A Mini Project

On

Online Car Parking Reservation System

Master of Computer Applications (MCA)

Submitted By

NALLURI DINESH SHARATH

(Regd No: 1223715120)

Under the guidance of

Mr. G. Babu Rao

Assistant Professor

Department of Computer Science

GITAM Institute of Science

GITAM University

Visakhapatnam-530 045, A.P..

2015 – 2018

CERTIFICATE

This is to certify that the Mini Project entitled **"Online Car Parking Reservation System"** is the bonafied work done **by N. DINESH SHARATH,** Reg No. 1223715120 during 2016-2017 in partial fulfillment of the requirement of the 3rd Semester Mini Project of Master of Computer Applications, in GITAM University, Visakhapatnam, under my supervision and guidance.

Signature
(Internal Guide)
Mr. G. Babu Rao

Signature
(Head of the Department)
Dr. V. Nagalakshmi

DECLARATION

I, **Nalluri Dinesh Sharath** hereby declare that the mini project report entitled **"Online Car Parking Reservation System"** submitted by me, in partial fulfillment of the requirement of the 3rd semester Mini Project of MCA course in Computer Science Department, GIS, to GITAM UNIVERSITY, Visakhapatnam is a bonafied mini project work carried out by me under the guidance of **Mr. G. Babu Rao**.

NALLURI DINESH SHARATH

Regd no: 1223715120

ACKNOWLEDGEMENT

I would like to whole heartedly thank and express my sincere gratitude to **Mr. G. Babu Rao** for his constant encouragement in the development of this Mini project. I also wish to thank **Dr. V Naga Lakshmi** HOD, Department of Computer Science, GIS, GITAM UNIVERSITY for her valuable advice and support. Above all, I extend my sincere thanks to all the faculty members of the department, for their continuous encouragement, guidance and support throughout the development of this Mini project.

NALLURI DINESH SHARATH

Regd no: 1223715120

ABSTRACT

Now–a– days in all the cities in India face severe parking problems. Off street parking is common in all Organizations, big shopping malls and theatres and huge offices which are used by employees & customers, on the other hand on-street parking is traffic problem causing as the parking is directly controlled by market forces, with individual parking and hence has high demand.

With this facility the end user will be provided with a wide range of user-friendly parking facilities in all major organizations of various cities. The user can book the parking slot through online. This service is continually provide the highest levels of customer service at every location, ensuring every experience. Achieving this has been one of the key factors in making Secure Parking. With this everyone will have no parking worries.

CONTENTS

INTRODUCTION

1. INTRODUCTION

1.1 Introduction to project:

This Web Application mainly deals with the parking slots in the buildings like clubs, hotels, malls and many more. In this Web Application we can access the information of parking slots in the building where is free. By finding the empty space the user is able to block the slot. This application also provides information about the user like Car No, License No. and mobile number the administration will able to notify the user if there is any problem. After selecting the empty space in the parking slot the user is able to pay the amount and confirm his/her booking.

1.2 Purpose of the system:

The purpose of this project is to ease the process of parking through the web application. In this we mainly concentrate on parking slot in the building and the user is able to block the slot before entering into the building. The web application is designed for client side. It contains only user level.

1.3 Existing System:

We already have many Car Parking management systems which can provide services for the users only with the help of RFID tags or Sensors. In the Existing System, by using the electrical equipment only we can find the empty slot by an indication method. We can only identify the free slot but we can't book the slot before.

Limitations in Existing System:

The following are the disadvantages of the existing system

- It is difficult to maintain the whole database (images of the parking space).
- Users have to spend 1-2 minutes for booking the slot (only registered).

1.4 Proposed System:

Our web application contains a continuous list of free slots, the user can select any of the slots and can able to book the slot. After entering in to the building the user will have an option to see route/path to selected slot only if the database has map. The functionalities of the car parking system can be automated.

Advantages:

- A fast and efficient service to the user regarding to the
- Searching process is an easy task.
- Selecting and booking a slot is also an easy process.
- Saves lot of time while searching a free slot in the building and even may avoid the minor accidents.

1.5 Scope of the project:

A detailed study of the existing system is necessary. The functions of the system, requirements for the users, structure of the current system is made through the system study. The problems faced in the current system are found and solution pertaining to it is done in the system study.

SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

2.1 Study of the system:

We present the proposed Online Car Parking Reservation System, which implements a reservation policy and dynamic pricing scheme.

System Architecture:

There are three components in the system, including parking lots, users and the management system. The management system determines the dynamic parking prices based on real-time parking information, and broadcast live parking prices to users (also drivers). The price reflects the relationship between demand and supply, and implies the congestion level. Upon receiving dynamic parking prices, the user selects a desired parking lot and reserves a space in the parking lot. According to his budget constraint and convenience degree, the parking decision would vary by user. As a result, the state of parking resources is changed by users.

The parking lot consists of a group of parking spaces. The on-street parking can also be considered as a virtual parking lot. The state of a parking lot is the number of occupied spaces versus total spaces. Every parking lot has access to the Internet to communicate with the management system and users, and share parking information with other parking lots. In each parking lot, the reservation authority is deployed for authenticating the individual user's identity and reservation request. In this case, the reservation authority in the parking lot communicates with the specific user individually. Once the reservation order is confirmed, the reservation authority updates reservation information to hold the related space for the user. The sensor system deployed in parking lot is responsible for monitoring the real-time condition of parking lots and delivers the live aggregated sensing information (the number of available spaces or occupancy rate) to the management system. The sensing information is updated on demand.

By placing the reservation authority in individual parking lots, we simplify a lot of issues related to the implementation, including communication overhead, reservation synchronization and load balancing. Since each user only has to communicate with his desired parking lot to make his reservation, rather than the centralized management system, the communication overhead of reservation is highly reduced. Also, since each parking lot manages its own reservation information, it makes the reservation requests from users easily to be synchronized, comparing with reservation synchronization in the management system.

SYSTEM DEVELOPMENT LIFE CYCLE MODEL (SDLC MODEL)

This is also called as Classic Life Cycle Model (or) Linear Sequential Model (or) Waterfall Method. This model has the following activities

1. System/Information Engineering and Modelling
2. Software Requirements Analysis
3. Systems Analysis and Design
4. Code Generation
5. Testing
6. Maintenance

1) System/Information Engineering and Modelling

As software development is a large process, so work begins by establishing requirements for all system elements and then allocating some subset of these requirements to software. The view of this system is necessary when software must interface with other elements such as hardware, people and other resources. System is the very essential requirement for the existence of software in any entity. In some cases for maximum output, the system should be re-engineered and spruced up. Once the ideal system is designed according to requirement, the development team studies the software requirement for the system.

2) Software Requirement Analysis

Software Requirement Analysis is also known as feasibility study. In this requirement analysis phase, the development team visits the customer and studies their system requirement. They examine the need for possible software automation in the given software system. After feasibility study, the development team provides a document that holds the different specific recommendations for the candidate system. It also consists of personal assignment, costs of the system, project schedule and target dates.

The requirements analysis and information gathering process is intensified and focused especially on the software. To understand what type of the programs to be built, the system analyst must study the information domain for the software as well as understand requirement function, behavior, performance and interfacing. The main purpose of the requirement analysis phase is to find the need and to define the problem that needs to be solved.

3) System Analysis and Design

In System Analysis and Design phase, the whole software development process, the overall software structure and its outlay are defined. In case of the client/server processing

technology, the number of tiers required for the package architecture, the database design, the data structure design etc. are all defined in this phase. After designing part a software development model is created. Analysis and Design are very important in the whole development cycle process. Any fault in the design phase could be very expensive to solve in the software development process. In this phase, the logical system of the product is developed.

