

MANAGEMENT OF CAR PARKING SYSTEM USING WIRELESS SENSOR NETWORK

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Electronics and Communication Engineering.

by

Akhila Gandrotula (37130134),

Harshini Gannamani (37130139).



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI - 600119, TAMILNADU

APRIL-2021



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **AKHILA GANDROTULA (Reg. no. 37130134)** and **HARSHINI GANNAMANI (Reg. no. 37130139)** who carried out the project entitled **"MANAGEMENT OF CAR PARKING SYSTEM USING WIRELESS SENSOR NETWORK"** under my supervision from December 2020 to April 2021.

Internal Guide

Mrs. R. Sakthi Prabha, M.E., (Ph.D)

Head of the Department

Dr. T. Ravi, M.E., Ph.D.

Submitted for Vivavoce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

We, **AKHILA GANDROTULA** (Reg. No. 37130134) and **HARSHINI GANNAMANI** (Reg. No.37130139) hereby declare that the Project Report entitled “**MANAGEMENT OF CAR PARKING SYSTEM USING WIRELESS SENSOR NETWORK**” done by us under the guidance of **Ms. R. Sakthi Prabha, M.E., (Ph.D.)**, Assistant Professor, Department of Electronics and Communication Engineering, at SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY, CHENNAI is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in ELECTRONICS AND COMMUNICATION.

1.

Gr. Akhila

Gr. Harshini

DATE: 17-04-2021

2.

PLACE:CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. N.M. Nandhitha, M.E., Ph.D.**, Dean, School of Electrical and Electronics Engineering, and **Dr. T. RAVI, M.E., Ph.D.**, Head of the Department, Dept. of Electronics and Communication for providing us necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to our Project Guide **Mrs. R. Sakthi Prabha, M.E.(Ph.D.)**, Assistant Professor, Department of Electronics and Communication Engineering, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication who were helpful in many ways for the completion of the project.

ABSTRACT

In recent trends traffic has become a major problem in most of the cities. There are limited car parking facilities and less road safety precautions observed. So, in order to address the traffic problems are being made in the field of IoT. This paper will present the basic idea of cloud based parking system. Cloud will store and process the data between device and mobile. In cloud, stored files will be obtained from anywhere through the network connection. By using the latest technology called IoT for smart cities, we can book the parking slot of a vehicle from anywhere. We need to take registration details for reserving the parking slot and for security purpose we are generating an OTP. By using the OTP we can book our slot through an android application. The main aim of this work was to design and develop a smart parking system using software solution with sensors linked to the via IOT technologies, which gives a solution to reduce traffic congestions and related concerns due to them. As a proof-of concept of this proposed method, a prototype will be built with Android and Python to demonstrate the proposed system. Additionally, an python application using PYTHON IDLE will be developed for the users to facilitate them with bookings, editing, viewing status of car parks, etc. The application works on the smart slot suggestion mechanism and the system incorporates smart features such as entry/exit.

CONTENTS

CHAPTER NO	TITLE	PAGE
1	Introduction	1
2	Literature Survey	12
3	Methodology	17
	3.1 Existing Methodology	17
	3.2 Proposed System	18
	3.2.1 Implementation	19
	3.2.2 Managing Locations	20
	3.2.3 Application Framework	20
	3.2.4 Libraries	21
	3.2.5 Android runtime	22
	3.2.6 System Requirements	23
	3.2.7 Version History	27
4	Software Environment	40
	4.1 Android Architecture	40
5	Simulation results	43
6	6.1 Conclusion	48
	6,2 Future Scope	49
	References	50

LIST OF ABBREVIATION

SDK	Software Development Kit
USB	Universal Serial Bus
IoT	Internet of things
API	Application Programming Interface
AVD	Android Virtual Device
Os	Operating system
HTML	Hypertext Markup Language
ADB	Android Debug Bridge
DDMS	Dalvik Debug Monitor Service
VGA	Video Graphics Array
GSM	Global System for Mobile Communications
EDGE	Enhanced Data rates for GSM Evolution
IDEN	Integrated Digital Enhanced Network
CDMA	Code Division Multiple Access
EVDO	Evolution-Data Optimized
UMTS	Universal Mobile Telecommunications System
WiMAX	Worldwide Interoperability for Microwave Access
C2DM	Android Cloud to Device Messaging
SMS	Short Message Service
MMS	Multimedia Messaging Service
CPU	Central Processing Unit
JPEG	Joint Photographic Experts Group
PNG	Portable Network GraphicsGIF
GIF	The Graphics Interchange Format
HE-AAC	High-Efficiency Advanced Audio Coding
AMR	Antimicrobial resistance
MIDI	Musical Instrument Digital Interface
ISMA	Inhibit Sense Multiple Access
RTMP	Real Time Messaging Protoco

HTTP	Hypertext Transfer Protocol
RTSP	Real Time Streaming Protocol
HID	Human Interface Devices
HDD	Hard Disk Drive
NTFS	New Technology File System

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
1.1	Management of parking system	2
3.1.1	Board showing available parking slots	17
3.2.1	Block diagram for proposed system	18
5.1	Admin add parking slot page	43
5.2	Admin home page	43
5.3	Slot booking page	44
5.4	Login page	45
5.5	Welcome page	45
5.6	Manage your locations	46
5.7	Registration page for new user	46
5.8	Text view	47

CHAPTER - 1

INTRODUCTION

Smart Parking sensing technology is leading the way in the delivery of proven, fully integrated, end-to-end solutions, resulting in a transformation of the complete parking experience. Generally SmartPark system is made up of interchangeable components and fully integrates parking, guidance, payment and analytics as well as a host of other complementary services and options. This makes the complete SmartPark solution truly a sum of its parts, and an industry leader in the technology and parking sector. As the population multiplied within the metropolitan cities, the usage of vehicles got multiplied. Finding a automobile parking zone in most metropolitan areas, particularly throughout the push hours, is troublesome for drivers. Indisciplinary parking could lead to injury to the automobile. SmartCloud is Smart Parking's real-time, global scale, intelligent IoT services platform that enables sophisticated and flexible services to be created using open web interfaces. So there's a requirement to provide comfortable parking areas let alone lots of slots to assist the user park his automobile safely. Essentially parking system is one in every of the foremost adopted and quickest growing answer of sensible town. Currently, most of the present automobile parks don't have a scientific system. Most of them square measure manually managed and a bit inefficient. Each user's demands ought to be i. Users friendly ii. Ought to be additional economical iii. They must give additional security. The thought behind our humanoid Application-“validspot” is to assist the user for on-line parking booking. During this application user will read numerous parking areas conjointly he will choose it to look at whether or not parking slot is accessible or not. If the parking slot is accessible in parking then user will book it for a few specific time interval conjointly, this technique provides a further feature of cancelling the bookings. It conjointly utilize the open ground for parking with security. So, it'll solve the parking and traffic drawback. During this there's no got to use extra expensive camera and scanner device for verification purpose. During this system Registration certificate (R.C) book is use for verification purpose, thus it cut back the

