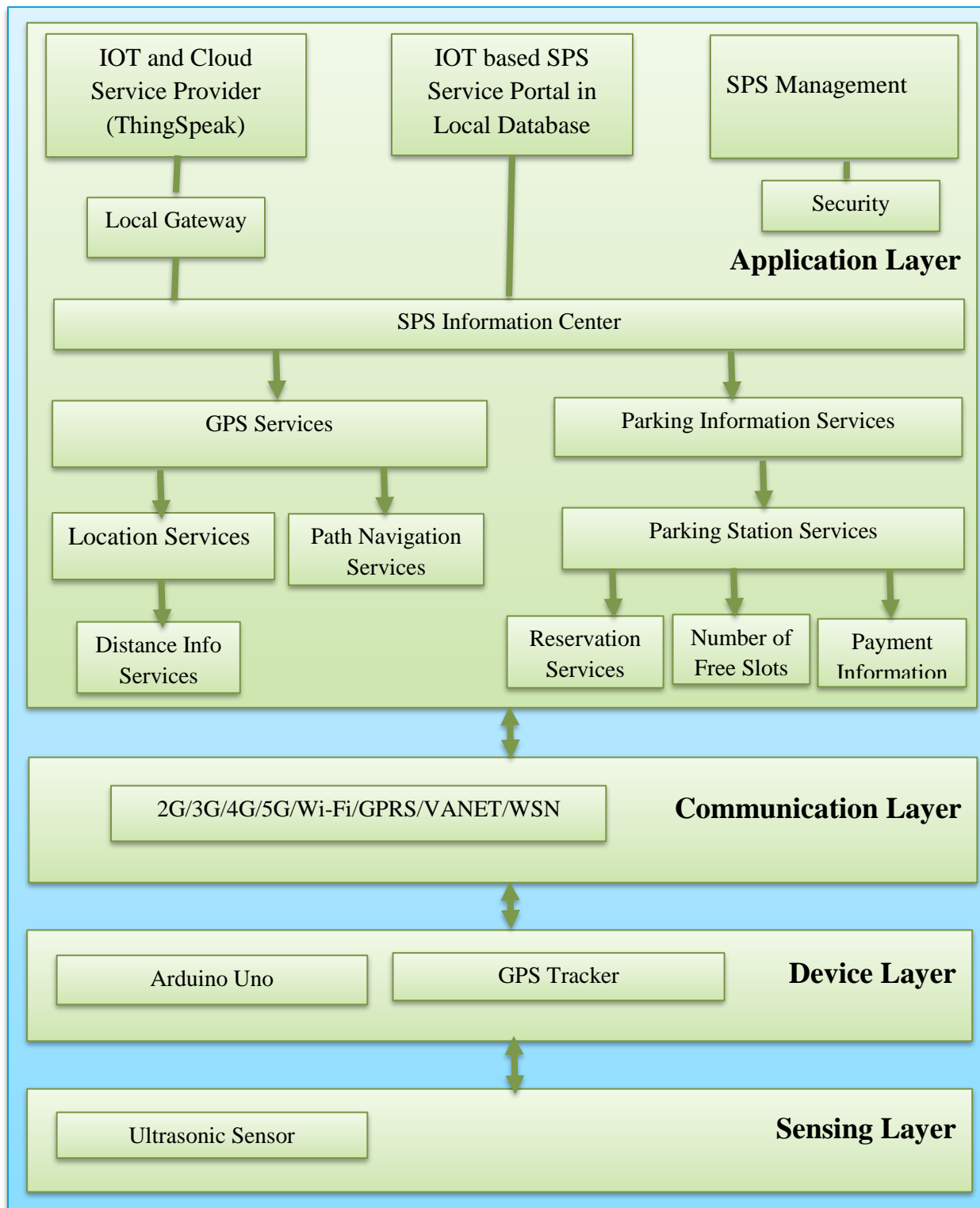


Figure 5. 2. Architecture of Smart Parking System (my own work)

In the figure 5.2 reference architecture of smart parking system sensing layer was found in the bottom of the architecture to identifying the number of free slots in the parking stations by installed Ultrasonic sensor in each parking stations. The sensing layer was direct communicated with device layer because the identified distance using Ultrasonic sensor was analyzed using Arduino Uno.

Device layer is a bridge between the sensing layer and the communication layer. The devices layer contains the GPS tracker to get path navigation and location services, Arduino UNO, which is the brain to develop smart parking system. In the device layer we were also used GPS tracker to get the location co-ordination of parking stations in Addis Ababa city then the status of free parking slots and parking station locations was transmitted to application layer using communication network in the communication layer. Arduino Uno accepts data from ultrasonic sensor and sends data to local servers. In this layer, devices were integrated and communicated in more secure, convenient and easy way.

Communication layer is the connectivity networks of the SPS. It is medium to transmit or receive data over the physical layer, networking layer and application layer.

Application layer contains the all the smart parking services and smart parking managements. IoTCloud service providers were used to store data and access data directly from ThingSpeak. IoTCloud service providers were directly communicated with IOT based SPS service portal in local database and SPS management. IOT based SPS service portal in local database contains all the services in SPS information center (GPS services, parking information services). All this service stored in service portal of local database and which was directly communicated IoTCloud service provider of server. Data from SPS information center or Integrated IOT based SPS service portal in local database was been analyzed and interpreted at scale to generate perceptions and perform and status of parking in each area in the city before send information's to users in a secure and efficient way. The SPS managements are a central layer with continuous processes required to keep the service availability and the security of smart parking system up-to-date to ensuring sufficient services to user and the security of smart parking system. The smart parking system management combines all functionalities that were needed to govern system using IoT technologies. In the smart parking management system [35] [36] "System allocates available parking space to a given driver to park their vehicle" but in the managing part of smart parking system [37] IoT concept contains flexibility of the system because the system may need change

based on the requirements of users and owners and also the security [38] management services, networks and other application services of the system must be ensured to identify an authorized and authenticated user even if IoT device has venerable to hackers. So, it is very challenging to manage using intelligent parking system is the best way [39] .The in the smart parking system manager of smart parking system was also responsible for failure handling of devices, systems, networks. The failure may be due to human or environmental factors. In case of device security level, End-End encryption and platform and application integrity verification have been used, but the End-to-End encryption security was the most vital because the architecture was based on device-to-cloud communication model.

Smart parking system information center was decomposed into GPS services and parking information services. The GPS services also decomposed into location service and path navigation service to get the location of parking stations and to get the shortest path to go free parking stations respectively. The current research path navigation was not developed But, simply to give the future researcher as reference in case of developing smart parking system by also analyzing free traffic congestions in addition to shortest roots. In case of identifying parking station location the GPS co-ordination was used for each parking area and the data was stored in the database of smart parking system so, the parking users was could easily access parking station locations including the distance from their current location to parking stations.

Smart parking information services were containing smart parking stations. The smart parking stations contains also contains the total number of parking stations, number of free slots and the payment information in each parking stations. The information was analyzed in the smart parking information services in each parking stations and then stored in the central database.

CHAPTER SIX: IMPLEMENTATION OF SMART PARKING SYSTEM

6.1 Implementation of Smart Parking System

Implementation of smart parking system for real time application was done to monitor the status of parking slots, location of parking area and distance from the current location of user to each parking stations. The implementation of the smart parking system was developed based on the real-time requirements of parking services for the parking user and parking owners which was stated on the reference architecture that contains different components and layers [40] by using IoT technologies like hardware and software, each component. The implementation of the smart parking system was dividing into two the hardware implementation and software implementation. The hardware implementation was free slot identification, location identification and data transfer from Arduino Uno to database. The software implementation was divided into two, which are mobile application implementation and web server implementation. The mobile application implementation covers location tracking and parking information. The web server implementation included the restful web services implementation for the communication of mobile application implementation and the central system. In case of hardware was implemented and the data was transferred to database using proteus professional software.

6.2 Simulation of Smart Parking System

The hardware of smart parking system was simulated on Proteus 8.5 professional PCB design software tool and the implementation was programmed, developed and compiled in Arduino Integrated development environment (IDE) using embedded C++ programming language. The Proteus software is a procedural programming language created in 1998 by Simone Zanella. Proteus is a multiplatform that supports windows, DOS and UNIX. But for the simulation of smart parking system the proteus software was installed on windows and the interface of the proteus was showed in figure 6.1.

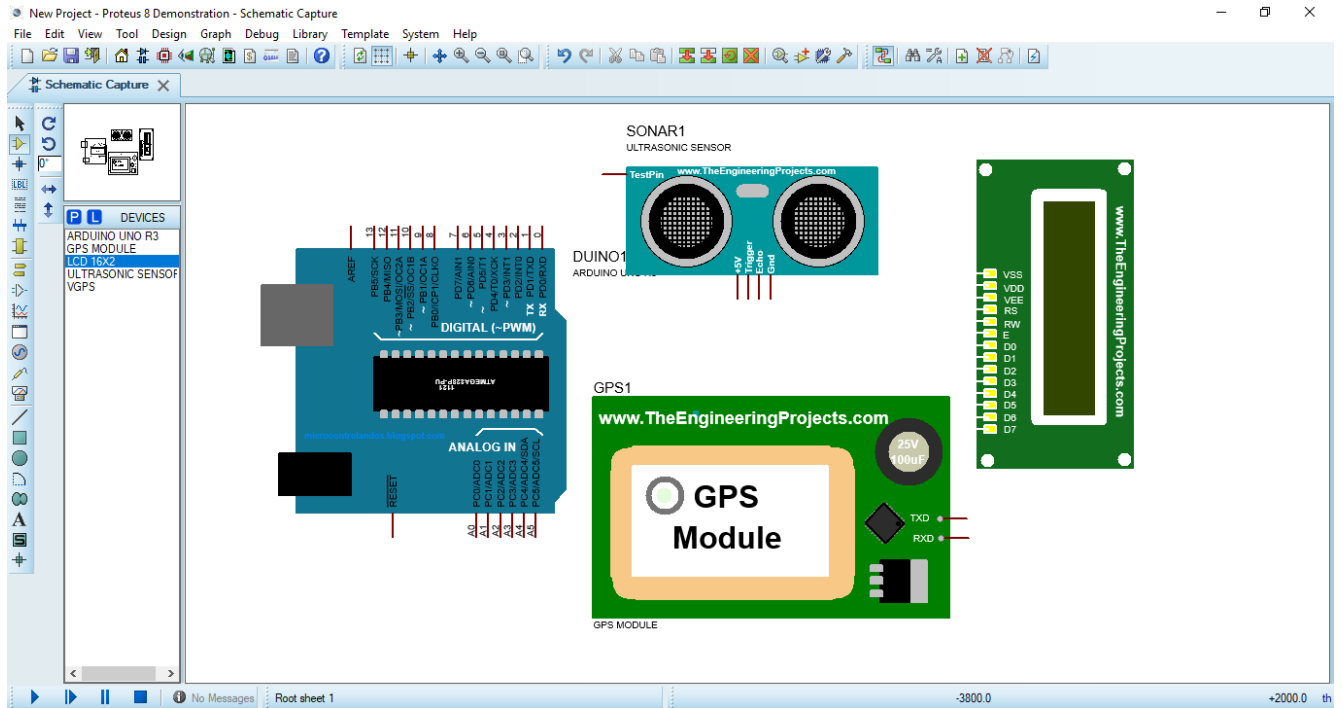


Figure 6. 1. Proteus Simulator

All the necessary tools were installed in the proteus library for the development of smart parking system. The tools were simulated to identify availability of free slot, to track locations, and to transfer the free slot data from Arduino Uno to the database. The device was connected and interfaced using wire and their pins.