4) Code Generation

In Code generation phase, the design must be decoded into a machine-readable form. If the design of software product is done in a detailed manner, code generation can be achieved without much complication. For generation of code, programming tools like Compilers, Interpreters, and Debuggers are used. For coding purpose high level programming language php is used. The right programming language is chosen according to the type of application.

5) Testing

After code generation phase the software program testing begins. Different testing methods are available to detect the bugs that were committed during the previous phase. A number of testing tools and methods are available for testing purpose.

6) Maintenance

Software will definitely go through change once when it is delivered to the customer. There are large numbers of reasons for the change. Change could happen due to some unpredicted input values into the system. In addition to this the changes in the system directly have an effect on the software operations. The software should be implemented to accommodate changes that could happen during the post development period.

2.2 Modules of the System:

Admin/ Guard Module: Guard will see reserved slots only for the today / current date. Once any vehicle leave from parking lot, guard can vacate that slot. Once guard will vacate slot, it will become available for new user to book that slot.

User Module: User can make reservation on site after signing up and logging into the site. Once user reserve the parking slot for the desired date. User will get confirmation email from site confirming reservation is complete successfully. Once user book the slot for particular date it will become unavailable for others users. Means no other user can book that slot for that particular date.

Payment Module: The payment module consists of payment gateway which facilitates the users to make the payments for slot confirmation.

REQUIREMENT ANALYSIS

3. REQUIREMENT ANALYSIS

3.1 System Requirement Specification (SRS)

Software Requirement specifications (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easier comprehended. Hence the nedd for the requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase.)

Role of SRS:

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium through which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties involved in the system

Requirement Specifications:

The focus is on specifying what has been found giving analysis such as representation, specification languages and tools and checking the specifications are addressed during this activity. The Requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic goal of this phase.

Hardware Requirements:

- Pentium Processor
- 2 GB Ram
- 50 GB Hard disk

Software Requirements:

- Operating System: Windows 8.1 or above
- Front End: PHP and HTML
- Back End: MySQL

3.2 Overview of the Front end:

PHP

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. PHP is a server-side scripting language designed primarily for web development but it is also used as a general-purpose programming language. PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications. The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as Ade facto standard. Since 2014 work has gone on to create a formal PHP specification. The fact that PHP lacked an original overall design but instead developed organically has led to inconsistent naming of functions and inconsistent ordering of their parameters.[1] In some cases, the function names were chosen to match the lower-level libraries which PHP was "wrapping", while in some very early versions of PHP the length of the function names was used internally as a hash function, so names were chosen to improve the distribution of hash values.

HyperText Markup Language (HTML)

Hyper Text Markup language is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS), and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML can embed programs written in a scripting language such as JavaScript which affect the behavior and content of web pages. Instead of lots of commands to output HTML (as seen in C or Perl), PHP pages contain HTML with embedded code. The PHP code is enclosed in special start and end processing instruction <? php and ?> that allow you to jump into and out of "PHP mode". What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the result of running that script, but would not know what the underlying code was. The best thing in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer. Although PHP's development is focused on server-side scripting, we can do much more with it.

- **Hypertext** refers to the way in which Web pages (HTML documents) are linked together. Thus the link available on a webpage are called Hypertext.
- As its name suggests, HTML is a **Markup Language** which means you use HTML to simply "mark up" a text document with tags that tell a Web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.

Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

3.3 Overview of the Back end:

MYSQL

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality. MySQL is a central component of the LAMP open-source web application software stack.

The MySQL server software itself and the client libraries use dual-licensing distribution. They are offered under GPL version 2, beginning from 28 June 2000 (which in 2009 has been extended with a FLOSS License Exception) or to use a proprietary license.

Support can be obtained from the official manual. Free support additionally is available in different IRC channels and forums. Oracle offers paid support via its MySQL Enterprise products. They differ in the scope of services and in price. Additionally, a number of third party organizations exist to provide support and services, including MariaDBand Percona.

MySQL has received positive reviews, and reviewers noticed it "performs extremely well in the average case". And that the "developer interfaces are there, and the documentation (not to mention feedback in the real world via Web sites and the like) is very, very good".] It has also been tested to be a "fast, stable and true multi-user, multi-threaded sql database server".

FEASIBILITY REPORT

4. FEASIBILITY REPORT

4.1 Technical Feasibility:

Technical feasibility includes the software's and hardware that are needed to develop the system. Software's and hardware have to be chosen according to the client requirements. We have to be very clear about what are the technologies that are to be required for the development of the new system. Find out whether the organization currently processes the required technologies. Is the required technology available with the organization?

4.2 Economical Feasibility:

Economic feasibility attempts to weigh the costs of developing and implementing a new system. To develop the project from top to bottom, the estimated cost will comes under this feasibility. It defines whether the client is able to pay the estimated cost or not. If the client is unable to spend then the software's have to be changed.

4.3 Operational Feasibility:

Proposed system is beneficial only if it will meet the organizations operating requirements. The current business operations are considered. This test of feasibility asks if the system will work when it is developed and installed. Here are questions that will help test the operational feasibility of a project.

Is there sufficient support for the project from management from users? What are the operation that are performing in the existing system? Are the current business methods acceptable to the user?

If they are not, Users may welcome a change that will bring about a more operational and useful systems. Have the user been involved in the planning and development of the project?

DATA FLOW DIAGRAMS

DATA FLOW DIAGRAM

A data flow diagram is a graphical tool used to describe and analyze movement of data through a system. They are the central tool and forms the basis from which the other components are developed. The transformation of data from input to output may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movements of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagram. Each component in a DFD is labelled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD's is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The top level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