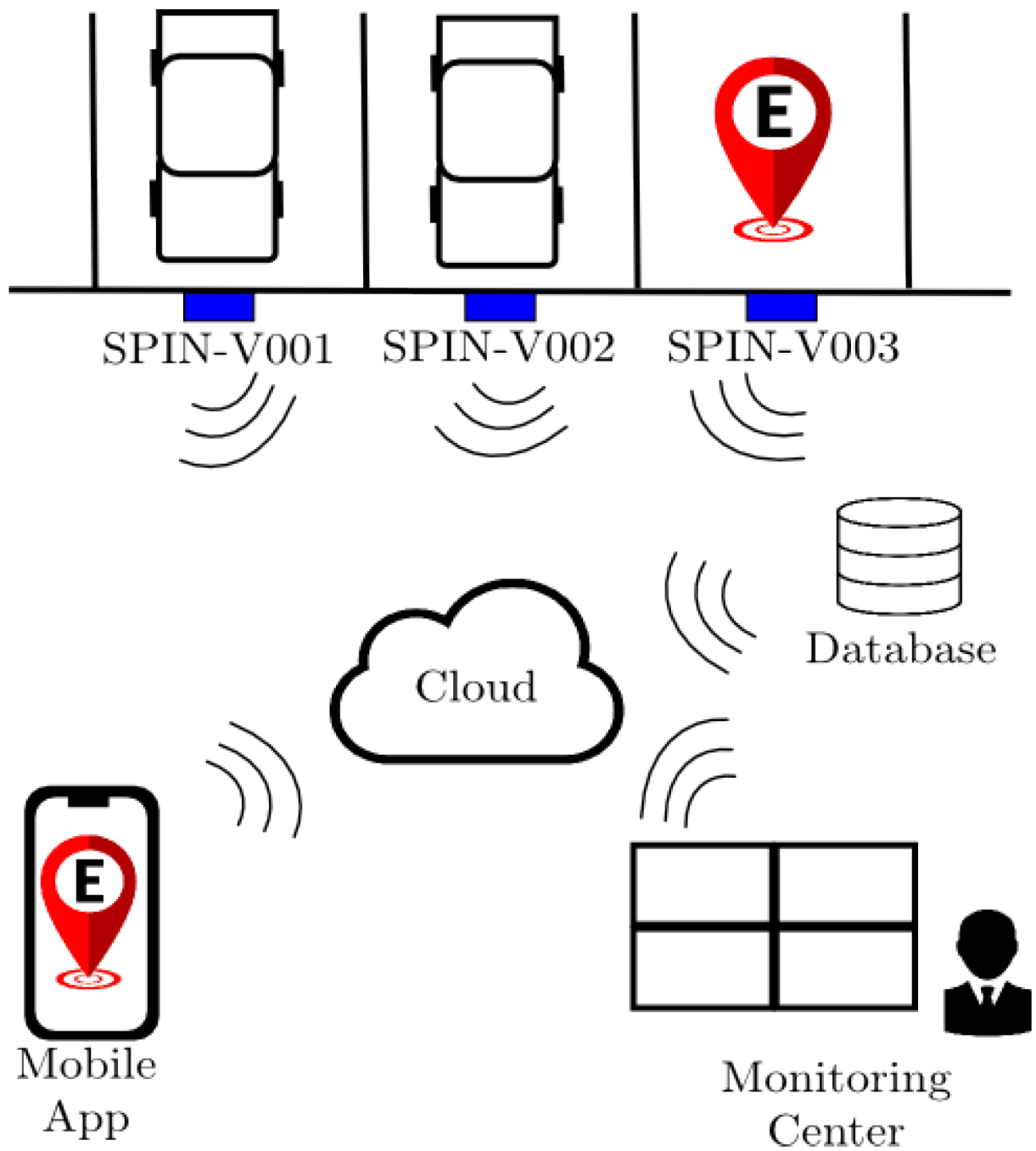


Figure 1.1. Management of Parking System

additional value conjointly. Smart parking development implies an IoT-based system that sends data about free and occupied parking places via web/mobile application. The IoT-device, including sensors and microcontrollers, is located in each parking place. This allows a complete solution for connecting, managing and absorbing data from IoT devices, and has meant we have been able to completely redefine and streamline how we manage the deployment, activation and administration of a range of smart sensors and devices. In this project the main role will be of cloud because, the data of all the parking slots will be available on the cloud. Cloud Computing is the delivery of computing services such as servers, storage, databases, networking, software, analytics, intelligence, and more, over the Cloud. Cloud Computing provides an alternative to the on-premises datacentre. With an on-premises datacentre, we have to manage everything, such as purchasing and installing hardware, virtualization, installing the operating system, and any other required applications, setting up the network, configuring the firewall, and setting up storage for data. After doing all the set-up, we become responsible for maintaining it through its entire lifecycle. We all know that one of the most common problems today is a saturation of parking spaces. Vehicles continue to outnumber existing parking spaces, thus clogging roads. Incidences of violence over occupancy, deformed cars due to a space crunch, and overcharging for parking are some problems that result. Smart parking development implies an IoT-based system that sends data about free and occupied parking places via web/mobile application. The IoT-device, including sensors and microcontrollers, is located in each parking place.

COMPONENTS

Available Components

By default, there are two repositories of components for your SDK: *Android Repository* and *Third party Add-ons*.

The *Android Repository* offers these types of components:

- **SDK Tools** — Contains tools for debugging and testing your application and other utility tools. These tools are installed with the Android SDK starter package and receive

periodic updates. You can access these tools in the <sdk>/tools/ directory of your SDK. To learn more about them, see SDK Tools in the developer guide.

- **SDK Platform-tools** — Contains platform-dependent tools for developing and debugging your application. These tools support the latest features of the Android platform and are typically updated only when a new platform becomes available. You can access these tools in the <sdk>/platform-tools/ directory. To learn more about them, see Platform Tools in the developer guide.
- **Android platforms** — An SDK platform is available for every production Android platform deployable to Android-powered devices. Each SDK platform component includes a fully compliant Android library, system image, sample code, and emulator skins. To learn more about a specific platform, see the list of platforms that appears under the section "Downloadable SDK Components" on the left part of this page.
- **USB Driver for Windows** (Windows only) — Contains driver files that you can install on your Windows computer, so that you can run and debug your applications on an actual device. You *do not* need the USB driver unless you plan to debug your application on an actual Android-powered device. If you develop on Mac OS X or Linux, you do not need a special driver to debug your application on an Android-powered device. See Using Hardware Devices for more information about developing on a real device.
- **Samples** — Contains the sample code and apps available for each Android development platform. If you are just getting started with Android development, make sure to download the samples to your SDK.
- **Documentation** — Contains a local copy of the latest multiversion documentation for the Android framework API.

The *Third party Add-ons* provide components that allow you to create a development environment using a specific Android external library (such as the Google Maps library) or a customized (but fully compliant) Android system image. You can add additional Add-on repositories by clicking **Add Add-on Site**.

Recommended Components

The SDK repository contains a range of components that you can download. Use the table below to determine which components you need, based on whether you want to set up a basic, recommended, or full development environment:

Environment	SDK Component	Comments
Basic	SDK Tools	If you've just installed the SDK starter package, then you already have the latest version of this component. The SDK Tools component is required to develop an Android application. Make sure you keep this up to date.
	SDK Platform-tools	This includes more tools that are required for application development. These tools are platform-dependent and typically update only when a new SDK platform is made available, in order to support new features in the platform. These tools are always backward compatible with older platforms, but you must be sure that you have the latest version of these tools when you install a new SDK platform.
	SDK platform	You need to download at least one platform into your environment, so that you will be able to compile your application and set up an Android Virtual Device (AVD) to run it on (in the emulator). To start with, just download the latest version of the platform.

		Later, if you plan to publish your application, you will want to download other platforms as well, so that you can test your application on the full range of Android platform versions that your application supports.
+		
Recommended (plus Basic)	Documentation	The Documentation component is useful because it lets you work offline and also look up API reference information from inside Eclipse.
	Samples	The Samples components give you source code that you can use to learn about Android, load as a project and run, or reuse in your own app. Note that multiple samples components are available — one for each robot platform version. When you are choosing a sample component to download, select the one who's API Level matches the API Level of the machine platform that you plan to use.
	Usb Driver	The Usb Driver module is needed only if you are developing on Windows and have an Android-powered device on which you want to install your application for debugging and testing. For Mac OS X and Linux platforms, no special driver is needed.
+		
Full (plus Recommended)	Google APIs	The Google APIs add-on gives your application access to the Maps external library, which makes it easy to display and

		manipulate Maps data in your application.
	Additional SDK Platforms	If you plan to publish your application, you will want to download additional platforms corresponding to the Android platform versions on which you want the application to run. The recommended approach is to compile your application against the lowest version you want to support, but test it against higher versions that you intend the application to run on. You can test your applications on different platforms by running in an Android Virtual Device (AVD) on the Android emulator.

Once you've installed at least the basic configuration of SDK components, you're ready to start developing Android apps. The next section describes the contents of the Android SDK to familiarize you with the components you've just installed.

For more information about using the Android SDK and AVD Manager, see the [Adding SDK Components](#) document.

Step 5. Exploring the SDK

Once you've installed the SDK and downloaded the platforms, documentation, and add-ons that you need, we suggest that you open the SDK directory and take a look at what's inside.

The table below describes the full SDK directory contents, with components installed.

Name	Description
add-ons/	Contains add-ons to the Android SDK development environment, which let you develop against external libraries that are available on some devices.

docs/	A full set of documentation in HTML format, including the Developer's Guide, API Reference, and other information. To read the documentation, load the file offline.html in a web browser.
platform-tools/	Contains platform-dependent development tools that may be updated with each platform release. The platform tools include the Android Debug Bridge (adb) as well as other tools that you don't typically use directly. These tools are separate from the development tools in the tools/ directory because these tools may be updated in order to support new features in the latest Android platform.
platforms/	Contains a set of Android platform versions that you can develop applications against, each in a separate directory.
<platform>/	Platform version directory, for example "android-11". All platform version directories contain a similar set of files and subdirectory structure. Each platform directory also includes the Android library (android.jar) that is used to compile applications against the platform version.
samples/	Sample code and apps that are specific to platform version.
tools/	Contains the set of development and profiling tools that are platform-independent, such as the emulator, the Android SDK and AVD Manager, ddms, hierarchy viewer and more. The tools in this directory may be updated at any time using the Android SDK and AVD Manager and are independent of platform releases.
SDK Readme.txt	A file that explains how to perform the initial setup of your SDK, including how to launch the Android SDK and AVD Manager tool on all platforms.
SDK Manager.exe	Windows SDK only. A shortcut that launches the Android SDK and AVD Manager tool, which you use to add components to your SDK.

Optionally, you might want to add the location of the SDK's tools/ and platform-tools to

your PATH environment variable, to provide easy access to the tools.

Android's Features

Handset layouts

The platform is adaptable to larger, VGA, 2D graphics library, 3D graphics library based on OpenGL ES 2.0 specifications, and traditional smart phone layouts.

Storage

SQLite, a lightweight relational database, is used for data storage purposes.

Connectivity

Android supports connectivity technologies including GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.

Messaging

SMS and MMS are available forms of messaging, including threaded text messaging and now Android Cloud To Device Messaging (C2DM) is also a part of Android Push Messaging service.

Multiple language support

Android supports multiple languages.

Web browser

The web browser available in Android is based on the open-source WebKit layout engine, coupled with Chrome'sV8 JavaScript engine. The browser scores 100/100 on the Acid3 test on Android 4.0.

Java support

While most Android applications are written in Java, there is no Java Virtual Machine in the platform and Java byte code is not executed. Java classes are compiled into Dalvik executables and run on Dalvik, a specialized virtual machine designed specifically for Android and optimized for battery-powered mobile devices with limited memory and CPU. J2ME support can be provided via third-party applications.

Media support

Android supports the following audio/video/still media formats: WebM, H.263, H.264 (in 3GP or MP4container), MPEG-4 SP, AMR, AMR-WB (in 3GP container), AAC, HE-AAC (in MP4 or 3GP container), MP3, MIDI, OggVorbis, FLAC, WAV, JPEG, PNG, GIF,

BMP.^[71]

Streaming media support

RTP/RTSP streaming (3GPP PSS, ISMA), HTML progressive download (HTML5 <video> tag). Adobe Flash Streaming (RTMP) and HTTP Dynamic Streaming are supported by the Flash plugin. Apple HTTP Live Streaming is supported by RealPlayer for Android, and by the operating system in Android 3.0 (Honeycomb).