6.3 Free Slot Identification of Smart Parking System

Identifying the free slots were one objective in the smart parking system, because user need to access the available free parking stations using mobile phone, so the status of free slot in each parking area in Addis Ababa city was identified using ultrasonic sensor and Arduino Uno. The Ultrasonic sensor and Arduino Uno was connected in figure 6.2.



Figure 6. 2. free slot identification

HC-SR04 sensor was used to detect car within a range and Arduino Uno R3 used to calculate the distance. Ultrasonic sensor connected with Arduino Uno to analyze and identify the status of free parking slots in each parking stations. To identify the free slots the ultrasonic sensors were installed in front of the parking area with an angle of 7° - 15° for more accuracy and the slot must be within 4meters in the front direction because ultrasonic sensor detects a car from 2mm to 4m in maximum range. But the installation may be depending on the size of the slots because it may detect the adjacent slots in the nearby. Ultrasonic sensor has loudspeaker (transmitter) and microphone (receiver). So, to detect a car with in 4meter range loudspeaker sent a short ultrasonic sound burst then the microphone starts to listen. This sound travels to the car and reflected by the same angle back to microphone so Emergent angle equals to incident angle. When sound travels from loudspeaker to the car and back to the microphone, this time Arduino measure and calculate the distance. To check whether the free slots were available or not within a range, the distance was calculated using the following formula.

$$\text{Distance(s)} = \text{Speed of sound (v)} \times \text{Time taken (t)} / 2$$

For example, if distance is one meter from ultrasonic sensor to car in the smart parking system area then the time was calculated to hear by microphone using the following formula.

$$t = 2 \times s / v = 2 \times 1 / 330 = 0.0060616060 \sim 6\text{ms}$$

Therefore, for every 1m distance the microphone hears the sound in 6milliseconds. If the maximum ranges 4ms are takes, the microphone heard after 24millisecond. This speed of sound

may have affected by temperature. If temperature increases by 5⁰c, the speed of sound changes nearly by 1%. Based on the above formula the microphone reads the distance and Arduino calculates the distance then the availability of slots was determined. But the ultrasonic sensor detects any object so we must avoid any objects in the parking slot area and the distance and angle of Ultrasonic sensor must be considered before install in in the parking stations.

6.3.1. Circuit Design of Slots Identification in Smart Parking System

The circuit design was done to check the working functionality of Arduino Uno and Ultrasonic sensor in case of identifying and analyzing the free slot availability in each parking stations. The circuit was simulated on proteus software and programmed in Arduino IDE using C++ language. The circuit was done using three Ultrasonic sensors, one Arduino Uno, one virtual terminal, test buttons and connected wire.

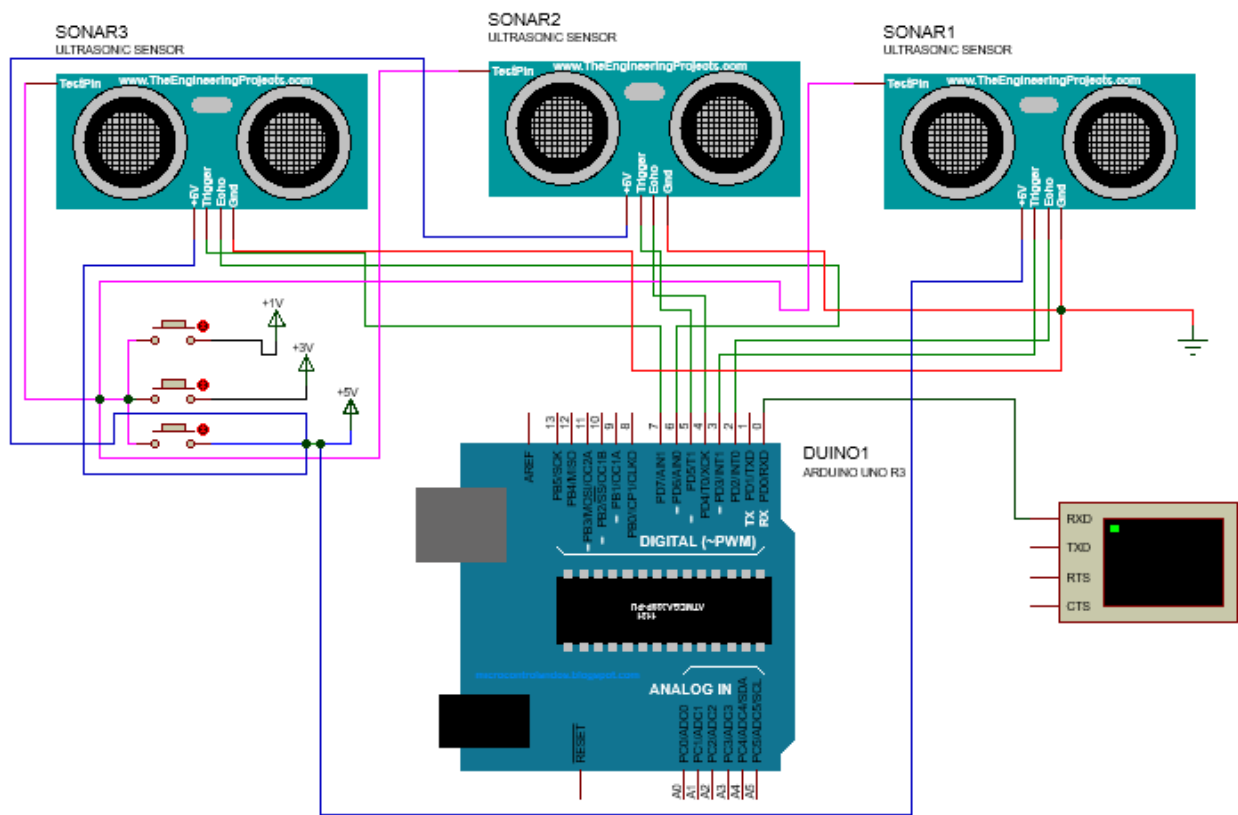


Figure 6. 3. Ultrasonic sensor and Arduino Uno circuit design

We were used three Ultrasonic sensors to test the systems functionality in the three slots by making the test buttons on and off after connecting the system using wire on their pins the result

of parking slots was displayed in the virtual terminals. The pins connection using wire was been summarized in table 6.1.

Table 6. 1. Ultrasonic sensor and Arduino Uno circuit design pins connection

<i>Device</i>	<i>Pins</i>						
Arduino Uno	0	2	3	4	5	6	7
Ultrasonic sensor#1		Echo	Trigger				
Ultrasonic sensor#2				Echo	Trigger		
Ultrasonic sensor#3						Echo	Trigger
Virtual terminal	RXD						

Test pin of the entire three ultrasonic sensor was been connected with TTSL push button to test whether the circuit design was working or not on virtual terminal. When the push button was pressed the simulation was seen in virtual terminal. The power pin (+5v) of the ultrasonic sensors was been connected with the push button which has +5v and all the GND pins of ultrasonic sensors were connected with ground.

After connecting Ultrasonic sensor and Arduino Uno using wire, the result of free slot was displayed on virtual terminal using C++ programing language. The following sample code is used to identify the free parking slots. In table 6.1 echo pins and the trig pins of ultrasonic sensor was interfaced with Arduino Uno to get input and output of data respectively based on the interfacing the pins value was assigned in line2. The three Ultrasonic sensors were used to check three parking slots and the slot and distance was declared in line3 and the pinMode of each echo and trig was read in void setup () method and write in the void slot () method before sending pulse sound in line 16 to the car to calculate the distance. In the void loop () method in line19 the distance was calculated based on speed of sound. The transmitter sends pulse sound to the car from the three

Ultrasonic sensors and the distance was reflected back from the car to receiver then the Arduino was analyzed the distance. The maximum range of Ultrasonic sensor detect a car up to 400 cm distance so, we were used looping statement to check the status of slots repeatedly.

```
1. #include <Ultrasonic.h>
2. int echo1=2, int trig1=3, int echo2=4, int trig2=5, int echo3=6 trig3=7;
3. long duration, distance, slot1, slot2, slot3;
4. void setup() {
5.   Serial.begin(9600);
6.   pinMode(trig1, OUTPUT);pinMode(echo1,INPUT);
7.   pinMode(trig2, OUTPUT);pinMode(echo2,INPUT);
8.   pinMode(trig3, OUTPUT);pinMode(echo3,INPUT);
9. }
10. void slot(int trig, int echo){
11.   digitalWrite(trig, LOW);
12.   delay(10)
13.   digitalWrite(trig, HIGH);
14.   delay(10);
15.   digitalWrite(trig, LOW);
16.   duration=PulseIN(echo, HIGH);
17.   distance=((duration/2)*0.034);
18. }
19. void loop() //calculate the distance based on speed of sound
20.   slot(trig1, echo1); slot1=distance; //Serial.println(slot1);
21.   slot(trig2, echo2); slot2=distance; //Serial.println(slot2);
22.   slot(trig3, echo3); slot3=distance; //Serial.println(slot3);
23.   if(slot1>400){
24.     Serial.println("slot1 is free");
25.   Else {
26.     Serial.println("slot1 is full");
27.   }
28.   if(slot2>400){
29.     Serial.println("slot2 is free");
30.   } else {
31.     Serial.println("slot2 is full");
32.   }
33.   if(slot3>400){
34.     Serial.println("slot3 is free");
35.   }else{
36.     Serial.println("slot3 is full");
```

```

37. }
38. delay(1000);
39. }

```

6.4. Location Identification of Smart Parking System

Parking location and path navigation is very important for users and drivers, especially in case of short period. The GPS system was installed in the smart parking areas in Addis Ababa city to get location coordination by drivers and users. The location coordination of each parking stations was stored in the database of smart parking system and the simulation was done on proteus software by adding the necessary library and also was programmed on Arduino IDE in C++ language.