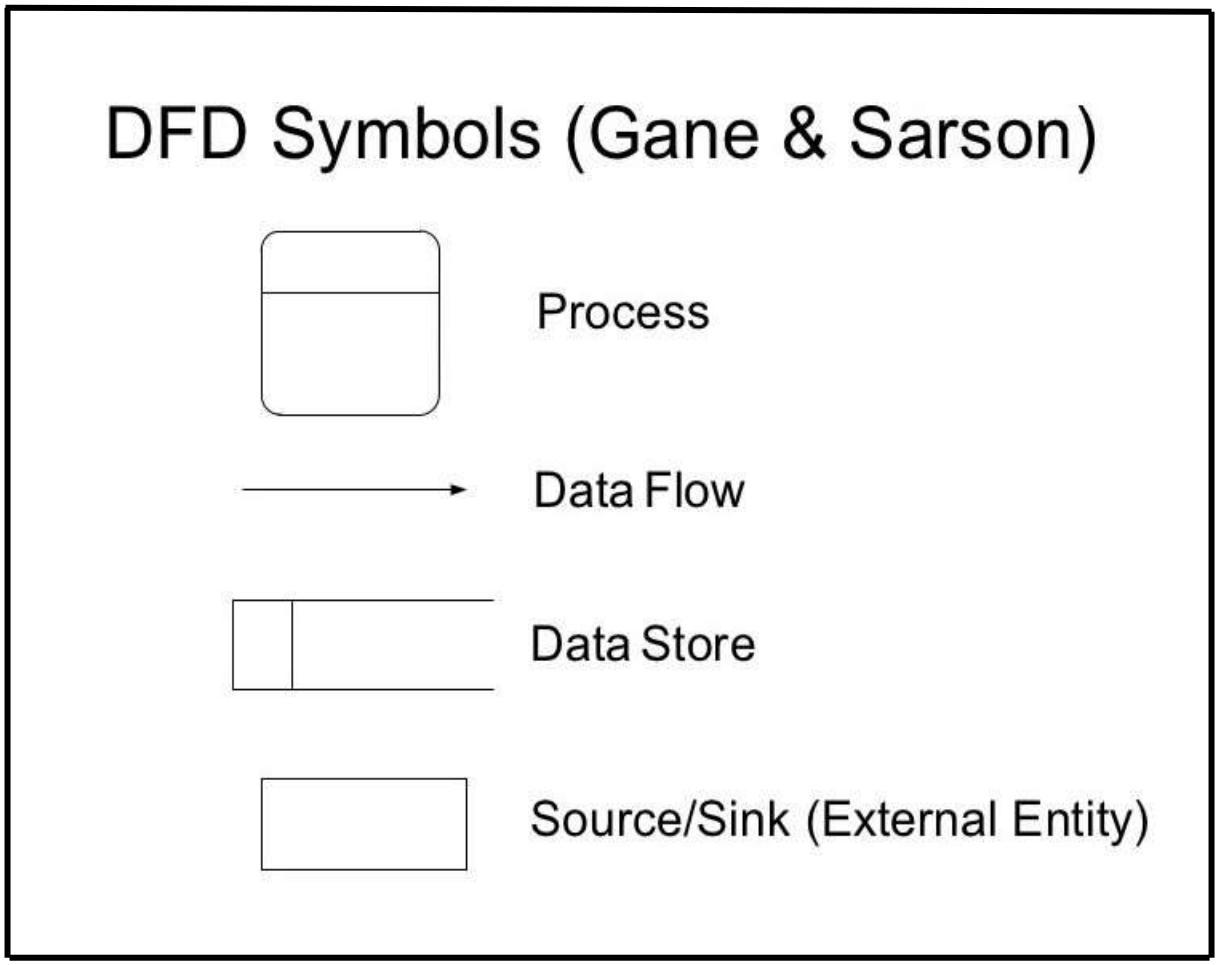
Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical form, this lead to the modular design.

A DFD is also known as a "bubble Chart" has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

DFD SYMBOLS

In the DFD, there are four symbols

1. A square defines a source (originator) or destination of system data.
2. An arrow defines data flow. It is the pipeline through which the information flows.
3. A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data.



CONSTRUCTING A DFD

There are several rules of thumb that are used in drawing DFD's. They are

1. Process should be named and numbered for an easy reference. Each name should be representative of the process.
2. The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD, it is marked with a short diagonal.
3. When a process is exploded into lower level details, they are numbered.

RULES GOVERNING THE DFD's

PROCESS

1. No process can have only outputs.
2. No process can have only inputs. If an object has only inputs then it must be a sink.
3. A process has a verb phrase label.

DATA STORE

1. Data cannot move directly from one data store to another data store, a process must move data.
2. Data cannot move directly from an outside source to a data store. Data must be moved by a process that receives data from the source and places the data into the data store.
3. Data cannot move directly to an outside sink from a data store. Data must be moved by a process.
4. A data store has a noun phrase label.

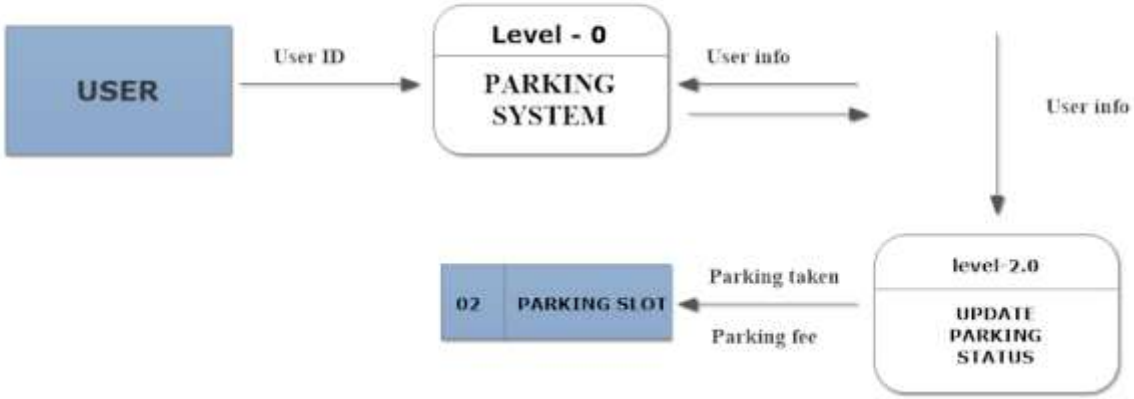
SOURCE OR SINK

1. Data cannot move directly from a source to a sink. It must be moved by a process.
2. A source or sink has a noun phrase label.

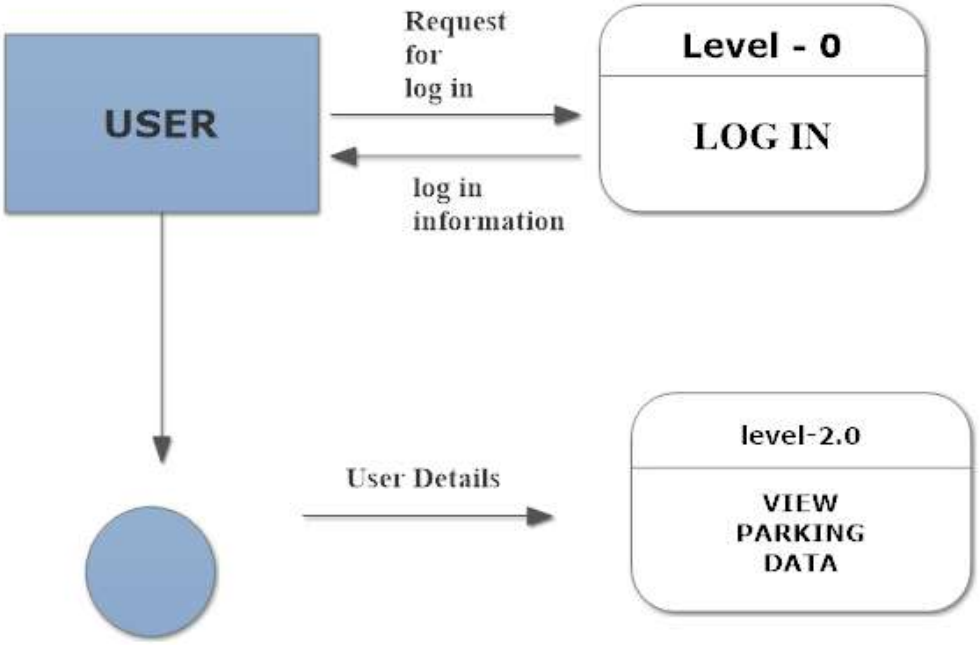
DATA FLOW

1. A data flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update.
2. A fork in a data flow means that exactly the same data goes from a common location to two or more different processes, data stores, or source/sinks.
3. A join in a data flow means that exactly the same data come from any of two or more different processes, data stores, or source/sinks to a common location
4. A data flow cannot go directly back to the same process it leaves.
5. A data flow to a data store means update.
6. A data flow from a data store means retrieve or use.
7. A data flow has a noun phase label.