Additional hardware support

Android can use video/still cameras, touchscreens, GPS, accelerometers, gyroscopes, barometers, magnetometers, dedicated gaming controls, proximity and pressure sensors, thermometers, accelerated 2D bit blits (with hardware orientation, scaling, pixel format conversion) and accelerated 3D graphics.

Multi-touch

Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero. The feature was originally disabled at the kernel level (possibly to avoid infringing Apple's patents on touch-screen technology at the time). Google has since released an update for the Nexus One and the Motorola Droid which enables multi-touch natively.

Bluetooth

Supports A2DP, AVRCP, sending files (OPP), accessing the phone book (PBAP), voice dialing and sending contacts between phones. Keyboard, mouse and joystick (HID) support is available in Android 3.1+, and in earlier versions through manufacturer customizations and third-party applications.

Video calling

Android does not support native video calling, but some handsets have a customized version of the operating system that supports it, either via the UMTS network (like the Samsung Galaxy S) or over IP. Video calling through Google Talk is available in Android 2.3.4 and later. Gingerbread allows Nexus S to place Internet calls with a SIP account. This allows for enhanced VoIP dialing to other SIP accounts and even phone numbers. Skype 2.1 offers video calling in Android 2.3, including front camera support.

Multitasking

Multitasking of applications is available.

Voice based features

Google search through voice has been available since initial release. Voice actions for calling, texting, navigation, etc. are supported on Android 2.2 onwards.

Tethering

Android supports tethering, which allows a phone to be used as a wireless/wired Wi-Fi hotspot. Before Android 2.2 this was supported by third-party applications or manufacturer customizations.

Screen capture

Android supports capturing a screenshot by pressing the power and volume-down buttons at the same time. Prior to Android 4.0, the only methods of capturing a screenshot were through manufacturer and third-party customizations or otherwise by using a PC connection (DDMS developer's tool). These alternative methods are still available with the latest Android.

External storage

Most Android devices include microSD slot and can read microSD cards formatted with FAT32, Ext3fs or Ext4fs file system. To allow use of high-capacity storage media such as USB flash drives and USB HDDs, many Android tablets also include USB 'A' receptacle. Storage formatted with FAT32 is handled by Linux Kernel VFAT driver, while 3rd party solutions are required to handle other popular file systems such as NTFS, HFS Plus and exFAT.

CHAPTER - 2

LITERATURE SURVEY

Jung, H. G., Choi, C. G., Kim, D. S., and Yoon, P. J. 2006. "System configuration of intelligent parking assistant system," Proc. 13th World Congr. Intell. Transp. Syst. Services, London, UK, Oct).they This proposed a novel light-stripe-projection-based target position-designation method for an intelligent parking assist system, providing an economical free-space-based target position-designation method for poorly lit indoor parking spaces without degradation of the external appearance when integrated into a vehicle's body. The proposed method can be applied by adding a low-cost light plane projector to existing backward-parking cameras. Light stripe features are detected by the difference between an image with the light plane projector on and off. Three-dimensional information of the parking area is reconstructed using the light-stripe-projection theory. By orientation normalization and depth-map construction, free spaces are detected, and the nearest is selected as the target position. Experimental results show that the proposed method can successfully designate the target position, in spite of poor lighting conditions and the black reflective surface of adjacently parked vehicles.

Kianpisheh, A., Mustaffa, N., Limtrairut, P., and Keikhosrokiani, P. 2012. "Smart parking system (SPS)architecture using ultrasonic detector," International Journal of Software Engineering and ItsApplications (6:3), pp 55-58. With the increase in vehicle production and world population, more and more parking spaces and facilities are required. In this paper a new parking system called Smart Parking System (SPS) is proposed to assist drivers to find vacant spaces in a car park in a shorter time. The new system uses ultrasonic (ultrasound) sensors to detect either car park occupancy or improper parking actions. Different detection technologies are reviewed and compared to determine the best technology for developing SPS. Features of SPS include vacant parking space detection, detection of improper parking, display of available parking spaces, and directional indicators toward vacant parking spaces, payment facilities and different types of parking spaces (vacant, occupied, reserved and handicapped) through

the use of specific LEDs. This paper also describes the use of an SPS system from the entrance into a parking lot until the finding of a vacant parking space. The system is designed for a four-level parking lot with 100 parking spaces and five aisles on each floor. The system architecture defines the essential design features such as location of sensors, required number of sensors and LEDs for each level, and indoor and outdoor display boards.

Panayappan, R., Trivedi, J. M., Studer, A., and Perrig, A. Year. "VANET-based approach for parking space availability," Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks, ACM2007, pp. 75-76. Parking availability notification and parking spot locator are two useful VANET applications. In this paper, we provide a network model through which the available parking lots and parking spots can be located efficiently and without extensive infrastructure support. The various security vulnerabilities of the model are noted and an appropriate security architecture is proposed.

Pham, T. N., Tsai, M.-F., Nguyen, D. B., Dow, C.-R., and Deng, D.-J. 2015. "A Cloud-Based Smart-Parking System Based on Internet-of-Things Technologies," IEEE Access (3), pp 1581-1591. they discussed the revolution of cycles and rhythms that is expected to take place in different branches of science and engineering in the 21st century, with a focus on communication and information processing. It presents high-quality papers in vibration sciences, rhythms and oscillations, neurosciences, mathematical sciences, and communication. It includes major topics in engineering and structural mechanics, computer sciences, biophysics and biomathematics, as well as other related fields. Offering valuable insights, it also inspires researchers to work in these fields. The papers included in this book were presented at the 1st International Conference on Engineering Vibration, Communication and Information Processing (ICoEVCI-2018), India.

Solic, P., Marasovic, I., Stefanizzi, M. L., Patrono, L., and Mainetti, L. Year. "RFID-based efficient method for parking slot car detection," Software,

Telecommunications and Computer Networks (SoftCOM),2015 23rd International Conference on, IEEE2015, pp. 108. The literature review conducted for this paper offers an in-depth review of the recent advances in sensing and communication technology concerning parking systems. In addition, this paper presents a survey and analysis of an academic, qualitative literature review. It includes an in-depth study of the selected topics and provides a step by step implementation process. It reviews different smart parking systems used for parking guidance and parking facility management and gives an insight into the technical aspects and specifications analysis of such systems that have been published in academia during the last 15 years.

Song, I., Gowan, K., Nery, J., Han, H., Sheng, T., Li, H., and Karray, F. Year. "Intelligent parking system design using FPGA," 2006 International Conference on Field Programmable Logic and Applications,IEEE2006, pp. 1-6. In this research, we introduce FPGA based fuzzy logic controller (FLC. [...] Key MethodWe build a small-scaled robot car and test on a real environment with VHDL code for wall following and parking. This paper describes the background of fuzzy logic system, the design of fuzzy logic system with FPGA and the experimental results.

Xu, J., Chen, G., and Xie, M. Year. "Vision-guided automatic parking for smart car," Proceedings of the IEEE Intelligent Vehicles Symposium2000, pp. 725-730.This describes about how to effectively and efficiently locating parking-slots using a vision-based system is still an unresolved issue. Even more seriously, there is no publicly available labeled benchmark dataset for tuning and testing parking-slot detection algorithms. In this paper, we attempt to fill the above-mentioned research gaps to some extent and our contributions are twofold. Firstly, to facilitate the study of vision-based parking-slot detection, a large-scale parking-slot image database is established. This database comprises 8600 surround-view images collected from typical indoor and outdoor parking sites. For each image in this database, the marking-points and parking-slots are carefully labeled. Such a database can serve as a benchmark to design and validate parking-slot detection algorithms. Secondly, a learning-based parking-slot detection approach, namely PSDL , is proposed. Using PSDL , given a surround-view

image, the marking-points will be detected first and then the valid parking-slots can be inferred. The efficacy and efficiency of PSDL have been corroborated on our database. It is expected that PSDL can serve as a baseline when the other researchers develop more sophisticated methods

Yang, L., Rongguo, M., and Longfei, W. Year. "Intelligent Parking Negotiation Based on Agent Technology," Information Engineering, 2009. ICIE'09. WASE International Conference on, IEEE2009, pp. 265-268 The literature review conducted for this paper offers an in-depth review of the recent advances in sensing and communication technology concerning parking systems. In addition, this paper presents a survey and analysis of an academic, qualitative literature review. It includes an in-depth study of the selected topics and provides a step by step implementation process. It reviews different smart parking systems used for parking guidance and parking facility management and gives an insight into the technical aspects and specifications analysis of such systems that have been published in academia during the last 15 years.

Yeh, H.-T., Chen, B.-C., and Wang, B.-X. 2016. "A City Parking Integration System Combined with CloudComputing Technologies and Smart Mobile Devices," Eurasia Journal of Mathematics, Science &Technology Education (12:5), pp 1231-1242. The current study applied cloud computing technology and smart mobile devices combined with a streaming server for parking lots to plan a city parking integration system. It is also equipped with a parking search system, parking navigation system, parking reservation service, and car retrieval service. With this system, users can quickly find parking lots near their destination, using various parking servers and the system center to connect to and integrate the cloud server. The computing power of the cloud servers transmits five of the closest parking lots with available parking spaces to a user's smart mobile device so the user can choose the most suitable parking lot. After users arrive at the parking lot, they can scan the information in the parking lot using an RFID indoor positioning system and the users' current location, so that the location of vacant parking spaces in the parking lot and moving vehicles will be displayed on their smart mobile device. In addition, the stored traffic flow record can provide local

governments with statistics for building parking lots in the city in the future

Zhao, Y., and Collins, E. G. 2005. "Robust automatic parallel parking in tight spaces via fuzzy logic," *Robotics and Autonomous Systems* (51:2), pp 111-127. This paper develops and experimentally demonstrates a robust automatic parallel parking algorithm for parking in tight spaces. Novel fuzzy logic controllers are designed for each step of the maneuvering process. The controllers are first demonstrated by simulation using the kinematic model of a skid steering autonomous ground vehicle (AGV). They are then demonstrated experimentally on a skid steering AGV and it is shown that the developed algorithm has the ability to parallel park AGVs in tight spaces under both vehicle localization errors and parking space detection errors. This paper also presents a genetic fuzzy system which uses a genetic algorithm's learning ability to determine effective parameters for the developed fuzzy logic controllers. The genetic fuzzy system is used to tune the fuzzy logic algorithm for both a skid steering AGV and a front-wheel steering AGV.

