Local Service Search Engine Management System (LSSEMS)

Kaushik Mishra, Aditya Sharma, Mohak Gund

Computer Science & Engineering Department, Uttaranchal University, Dehradun, Uttarakhand, India

ABSTRACT

Local Services Search Engine Management System (LSSEMS) is a web based application which helps user to find serviceman in a local area such as maid, tuition teacher, plumber etc. LSSEMS contain data of serviceman (maid, tuition teacher, plumber etc.). The main purpose of LSSEMS is to systematically record, store and update the serviceman records.

How to cite this paper: Kaushik Mishra | Aditya Sharma | Mohak Gund "Local Service Search Engine Management System (LSSEMS)" Published in International Journal of Trend in Scientific Research and Development (Ijtsrd), ISSN: 2456-6470, Special Issue | International Conference on Advances in Engineering, Science and Technology – 2021, May 2021, pp.29-41, URL: www.ijtsrd.com/papers/ijtsrd42462.pdf

Copyright © 2021 by author(s) and International Journal of Trend in Scientific Research and Development Journal. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0) (http://creativecommons.org/licenses/by/4.0)

1. INTRODUCTION

LSSEMS is a web based application which provides technology based platform to users to take care of their daily needs. This application manages all critical minor concern. It can help user to get the serviceman of locality at doorstep. It is an web application which serving as a platform for users and service providers to interact each other about delivering the desired service. In this project there are two modules i.e. admin and user.

1.1. Admin

1. Admin Setting: In this section, admin can update his/her profile, Change password and logout.
2. Dashboard: In this section, admin can briefly view total number of category and total number of person.
3. Service Category: In this section, admin can manage category (Add/Update).
4. Person List: In this section, admin can manage person (Add/Update).
5. Pages: In this section admin can manage about us and contact us pages.

Admin can update his profile, change password and recover password.

1.2. User

1. Home Page: User can visit home page and view category wise serviceman details.

2. Categories: User can view category wise serviceman details.
3. About Us: User sees the details of website administrator.
4. Contact Us: User can contact with website administrator.

1.3. Purpose:-

The main purpose of LSSEMS to solve the problem of users who search serviceman in their own locality by providing a platform for users and serviceman (maid, tuition teacher, plumber etc.).

1.4. Scope:-

This web application solve all household problems by providing a platform for Users and service providers to interact each other about delivering the desired service.

2. Requirement Specification

2.1. Hardware Configuration:

2.1.1. Client Side:
RAM 512 MB
Hard disk 10 GB
Processor 1.0 GHz

2.1.2. Server side:
RAM 1 GB
Hard disk 20 GB
Processor 2.0 GHz
2.2. Software Requirement:

2.2.1. Client Side:

<table>
<thead>
<tr>
<th>Web Browser</th>
<th>Google Chrome or any compatible browser</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Windows or any equivalent OS</td>
</tr>
</tbody>
</table>

2.2.2. Server Side:

<table>
<thead>
<tr>
<th>Web Server</th>
<th>APACHE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server side Language</td>
<td>PHP5.6 or above version</td>
</tr>
<tr>
<td>Database Server</td>
<td>MYSQL</td>
</tr>
<tr>
<td>Web Browser</td>
<td>Google Chrome or any compatible browser</td>
</tr>
<tr>
<td>Operating System</td>
<td>Windows or any equivalent OS</td>
</tr>
</tbody>
</table>

**APACHE**
The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards.

The Apache HTTP Server ("http") was launched in 1995 and it has been the most popular web server on the Internet since April 1996. It has celebrated its 20th birthday as a project in February 2015.

**PHP**
PHP stands for PHP: Hypertext Preprocessor:
- PHP is a server-side scripting language, like ASP.
- PHP scripts are executed on the server.
- PHP supports many databases (MYSQL, Informix, Oracle, Sybase, Solid, Generic ODBC, etc.).
- PHP is open source software.
- PHP is free to download and use.

**MYSQL**
- MYSQL is a database server
- MYSQL is ideal for both small and large applications
- MYSQL supports standard SQL
- MYSQL compiles on a number of platforms
- MYSQL is free to download and use

How to access MySQL: http://localhost/phpmyadmin

3. Analysis and Design

3.1. Analysis:
In present it is too difficult to search serviceman in local area. We can't get serviceman without any personal contact and searching them is waste of time and very time consuming. This application solves this issues at one single click.

Disadvantage of present system:
- **Not user friendly**: The present system not user friendly because data is not stored in structure and proper format.
- **Manual Control**: All report calculation is done manually so there is a chance of error.
- **Lots of paper work**: Person record maintain in the register so lots of paper require storing details.
- **Time consuming**

3.2. Design Introduction:
Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system.

Design is the only way to accurately translate the customer's requirements into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

**UML Diagrams:**

**Actor:**
A coherent set of roles that users of use cases play when interacting with the use case.

Use case: A description of sequence of actions, including variants, that a system performs that yields an observable result of value of an actor.

UML stands for Unified Modeling Language. UML is a language for specifying, visualizing and documenting the system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built. The representation of the entities that are to be used in the product being developed need to be designed.