LEVEL 0 DFD (Context level DFD)



LEVEL-1 DFD (Context level DFD)



LEVEL - 1.0 DFD (USER)

DATA BASE DESIGNS

7. DATABASE DESIGNS

7.1 Introduction

Database Design is a collection of processes that facilitate the designing, development, implementation and maintenance of enterprise data management systems. It helps produce database systems

That meet the requirements of the users

Have high performance.

The main objectives of database designing are to produce logical and physical designs models of the proposed database system. The logical model concentrates on the data requirements and the data to be stored independent of physical considerations. It does not concern itself with how the data will be stored or where it will be stored physically.

The physical design model involves translating the logical design of the database onto physical media using hardware resources and software systems such as database management systems (DBMS).

7.2 Normalized Database Tables

Normalization of data can be considered a process of analyzing the given relation schemas based on their FDs and primary keys to achieve the desirable properties of

1) Minimizing redundancy.
2) Minimizing the insertion, deletion and update anomalies.

FIRST NORMAL FORM

First normal form states that the domain of an attribute must include only atomic (simple, indivisible) values and that the value of any attribute in a tuple must be a single value from the domain of that attribute.

SECOND NORMAL FORM

A relation schema R is in second normal form if every nonprime attribute in R is fully functionally dependent on the primary key of R.

A functional dependency X->Y is a full functional dependency if removal of any attribute A from X means that the dependency does not hold any more.

THIRD NORMAL FORM

A relation schema R is in third normal form if it satisfies second normal form and no nonprime attribute of R is transitively dependent on the primary key.

BOYCE-CODD NORMAL FORM

A relation schema R is in BCNF if whenever a non-trivial functional dependency X->A holds in R, then X is a super key of R.

FOURTH NORMAL FORM

A relation schema R is in fourth normal form with respect to a set of dependencies F if, for every nontrivial multivalued dependency X->->Y in F^+, X is a super key for R.

FIFTH NORMAL FORM

A relation schema R is in fifth normal form with respect to a set F of functional, multivalued, and join dependencies if, for every nontrivial join dependency JD(R_1,$R_{2,\ldots\ldots,}$$R_{n)}$ in F^+, every R_i is a super key of R.

DATABASE TABLES

Table Name : Register

Function: Displays user details.

Fields	Data Type
Mobile No	Number
Name	Varchar2
Email	Varchar2
License	Varchar2
Car No	Varchar2

Table Name: Booked Slot Table

Function: Display cars information after booking the slot

Fields	Data Type
Car No	Varchar2
Duration	Varchar2
Floor No	Varchar2
Slot No	Varchar2

Registration Table:

Mobile No	Name	Email	License	Car No
9963045007	Madhu	madhu06@gmail.com	LASD12237151	AP AB-156-CD
9160262550	Dinesh	nalluri06@gmail.com	LASD15524585	AP AB-524-CD
9493877487	Teja	teja007@gmail.com	LASD25455556	AP AB-558-CD

Booked Slots Table:

Car No	Duration	Floor No	Slot No
AP AB-456-CD	0-5 hrs	F1	S1
AP AB-524-CD	1-2 hrs	F4	S12
AP AB-558-CD	0-30 mins	F2	S2

SAMPLE SCREEN SHOTS

Log in Page:


ONLINE CAR PARKING RESERVATION SYSTEM
Log In
Safe Parking, wherever you are.

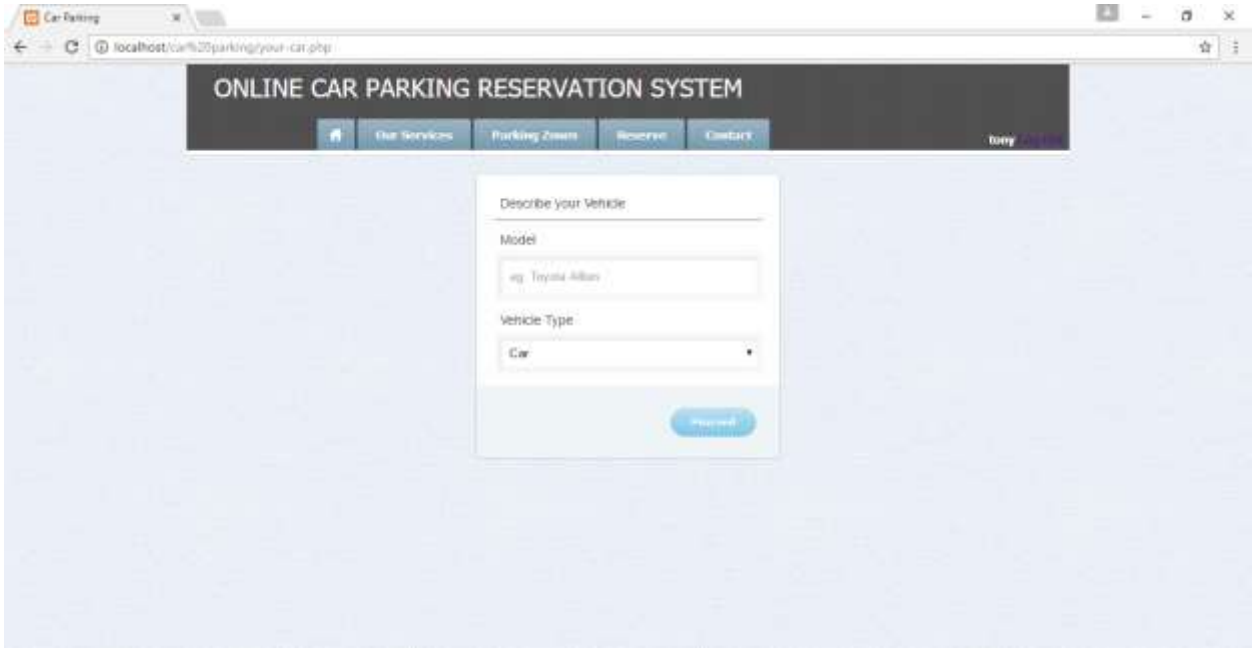
Contact us:


Car Parking
localhost/car%20parking/contact.php
ONLINE CAR PARKING RESERVATION SYSTEM
Our Services
Parking Zones
Reserve
Contact
We are at your service. Call our support team any time as from 7.00am - 10.00pm
Call +91-9160262550
or e-mails us : sallur06@gmail.com
Safe Parking, wherever you are.

Sign up :

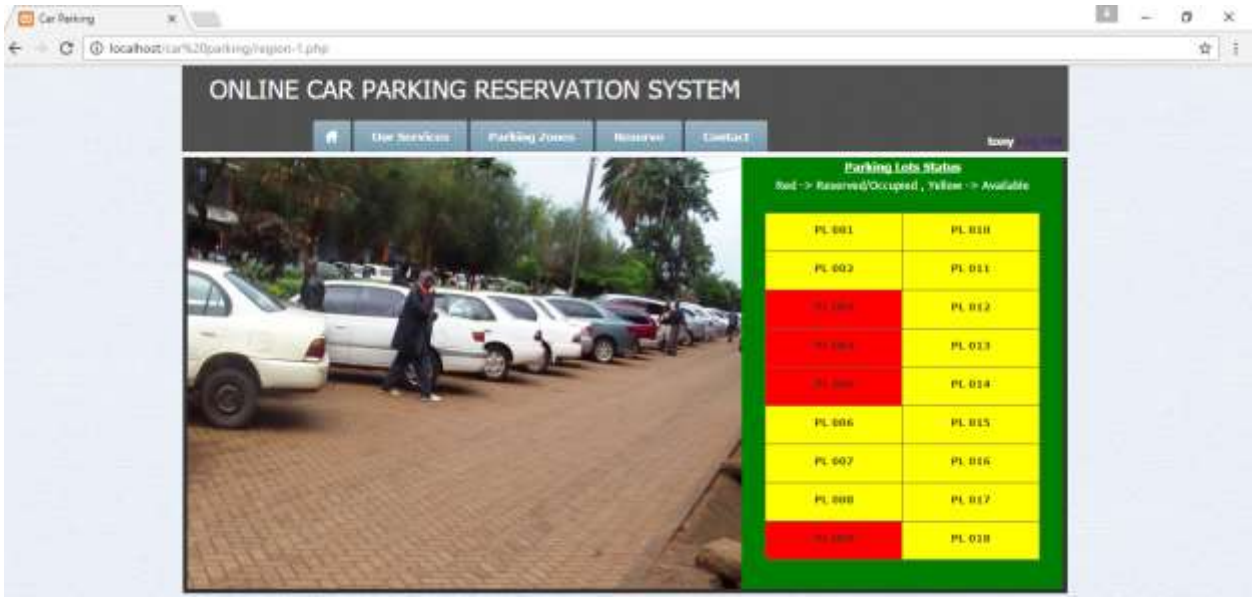
Log in Success:

Car Details:

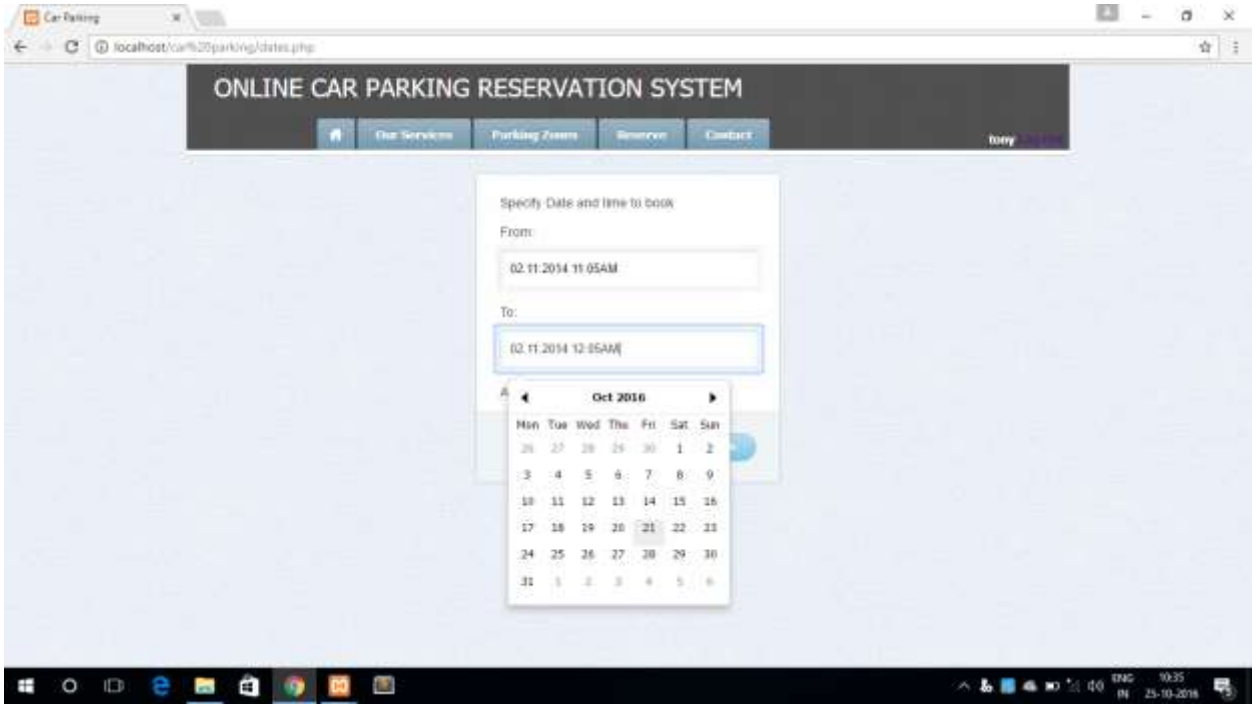

ONLINE CAR PARKING RESERVATION SYSTEM
Describe your Vehicle
Model
Vehicle Type
Car