Zheng, Y., and Cao, J. Year. "An intelligent car park management system based on wireless sensor networks," 2006 First International Symposium on Pervasive Computing and Applications, IEEE2006, pp. 65-70. This describes about a WSN-based intelligent car parking system. In the system, low-cost wireless sensors are deployed into a car park field, with each parking lot equipped with one sensor node, which detects and monitors the occupation of the parking lot. The status of the parking field detected by sensor nodes is reported periodically to a database via the deployed wireless sensor network and its gateway. The database can be accessed by the upper layer management system to perform various management functions, such as finding vacant parking lots, auto-toll, security management, and statistic report. We have implemented a prototype of the system

CHAPTER 3

METHODOLOGY

3.1.EXISTING METHODOLOGY

currently in use In general, the parking area is managed by human resources.As a consequence, there are traffic queues. It is impossible to know the number of open slots ahead of time. In emergency cases, the user cannot book their slot using the current technique. When the city becomes congested, everyone will be unable to find a parking spot. Proposed Methodology The proposed framework is an Android application that is made up of three key components.



Figure 3.1.1. Board showing available Parking Slots

One of the most common problems today is a saturation of parking spaces. Vehicles continue to outnumber existing parking spaces, thus clogging roads. Incidences of violence over occupancy, deformed cars due to a space crunch, and overcharging for parking are some problems that result. Admist this people are still not able to perfectly find the available slots for parking their vehicles.

3.2.PROPOSED SYSTEM

After logging into the app, the user can choose a parking space that is closest to his destination. When a consumer books a parking space, the administrator changes the status of that parking space to "RESERVED." If the user does not arrive at the parking slot within 20 minutes of making the reservation, the reservation is cancelled and the status is changed to "EMPTY."The client-server architecture underpins the Intelligent Parking System (IPS). It's economically beneficial since it doesn't require any heavy infrastructure. It is neither sensitive to temperature change nor affected by any extreme air turbulence.

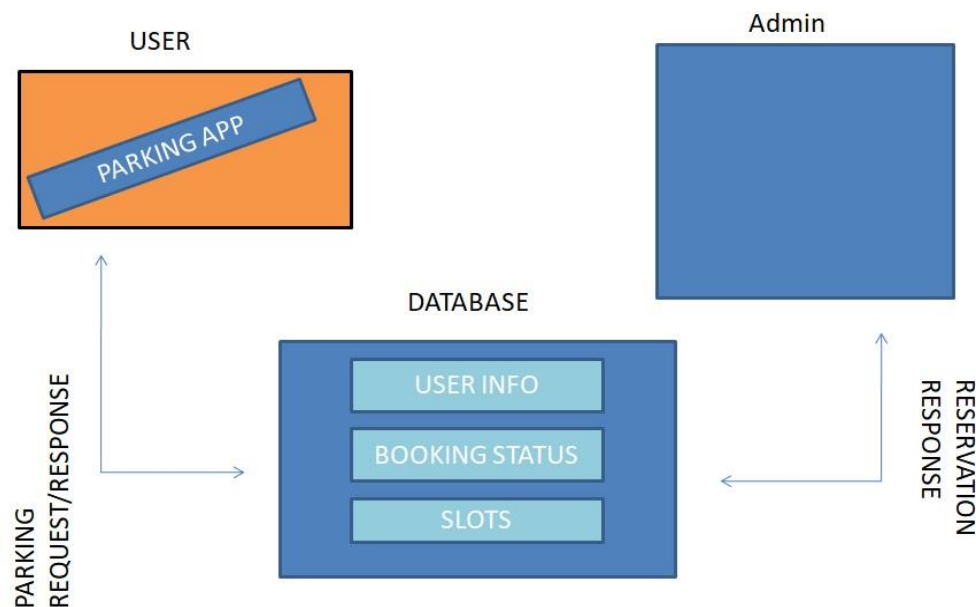


Figure 3.2.1. Block Diagram for proposed system

The following are the key goals of the Intelligent Parking System(IPS)application:

1. An intelligent, ubiquitous, and user-friendly automated parking system application that saves users time and helps them escape traffic congestion in major cities.
2. To ensure that parking spaces are safe and protected within a restricted area.

3.2.1.IMPLEMENTATION

The following is the sequence of the slot allocation method:

Step 1: Initially, the user selects the slot from his mobile phone. He looks for a parking spot that is close to his location and sees if one is available. If it is open, he advances to the next stage; otherwise, he returns to the previous state.

Step 2: Using an Android application, sends a request for a parking spot from the phone.

Step3:The Parking Control Unit (PCU) receives the user's requested slot number

Step 4: If the payment is completed successfully, the requested parking space will be reserved.

Step5: After the user has reserved a specific slot, the status of that slot is labeled

Step 6: As soon as the vehicle is parked in the designated space

Step7: As soon as the vehicle leaves the parking spot, the total cost will be shown. A module is a component of a larger device.

The Intelligent Parking System is made up of three modules. They are the following three:

1: Administrator Module

2: User Module

3: Booking Module Server Side

The administrator first enters his username and password into the application.

Server Side Initially the administrator logs in the application by using his username and password. The administrator has authority to add new users and stores their details in the database which are used for further purpose.

Login: The administrator can login to the application by giving email and password. If the administrator gets successfully login then the administrator is said to be authorized.

After getting login to the application the administrator can carry out many tasks such as:

- Adding Parking Locations
- View Parking Locations
- View All Users
- View All Bookings

3.2.2.MANAGING LOCATIONS

Add and view parking locations:

The administrator may check the status of different parking slots as well as display different locations where parking slots are open. View All Users and Their Scheduled Slots: The administrator will see all of the users who are using the app and review their booking information, such as the time and date the user needs a slot, the number of hours a person is using the assigned slot, the position at which he needs a slot, and so on. The reserved slots of all authorised users are visible to the administrator. The administrator takes this as a reference for further allocation.

3.2.3.Application Framework

Framework for Applications By making an open development platform available,Android allows developers to create apps that are incredibly rich and creative. Developers can use the device's hardware to access location data, run background services, set alarms, display alerts in the status bar, and much more.

The same platform APIs that the core applications use are available to developers. The application architecture is designed to facilitate component reuse; any application can publish its capabilities, which can then be used by other applications (subject to security constraints enforced by the framework). The consumer can substitute components using the same mechanism.

- All applications are supported by a range of facilities and systems, which include Lists, grids, text boxes, icons, and even an embeddable web interface are among the Views that can be used to build an application.
- Content Providers allow apps to access data from other apps (like Contacts), as well as share their own data.
- A Resource Manager is someone who manages resources
- A Notification Manager that allows all apps to display custom notifications in the status bar
- An activity manager that handles application lifecycles and offers a unified navigation backend

3.2.4. Libraries

Android contains a collection of C/C++ libraries that are used by various Android components. The Android application platform makes these features available to developers.

The following are some of the most important libraries:

- **System C library** (libc) is a BSD-derived implementation of the regular C system library (libc) that is optimised for embedded Linux devices.
- **Media Libraries** - The libraries are based on PacketVideo's OpenCORE and allow playback and recording of a variety of audio and video formats, as well as static image
- **Surface Manager** - handles access to the display subsystem and seamlessly composites 2D and 3D graphic layers from various applications.
- **LibWebCore** - a modern web browser engine which powers both the Android browser and an embeddable web view
- **SGL** - the underlying 2D graphics engine
- **3D libraries** - an implementation based on OpenGL ES 1.0 APIs; the libraries use either hardware 3D acceleration (where available) or the included, highly optimized 3D software rasterizer
- **FreeType** - bitmap and vector font rendering

- **SQLite** - a powerful and lightweight relational database engine available to all applications

3.2.5.Android Runtime

Android includes a collection of core libraries that has most of the practicality offered within the core libraries of the Java programming language. Every mechanical man application runs in its own method, with its own instance of the virtual machine referred to as Dalvik virtual machine. Dalvik has been written in order that a tool will run multiple VMs with efficiency. The Dalvik VM executes files within the Dalvik feasible (.dex) format that is optimized for lowest memory footprint. The VM is register-based, and runs categories compiled by a Java language compiler that are reworked into the .dex format by the enclosed "dx" tool.

The Dalvik VM depends on the Linux kernel for underlying practicality like threading and low-level memory management.

Linux Kernel

Android's core system services, such as security, memory management, process management, network stack, and driver model, are all based on Linux version 2.6. Between the hardware and the rest of the software stack.

ANDROID SDK

Rather than installing a new SDK starter kit, you can use the Android SDK and AVD Manager to upgrade to the current tools or framework

http://dl.google.com/android/android-sdk_r16-macosx.zip

Here's a rundown of the steps you'll need to take to instal the Android SDK:.

1. Prepare your development computer and ensure it meets the system requirements.
2. Install the SDK starter package from the table above. (If you're on Windows, download the installer for help with the initial setup.)
3. you'll need to instal the ADT Plugin for Eclipse.