3.3. **USECASE DIAGRAMS:**

Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what's called an actor.

Use case diagram can be useful for getting an overall view of the system and clarifying who can do and more importantly what they can't do.

Use case diagram consists of use cases and actors and shows the interaction between the use case and actors.
- The purpose is to show the interactions between the use case and actor.
- To represent the system requirements from user's perspective.
- An actor could be the end-user of the system or an external system.
3.4. USECASE DIAGRAM:
A Use case is a description of set of sequence of actions. Graphically it is rendered as an ellipse with solid line including only its name. Use case diagram is a behavioral diagram that shows a set of use cases and actors and their relationship. It is an association between the use cases and actors. An actor represents a real-world object. Primary Actor – Sender, Secondary Actor Receiver.

Use Case Diagrams:

![Use Case Diagrams](image)

**Admin**
- Admin Setting
- Update Profile
- Change Password
- Dashboard
- Add Category
- Manage Category (Update Details)
- Person List (Add Person)
- Manage Person (Update Detail)
- Pages
- About Us
- Contact Us
- Password Recovery

Fig 3.1 Use case Diagram (Admin)

**User**
- View Home Page
- View Categories
- View Service Provider details
- View About Us
- View Contact Us

Fig 3.2 Use case Diagram (User)

3.5. Class Diagram:
A description of set of objects that share the same attributes operations, relationships, and semantics.
3.6. ER Diagram:
The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design for the database designer, the utility of the ER model is:
- It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.
- It is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user.
- In addition, the model can be used as a design plan by the database developer to implement a data model in specific database management software.

3.7. ER Notation
There is no standard for representing data objects in ER diagrams. Each modeling methodology uses its own notation. The original notation used by Chen is widely used in academic texts and journals but rarely seen in either CASE tools or publications by non-academics. Today, there are a number of notations used; among the more common are Bachman, crow’s foot, and IDEFIX.

All notational styles represent entities as rectangular boxes and relationships as lines connecting boxes. Each style uses a special set of symbols to represent the cardinality of a connection. The notation used in this document is from Martin. The symbols used for the basic ER constructs are:
- Entities are represented by labeled rectangles. The label is the name of the entity. Entity names should be singular nouns.
- Relationships are represented by a solid line connecting two entities. The name of the relationship is written above the line. Relationship names should be verbs.
- Attributes, when included, are listed inside the entity rectangle. Attributes which are identifiers are underlined. Attribute names should be singular nouns.
- Cardinality of many is represented by a line ending in a crow’s foot. If the crow's foot is omitted, the cardinality is one.

Existence is represented by placing a circle or a perpendicular bar on the line. Mandatory existence is shown by the bar (looks like a 1) next to the entity for an instance is required. Optional existence is shown by placing a circle next to the entity that is optional.
3.8 MySQL Data Tables:

**Admin Table:** (Table name is tbladmin)
This store admin personal and login details.

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Type</th>
<th>Collation</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Comments</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ID</td>
<td>int(10)</td>
<td></td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td>AUTO_INCREMENT</td>
</tr>
<tr>
<td>2</td>
<td>AdminName</td>
<td>varchar50</td>
<td>utf8mb4_general_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>UserName</td>
<td>varchar50</td>
<td>utf8mb4_general_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MobileNumber</td>
<td>bigint10</td>
<td></td>
<td></td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Email</td>
<td>varchar120</td>
<td>utf8mb4_general_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Password</td>
<td>varchar200</td>
<td>utf8mb4_general_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>AdminStartDate</td>
<td>timestamp</td>
<td></td>
<td></td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td>current_timestamp()</td>
</tr>
</tbody>
</table>

**Category Table:** (Table name is tblcategory)

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Type</th>
<th>Collation</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Comments</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ID</td>
<td>int(10)</td>
<td></td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td>AUTO_INCREMENT</td>
</tr>
<tr>
<td>2</td>
<td>Category</td>
<td>varchar100</td>
<td>utf8mb4_general_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>CreationDate</td>
<td>timestamp</td>
<td></td>
<td></td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td>current_timestamp()</td>
</tr>
</tbody>
</table>

**Person Table:** (Table name is tblperson)

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Type</th>
<th>Collation</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Comments</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ID</td>
<td>int(10)</td>
<td></td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td>AUTO_INCREMENT</td>
</tr>
<tr>
<td>2</td>
<td>Category</td>
<td>varchar200</td>
<td>utf8mb4_general_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Name</td>
<td>varchar200</td>
<td>utf8mb4_general_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Picture</td>
<td>varchar200</td>
<td>utf8mb4_general_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>MobileNumber</td>
<td>bigint10</td>
<td></td>
<td></td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Address</td>
<td>mediumtext</td>
<td>utf8mb4_general_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>RegDate</td>
<td>timestamp</td>
<td></td>
<td></td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td>current_timestamp()</td>
</tr>
</tbody>
</table>