Slot Selection in Area-1

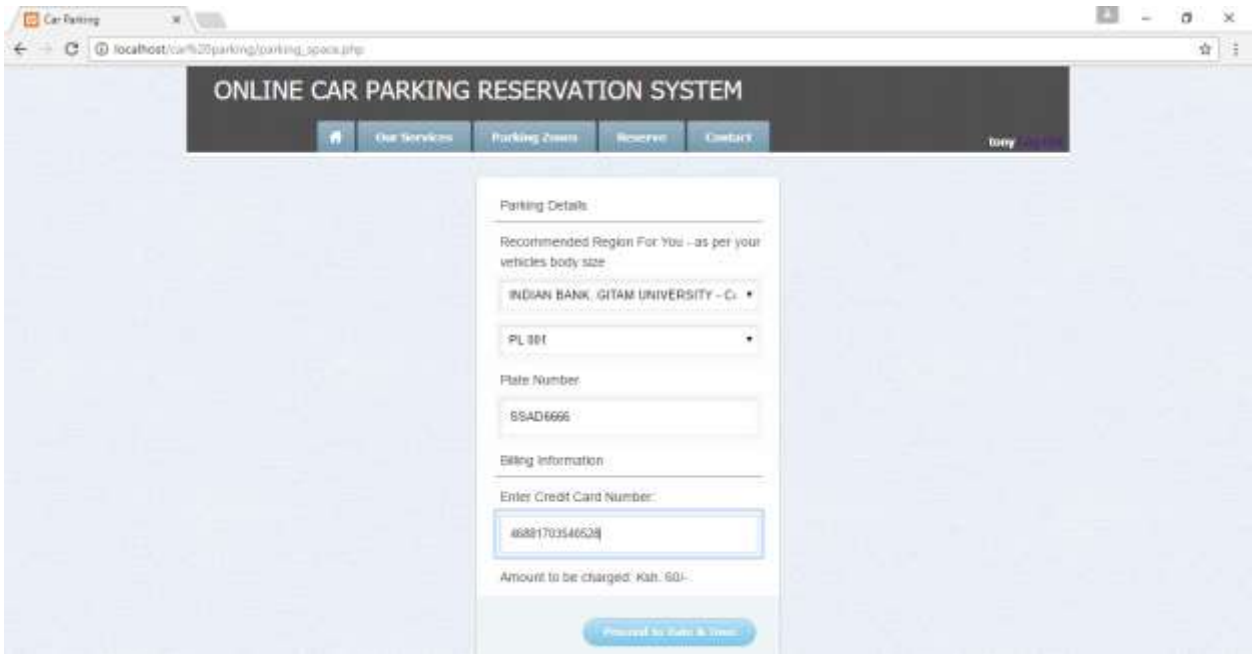
Slot Selection in Area-2



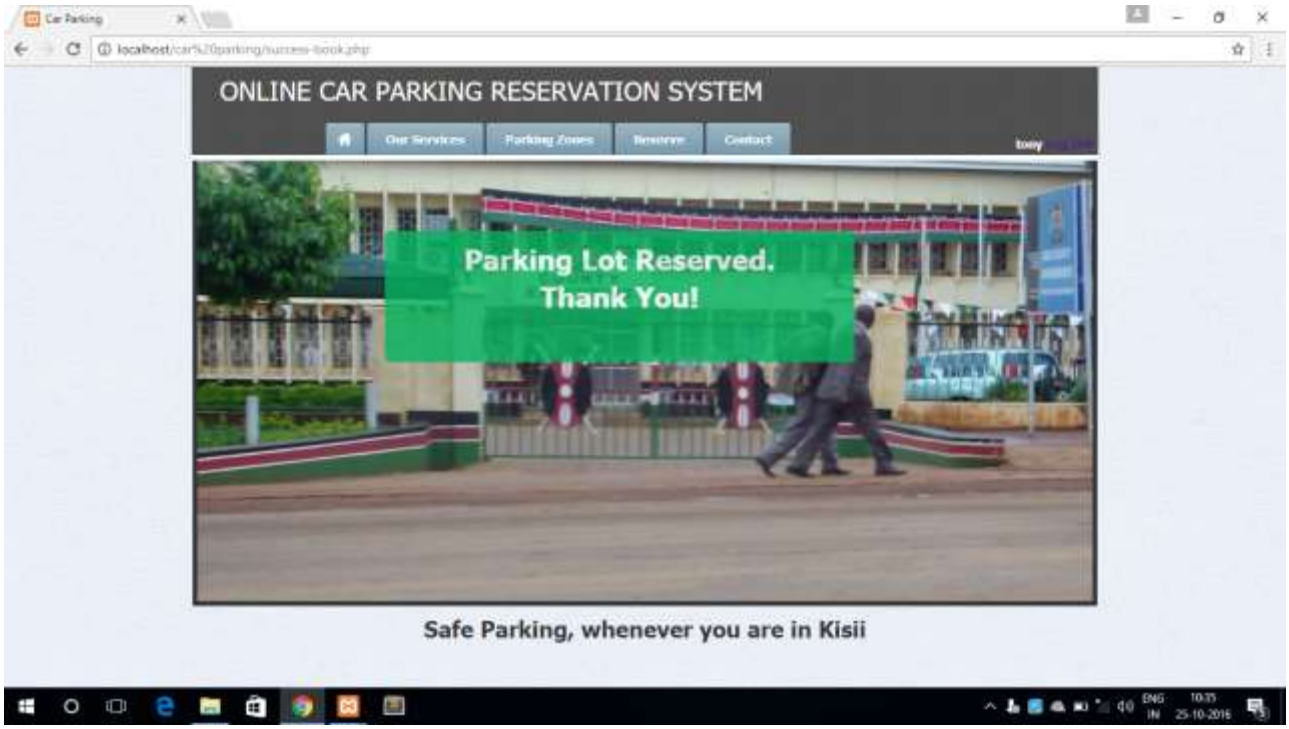
Selecting Time and Date:



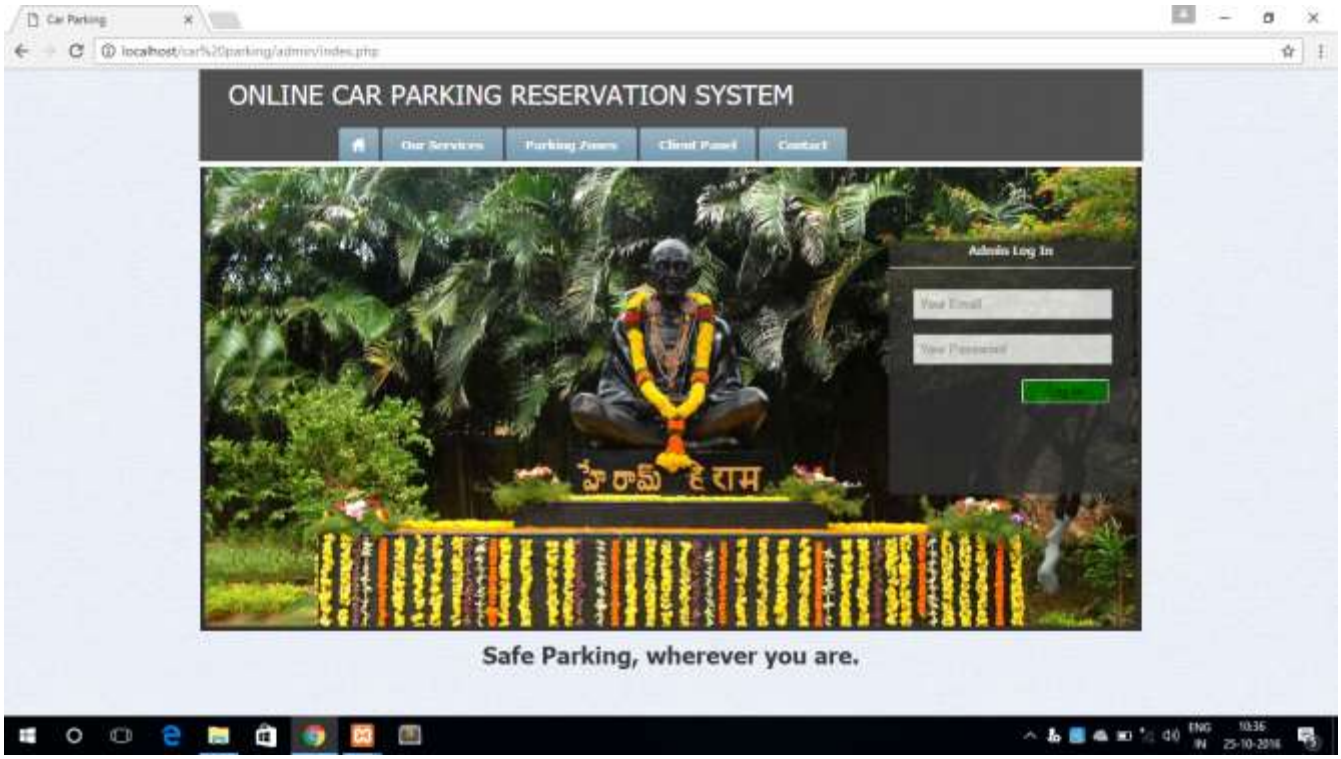
Payment:


ONLINE CAR PARKING RESERVATION SYSTEM
Parking Details
Recommended Region For You - as per your vehicles body size
INDIAN BANK, GITAM UNIVERSITY - C
PL 9H
Plate Number
SSAD6666
Billing Information
Enter Credit Card Number:
Amount to be charged: Ksh. 60/-

Slot Reserved Success:



Admin Login:


ONLINE CAR PARKING RESERVATION SYSTEM
Admin Log In
Safe Parking, wherever you are.