4. To your SDK, Add Android platforms and other components.
5. Examine the Android SDK's contents.

3.2.6. System Requirements

Supported Operating Systems

- Windows XP (32-bit), Vista (32- or 64-bit), or Windows 7 (32- or 64-bit)
- Mac OS X 10.5.8 or later (x86 only)
- Linux (tested on Ubuntu Linux, Lucid Lynx)
 - GNU C Library (glibc) 2.7 or later is required.
 - On Ubuntu Linux, version 8.04 or later is required.
 - 64-bit distributions must be capable of running 32-bit applications. For information about how to add support for 32-bit applications

Supported Development Environments for Android SDK

Eclipse IDE

- Eclipse 3.6 (Helios) or greater
- Note:** Eclipse 3.5 (Galileo) is no longer supported with the latest version of ADT.
- Eclipse JDT plugin (included in most Eclipse IDE packages)
 - If you need to install or update Eclipse, you can download it from <http://www.eclipse.org/downloads/>.

Several types of Eclipse packages are available for each platform. For developing Android applications, we recommend that you install one of these packages:

- Eclipse IDE for Java Developers
- Eclipse Classic
- Eclipse IDE for Java EE Developers
- JDK 5 or JDK 6 (JRE alone is not sufficient)
- Android Development Tools plugin (recommended)
- Notcompatible with Gnu Compiler for Java (gcj)

Other development environments or IDEs

- JDK 5 or JDK 6 (JRE alone is not sufficient)
- Apache Ant 1.8 or later
- Not compatible with Gnu Compiler for Java (gcj)

Hardware requirements for Android SDK

The Android SDK requires disk storage for all of the components that you choose to install. The table below provides a rough idea of the disk-space requirements to expect, based on the components that you plan to use.

Component type	Approximate size	Comments
SDK Tools	35 MB	Required.
SDK Platform-tools	6 MB	Required.
Android platform (each)	150 MB	At least one platform is required.
SDK Add-on (each)	100 MB	Optional.
USB Driver for Windows	10 MB	Optional. For Windows only.
Samples (per platform)	10M	Optional.
Offline documentation	250 MB	Optional.

Note that the disk-space requirements above are *in addition to* those of the Eclipse IDE, JDK, or other prerequisite tools that you may need to install on your development computer.

INSTALLING THE SDK

Step 1. Preparing Your Development Computer

Before getting started with the Android SDK, take a moment to confirm that your development computer satisfies the system's requirements. If you don't already have it, you'll probably need to install the JDK. If you're going to build in Eclipse with the Android Development Tools (ADT) Plugin—which is the recommended route if you're new to Android—make sure you have the right version of Eclipse installed on your device, as mentioned in the System Requirements. document.

If you need to install Eclipse, you can download it from this location:

<http://www.eclipse.org/downloads/>

The "Eclipse Classic" version is recommended. Otherwise, a Java or RCP version of Eclipse is recommended.

Step 2. Downloading the SDK Starter Package

The SDK starter package is not a full development environment—it includes only the core SDK Tools, which you can use to download the rest of the SDK components (such as the latest Android platform).

If you haven't already, get the latest version of the SDK starter package from the [SDK download page](#).

If you downloaded a .zip or .tgz package (instead of the SDK installer), unpack it to a safe location on your machine. By default, the SDK files are unpacked into a directory named android-sdk-<machine-platform>.

If you downloaded the Windows installer (.exe file), run it now and it will check whether the proper Java SE Development Kit (JDK) is installed (installing it, if necessary), then install the SDK Tools into a default location (which you can modify).

Make a note of the name and location of the SDK directory on your system—you will need to refer to the SDK directory later, when setting up the ADT plugin and when using the SDK tools from the command line.

Step 3. Installing the ADT Plugin for Eclipse

Android offers a custom plugin for the Eclipse IDE, called Android Development Tools (ADT), that is designed to give you a powerful, integrated environment in which to build Android applications. It extends the capabilities of Eclipse to let you quickly set up new Android projects, create an application UI, debug your applications using the Android SDK tools, and even export signed (or unsigned) APKs in order to distribute your application. In general, developing in Eclipse with ADT is a highly recommended approach and is the fastest way to get started with Android.

If you'd like to use ADT for developing Android applications, install it now. Read [Installing the ADT Plugin](#) for step-by-step installation instructions, then return here to continue the last step in setting up your Android SDK.

If you prefer to work in a different IDE, you do not need to install Eclipse or ADT. Instead, you can directly use the SDK tools to build and debug your application. The [Introduction](#) to Android application development outlines the major steps that you need to complete when developing in Eclipse or other IDEs.

Step 4. Adding Platforms and Other Components

The last step in setting up your SDK is using the Android SDK and AVD Manager (a tool included in the SDK starter package) to download essential SDK components into your development environment.

The SDK uses a modular structure that separates the major parts of the SDK—Android platform versions, add-ons, tools, samples, and documentation—into a set of separately installable components. The SDK starter package, which you've already downloaded, includes only a single component: the latest version of the SDK Tools. To develop an Android application, you also need to download at least one Android platform and the

associated platform tools. You can add other components and platforms as well, which is highly recommended.

If you used the Windows installer, when you complete the installation wizard, it will launch the Android SDK and AVD Manager with a default set of platforms and other components selected for you to install. Simply click **Install** to accept the recommended set of components and install them. You can then skip to Step 5, but we recommend you first read the section about the Available Components to better understand the components available from the Android SDK and AVD Manager.

You can launch the Android SDK and AVD Manager in one of the following ways:

- From within Eclipse, select **Window > Android SDK and AVD Manager**.
- On Windows, double-click the SDK Manager.exe file at the root of the Android SDK directory.
- On Mac or Linux, open a terminal and navigate to the tools/ directory in the Android SDK, then execute:
android

To download components, use the graphical UI of the Android SDK and AVD Manager to browse the SDK repository and select new or updated components (see figure 1). The Android SDK and AVD Manager installs the selected components in your SDK environment. For information about which components you should download, see Recommended Components.

3.2.7.VERSION HISTORY

mechanism has seen variety of updates since its original unleash, every fixing bugs and adding new options. every version is called, in alphabetical order, when a sweet.

- 2.3 cake refined the computer programme, improved the soft keyboard and copy/paste options, improved recreation performance, additional SIP support (VoIP calls), and additional support for close to Field Communication.
- 3.0 Honeycomb was a tablet-oriented unleash that supports larger screen

devices and introduces several new computer programme options, support for multi-core processors, hardware acceleration for graphics and full system encoding. the primary device that includes this version, the Motorola Xoom pill, went on sale in Feb 2011.

□ 3.1 Honeycomb, discharged in could 2011, additional support for further input devices, USB host mode for transferring info directly from cameras and alternative devices, and therefore the Google Movies and Books apps.

□ 3.2 Honeycomb, discharged in July 2011, additional improvement for a broader vary of screen sizes, new "zoom-to-fill" screen compatibility mode, loading media files directly from Coyote State card, associated an extended screen support API. Huawei MediaPad is that the initial seven in. pill to use this version.

- 4.0 frozen dessert Sandwich, declared on Gregorian calendar month nineteen, 2011, brought Honeycomb options to smartphones and additional new options together with identity verification unlock, network information usage watching and management, unified social networking contacts, photography enhancements, offline email looking, app folders, and knowledge sharing exploitation NFC. Android 4.0.3 frozen dessert Sandwich is that the latest automaton version that's out there to phones. The ASCII text file of automaton four.0.1 was discharged on Gregorian calendar month one4, 2011.



Prefer a video? We have one for you.

Python Program

-
- Python programs are composed of modules
 - Modules contain statements

- Statements contain expressions
- Expressions create and process objects

Features of Python

Open source: Python is publicly available open source software,any one can use source code that doesn't cost anything.

Easy-to-learn: Popular (scripting/extension) language, clear and easy syntax, no type declarations, automatic memory management, high-level data types and operations, design to read (more English like syntax) and write (shorter code compared to C, C++, and Java) fast.

High-levelLanguage:

High-level language (closer to human) refers to the higher level of concept from machine language (for example assembly languages). Python is an example of a high-level language like C, C++, Perl, and Java with low-level optimization.

Portable:

High level languages are portable, which means they are able to run across all major hardware and software platforms with few or no change in source code. Python is portable and can be used on Linux, Windows, Macintosh, Solaris, FreeBSD, OS/2, Amiga, AROS, AS/400 and many more.

Object-Oriented: Python is a full-featured object-oriented programming language, with features such as classes, inheritance, objects, and overloading.

Python has an interactive console where you get a Python prompt (command line) and interact with the interpreter directly to write and test your programs. This is useful for mathematical programming.

Interpreted : Python programs are interpreted, takes source code as input, and then compiles (to portable byte-code) each statement and executes it immediately. No need to compiling or linking

Extendable : Python is often referred to as a "glue" language, meaning that it is capable to work in mixed-language environment. The Python interpreter is easily extended and can add a new built-in function or modules written in C/C++/Java code.

Libraries : Databases, web services, networking, numerical packages, graphical user interfaces, 3D graphics, others.

Supports :Support from online Python community

Python Interpreter

- In interactive mode, type Python programs and the interpreter displays the result:
- Type python into your terminal's command line
- After a short message, the >>> symbol will appear
- The above symbol signals the start of a Python interpreter's command line.
- Python interpreter evaluates inputs (For example >>> 4*(6-2) return 16)

How stable is Python?

Very stable. New, stable releases are popping out roughly each half dozen to eighteen months since 1991, and this appears seemingly to continue. presently there ar sometimes around eighteen months between major releases. The latest stable releases will invariably be found on the Python transfer page. There ar 2 counseled production-ready versions at this time in time, as a result of at the instant there ar 2 branches of stable releases: a pair of.x and 3.x. Python 3.x could also be less helpful than a pair of.x, since presently there's additional third party code accessible for Python a pair of than for Python three. Python a pair of code can typically not run unchanged in Python three.