**Pages Table:** (table name tblpage)

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Type</th>
<th>Collation</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Comments</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ID</td>
<td>int(10)</td>
<td></td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td>AUTO_INCREMENT</td>
</tr>
<tr>
<td>2</td>
<td>PageType</td>
<td>varchar50</td>
<td>utf8mb4_general_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>PageTitle</td>
<td>varchar200</td>
<td>utf8mb4_general_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>PageDescription</td>
<td>mediumtext</td>
<td>utf8mb4_general_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Email</td>
<td>varchar200</td>
<td>utf8mb4_general_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>MobileNumber</td>
<td>bigint10</td>
<td></td>
<td></td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td>current_timestamp()</td>
</tr>
<tr>
<td>7</td>
<td>UpdateDate</td>
<td>timestamp</td>
<td></td>
<td></td>
<td>No</td>
<td>NULL</td>
<td></td>
<td>ON UPDATE CURRENT_TIMESTAMP()</td>
</tr>
</tbody>
</table>
4. Implementation and System Testing
After all phase have been perfectly done, the system will be implemented to the server and the system can be used.

4.1. System Testing
The goal of the system testing process was to determine all faults in our project. The program was subjected to a set of test inputs and many explanations were made and based on these explanations it will be decided whether the program behaves as expected or not. Our Project went through two levels of testing
1. Unit testing
2. Integration testing

4.1.1. UNIT TESTING
Unit testing is commenced when a unit has been created and effectively reviewed. In order to test a single module we need to provide a complete environment i.e. besides the section we would require
1. The procedures belonging to other units that the unit under test calls
2. Non local data structures that module accesses
3. A procedure to call the functions of the unit under test with appropriate parameters

1. Test for the admin module
   ➢ Testing admin login form-This form is used for log in of administrator of the system. In this form we enter the username and password if both are correct administration page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask the details.
   ➢ Report Generation: admin can generate report from the main database.

4.1.2. INTEGRATION TESTING
In the Integration testing we test various combination of the project module by providing the input.

The primary objective is to test the module interfaces in order to confirm that no errors are occurring when one module invokes the other module.

5. EVALUATION
Project URL: http://localhost/lssems

5.1. Home Page
5.2. Categories

**LOCAL SERVICE CATEGORY**

- Car Cleaner (1)
- Cook (2)
- Driver (3)
- Gym Instructor (2)
- Maid (3)
- Milkman (2)
- Nanny (1)
- Paperboy (1)
- Plumber (1)
- Tuition Teacher (4)

Fig 5.2 Local Services Category

5.3. About Us

**About Us**

Local search is the use of specialized Internet search engines that allow users to submit geographically constrained searches against a structured database of local business.

Fig 5.3 About Us Page
5.4. Contact Us

Fig 5.4 Contact Us Page

5.5. 5.5 Admin Login

Fig 5.5 Admin Login Page

5.6. 5.6 Dashboard

Fig 5.6 Dashboard Page
5.6.1. **Add Category**

![Add Category Section](image1)

**Fig 5.6.1 Add Category Section**

5.6.2. **Manage Category**

![Manage Category Section](image2)

**Fig 5.6.2 Manage Category Section**

5.6.3. **Update Category**

![Update Category Section](image3)

**Fig 5.6.3 Update Category Section**
5.6.4. Add Person

![Add Person Section](image)

5.6.5. Manage Person

![Manage Person Section](image)
5.6.6. Update Person

Add Person

Update Person Detail

Service Category
Male

Name
Kanta Agarwal

Profile Pics
Edit Image

Mobile Number
4545454545

Address
H-0986997, Harkeness ghelab

Update

Fig 5.6.6 Update Person Section

5.6.7. Update Image

Update Image

Update Image

Name
Kanta Agarwal

Profile Pics

New Profile Pics
Choose File: no file chosen

Update

Fig 5.6.7 Update Image Section

5.6.8. About Us

About Us

About Us

Page Title
About Us

Page Description
Local search is the use of specialized internet search engines that allow users to submit geographically constrained searches against a structured database of local business

Update

Fig 5.6.8 About Us Section in Dashboard
5.6.9. Contact Us

Fig 5.6.9 Contact Us Section in Dashboard

5.6.10. Admin Profile

Fig 5.6.10 Admin Profile Section
5.6.11. Change Password

![Change Password Section](image)

5.6.12. Forgot Password

![Forgot Password Page](image)

6. CONCLUSION
This Application provides an online version of Local Service System which will benefit the system administrator who wants to maintain records of serviceman and also help to users who search serviceman according to his/her requirement in their own locality.

It makes entire process online and can generate reports.

The Application was designed in such a way that future changes can be done easily. The following conclusions can be deduced from the development of the project:
- Automation of the entire system improves the productivity.
- It provides a friendly graphical user interface which proves to be better when compared to the existing system.
- It gives appropriate access to the authorized users depending on their permissions.
- It effectively overcomes the delay in communications.
- Updating of information becomes so easier.
- System security, data security and reliability are the striking features.
- The System has adequate scope for modification in future if it is necessary.

7. References
For PHP

For MySQL

For XAMPP
[3] https://www.mysql.net/