CODING

INDEX PAGE

```
<?php
session_start();
if( isset( $_SESSION['email'] ) )
    {
        header("location: logged-index.php");
        exit;
 }
  ?>
<?php include 'main.php';?>
<!DOCTYPE HTML>
<html>
<head>
<link rel="stylesheet" type="text/css" href="inc/css/structure.css">
<link rel="stylesheet" type="text/css" href="inc/css/mystyles.css">
</head>
<body>
<section class="HomeKisii">
<img src="img/gitam1.jpg" alt="Mountain View" style="width:954px;height:470px">
<section class="account">
 <h1>Log In</h1>
```

```
<hr>

 <form action="process-log-in.php" method="post">

   <input type="text" id="FullName" name="email"placeholder="Your Email" maxlength="31" required />

        <input type="password" id="FullName" name="password" placeholder="Your Password" maxlength="10" required />

        <input type="submit" id="LogIn" value="Log In">

        </form>

        <p><a href="signup.php" class="Link">Or Sign Up</a></p>

</section>

</section>

<section class="kisii-bottom">

<p>Safe Parking, wherever you are.</p>

</section>

</body>

</html>
```

SIGN UP

```
<?php include 'main.php';?>

<!DOCTYPE HTML>

<html>

<head>

<link rel="stylesheet" type="text/css" href="inc/css/structure.css">

<link rel="stylesheet" type="text/css" href="css/mystyles.css">

</head>

<body>
```

```
<section class="HomeKisii">

<img src="img/gitam2.jpg" alt="Mountain View" style="width:954px;height:470px">

<section class="account">

 <h1>Sign Up</h1>

<hr>

 <form action="process-sign-up.php" method="post">

        <input type="text" id="FullName" name="name" placeholder="Your Name" maxlength="31" required />

    <input type="text" id="FullName" name="email"placeholder="Your Email" maxlength="31" required />

        <input type="password" id="FullName" name="password" placeholder="Your Password" maxlength="10" required />


        <input type="text" name="phone" id="FullName"required pattern="(\+?\d[- .]*){7,13}" title="phone number" maxlength="10" placeholder="Your Phone Number" />

        <input type="submit" id="LogIn" value="Sign Up">

        </form>

        <p><a href="index.php" class="Link">Or Log In</a></p>

</section>

</section>

<section class="kisii-bottom">

<p>Safe Parking, whenever you are in Kisii</p>

</section>

</body>

</html>
```

PARKING SQL

```
-- phpMyAdmin SQL Dump
-- version 4.2.7.1
-- http://www.phpmyadmin.net
--
-- Host: 127.0.0.1
-- Generation Time: Apr 01, 2015 at 04:47 PM
-- Server version: 5.5.39
-- PHP Version: 5.4.31

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS
*/;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Database: `parking`
--

-- --------------------------------------------------------
```

```
--
-- Table structure for table `transactions`
--

CREATE TABLE IF NOT EXISTS `transactions` (
  `street` text NOT NULL,
  `plot` text NOT NULL,
  `status` text NOT NULL,
  `model` text NOT NULL,
  `vehicle` text NOT NULL,
  `platenumber` text NOT NULL,
  `email` text NOT NULL,
  `account` text NOT NULL,
  `d1` text NOT NULL,
  `d2` text NOT NULL,
  `charge` text NOT NULL,
`id` int(5) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;

--
-- Dumping data for table `transactions`
--

INSERT INTO `transactions` (`street`, `plot`, `status`, `model`, `vehicle`, `platenumber`, `email`,
`account`, `d1`, `d2`, `charge`, `id`) VALUES
```

('OGEMBO STREET', 'PL 002', 'RESERVED', 'MAZDA', 'volvo', 'KAB',
'vinnymosh@gmail.com', '40204304', '02.11.2014 11:05AM', '02.11.2014 12:05AM', '60', 1);

-- --

--

-- Table structure for table `users`

--

CREATE TABLE IF NOT EXISTS `users` (

`name` text NOT NULL,

`email` text NOT NULL,

`password` text NOT NULL,

`phone` text NOT NULL,

`level` int(11) NOT NULL,

`status` text NOT NULL,

`joindate` text NOT NULL,

`id` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=15 ;

--

-- Dumping data for table `users`

--

INSERT INTO `users` (`name`, `email`, `password`, `phone`, `level`, `status`, `joindate`, `id`)
VALUES

```
('ibrahim', 'ibrahimond75@gmail.com', 'ondabu', '', 0, '', '', 2),
('vinny', 'vinny@yahoo.com', '9988', '0724229077', 0, 'Active', 'Array', 4),
('ibrahim maina', 'ibrahimond75@yahoo.com', 'ondabu', '0729667794', 1, '', '', 5),
('antony', 'bitmay2012@gmail.com', '14members', '0723136090', 0, 'Active', 'Array', 6),
('ruth', 'ruth@gmail.com', 'ruth', '0729667794', 0, 'Active', 'Array', 7),
('victor', 'victor@gmail.com', 'ogesi', '0704350482', 0, 'Active', 'Array', 8),
('onchaga', 'onchaga@gmail.com', 'onchaga', '0719788340', 0, 'Active', 'Array', 9),
('esther', 'esther@yahoo.com', 'kash', '0729667794', 0, 'Active', 'Array', 10),
('victor', 'ogesi@gmail.com', 'ondabu', '0729667794', 0, 'Active', 'Array', 11),
('ASHA', 'aliasha94@yahoo.com', 'Mymdumba1', '0713009058', 0, 'Active', 'Array', 13),
('kashmir', 'kashmir@gmail.com', 'kash', '0729667794', 0, 'Active', 'Array', 14);

-- --------------------------------------------------------

--
-- Table structure for table `zones`
--

CREATE TABLE IF NOT EXISTS `zones` (
  `street` text NOT NULL,
  `plot` text NOT NULL,
  `status` text NOT NULL,
  `model` text NOT NULL,
  `vehicle` text NOT NULL,
  `platenumber` text NOT NULL,
```

```
  `email`  text NOT NULL,
  `account`  text NOT NULL,
  `d1` text NOT NULL,
  `d2` text NOT NULL,
  `charge`  text NOT NULL,
`id` int(5)  NOT NULL
) ENGINE=InnoDB   DEFAULT CHARSET=latin1  AUTO_INCREMENT=19 ;


--
-- Dumping  data for table  `zones`
--


INSERT INTO `zones` (`street`, `plot`, `status`, `model`, `vehicle`,  `platenumber`,  `email`,
`account`,  `d1`, `d2`, `charge`,  `id`) VALUES
('DARAJA MBILI/UHURU  PLAZA', 'PL 001', 'RESERVED',  'TOYOTA', 'volvo',  'BMW
600H', 'ibrahimond75@gmail.com',  '7777777', '02.11.2014 11:05AM',  '02.11.2014 12:05AM',
'60', 3),
('DARAJA MBILI/UHURU  PLAZA', 'PL 003', 'RESERVED',  'toyota nissan',  'volvo',  'kbz 220k',
'bitmay2012@gmail.com',  '6666666', '02.11.2014 11:05AM',  '02.11.2014 12:05AM',  '60', 4),
('OGEMBO  STREET',  'PL 005', 'RESERVED',  'toyota',  'volvo',  'KCA  899', 'victor@gmail.com',
'78889998888844', '02.11.2014 11:05AM',  '02.11.2014 12:05AM',  '60', 5),
('OGEMBO  STREET',  'PL 003', 'RESERVED',  'toyota',  'volvo',  'KCA  899',
'cmaubi.cm@gmail.cm',  '55889999999999', '02.11.2014 11:05AM',  '02.11.2014 12:05AM',  '60',
7),
('OGEMBO  STREET',  'PL 009', 'RESERVED',  'toyota',  'volvo',  'KCA  899',
'ibrahimond75@yahoo.com',  '23456789', '02.11.2014 11:05AM',  '02.11.2014 12:05AM',  '60', 8),
('NATIONAL BANK/LEVEL  FIVE STREET',  'PL 001', 'RESERVED',  'premier',  'volvo',  'kca
700j', 'ogesi@gmail.com',  '19999999999999999999999999999999999999999999999', '02.11.2014
11:05AM',  '02.11.2014 12:05AM',  '60', 9),
```

('OGEMBO STREET', 'PL 001', 'RESERVED', 'premier', 'volvo', 'kca 700j', 'ogesi@gmail.com', '19999999999999999999999999999999999999999999', '02.11.2014 11:05AM', '02.11.2014 12:05AM', '60', 10),