History

The name Python was elect from "Monty Pythons Flying Circus" that was a British sketch comedy series created by the comedy cluster Monty Python and broadcast by the BBC from 1969 to 1974. Python was created within the early Nineties by Guido van Rossum at the National analysis Institute for arithmetic and technology in Holland. Python was created as a successor of a language known as rudiment (All Basic Code) and discharged publically in1991. Guido remains Python's principal author, though it includes several contributions from active user community. Between 1991 and 2001 there area unit many versions discharged, current stable unharness is three.2. In 2001 the Python software system Foundation (PSF) was shaped, a non-profit organization created specifically to possess Python-related holding. Zope Corporation could be a sponsoring member of the terrorist organization. All most all Python releases area unit Open supply. to check the main points of unharness versions and licence agreement of Python [check here](#).

Python Is Easy to Use

A simple program written in C++, C, Java and Python. All program prints "Hello world".

C++ Program :

```
#include<iostream>
intmain()
{
std::cout<<"Hello World"<< std::endl;
return0;
}
Copy
```

C Program :

```
#include<stdio.h>
intmain(intargc,char**argv)
{
printf("Hello, World!\n");
}
```

Copy

Java Program :

```
publicclassHello
{
publicstaticvoidmain(String argv[])
{
System.out.println("Hello, World!");
}
}
```

Copy

Python Program:

```
print("Hello World")
```

Copy

Python introduction video

Python Environment

AIX	AROS	AS/400 (OS/400)	BeOS
MorphOS	MS-DOS	OS/2	OS/390 and z/OS
Palm OS	PlayStation and PSP	Psion	QNX
RISC OS	Series 60	Solaris	VMS
Windows CE or Pocket PC	HP-UX	Linux	

Major uses of Python

- System utilities (system admin tools, command line programs).
- Web Development.
- Graphical User Interfaces (Tkinter, gtk, Qt).
- Internet scripting.
- Embedded scripting.
- Database access and programming.
- Game programming.
- Rapid prototyping and development.
- Distributed programming

Organizations Using Python (sector wise)

Web Development : Yahoo Maps, Yahoo teams, Google, Zope Corporation, Ultraseek,
Linux Weekly News, ElasticHosts Cloud Servers, Mojam.com, hunch, Shopzilla,

Movieplayer.it, Multiplayer.it. Games: field of battle a pair of, Crystal area, Star Trek Bridge Commander, The Temple of Elemental Evil, Vampire: The Masquerade: Bloodlines, Civilization four, QuArK (Quake Army Knife) Graphics : Industrial light-weight & Magic, filmmaker Feature Animation, HKS, Inc. (ABAQUS/CAE), RoboFog, Caligari Corporation, Blender 3D, Jasc code, Paint look professional. Financial :Altis Investment Management, ABN AMRO Bank, Treasury Systems, Bellco bank, Journyx Timesheet and Resource Management code. Science : National Weather Service, measuring device Remote Sensing cluster, Applied Maths, Biosoft, The National analysis Council of Canada, town National Laboratory (LANL) Theoretical Physics Division, AlphaGene, Inc., LLNL, NASA, Swedish earth science and Hydrological Institute (SMHI), Environmental Systems analysis Institute (ESRI), Objexx Engineering, Nmag procedure Micromagnetics Electronic style Automation: Ciranova, Productivity style Tools, Object Domain, Pardus, Red Hat, SGI, Inc., MCI Worldcom, Nokia, Education : University of Calif., Irvine, Smeal school of Business, The Pennsylvania State University, New island Digital Library, IT Certification examination preparation, SchoolTool, Business code : Raven Bear Systems Corporation, Thawte Consulting, Advanced Management Solutions opposition., IBM, Arakn, RealNetworks, dSPACE, Escom, the little Company, Nexedi, Piensa Technologies - Bufete Consultor First State United Mexican States, Nektra, WuBook.

Is Python a good language for beginning programmers?

Yes. it's still common to start out students with a procedural and statically written language like Pascal, C, or a set of C++ or Java. Students could also be higher served by learning Python as their natural language. Python contains a terribly straightforward and consistent syntax and an outsized normal library and, most significantly, victimization Python in a very starting programming course lets students focus on vital programming skills like drawback decomposition and knowledge sort style. With Python, students may be quickly introduced to basic ideas like loops and procedures. they will in all probability even work with user-defined objects in their terribly 1st course. For a

student World Health Organization has ne'er programmed before, employing a statically written language appears unnatural. It presents extra complexness that the scholar should master and slows the pace of the course. the scholars are attempting to be told to assume sort of a pc, decompose issues, style consistent interfaces, and encapsulate knowledge. whereas learning to use a statically written language is very important within the future, it's not essentially the most effective topic to deal with within the students' 1st programming course. Many different aspects of Python build it an honest natural language. Like Java, Python contains a massive normal library so students may be assigned programming comes terribly early within the course that do one thing. Assignments aren't restricted to the quality four-function calculator and check equalization programs. By victimization the quality library, students will gain the satisfaction of acting on realistic applications as they learn the basics of programming. victimization the quality library additionally teaches students regarding code apply. Third-party modules like PyGame are useful in extending the students' reach.

SYSTEM TESTING

The purpose of testing is to seek out errors. Testing is that the tactic of Associate to seek out every conceivable fault or weakness in an passing work product. It provides how to ascertain the practicality of elements, sub assemblies, assemblies and/or a finished product it's the method of exercise code with the intent of guaranteeing that the Software system meets its necessities associated user expectations and doesn't fail in an unacceptable manner. There are varied forms of take a look at every take a look at kind addresses a particular testing demand.

TYPES OF TESTS

5.1.1. Unit testing

Unit checking involves the design of check cases that validate that the within program logic is functioning properly, that program inputs prove valid outputs. All call branches

and internal code flow need to be valid. it is the testing of individual software units of the appliance .it is done once the completion of a private unit before integration. this may be a structural testing, that depends on info of its construction and is invasive. Unit checks perform basic checks at half level and test a selected business process, application, and/or system configuration. Unit tests confirm that each unique path of a business methodology performs accurately to the documented specifications and contains clearly printed inputs and expected results.

5.1.2.Integration testing

Integration tests ar designed to examine integrated package components to work out if they terribly lie program. Testing is event driven and is extra involved the essential outcome of screens or fields. Integration tests demonstrate that tho' the components were one by one satisfaction, as shown by successfully unit testing, the mixture of components is correct and consistent. Integration testing is specifically aimed toward exposing the issues that arise from the mixture of components.

5.1.3.Functional check

Functional tests supply systematic demonstrations that functions tested ar gettable as nominal by the business and technical requirements, system documentation, and user manuals. Functional testing is targeted on the next items:

Valid Input : noted classes of valid input ought to be accepted.

Invalid Input : noted classes of invalid input ought to be rejected. Functions : noted functions ought to be exercised.

Output : noted classes of application outputs ought to be exercised.

Systems/Procedures: interfacing systems or procedures ought to be invoked.

Organization and preparation of helpful tests is targeted on requirements, key functions, or special check cases. in addition, systematic coverage relating to verify Business methodology flows; data fields, predefined processes, and serial processes must be thought of for testing. Before helpful testing is complete, more tests ar identified and so the effective price of current tests is set .

SYSTEM CHECK

System testing ensures that the total integrated computer code meets necessities.

It tests a configuration to form certain famed and inevitable results. AN example of system checking is that the configuration minded system integration test. System testing is predicated on method descriptions and flows, action pre-driven method links and integration points.

White Box Testing

White Box Testing is also a testing throughout that throughout that the computer code tester has knowledge of the inner workings, structure and language of the computer code, or a minimum of its purpose. it's purpose. it's accustomed check areas that cannot be reached from a recorder level.

Black Box Testing

Black Box Testing is testing the computer code with none data of the inner workings, structure or language of the module being tested. recording machine tests, as most alternative forms of tests, should be written from a definitive supply document, like specification or

requirements document, like specification or necessities document. it's a testing during that the computer code underneath check is treated, as a recorder .you cannot "see" into it. The check provides inputs and responds to outputs while not considering however the computer code works.

Unit Testing:

Unit checking is often conducted as a vicinity of a combined code and unit test section of the computer code lifecycle, though it isn't uncommon for writing and unit testing to be conducted as 2 distinct phases.

Test strategy and approach

Field testing ar reaching to be performed manually and purposeful tests ar reaching to

be written intimately .

Test objectives

- All field entries should work properly.
- Pages should be activated from the known link.
- The entry screen, messages and responses should not be delayed.

Features to be tested

- Verify that the entries are of the proper format
- No duplicate entries ought to be allowed
- All links ought to take the user to the proper page.

INTEGRATION TESTING

Software integration testing is that the incremental integration testing of two or more integrated software components on one platform to supply failures caused by interface defects.

The task of the mixing test is to see that components or software applications, e.g. components during a software or – one intensify – software applications at the corporate level – interact without error.

Test Results: All the test cases mentioned above are passed successfully. No defects are encountered.

Acceptance Testing

User Acceptance Testing may be a critical phase of any project and requires significant participation by the top user. It also ensures that the system meets the functional

requirements.

Test Results: All the test cases mentioned above are passed successfully. No defects are encountered.