6.4.1. Circuit design for Location identification of smart parking system

The location identification of circuit design of smart parking system was developed using Arduino Uno and GPS. Circuit was designed to track the location of smart parking area in Addis Ababa city.

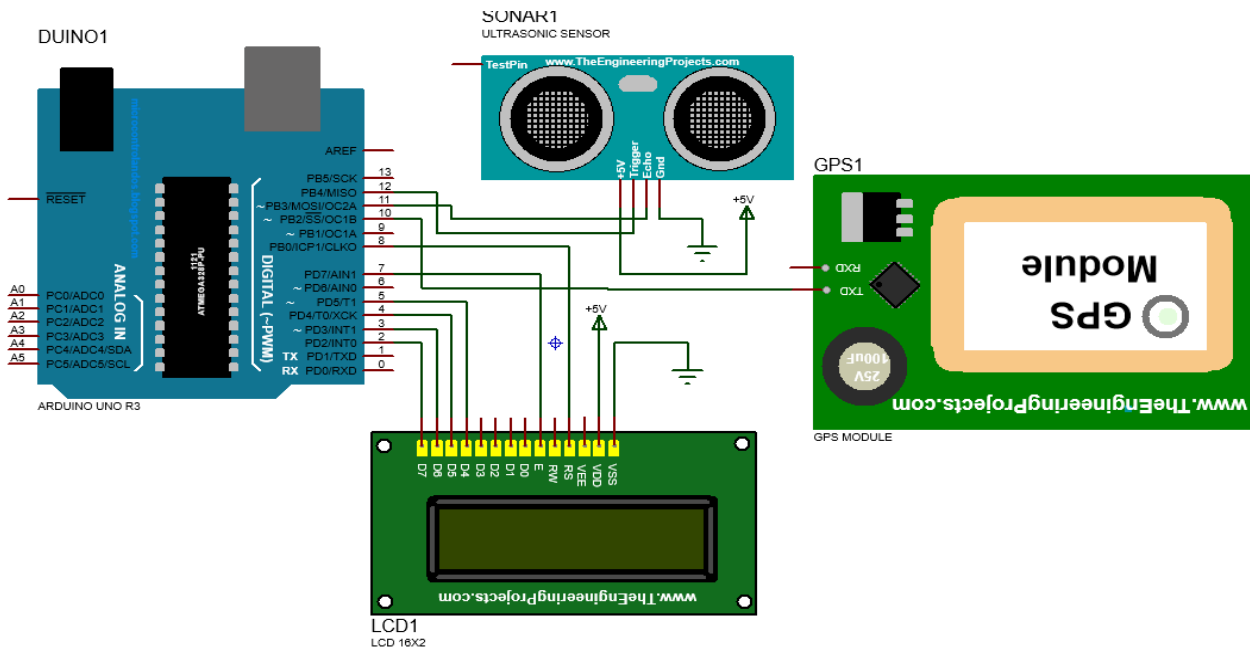


Figure 6. 4. Location tracking circuit design

The pin connection was summarized in table 6.2.

Table 6. 2.Location tracking circuit design pins connection

<i>Device</i>	<i>Pins</i>								
Arduino Uno	2	3	4	5	7	8	10	11	12
Ultrasonic sensor								Echo	Trigger
GPS							TXD		
LDC	D7	D6	D5	D4	E	RS			

The power of ultrasonic sensor and LCD was connected with +5v and the VSS pin of LCD and GND of ultrasonic sensor was connected with ground. The status of free slots of park was identified by ultrasonic sensor and the distance from ultrasonic sensor to car was been calculated by Arduino Uno also the status was displayed in 16 *2 LCD display and the location was done using GPS tracker.

To program GPS with Arduino Uno the Thiny library of GPS was installed in Arduino IDE and the GPS module was installed on proteus software then we were programmed using C++ language on Arduino IDE. The sample code that used to identify the location of parking stations was put at appendix 3.

6.5. Send Data from Arduino Uno to Database

To send data from Arduino Uno to database we were directly connect Arduino Uno to a MySQL server by installing MySQL Connector/Arduino library on Arduino IDE without using an intermediate computer, a web application- or cloud-based service applications. The connection of Arduino and MYSQL sample code (see Appendix 3). The data from Arduino Uno was inserted, updated, delated and selected in the parking database that was create using XAMPP server on PHPMyadmin. The sample code to select data from MSQL database was (in Appendix 4). To send data from Arduino Uno to MYSQL database we were used EPS8266 Wi-Fi module or Arduino Internet module. The data was continuously sent from Arduino Uno to database about the status of parking stations and updated also continuously in the database. The sample updated web client Arduino and MYSQL code about status of parking station code (in appendix 5) and the back-end PHP and MYSQL code (in appendix 5.1).

6.6 Mobile Application Implementation of Smart Parking System

6.6.1. Hybrid Application

The hybrid mobile app has a nature of native and HTML5 mobile applications. A native mobile application was built for a particular or distributed platform from the app store. Various native mobile platforms use free development tools and programming language. For example, Native iOS platform uses Objective-C, Android platform uses Java, and Windows platform uses C #, and so on using their integrated development environment (IDE) to develop the mobile applications. They have advanced interactive interfaces and best performance. Even if the native mobile applications have best performance they require more resources, efforts and time to develop. But the html5 application is also known as a web application and it is stored in a remote server. The application is distributed over the internet through a browser as an interface. The web application uses web technologies [41] like, HTML5, CSS, and JavaScript. It is possible to develop advanced and dynamic web applications that can run in the web browsers of desktop or mobile devices. However, the web application does not give the feel and the app store distribution capabilities as the native apps do. So, the hybrid application is an HTML5/web application which is wrapped inside a native container so that it was possible to run on any operating system or devices. The main advantages of the hybrid applications are that they take relatively less time to build, the cost is low and they are easy to develop and maintain, especially for the lighter applications than the native apps. Nevertheless, there might be an unkind skill if the application grows bigger and contains more features. Native applications give high performance, best looking UIs, and other high-end features but they also require more time and cost while building an app. With the hybrid application development, we were focused on web applications using Apache Cordova platform to run the application in many mobiles operating because Apache Cordova supports many mobile operating systems like, Windows phone, android, iOS, iPhone, etc.

6.6.2. Apache Cordova

Apache Cordova is a mobile development framework to develop hybrid mobile application. It supports and networks with different device-specific applications and also it combines the features of all the native applications into single JavaScript applications that are accessible by the hybrid application. It simply loads up the JavaScript API when the mobile application is started and the output changes according to the platform of the device. Inside the web view, hybrid applications

are executed in the presence of Cordova. Cordova on top provides the concept of plugins which has the method to interact with single or multiple native APIs and enables to use native device resources, like camera, notifications, basic file access, etc.

The hybrid mobile application was developed for smart parking system using Apache Cordova. Apache Cordova was first installed on a local system to package an HTML5 application as a native mobile application with NetBeans IDE and we were use npm for NodeJS package manager, to install and to update Cordova. npm can manage packages that are local dependencies of a particular system, as well as globally-installed JavaScript tools and also installed java SDK and android SDK. The necessary plugins were also installed like mobile operating platforms, cameras and locations. After installing all the necessary tools on local device, the mobile applications were implemented for smart parking system. The mobile application implementation of smart parking system was developed to access parking information using mobile phone. The interface of mobile application was developed using HTML5, CSS, bootstrap and Java script, and jQuery as showed in figure 6.5 and another sample JS sample code [see appendix 7]. After implementing the mobile application, the drivers or users installed the applications on their mobile phone to access smart parking services like location distance and number of free slots information. When drivers or users open the smart parking mobile application, the system automatically fetches the drivers or users' current location and all available parking stations. After fetching the current location of drivers or users the system calculates the distance from their current location to all available parking stations in Addis Ababa city. Then the drivers can read the distance and free parking slots from their current location to each available parking stations. After selecting the free parking stations from the list of parking stations drivers or users first register in the system before parking the cars. After registering cars plate number then the system automatically permits them based on the available free parking slots. In the figure 6.5 the sample parking stations with available free parking lots was displayed. For more information drivers can use Google Map to see the parking locations.

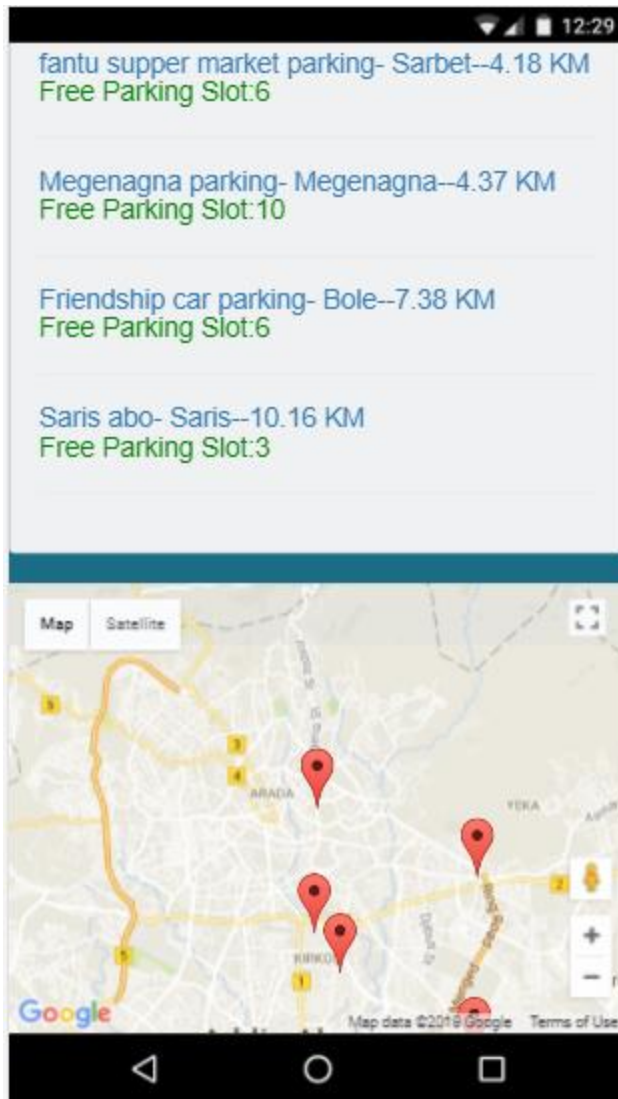


Figure 6. 5. Mobile application home screen

If the drivers or users click on the selected parking stations the system asks to enter car plate numbers including car codes then the user see notification about parking on their mobile. because all parking information was stored in the database of smart parking system. Then to fetch the available information mobile application and web application was connected using restful services applications. The mobile application implementation was developed using HTML and Java Script applications. In the figure 6.6 shows the sample Java Script code that was used to implement paring stations.

```

function accessParkingStaions() {
  var GetParkingStations =[]
  $.ajax({
    url: "http://localhost/BuildSparkingSystem/parkingStations/getPstations.php",
    type: "POST",
    dataType: "json",
    contentType: 'application/json',
    data: '{ "name": "Cordova", "another_thing" : "thing" }',
    success: function() {
      navigator.notification.alert("Success!");
    },
    error: function(jqXHR, textStatus, errorThrown) {
      console.log(textStatus + jqXHR.responseText);
    }
  });
}