('OGEMBO STREET', 'PL 001', 'RESERVED', 'premier', 'volvo', 'kca 700j', 'ogesi@yahoo.com', '77777777777777', '02.11.2014 11:05AM', '02.11.2014 12:05AM', '60', 11),

('MAIN BUS/MATATU STAGE', 'PL 001', 'RESERVED', 'premier', 'volvo', 'kca 700j', 'ibrahimond75@gmail.com', '77777777777777', '02.11.2014 11:05AM', '02.11.2014 12:05AM', '60', 12),

('OGEMBO STREET', 'PL 004', 'RESERVED', 'lexus', 'volvo', 'kca 700j', 'aliasha94@yahoo.com', '785685789', '02.11.2014 11:05AM', '02.11.2014 12:05AM', '60', 13),

('AGAKHAN STREET', 'PL 001', 'RESERVED', 'premier', 'volvo', 'kca 700j', 'ibrahimond75@gmail.com', '785685789', '02.11.2014 11:05AM', '02.11.2014 12:05AM', '60', 14),

('OGEMBO STREET', 'PL 001', 'RESERVED', 'premier', 'volvo', 'kca 700j', 'ibrahimond75@gmail.com', '785685789', '02.11.2014 11:05AM', '02.11.2014 12:05AM', '60', 15),

('AGAKHAN STREET', 'PL 002', 'RESERVED', 'premier', 'volvo', 'kca 700j', 'ibrahimond75@gmail.com', '785685789', '02.11.2014 11:05AM', '02.11.2014 12:05AM', '60', 16),

('OGEMBO STREET', 'PL 001', 'RESERVED', 'premier', 'volvo', 'KCC 800H', 'kashmir@gmail.com', '8889999444444', '02.11.2014 11:05AM', '02.11.2014 12:05AM', '60', 17),

('OGEMBO STREET', 'PL 001', 'RESERVED', 'premier', 'volvo', 'KCC 800H', 'ibrahimond75@gmail.com', '77886766666556', '02.11.2014 11:05AM', '02.11.2014 12:05AM', '60', 18);

--

-- Indexes for dumped tables

--

--

```
-- Indexes for table `transactions`
--
ALTER TABLE `transactions`
 ADD UNIQUE KEY `id` (`id`);


--
-- Indexes for table `users`
--
ALTER TABLE `users`
 ADD PRIMARY KEY (`id`);


--
-- Indexes for table `zones`
--
ALTER TABLE `zones`
 ADD UNIQUE KEY `id` (`id`);


--
-- AUTO_INCREMENT for dumped tables
--


--
-- AUTO_INCREMENT for table `transactions`
--
ALTER TABLE `transactions`
```

```
MODIFY `id` int(5) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=2;
--
-- AUTO_INCREMENT for table `users`
--
ALTER TABLE `users`
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=15;
--
-- AUTO_INCREMENT for table `zones`
--
ALTER TABLE `zones`
MODIFY `id` int(5) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=19;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Process login

```
<?php

require('inc\dbcon.php');

// username and password sent from form
$email=$_POST['email'];
$password=$_POST['password'];

// To protect MySQL injection (more detail about MySQL injection)
```

```
$email  = stripslashes($email);
$password = stripslashes($password);
$email  = mysql_real_escape_string($email);
$password = mysql_real_escape_string($password);

$sql="SELECT  * FROM users WHERE email='$email'  and password='$password'";
$result=mysql_query($sql);

// Mysql_num_row  is counting  table row
$count=mysql_num_rows($result);

// If result matched  $myusername  and $mypassword,  table row must be 1 row

if($count==1){
//Name variable
while($row  = mysql_fetch_assoc($result))
{
  $name  = $row["name"];
}
// Register  $myusername,  $mypassword  and redirect to file  "login_success.php"
// Start the session
session_start();
// Set session  variables
$_SESSION["email"]  = $email;
$_SESSION["password"]  = $password;
```

```
$_SESSION["name"]  = $name;
//echo "Session  variables  are set.";
header("location:success-log-in.php");
}
else {
echo "Wrong  Username  or Password";
}
?>
```

TESTING

10. TESTING

TESTING

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. Testing is the exposure of the system to trail input to see whether it produces correct output.

Testing Phases:-

Software testing includes the following:

- Test activities are determined and test data selected.
- The test is conducted and test results are compared with the expected results.

The testing phase is an important part of software development. It is the computerized system will help in automate process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied.

Software testing is carried out in three steps:

UNIT TESTING:

The first includes unit testing, where in each module is tested to provide its correctness, validity and also determine any missing operations and to verify whether the objectives have been met. Errors are noted down and corrected immediately. Unit testing is the important and major part of the project. So errors are rectified easily in particular module and program clarity is increased. In this project entire system is divided into several modules and is developed individually. So unit testing is conducted to individual modules.

INTEGRATION TESTING:

The second step includes Integration testing. it need not be the ease, the software whose modules when run individually and showing perfect results, will also show perfect results when run as a whole. The individual modules are clipped under this major module and tested again and verified the results. This is due to poor interfacing, which may results in data being lost across an interface. A module can have inadvertent, adverse effect on any other or on the global data structures, causing serious problems

VALIDATION TESTING:

The final step involves validation and testing which determines which the software functions as the user elected. Here also some modifications were. In the completion of the protect it is satisfied fully by the end user.

ACCEP TANCE TESTING:

It is performed with realistic data of the client to demonstrate that the software is working, satisfactorily.

SYSTEM TESTING:

It is mainly used if the software meets its requirements. The reference document for this process is the requirement document.

TESTING FUNDAMENTALS

Testing is a process of executing program with the intent of finding error. A good test case is one that has high probability of finding an undiscovered error. If testing is conducted successfully it uncovers the errors in the software. Testing cannot show the absence of defects, it can only show that software defects present

TEST CASES

A Test case in software engineering is a set of conditions or variables under which a tester will determine whether an application or software system is working correctly or not

Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

A test case is a detailed procedure that fully tests a feature or an aspect of a feature. Whereas the test plan describes what to test, a test case describes how to perform a particular test

Unit Testing methods, the software engineer can drive test cases that

- Guarantee that logical decisions on their true and false sides.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structure to assure their validity.

Test Case	Expected Result	Actual Result	Result(Pass/Fail)
Main Page	Go to BookSlot	Redirects to BookSlot	Pass
View Slot Page	Go to ViewSlot	Going to ViewSlot	Pass
LeaveSlot	On clicking, go to LeaveSlot page	Going to LeaveSlot Page	Pass
Application	Data saved in database	Data is saving in Database	Pass

CONCLUSION

From this concept I conclude that these application will help many people from finding the empty space without any collision or minor accident. This application is easily accessible and easy to use. Hence this is the best alternative to Car Parking Management System.

FURTHER ENHANCEMENTS

- Based on the various parameters and properties files everything from the look and feel to the functionalities can be customized. Thus this project is developed from the beginning with reuse in mind and implicitly uses several design patterns.
- The features provided by use for more interactive enhancement of the screens and inclusion of more data.

BIBLIOGRAPHY

- Modern System Analysis and Design: by Jeffery A.Hoffer, 5th Edition.
- Sofrware Engineering, Practitioner's Approach-Roger S.pressman, 6th Edition.
- Fundamentals of Data base Systems Concepts: by Shamkant B.Navathe
- Web Database Applications with PHP and MySQL: by Hugh E. Williams & David lane
- www.w3shools.com
- www.programiz.com