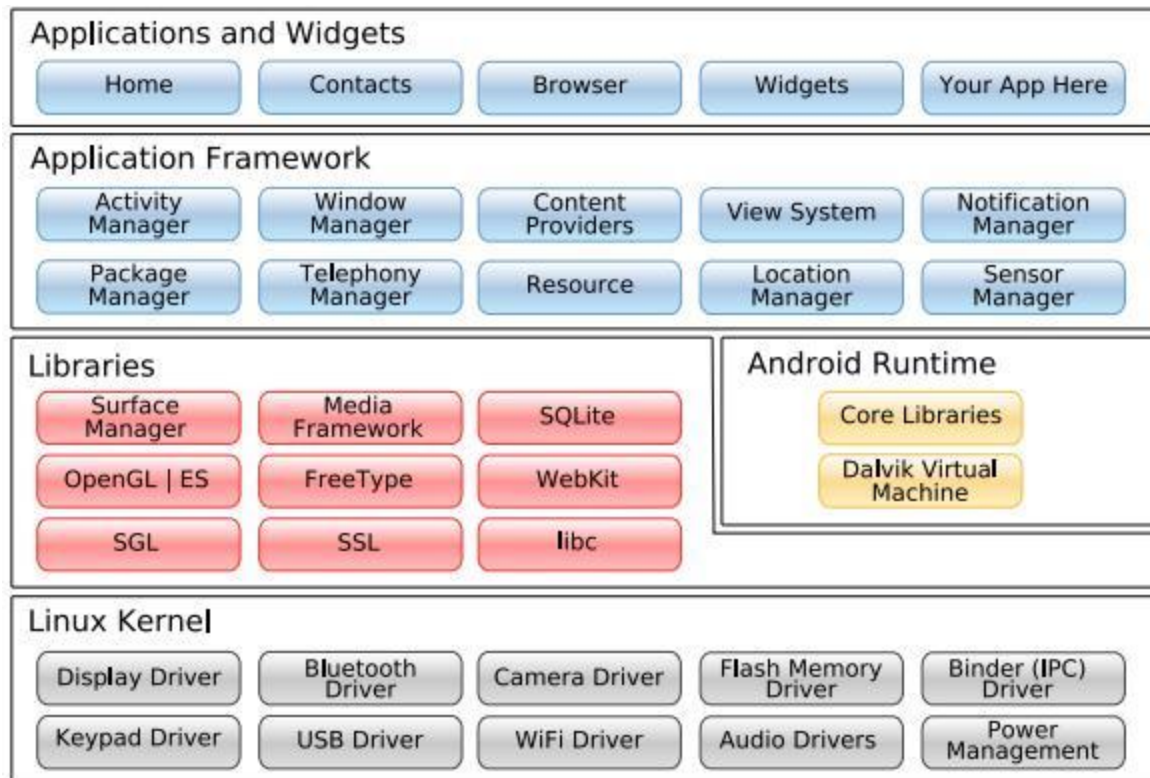
CHAPTER – 4

SOFTWARE ENVIRONMENT

Android is a mobile operating system based on Linux that runs on smartphones and tablets. The Open Handset Alliance, which is headed by Google, is working on it. Android has a broad developer community that creates applications (or "apps") that enhance the device's functionality. The majority of the code is written in a customised version of Java. Android's structure

4.1.Android architecture

Android is made up of a kernel based on the Linux kernel, middleware, libraries, and APIs written in C, and application software running on an application system based on Apache Harmony that contains Java-compatible libraries. To run Dalvik dex-code (Dalvik Executable), which is normally translated from Java bytecode. Android uses the Dalvik virtual machine with just-in-time compilation. Android uses the ARM architecture as its primary hardware platform. The Android x86 project provides support for x86, and Google TV runs on an x86-only version of Android.



The Android kernel is based on the Linux kernel, but Google has made additional architectural improvements outside of the normal Linux kernel development cycle. Since Android lacks a native X Window System and does not support the full collection of standard GNU libraries, porting existing Linux applications or libraries to Android is difficult.

Certain Google contributions to the Linux kernel, most notably a power management feature called wakelocks, were rejected by mainline kernel developers, partially because kernel maintainers believed Google did not want to maintain their own code. Despite Google's announcement in April 2010 that two employees would be hired to work with the Linux kernel community, Greg Kroah-Hartman, the new Linux kernel maintainer for the -stable branch, expressed concern in December 2010 that Google was no longer attempting to bring their code improvements into mainstream Linux. Since they were a small team with more pressing Android work to do, some Google Android developers hinted that "the Android team was getting fed up with the operation."

Rafael J. Wysocki, a Linux kernel developer, added a patch in September 2010 that enhanced the mainline Linux wakeup events framework. Android system drivers that use wakelocks, he claims, can now be easily merged into the kernel. The opportunistic suspend features of Android, on the other hand, should not be included in the mainline kernel. "Eventually, Android and Linux will come back to a similar kernel," Linus Torvalds said in 2011, "but it will probably not be for four to five years."

Greg Kroah-Hartman announced the launch of the Android Mainlining Project in December 2011, with the aim of reintegrating some Android drivers, updates, and features into the Linux kernel beginning with Linux 3.3. For Linux Kernel 3.4, further integration is required

CHAPTER - 5

SIMULATION RESULTS

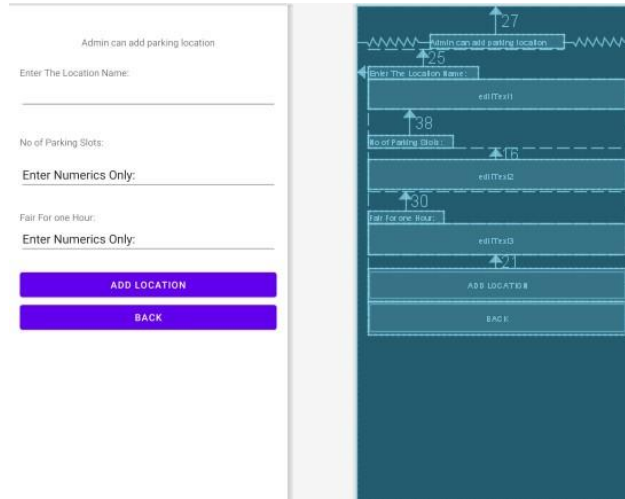


Figure 5.1.Admin add parking slot page

Above image shows admin add parking location, at first space provided, admin will be entering location to be added followed by the slots to be added. Fair will be estimated based on parking hours and will be displayed accordingly



Figure 5.2.Admin home page

Above figure shows admin's home page .All the specifications of the application will be

available in this page, admin can add parking locations ,if admin wants to manage location, can manage in the next block, if admin wants to view all the parking locations available can be able to check in the view locations option.

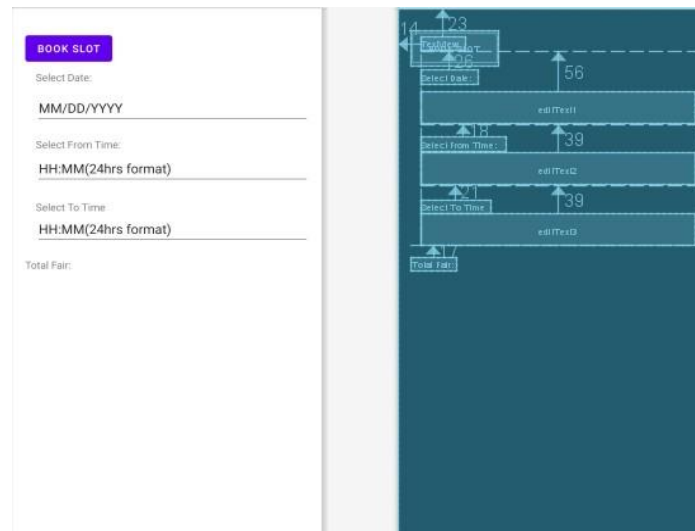


Figure 5.3. Slot booking page

Above figure shows booking slot page, date in format of MM/DD/YYYY of the particular day when the user want to book the slot. Next user need to enter the time for which he wants to park the vehicle. After entering all the required fields immediately, total fair will be displayed on the screen based on parking hours.

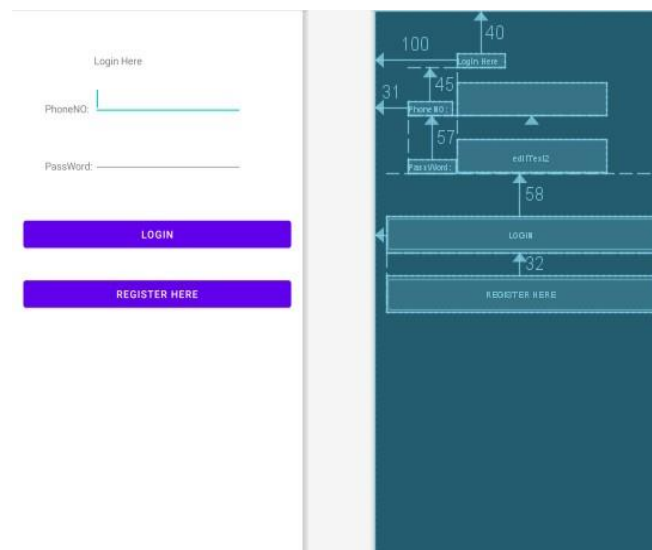


Figure 5.4. Login page

If user has already registered in the app then user need to enter the login details to go through the app and book the slot for parking the vehicle. If user is new to the app, first user need to register in the app by entering the personal mobile number and password to login.