```

Figure 6. 6. Sample snapshot JS code to get parking stations

In the figure 6.6 the sample snapshot code was used to access parking stations using ajax and Json services in http protocol. So, if the parking stations were got from central database the driver can easily access all parking information with short period of time in Google Map. The drivers or users used mobile networks, Internet, over Wi-Fi, 2G, 3G, 4G and in the future 5G cellular network, to obtain the information of parking availability, parking station locations, distance from the current location of drivers or users and payment information in each parking stations.

6.7. Web application Implementation of Smart Parking System

The web application of smart parking system was implemented using html, CSS, bootstrap, jQuery and Java script language for front-end implementation and the back-end implementation was done on NetBeans IDE 8.2 using PHP and MySQL through Glassfish and JSON on XAMPP control panel V 3.2.2. The web application implementation was a bridge between the mobile application and hardware implementation of smart parking system in the local database. The web application implementation was containing all the parking stations information's in each distinct, locations of smart parking system and the mobile based application in Addis Ababa city. The web application was continuously communicating with the local server via Internet in each parking stations. The web server was send or receives smart parking system data to /from local server using http protocol. In the smart parking system each parking station have a manager to manage the systems such as the available free parking lots, system security, payments and user and car profiles. Each manager in each parking station has their own account to manage their parking stations. The

sample code for the login page of the manager was [see in appendix 8]. When the manager login the system shows only his/her parking stations. The managers in each parking stations have privileges to access or not to access the parking information in other parking stations. The manager has no authority to access unauthorized information of other parking system in the central server. The web application implementation gave privilege for each managers, customers and drivers. The drivers or users could access only the parking stations, number of free parking slots and payments. The drivers or users could first check parking stations, locations, free slots in each parking area and payments in an hour using smart phone. in the figure 6.7 the status of Megenagna parking slots was displayed. Based on the status of the slots drivers/ customers can park their car by sending plate numbers and including car codes.

#	Slot	Status	Duration
1	Slot 1	Free	
2	Slot 2	Free	
3	Slot 3	Free	
4	Slot 4	Free	
5	Slot 5	Occupied	2:00:6:55
6	Slot 6	Free	

Figure 6. 7.Status of parking Slots

From the status of parking in Megenagna parking stations only one slot is occupied from 2:00-6.55. SO, with in the specified time drivers select free parking slots from the park stations in Addis Ababa city. The drivers can also select free parking stations to reserve his/her cars by registering car plate numbers see in figure 6.8 and customers profiles. After the car was exit from the slot, the slot was automaticity updated.

Megenagna Parking Station available slots







Date Time								
#	Slot	Slot Status	Customer Full Name	Phone	Plate Number	Reserved Time duration	Reserved Date	
1	Slot 1	reserved	Duhlay ardona	+2519080089	008912	2:10-4:45	1-29-2018 2:15	
2	Slot 2	reserved	Feyima Nami	+2519979990	089123	For a week	2-29-2018 12:50	
3	Slot 3	reserved	Shifera alka	+2519206443	009126	7:00-10:00	3-29-2018 7:01	
4	Slot 4	reserved	Fatmagul kagnew	+2519121013	0091256	1:00-12:00	4-29-2018 1:02	
5	Slot 5	reserved	Dawit Bekele	+2519942123	001245	3:15-6:45	5-29-2018 3:16	

Figure 6. 8. Sample parking reservation in Megenagna parking

The driver or user could send request for parking by comparing the distance from his/her current location, free slots and payment. Then the system after accepting the request of drivers or users was waiting 10 minutes for parking else the request was expired except the customers asked reserve in a specific period. Since the local and central server communicates continuously all the available information in each parking station was updated in the central server.

The data in central server was been managed in database management system (DBMS) software. Because DBMS helps the user or the programmer to update, retrieve, create and manage the data in a proper way, DBMS interacts with the user, other applications, and the database itself to capture and analyze data. The data in the smart parking system database was updated, created, edited, inserted, and deleted periodically in the local and central server based on the availability of the system. The smart parking system data was stored in the innoDB storage engine on XAPP local server by PHPMyadmin. The smart parking system ER database design was based on the objective and the goal of the research.



Figure 6. 9. Database Design of smart parking system

The database table was design on MySQL workbench and the relationship of the table was done based on the user interactions with the system. The web based application implementation of smart parking system contains parking stations, user account, parking customers, number of slots, payment and parking station income customer tables. Each park in the parking stations was identified using parking station ID and each park has its own location, latitude and longitude coordination, number of slots and registration date Based on latitude and longitude coordination users could easily access the location of each parking station on Google map because each information in the parking station was stored in the. If users had account in the smart parking system, they simply log in the system by their user account and password because the system identified them by their account ID. In the user account them was account ID, user name password, first name, last name and parking station ID to interact to the park stations. The parking customers

were identified by customer ID and interact with parking stations using parking station ID. The parking customers table also contains parking slot ID, car ID, phone and parking date and time for parking access or reserve parking. Each slot in the park was identified by slot ID. The slot was taking parking station ID for interaction and a slot has its own name and status about occupancy whether it was free or full. The parking customers interact with the parking station in come customers using customer ID and parking station in come customer ID. The park station income customers were payed money for parking based on waiting time. Payment of parking was described in the payment table. The amount of fee in each parking station was being identified so users could pay based on waiting time in the parking station.

6.7.1. Web Service Applications

The mobile application was connecting to a web service to pull or push data in order to update the app or provide real time smart parking system services with centralized data. For web application and mobile application, we were used PHP REST API in the backed with MySQL database. Representational State Transfer (REST) is an architecture style in the web services. Web services follow the RESTful principles were RESTful services. In RESTful services, URIs is used to access the resources. The mobile application and the web application could communicate using web service application by sending request and receiving response. In the web app implementation, we were creating a PHP RESTful service to read data from a MySQL database table. Also, I am providing an example Android project code for invoking this RESTful service by calling the PHP REST API from mobile application then in server side, the API service reads data from the database and sends the response in JSON format. After receiving the response, the mobile application displays the parking stations and locations. REST API contains the REST controller, service class, and the DAO the REST calls the service to prepare a response. Service class raises DAO function to read data from the database table. Since application requests JSON type data, the service class fixes the response in JSON format. HTTPConnectionRequest class was used to set parameters and send the request to the API.

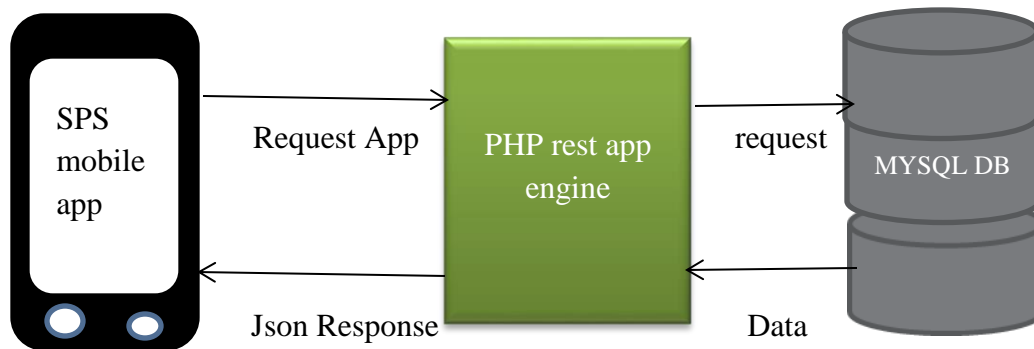


Figure 6.10. Web service application

The SPS mobile app send request application to PHP rest app engine to get response, then the reset engine also send request to MYSQL database to get data after that the data send to SPS mobile app by Json format response.

6.8. IOT Platforms of Smart Parking System

To select IoT software platform, device management, integration, security, protocols for data collection, types of analytics, and support for visualizations was considered.

Device management is one of the most important features expected from any IoT software platform. The IoT platform should maintain a list of devices connected to it and track their operation status; it should be able to handle configuration, firmware and also updates and provide device level error reporting and error handling. The API should provide access to the important operations and data that needs to be exposed from the IoT platform. It's common to use REST APIs to achieve device management.

The information security measures required to operate an IoT software platform are much higher than general software applications and services, the network connection between the IoT devices and the IoT software platform would need to be encrypted with a strong encryption mechanism to avoid potential eavesdropping. The IoT software platform itself needs to implement alternative measures to handle such device level issues. requiring regular password updates and supporting updateable firmware.

Data Collection Protocols is the types of protocols used for data communication between the components of an IoT software platform. An IoT platform may need to be scaled to millions or

even billions of devices. Lightweight communication protocols should be used to enable low energy use as well as low network bandwidth functionality.

Data Analytics is the data collected from the sensors connected to an IoT platform needs to be analyzed in an intelligent manner in order to obtain meaningful insights. The most types of analytics which can be conducted on IoT data: real-time, batch, predictive, and interactive analytics. Real-time analytics conduct online analysis of the streaming data. Batch analytics runs operations on an accumulated set of data. Thus, batch operations run at scheduled time periods and may last for several hours or days. Predictive analytics is focused on making predictions based on various statistical and machine learning techniques. Interactive analytics runs multiple exploratory analyses on both streaming and batch data. The last is real-time analytics, which weighs heavier on any IoT software platform. From the IOT software platforms the followings were the most common in the above criteria.

6.8.1. Thing Speak

Thing Speak is an open data platform for the Internet of things; Thing Speak enables real-time data collection with an Open API and sophisticated data processing and analysis capabilities. With plugins, data visualizations and the ability to integrate your data with a variety of third-party platforms, systems and technologies, including other leading IoT platforms such as IoTBridge and Arduino, key features:

- Open API
- Integrate with Raspberry Pi, Arduino, ioBridge/RealTime.io and more
- Geo-location data
- Data processing and visualizations
- Device status messages
- Social network integration
- Integrations with data analytics and third-party services
- Thing Speak API powered by Phusion Passenger Enterprise

6.8.2. Wovyn

Wovyn is an open, distributed architecture for delving into new opportunities and business models with the IoT; Wovyn is capable of connecting any sensor to any app using any protocol. Used for

personal, commercial, industrial and military applications, Wovyn is a powerful middleware platform for the Internet of Things.

6.8.3. IBM IoT

IBM IoT Foundation Device Cloud key feature:

- Device management ability
- Integration with REST and Real-time APIs
- Security Link Encryption (TLS), Authentication, Identity management
- Support MQTT, HTTPS protocol
- Support real-time analytics
- Support Web portal visualizations

6.8.4. AWS IoT

AWS IoT platform key feature

- Device management ability
- Integration with REST APIs
- Use Link Encryption (TLS), Authentication
- Support MQTT, HTTP1.1 protocol
- Used in real-time analytics
- Support visualizations

ThingSpeak platform was selected for smart parking system due to its easily Arduino integration, geolocation application, high device managements and good data analytics and ThingSpeak is an open source IOT application. ThingSpeak was used to store and retrieve data from ultrasonic sensor and Arduino using HTTP over the Internet or via a Local Area Network. It stores and retrieve SPS data from ThingSpeak. First thing speak setup was installed in Arduino IDE to programed the system. Then using ThingSpeak platform free slot and location of parking station was been Analyze and Visualize. Figure 7.11 shows how smart parking ultrasonic sensor data using Arduino Uno was access by user using mobile phone over thing speak platform.

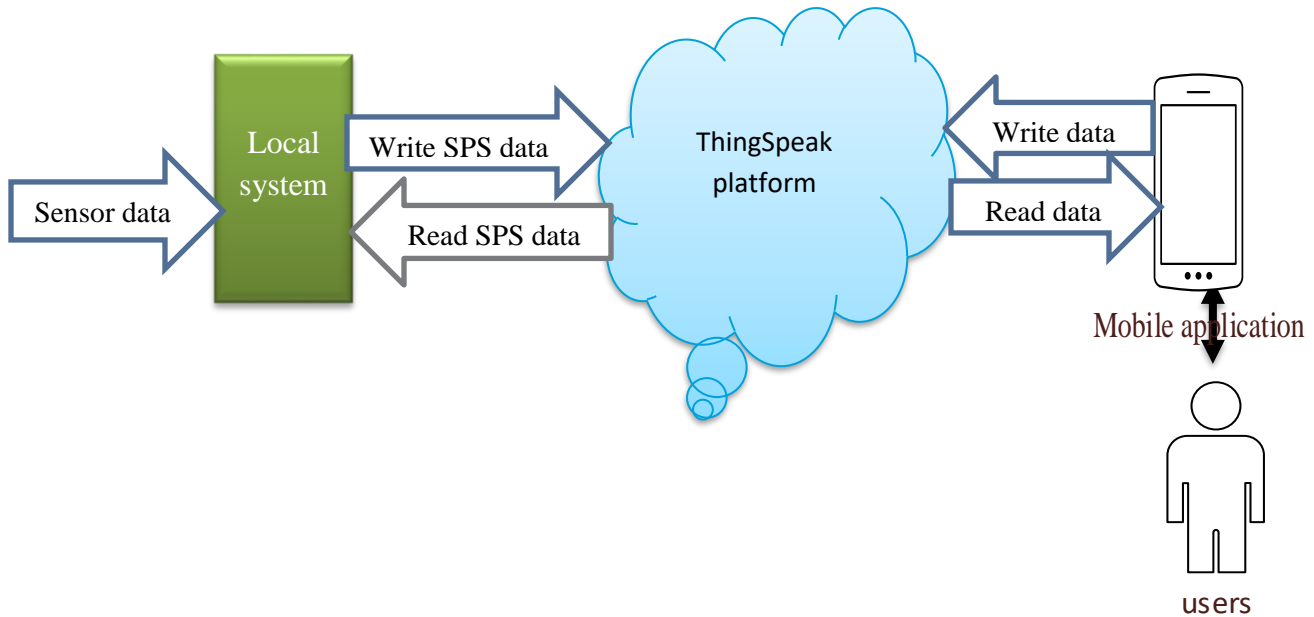


Figure 6. 11. ThingSpeak platform local system and mobile application communication

The available data from ultrasonic sensor about the status of free slot was sent to Arduino Uno then from Arduino Uno the data was writing to or read from Thing speak. The data that stored in the Thing speak was also read and write by smart mobile application users. Smart parking information can be monitored from anywhere in Addis Ababa city over the internet or EPS8266 Wi-Fi. Thing speak displays the data online so, users can access the smart parking system data from Thing speak at any time and at any place. To monitor and control smart parking system data in the Thing speak over Internet we could use Thing speak site, Channels and webpages provided by Thing speak. First the smart parking system account was created on Thingspeak.com, then Sign in and click on Get Started. After creating a for SPS account, we went to channels and we created a new channel. Then we wrote the name of the Channel and name of the Fields then we created the channel for public option lastly, we save the channel. Then the smart parking system channel was created.

CHAPTER SEVEN: EVALUATION AND RESULT ANALYSIS OF SMART PARKING SYSTEM

7.1. Architecture Evaluation of Smart Parking System (AESPS)

The architecture of smart parking system was evaluated to check the goal, the objective and the requirements of smart parking system whether that meet with the developed architecture or not. The preference based architecture of smart parking system was developed using hardware and software tools. The working functionality of hardware and software tools used in architecture was checked in implementation stage so, the architectural functionality of smart parking system was an indirectly evaluated from implementation.

7.2. Slot Identification Evaluation of Smart Parking System (SIESPS)

Evaluation of smart parking slot identification was done to check the availability of free parking space and how the system update periodically when the car enters to the parking and exit from the parking. Additionally, to evaluate the demand and the adequacy of parking area in Addis Ababa city is proportional or not. To identify the free slot, we were use three Ultrasonic sensors that install in the smart parking system, so, to evaluate the free slot we were use permutation formula.

Table 7. 1. Slot Identification Evaluation result of Smart Parking System

Slot1	checked	Slot2	checked	Slot3	Checked	Correct result in %
Free	✓	Free	✓	Free	✓	100%
Free	X	Free	✓	Full	✓	66.67%
Free	✓	Full	✓	Free	✓	100%
Free	✓	Full	✓	Full	X	66.67%
Full	✓	Full	✓	Full	✓	100%
Full	✓	Full	X	Free	✓	66.76%
Full	X	Free	X	Free	X	0%
Full	X	Free	✓	Full	✓	66.67%

When we were evaluating the free parking slots eight times we got three, two and three incorrect results in slot1, slot2 and slot3 respectively. At slot1 in the second evaluation case the car was parked, but the system showed the slot was free and also in test case seventh and eighth the cars

were not parked but the system showed the slot was full. In the same case at slot2 in evaluation case six the system showed slot2 is full, but the car was not parked and in case seven the system showed slot2 was free but the car was parked. At slot3 in evaluation case four the system showed slot3 was full, car was not parked and at case seven the system also showed slot3 was free, but car was parked. In table 7.1 the correct and the incorrect results was analysis by taking eight test cases for each three parking slots. So, the total test case was performed within the three slots was three times eight equals to twenty-four. Then the perfect working functionality of slot identification was calculated in percentage. From the eight tests the correct slot identification results was calculated using mean formula.

$$\text{Percentage of correct results} = \frac{\text{total correct test case in the three slots} * 100}{\text{total test cases in the three slots}}$$

$$\text{Percentage of correct results} = \frac{17 * 100}{24} = 70.833\%$$

The percentage of in correct result also calculated in the same way as follow.

$$\text{Percentage of incorrect results} = \frac{\text{total incorrect test case in the three slots} * 100}{\text{total test cases in the three slots}}$$

$$\text{Percentage of correct results} = \frac{7 * 100}{24} = 29.166\%$$

From the free slot identification evaluation in the eight-test cases the correct results obtained was 70.833% and 29.166% was also obtained. The incorrect results were from environmental factors of Ultrasonic sensors and object identification problems considering other means if any object was within 400 meters in maximum range in front of the parking slot the Ultrasonic sensor consider it like a car.

7.3. Location identification Evaluation of Smart Parking System (LIESPS)

The evaluation of location identification was carried out to check the working functionality of the smart parking system by analyzing the location of drivers/users and parking stations. So, to evaluate the location of users and parking stations, we were taking four random places in Addis Ababa city. These places were taking as parking station or as drivers/users' current location.

Table 7. 2. Location identification Evaluation result of Smart Parking System

Taken location	Systems showed	Checked	Accurate Percentage
Mexico1	Mexico	✓	100%
Mexico2	Mexico	✓	
Legehar1	Legehar	✓	50%
Legehar2	Mexico	X	
Aratkilo1	Aratkilo	✓	50%
Aratkilo2	Amstkilo	X	
Mekanisa1	Mekanisa	✓	100%
Mekanisa2	Mekanisa	✓	

$$\text{Percentage of accurate results} = \frac{\text{total accurate test case in the four locations} * 100}{\text{total test cases in the four locations}}$$

$$\text{Percentage of accurate results} = \frac{6 * 100}{8} = 75\%$$

$$\text{Percentage of inaccurate results} = \frac{2 * 100}{8} = 25\%$$

In the evaluation of location identification from eight test cases 75% was identified accurately but 25% was not identified accurately this may be the result of taking location coordination, as we checked the location of drivers or parking stations.

7.4. Speed Evaluation of Smart Parking System (SESPS)

The speed of smart parking system was evaluated using request and response time from local server to central server and from mobile phone to central server. The evaluation of speed of smart parking system was summarized in table 7.3.