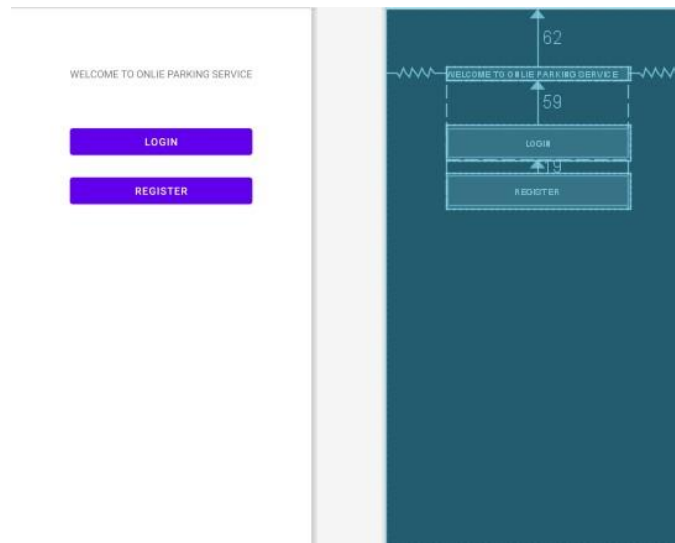


Figure 5.5. Welcome page

As you launch the app, Login page is displayed as shown in the above figure. If the user is an registered user, then he simply needs to login and continue the slot booking, if not, he needs to register in order to book the slot

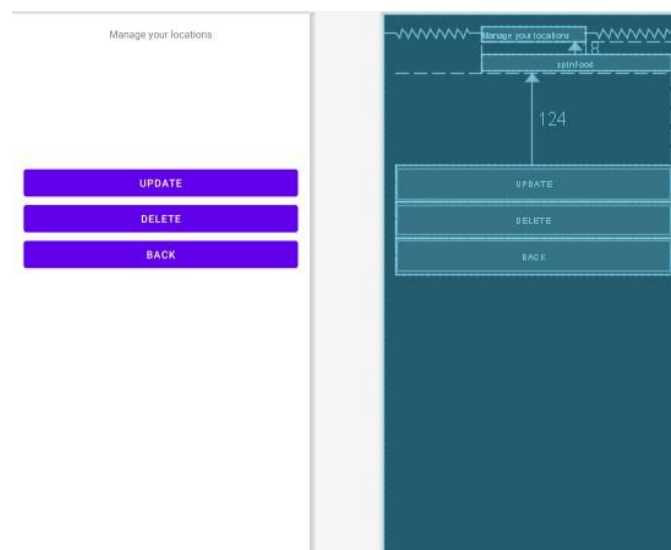


Figure 5.6.Manage your locations

While managing slots, if admin wants to add any slot, then he can add new slots from different locations using update option available and if he want to delete any slot from particular areas, then the admin can delete them using the delete option provided.

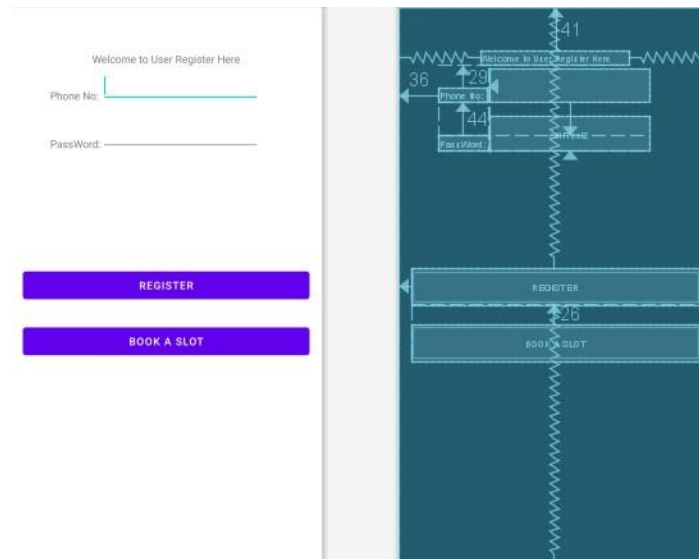


Figure 5.7. Registration page for new user

If admin is new user to the app, need to register in the app in order to login and book the slot. For registration we need to provide user personal mobile number and set the password and click on the register, now you are registered for the app and no need to go back for login again from where user can directly book the slot for parking the vehicle.

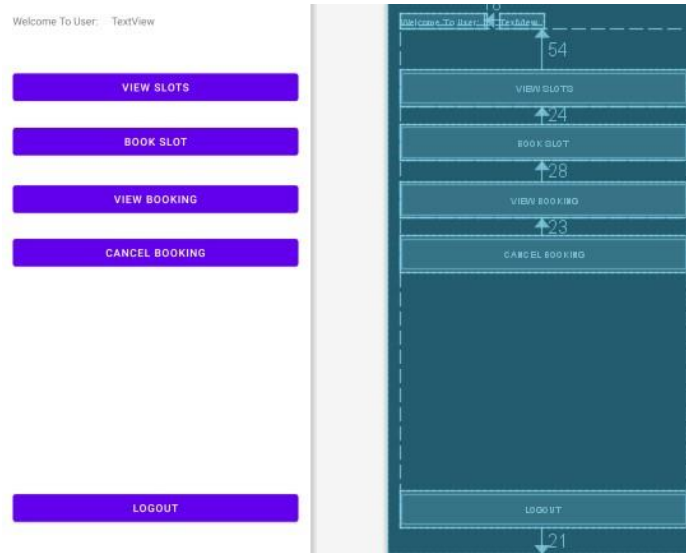


Figure 5.8. Text view

In this Text View Page, there are four options which are view slots, book slots, view booking and cancel booking. In view slots we can see whether slots are available for bookings. In book slots we can book any slot according to our choice and comfort. After booking we can view our booking slot for parking the vehicle and we can cancel the booking slot in necessary conditions within provided time.

CHAPTER – 6

6.1.CONCLUSION

The intellectual Parking System (IPS) allow users to reserve parking places with little effort using an Android smart phone. The consumer will check the condition of the parking slot and reserve a parking spot ahead of time. This will lead to the resolution of several issues that have arisen as a result of poor traffic management. This architecture was created for the Android mobile operating system since mobile computing has proven to be the most active field of study for record and data handing out researchers. This equipment can be used in every nook and cranny due to its ease of use and convenience. In this project the main role will be of cloud because, the data of all the parking slots will be available on the cloud. Cloud Computing is the delivery of computing services such as servers, storage, databases, networking, software, analytics, intelligence, and more, over the Cloud. Cloud Computing provides an alternative to the on-premises datacentre. With an on-premises datacentre, we have to manage everything, such as purchasing and installing hardware, virtualization, installing the operating system, and any other required applications, setting up the network, configuring the firewall, and setting up storage for data. After doing all the set-up, we become responsible for maintaining it through its entire lifecycle. We all know that one of the most common problems today is a saturation of parking spaces. Vehicles continue to outnumber existing parking spaces, thus clogging roads. Incidences of violence over occupancy, deformed cars due to a space crunch, and overcharging for parking are some problems that result. Smart parking development implies an IoT-based system that sends data about free and occupied parking places via web/mobile application. The IoT-device, including sensors and microcontrollers, is located in each parking place.

6.2 FUTURE SCOPE

With the help of existing methodologies and proposed methodology, the future work can be done on automating the app as well. In this proposed methodology, the user has to cancel the booked slots if he is not able reach the destination in specific time. Instead of this, this can be modified as automatic cancelling and thus results in the automation of

the app.

REFERENCES

[1]. Zhao, Y., and Collins, E. G. 2005. "Robust automatic parallel parking in tight spaces via fuzzy logic," *Robotics and Autonomous Systems* (51:2), pp 111-127.

[2]. Yang, L., Rongguo, M., and Longfei, W. Year. "Intelligent Parking Negotiation Based

on Agent Technology," Information Engineering, 2009. ICIE'09. WASE International Conference on, IEEE2009, pp. 265-268

[3]. Xu, J., Chen, G., and Xie, M. Year. "Vision-guided automatic parking for smart car," Proceedings of the IEEE Intelligent Vehicles Symposium2000, pp. 725-730

[4]. Zheng, Y., and Cao, J. Year. "An intelligent car park management system based on wireless sensor networks," 2006 First International Symposium on Pervasive Computing and Applications, IEEE2006, pp. 65-70

[5]. Jung, H. G., Choi, C. G., Kim, D. S., and Yoon, P. J. 2006. "System configuration of intelligent parking assistant system," Proc. 13th World Congr. Intell. Transp. Syst. Services, London, UK, Oct)

[6]. Kianpisheh, A., Mustaffa, N., Limtrairut, P., and Keikhosrokiani, P. 2012. "Smart parking system (SPS)architecture using ultrasonic detector," International Journal of Software Engineering and ItsApplications (6:3), pp 55-58.

[7].Panayappan, R., Trivedi, J. M., Studer, A., and Perrig, A. Year. "VANET-based approach for parking spaceavailability," Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks,ACM2007, pp. 75-76.

[8]. Pham, T. N., Tsai, M.-F., Nguyen, D. B., Dow, C.-R., and Deng, D.-J. 2015. "A Cloud-Based Smart-ParkingSystem Based on Internet-of-Things Technologies," IEEE Access (3), pp 1581-1591.

[9]. Solic, P., Marasovic, I., Stefanizzi, M. L., Patrono, L., and Mainetti, L. Year. "RFID-based efficient methodfor parking slot car detection," Software, Telecommunications and Computer Networks (SoftCOM),2015 23rd International Conference on, IEEE2015, pp. 108

- [10]. Song, I., Gowan, K., Nery, J., Han, H., Sheng, T., Li, H., and Karray, F. Year. "Intelligent parking system design using FPGA," 2006 International Conference on Field Programmable Logic and Applications, IEEE2006, pp. 1-6.
- [11]. Yang, L., Rongguo, M., and Longfei, W. Year. "Intelligent Parking Negotiation Based on Agent Technology," Information Engineering, 2009. ICIE'09. WASE International Conference on, IEEE2009, pp. 265-268.
- [12]. Yeh, H.-T., Chen, B.-C., and Wang, B.-X. 2016. "A City Parking Integration System Combined with CloudComputing Technologies and Smart Mobile Devices," Eurasia Journal of Mathematics, Science & Technology Education (12:5), pp 1231-1242.