Table 7. 3. Speed Evaluation Result of Smart Parking System

Requests and response	Request time	Response time	Waited time
Mobile to Central server	4:20:10	4:20:13:49	3:49micro second
Mobile to Central server	4:20:15	4:20:18:49	3:49micro second
Mobile to Central server	4:20:19	4:20:22:50	3:50micro second
Central Server to Mobile	4:20:20	4:20:24:05	3:05 micro second
Local server to Central server	4:20:30	4:20:33	3:00 micro second
Central Server to Local Server	4:20:42	4:20:45:20	3:2 micro second

The evaluation was done by sending request to central server from mobile to get parking services and the central server send response to mobile phone. We were check three times by sending request from mobile phone to central server. But the result we got was different that was due to mobile platforms and network availabilities. Finally, the whole system was evaluated by external user. The system was evaluated by software professionals, who work in INSA Software Company.

7.5. Test Design

To analyzed evaluation, test design was carried out by selecting random users or professionals. There were ten users selected to evaluate the smart parking systems and each user have equal chance. We were asked questions to the user or evaluators to evaluate the smart parking system architectures, slots, locations and speeds. After asking the questions, the average (ava) of their answers of the evaluators was calculated. The evaluators were checking the evaluation measures meet to the evaluated system or not. We were given the evaluation measures 1(poor), 2(good), 3(very good) and 4 (excellent).

Table 7. 4. Whole system users' evaluation result

Evaluation	Evaluation criteria	Evaluator1	Evaluator2	Evaluator3	Evaluator4	Evaluator5	ava
AESPS	goal& objectives	5	4	5	3	5	4.4
SIESPS	easily free slot accessibility	4	5	4	5	3	4.2
LIESPS	minimize cost, time & traffic problems	5	5	3	5	2	4.0
SESPS	Speed	3	5	5	2	5	4.0

7.6. Result Analysis

The smart parking system architecture, free slot identification, location identification and the whole system speed communication was analyzed based on the results of evaluators. From the evaluator's average results, the smart parking system architecture meets the objectives and goals of the system. In addition, drivers can easily identify the free slot area with the minimum time. The system minimized the traffic congestions because the drivers can access free parking location with a short period so, he/ she would not search parking area due to this the traffic problems were minimized in the evaluation of the objective of the study.

CHAPTER EIGHT: CONCLUSIONS, RECOMMENDATION AND FUTURE WORKS

8.1. Conclusions

Building smart parking system using IOT able to sense, analysis, connect and communicate over virtual and physical environment. Building smart parking system using IOT was a bridge for machine to machine (Arduino Uno), machine to environment (Ultrasonic sensor to environment) and machine to people (central server to user). Smart parking services was facilitated over the Internet communication infrastructure, using established Internet standards and technologies to make human beings behave smartly by providing applications that eliminate human labor using mobile devices. The goal of this study was to develop a reference-based architecture by analyzing parking stations infrastructures, that was to enable the selection of smart parking services from local or central server based on their request using mobile devices. The IoT, based smart parking system was been analyzed and integrated to achieve the goal of smart parking system services. These services were stored and published on a ThingSpeak platform. The SPS services consists of service consumer, service broker and service provider components. These services were adapted to the smart parking system. The proposed smart parking system architecture was developed, implemented and evaluated.

When we build IOT based smart parking system how to fetch data from parking environment, how to send data from Arduino Uno to local server, how to find the key components of architecture that could use for integrated IoT service selection and how to each local server communicate continuously with central server was big challenges. In the smart-parking system how to Identifying, integrating and selecting an IoT service is heterogeneity challenge to support dynamic service composition. The solution towards addressing the challenges send data from environment to local server a literature revealed that not much work has been done in design tools, simulating, identify IOT architectures, local and central server communications. IoT system considerations were identified and modelled by developing reference architecture flexible service applications. The functionality and the performance of the smart parking architecture, slot status and location of parking was evaluated, tested and analyzed by using parking services and also smart parking

system minimize traffic congestions because many drivers will not need to search free parking stations to park their cars. The main challenges and problems of parking stations including environmental influences were analyzed and solved. Any persons in Addis Ababa city were beneficiary directly or indirectly by smart parking systems because it minimizes air pollutions, traffic accidents, traffic congestions and drivers can access parking services in real-time.

8.2. Recommendation

However, in this research we tried to build smart parking system to manage parking problems for users by easily access parking locations, distances and free parking lots using mobile phone. The parking owners can also use smart parking services without developing web application for them by share a single application. To increase user's satisfaction a parking stations service requests also integrated and delivered with a short period of time and the performance and the quality of parking infrastructures were checked. Users could send parking service request (mobile based service request) with their phone via one application instead of developing several independent apps by each organization or service providers.

8.3. Future work

In the smart parking system, the location and slot were deified using GPS tracker and Ultrasonic sensor respectively. But especially ultrasonic sensor is affected by environment factors like rain, temperature, winds and so on and it recognize any object like a car this makes ultrasonic is not perfect detection of cars in the parking stations. For more perfect detection of parking slots and security of smart parking system in the future work CCTV camera based audio or voice parking services availability system will built. The Audio or voice based parking services availability system is beneficial for poor English language persons and minimize wastage of searching smart parking services.

References

- [1] Andrew Whitmore, Anurag Agarwal, and Li Da Xu., " The internet of things a survey of topics and trends.," Information Systems Frontiers, pp. 1-14, 2014.
- [2] Ms. Yogita Pundir, Ms. Nancy Sharma, Dr. Yaduvir Singh “, "Internet of Things (IoT) : Challenges and Future Directions," International Journal of Advanced Research in Computer and Communication Engineering, vol. 5, no. 3, 2016.
- [3] Arko Djajadi, Member. , "Ambient Environment quality monitoring Using IoT Sensor Network," Interworking Indonesia Journal, vol. 8, 2016.
- [4] Tschofenig, H., et. al., " Architectural Considerations in Smart Object Networking," Internet Architecture Board, , 2015.
- [5] Jyotiranjana Hota, Pritish Kumar Sinha , "Scope and challenges of Internet of Things," An Emerging Technological Innovation, 2015.
- [6] Zeinab Kamal Aldein Mohammed, Elmustafa Sayed Ali Ahmed , "Internet of Things Applications, Challenges and Related Future Technologies," WSN, vol. 67(2) , no. 2392-2192, pp. 126-148, 2017.
- [7] Indira Muhic, Migdat I. Hodzic, " Internet of Things: Current Technological Review," vol. 2, 2014.
- [8] Hancke, G.; Silva, B.; Hancke, G., Jr, "The Role of Advanced Sensing in Smart Cities.," vol. 13, p. 393–425., 2012.
- [9] R. C. a. C. B. Manoop Talasila, "Mobile Crowd Sensing;./Chapter4.pdf," <https://web.njit.edu/~mt57/publications>, New Jersey Institute of Technology, 2013.
- [10] P. Rawat, K. Singh, H. Chaouchi and J. Bonnin, "Wireless sensor networks: A survey on recent developments and potential synergies," . J. Supercomput., vol. 68, pp. 1-48, 2014.
- [11] Thanh NamPham, Ming-Fong Tsai1,Der-Jiunn Deng, "A CloudBased Smart-Parking System Based on Internet-of-Things Technologies," IEEE, pp. 2169-3536, 2015.

- [12] Saed Tarapiah, Shadi Atalla, Rajaa AbuHania , "Smart On-Board Transportation Management System Using GPS/GSM/GPRS technologies to Reduce Traffic Violation in Developing Countries," "International Journal of Digital Information and Wireless Communications 2, 2013.
- [13] Alamri, A.; Ansari, W.S.; Hassan, M.M.; Hossain, M.S.; Alelaiwi, A.; Hossain, M.A. A. Int. J. Distrib. Sens. Netw. , " Survey on Sensor-Cloud: Architecture, Applications, and Approaches," vol. 9, 2013.
- [14] Friedemann Mattern and Christian Floerkemeier, " From the Internet of Computers to the Internet of Things," Internet-of-things, 2016.
- [15] S.Pandikumar, R.S. Vetrivel, "Internet of Things Based Architecture of Web and Smart Home Interface Using GSM," International Conference on Innovations in Engineering and Technology, vol. 3, no. 3, 2014.
- [16] Wang Rui, Wang Jinguo, Wang Na, "Analysis of key technologies in the Internet of things," 3rd International Conference on Material, Mechanical and Manufacturing Engineering, 2015.
- [17] Nicola Piovesan, Leo Turi , Enrico Toigo, Borja Martinez and Michele Rossi , "Data Analytics for Smart Parking Applications.," 2016.
- [18] A. Gupta, V. Sharma, N. Ruparam, S. Jain, A. Alhammad and M. Ripon, " Integrating pervasive computing, InfoStations and swarm intelligence to design intelligent context-aware parking-space location mechanism.," In Proceedings of the International Conference, 2014.
- [19] He, W.; Yan, G.; Xu, L.D. , "Developing vehicular data cloud services in the IoT environment," IEEE Trans. Ind. Inform, vol. 19, p. 1587–1595, 2014.
- [20] Vlahogiannia, E.I.; Kepaptsogloua, K.; Tsetsosa, V.; Karlaftisa, M.G. , "A real-time parking prediction system for smart cities," J. Intell. Transp. Syst. Technol. Plan, vol. 20, p. 192–204., 2016.
- [21] Martinez, B.; Vilajosana, X.; Vilajosana, I.; Dohler, M. L, "lean sensing: Exploiting contextual information for most energy-efficient sensing," IEEE Trans. Ind. Inform, vol. 11, p. 1156–1165, 2016.

- [22] Prof.S.S, Thorat et.al, "IoT Based Smart Parking System Using RFID," International Journal Of Computer Engineering, vol. 4(1), no. Research Trends, pp. 9-12, 2017.
- [23] Rahul G. Chandore, Akshay P. Puneekar , Swati V. Suryawanshi ,Prof. P. S. Nawghare, "IoT Based Smart Parking System," International Journal of General Science and Engineering Research, vol. 3(1), no. 2455-510X,, pp. 17-19., 2017,.
- [24] Huang, J.D.; Lee, C.K.; Yeh, C.S.; Wu, W.J.; Lin, C.T. e ., "High-Precision Ultrasonic Ranging System Platform Based on Peak-Detected Self-Interference Techniqu," IEEE Trans. Instrum, vol. 60, p. 3775–3780, 2011.
- [25] Wan, D.Z.; Cheng, S.C.r., " Simulation and prototype testing of a low-cost ultrasonic distance measurement device in underwate," J. Mar. Sci. Technol, vol. 20, p. 142–154., 2014.
- [26] Group, Science Research, "Intelligent Parking Management System Based on Image Processing.," World Journal of Engineering and Technology, vol. 2, pp. 55-67, 2014.
- [27] Antoine Bagula, "on the design of smart parking networks in the smart cities: An optimal sensor Placement Model," 2015.
- [28] Jesús Martínez Abascal , "Open Technologies for Prototyping the Internet of Things," 2013.
- [29] Luis, G.D.J.; Antonio, M.F.; Jordi, L.; Andrés, C.; Blanco-Roldán, G.L.; Antonio, R.L. , "Testing accuracy of long-range ultrasonic sensors for olive tree canopy measurements," Sensors, vol. 15, p. 2902–2919, 2015.
- [30] Aswathy Natesh, Sudhi Sudharman, "An Intelligent Approach in Parking System for Car Parking Guidance and Damage Notification based on GPS," international journal for advance research in engineering and technology, vol. 3, no. III, pp. 2320-68, 2015.
- [31] Baharuddin Mustapha, Aladin Zayegh, Rezaul K. Begg , "Ultrasonic and Infrared Sensors Performance in a Wireless Obstacle Detection System," International Conference on Artificial Intelligence, Modelling & Simulation , 2013.
- [32] Luis, G.D.J.; Antonio, M.F.; Jordi, L.; Andrés, C.; Blanco-Roldán, G.L.; Antonio, R.L. , "Testing accuracy of long-range ultrasonic sensors for olive tree canopy measurements," vol. 15, p. 2902–2919, 2015.

- [33] Xu, G. , GPS: Theory, Algorithms and Applications (2nd Edition), Germany: Springer, 2007.
- [34] Al-Kharusi, H., "Intelligent Car Parking Management System.," 2014.
- [35] Xu Zhiqing, He Jialianga, and Li Haidong, "IoT-based Intelligent Parking Management System and Its Design," International Journal of Smart Home, vol. 10(5), pp. 217-230, 2016.
- [36] "The internet of things challenges and opportunities," 21 march 2016. [Online]. Available: <http://sandhill.com/article/the-internet-of-things-challenges-andopportunities/>.
- [37] Vrushali D. Ichake., Priya D. Shitole. and Mohsina Momin. Kanchan S. Thakare, "smart parking system based on IOT concepts," ijesi, vol. 3, no. 3 march, p. 54, 2016.
- [38] Saranya C. M., Nitha K. P., "Analysis of Security methods in Internet of Things," International Journal on Recent and Innovation Trends in Computing and Communication, vol. 3, no. 4, 2015.
- [39] Abdul Ahad, Zishan Raza Khan, Syed Aqeel Ahmad, "Intelligent Parking System," World Journal of Engineering and Technology, pp. 160-167, 2016.
- [40] PAUL FREMANTLE, "A REFERENCE ARCHITECTURE of Internet of Things," WSO2 White Papers and WSO2 Case Studies., p. 16, 20 OCTOBER 2015.
- [41] Rakesh Mahato, "Review of Cross-Platforms for Mobile Learning Application Development," International Journal of Advanced Computer Science and Applications, vol. Vol. 6, 2015.

Appendixes

Appendix1. Interview questions

Appendix1.1. Interview question for Ministry of Transport

1. What is the vision of Ministry of transport specially to reduce traffic congestion and special services for the public and the customer?
2. What is the cause for traffic congestion in Addis Ababa city?
3. Do you think that road side parking may be a cause to traffic congestion?
4. Do you hear about smart parking?
5. Is any smart parking in Addis Ababa city?
6. Is smart parking minimizing these problems?
7. Do you have a car? If yes where you park it specially in the working days?

Appendix 1.3. Interview Question for Traffic police

1. Is any traffic congestion in Addis Ababa? Is it high or low? At what time it become high?
2. Do you know the cause of traffic congestion?
3. Do you think traffic congestion leads traffic accident?
4. Is any roadside parking in the city? Do you think this is a cause for traffic problems?

Appendix 1.4. Interview question for Drivers

1. where you park your car?
2. Is any parking area in Addis Ababa city?
3. How you get parking?
4. How you pay for parking?

Appendix 1.5. Interview Question for Students

1. Are you late from school due to traffic congestion?
2. Do you have a car? Where you park it? Specially at the weekend?
3. What types of parking do you know?
4. Had you lost your friend due to traffic accident?

Appendix 2. Location Parking Station Sample Code

```
#include<LiquidCrystal.h>
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
#include <SoftwareSerial.h>
SoftwareSerial gps(10,11); // RX, TX
//String str="";
char str[70];
String gpsString="";
Char *test="$GPGGA";
String latitude="No Range ";
String longitude="No Range ";
int area=0,k;
boolean gps position=0;
void setup()
{
  lcd.begin(16,2);
  Serial.begin(9600);
  gps.begin(9600);
  lcd.print("parking location");
  lcd.setCursor(0,1);
  lcd.print(" System ");
  delay(2000);
  gsm_init();
  lcd.clear();

  Serial.println("AT+CNMI=2,2,0,0,0"
);
  lcd.print("GPS Preparing");
  lcd.setCursor(0,1);
  lcd.print(" out of GPS Range ");
  get_gps();
  delay(2000);
  lcd.clear();
  lcd.print("GPS Range got");
  lcd.setCursor(0,1);
  lcd.print("GPS is Prepared");
  delay(2000);
  lcd.clear();
  lcd.print("System Prepared");
  area=0;
}
void loop()
{
  serialEvent();
  if(area)
  {
    get_gps();
    location();
  }
}
void serialEvent()
{
  while(Serial.available())
  {
    if(Serial.find("location parking"))
    {
      area=1;
      break;
    }
  }
  else
  area=0;
}
```

```

    }
}
void gpsEvent()
{
    gpsString="";
    while(1)
    {
        while      (gps.available(>0)
//checking serial data from GPS
    {
        char inChar = (char)gps.read();
        gpsString+=      inChar;
//store data from GPS into gpsString
        k++;
        if (k < 7)
        {
            if(gpsString[k-1] != test[k-1])
//checking for $GPGGA sentence
            {
                k=0;
                gpsString="";
            }
        }
        if(inChar=='\r')
        {
            if(k>65)
            {
                gps_position=1;
                break;
            }
            else
        {
            k=0;
        }
    }
}
void gsm_init()
{
    lcd.clear();
    lcd.print("Finding Module..");
    boolean at_flag=1;
    while(at_flag)
    {
        Serial.println("AT");
        while(Serial.available(>0)
        {
            if(Serial.find("OK"))
                at_flag=0;
        }
        delay(1000);
    }
    lcd.clear();
    lcd.print("Module Connected..");
    delay(1000);
    lcd.clear();
    lcd.print("Disabling ECHO");
    boolean echo_flag=1;

```

```

while(echo_flag)
{
  Serial.println("ATE0");
  while(Serial.available()>0)
  {
    if(Serial.find("OK"))
      echo_flag=0;
  }
  delay(1000);
}
lcd.clear();
lcd.print("Echo OFF");
delay(1000);
lcd.clear();
lcd.print("Finding Network..");
boolean net_flag=1;
while(net_flag)
{
  Serial.println("AT+CPIN?");
  while(Serial.available()>0)
  {
    if(Serial.find("+CPIN:
READY"))
      net_flag=0;
  }
  delay(1000);
}
lcd.clear();
lcd.print("Network Found..");
delay(1000);
lcd.clear();
}

void get_gps()
{
  gps_position=0;
  int x=0;
  while(gps_position==0)
  {
    gpsEvent();
    int str_lenth=k;
    latitude="";
    longitude="";
    int comma=0;
    while(t<str_lenth)
    {
      if(gpsString[t]=='')
        comma++;
      if(comma==2) //extract
        latitude from string
        latitude+=gpsString[t+1];
      else if(comma==4) //extract
        longitude from string
        longitude+=gpsString[t+1];
      t++;
    }
    int l1=latitude.length();
    latitude[l1-1]=' ';
    l1=longitude.length();
    longitude[l1-1]=' ';
    lcd.clear();
    lcd.print("Lat:");
    lcd.print(latitude);
  }
}

```

```

    lcd.setCursor(0,1);
    lcd.print("Long:");
    lcd.print(longitude);
    k=0;t=0;
    str_lenth=0;
    delay(2000);
}
}
void init_sms()
{
    Serial.println("AT+CMGF=1");
    delay(400);

    Serial.println("AT+CMGS=\"+91**
*****\""); // use your 10 digit
cell no. here
    delay(400);
}
void send_data(String message)
{
    Serial.println(message);
    delay(200);
}
void send_sms()
{
    Serial.write(26);
}

}
void lcd_status()
{
    lcd.clear();
    lcd.print("Message Sent");
    delay(2000);
    lcd.clear();
    lcd.print("System Prepared");
    return;
}
void tracking()
{
    init_sms();
    send_data("parking location :");
    send_data(" area of parking
Location is:");
    Serial.print("Latitude:");
    send_data(latitude);
    Serial.print("Longitude:");
    send_data(longitude);
    send_data("you can park here
please");
    send_sms();
    delay(2000);
    lcd_position();
}

```

Appendix 3. Arduino and MYSQL connection sample code

```

#include <Ethernet.h>
#include <MySQL_Connection.h>
#include <Dns.h>

```

```

byte mac_addr[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
char hostname[] = "localhost"; // change to your server's hostname/URL
char user[] = "root";          // MySQL user login username
char password[] = "";         // MySQL user login password
IPAddress server_ip;
EthernetClient client;
MySQL_Connection conn((Client *)&client);
DNSClient dns_client; // DNS instance
void setup() {
  Serial.begin(1000);
  while (!Serial); // wait for serial port to connect
  Ethernet.begin(mac_addr);
  // Begin DNS lookup
  dns_client.begin(Ethernet.dnsServerIP());
  dns_client.getHostByName(hostname, server_ip);
  Serial.println(server_ip);
  // End DNS lookup
  Serial.println("Connecting...");
  if (conn.connect(127.0.0.1, 3306, root, "")) {
    delay(1000);
    // You would add your code here to run a query once on startup.
  }
  else
    Serial.println("Connection failed.");
  conn.close();
}

void loop() {
}

```

Appendix 4. Arduino Sample Code to Select Data from Database

```
#include <Ethernet.h>
#include <MySQL_Connection.h>
#include <MySQL_Cursor.h>
byte mac_addr[] = { 0xDE, 0xAD,
0xBE, 0xEF, 0xFE, 0xED };
IPAddress server_addr(127,0,0,1); //
IP of the MySQL *server* here
char user[] = "root"; // MySQL
user login username
char password[] = ""; // MySQL
user login password
// Sample query
char query[] = "SELECT *
ArduinoData";
EthernetClient client;
MySQL_Connection conn((Client
*)&client);
void setup() {
  Serial.begin(2000);
  while (!Serial); // wait for serial port
to connect
  Ethernet.begin(mac_addr);
  Serial.println("Connecting...");
  if (conn.connect(localhost, 3306,
root, "")) {
    delay(1000);
  }
  else
    Serial.println("Connection
failed.");
}
}
void loop() {
  delay(2000);
  Serial.println("\nRunning SELECT
and printing results\n");
  // Initiate the query class instance
  MySQL_Cursor *slot_num = new
MySQL_Cursor(&conn);
  // Execute the query
  slot_num->execute(query);
  // Fetch the columns and print them
  col_names *cols = slot_num-
>get_col();
  for (int s = 0; ss < cols->num_fields;
s++) {
    Serial.print(cols->fields[s]-
>name);
    if (f < cols->num_fields-1) {
      Serial.print(", ");
    }
  }
  Serial.println();
  // Read the rows and print them
  row_values *row = NULL;
  do {
    row = slot_num->get_next_row();
    if (row != NULL) {
      for (int s = 0; s < cols-
>num_fields; s++) {
        Serial.print(row->values[s]);
      }
    }
  }
}
```

```

    if (s < cols->num_fields-1) {
        Serial.print(", ");
    }
}
Serial.println();
}

} while (row != NULL);
// Deleting the cursor also frees up
memory used
delete slot_num;
}

```

Appendix 5. MYSQL and Arduino sample update slot status web client code

```

#include <SPI.h>
#include <Ethernet.h>
#include <Ultrasonic.h>
int echo1=2, int trig1=3, int echo2=4,
int trig2=5, int echo3=6 trig3=7;
long duration, distance, slot1, slot2,
slot3;
// Enter a MAC address for your
controller below.
// Newer Ethernet shields have a
MAC address printed on a sticker on
the shield
byte mac[] = { 0xDE, 0xAD, 0xBE,
0xEF, 0xFE, 0xED };
// if you don't want to use DNS (and
reduce your sketch size)
// use the numeric IP instead of the
name for the server:
IPAddress server(192,168,0,101); //
numeric IP for Google (no DNS)
// Set the static IP address to use if the
DHCP fails to assign
IPAddress ip(192, 168, 0, 177);
// Initialize the Ethernet client library
// with the IP address and port of the
server
// that you want to connect to (port 80
is default for HTTP):
EthernetClient client;
void setup() {
    // Open serial communications and
wait for port to open:
    Serial.begin(9600);
    while (!Serial) {
        ; // wait for serial port to connect.
        Needed for native USB port only
    }

    // start the Ethernet connection:
    if (Ethernet.begin(mac) == 0) {
        Serial.println("Failed to configure
Ethernet using DHCP");
        // try to configure using IP address
instead of DHCP:
        Ethernet.begin(mac, ip);
    }
    // give the Ethernet shield a second
to initialize:

```

```

delay(1000);
Serial.println("connecting...");
pinMode(trig1,
OUTPUT);pinMode(echo1,INPUT);
pinMode(trig2,
OUTPUT);pinMode(echo2,INPUT);
pinMode(trig3,
OUTPUT);pinMode(echo3,INPUT);
}
void slot(int trig, int echo){
digitalWrite(trig, LOW);
delay(10)
digitalWrite(trig, HIGH);
delay(10);
digitalWrite(trig, LOW);
duration=PulseIN(echo, HIGH);
distance=((duration/2)*0.034);
}
}
void loop() {
slot(trig1, echo1); slot1=distance;
//Serial.println(slot1);
slot(trig2, echo2); slot2=distance;
//Serial.println(slot2);
slot(trig3, echo3); slot3=distance;
//Serial.println(slot3);
// if you get a connection, report
back via serial:
if (client.connect(server, 80)) {
Serial.println("connected");
// Make a HTTP request:

```

```

client.println("GET
/BuildSparkingSystem/Arduinoslotd
ata/Pslotdata.php HTTP/1.1");
if(slot1>400)
{
Serial.println("slot1 is free");
Else
{
Serial.println("slot1 is full");
}
if(slot2>400)
{
Serial.println("slot2 is free");
}
else
{
Serial.println("slot2 is full");
}
if(slot3>400)
{
Serial.println("slot3 is free");
}
else
{
client.println("there is aproplem");
client.println("Connection:
close");
client.println();
} else {
// if you didn't get a connection to
the server:

```

```

Serial.println("connection failed");           while (true);
}                                               }
client.stop();                                 }

```

Appendix 5.1. Sample php code to update slot status

```

<?php
require_once 'ardudb.php';
$slot= mysqli_real_escape_string($_GET("slot"));
$sql="update slot SET slot='$slot' WHERE slotID=1||2||3";
$sql_query= mysqli_query($sql);
if ($sql_query)
{
    echo 'complet';
}
else
{
    echo 'error';
}
?>

```

Appendix 6. Parking station java script sample code

```

function AccessParkingStations() {
    var ParkingStations = [];

    var position = {
        "Latitude": Latitude,
        "Longitude": Longitude,
        "Altitude": 0,
        "Accuracy": 0,
        "countMore": 0
    };

    var server = Web_Service+ "/buildSparking/getparkingstationsbylocation"
    $.post(server, position, function (result) {

```

```

    console.log(result);
    var resultsRow = {};
    for (var i = 0; i < result.length; i++) {
        resultsRow = result[i];
        var rowrepresentation = "<p> <a      onclick='CustomerParking(); '> " +
resultsRow.Name + '(' + resultsRow.DistanceKM + " KM) </a></p>"
        $("#requestResult").append(rowrepresentation);
    }
    $("#span").html(result);
});
}

```

```

function CustomerParking() {
    var htmlString = "<div class='form-group'> <input type='text' class='form-control'
name='carModel'/>" +
        "</div> <div class='form-group'>          <input type='text' class='form-control'
name='Phone' value='+251'/>" +
        "</div> <div class='form-group'>          <input type='text' class='form-control'
name='carModel2'/>" +
        " </div> <div class='form-group'>          <button class='btn btn-primary'
id='saveRequest' onclick='SaveParkRequest();> Save</button> </div>"
    $("#requestResult").html(htmlString);
    return false;
}

```

```

function SaveParkRequest() {
    var Parkmodel = {
        "ID": 0,
        "SenderID": 0,
        "RecievedDate": "2017-12-24T06:44:55.204Z",
        "Message": "string"
    }
}

```

```

};
$.post(Web_Service, // url
      Parkmodel, // data to be submit
      function (data, status, jqXHR) { // success callback
        $('p').append('status: ' + status + ', data: ' + data);
      })
alert("welcome to our parking system");
}
app.initialize();

```

Appendix 7. Admin login and signup page sample code

```

<?php
//start session
session_start();
//load and initialize user class
include 'user.php';
$user = new User();
if(isset($_POST['signupSubmit'])){
//check whether user details are
empty
if(!empty($_POST['first_name']) &&
!empty($_POST['last_name']) &&
!empty($_POST['email']) &&
!empty($_POST['phone']) &&
!empty($_POST['password']) &&
!empty($_POST['confirm_password']
)){
//password and confirm password
comparison
if($_POST['password'] !==
$_POST['confirm_password']){
$sessionData['status']['type'] = 'error';
$sessionData['status']['msg'] = 'Confirm
password must match with the
password.';
}else{
//check whether user exists in the
database
$prevCon['where'] =
array('email'=>$_POST['email']);
$prevCon['return_type'] = 'count';
$prevUser = $user-
>getRows($prevCon);
if($prevUser > 0){
$sessionData['status']['type'] = 'error';
$sessionData['status']['msg'] = 'Email
already exists, please use another
email.';
}else{
//insert user data in the database
$userData = array(
'first_name' =>
$_POST['first_name'],

```

```

    'last_name' => $_POST['last_name'],
    'email' => $_POST['email'],
    'password' =>
    md5($_POST['password']),
    'phone' => $_POST['phone']
    );
    $insert = $user->insert($userData);
    //set status based on data insert
    if($insert){
        $sessData['status']['type'] = 'success';

    $sessData['status']['msg'] = 'You have
    registered successfully, log in with your
    credentials.';

        }else{
            $sessData['status']['type'] = 'error';
            $sessData['status']['msg'] = 'Some
            problem occurred, please try again.';
        }
    }
    }else{
        $sessData['status']['type'] = 'error';
        $sessData['status']['msg'] = 'All fields
        are mandatory, please fill all the
        fields.';
    }
    //store signup status into the session
    $_SESSION['sessData'] = $sessData;
    $redirectURL =
    ($sessData['status']['type'] ==

```

```

'success')?'index.php':'registration.ph
p';
//redirect to the home/registration
page
header("Location:".$redirectURL);
}elseif(isset($_POST['loginSubmit']))
){
//check whether login details are
empty
if(!empty($_POST['email']) &&
!empty($_POST['password'])){
//get user data from user class
$conditions['where'] = array(
'email' => $_POST['email'],
'password' =>
md5($_POST['password']),
'status' => '1'
);
$conditions['return_type'] = 'single';
$userData = $user-
>getRows($conditions);
//set user data and status based on
login credentials
if($userData){
    $sessData['userLoggedIn'] = TRUE;
    $sessData['userID'] = $userData['id'];
    $sessData['status']['type'] = 'success';
    $sessData['status']['msg'] =
    'Welcome
    '.$userData['first_name'].'!';
}else{

```

```

$sessData['status']['type'] = 'error';
$sessData['status']['msg'] = 'Wrong
email or password, please try again.';
}
}else{
$sessData['status']['type'] = 'error';
$sessData['status']['msg'] = 'Enter
email and password.';
}
//store login status into the session
$_SESSION['sessData'] = $sessData;
//redirect to the home page
header("Location:index.php");
}elseif(!empty($_REQUEST['logout
Submit']))){
//remove session data
unset($_SESSION['sessData']);
session_destroy();
//store logout status into the session
$sessData['status']['type'] = 'success';
$sessData['status']['msg'] = 'You
have logout successfully';
$_SESSION['sessData'] = $sessData;
//redirect to the home page
header("Location:index.php");
}else{ //redirect to the home page
header("Location:index.php");}

```