Global Linked Ticket Raising System

Mr.L.Sundarrajan[1], S.Karthiga[2], B.Sabeetha[3]
Assistant Professor Department of CSE[1], UG (Scholar) Department of CSE[2, 3]
N.S.N College of Engineering & Technology Manalmedu, Karur.[1, 2and3]
Anna University, Tamil nadu, India

Abstract— **Ticket raising system is an existing system of Global Linked Ticket Raising System.TRS does not provide the security for the communication. Proposed methodologies provide secured communication between client and administrator of the company. This methodology track all the customer complaints, events, failures and issues of the application developed by that company. The customer complaints originated from outside, that complaint are solved by the administrator of the company. Client can send multiple complaints to the administrator at a time. RT used to track all the client tickets. Client ticket status, priority, category is maintained by the Ticket manager and Ticket worker. Finally admin response to the customer complaints. This way customer complaints are resolved at the same time different versions of the same application will be enhanced. Client can send the feedback about the application developed by the company or else complaints in efficient way.**

***Index Terms*— Ticket, TRS, RT, GLTRS**

I. INTRODUCTION

The global linked ticket raising system is a computer software package that manages and maintains lists of issues as needed by an organization. GLTRS are commonly used in an organization support center to create, update, and resolve reported customer issues, or even issues reported by that organization's other employees or clients. A support center should include vital information for the account involved and the issue encountered. An GLTRS often also contain a knowledge base containing information on each customer, resolutions to common problems, and other such data. An GLTRS is similar to a "bug tracker", and often, a software company will sell both, and some bug trackers are capable of being used as an GLTRS, and vice versa. Consistent use of a bug trading system is considered one of the "hallmarks of a good software team".
A ticket element, within a GLTRS, is a running report on a particular problem, its status, and other relevant data. They are commonly created in the support center environment and almost always have a unique ticket ID, also known as an issue number which is used to allow the admin to quickly locate, add to or communicate the status of the user's issue or request. The Client can send problems, issues and friendly suggestion to the admin by using GLTRS that’s not strictly forwarded to the admin.Ticket then passed to Ticketing system and then ticket forwarded to the technical support manager. Technical support manager takes the personal details of the user from the database. If any bugs available in that application that was tracked by the bug trading system and then faults are recovered . The Final bug recovered application send to the client.

II. LITERATURE REVIEW

Client and admin communication is necessary for applications developers. If any bugs or any friendly suggestion about the application that was developed by the company that is forwarded to the ticketing system. Ticket system records all the details about the ticket in the database of the user. If any new ticket arrived, technical support manager collects the user details from the database. Technical support manager also takes the friendly suggestion about the application, that increase the standard of the company or application. Technical support manager sends the ticket to the ticket manager and ticket worker. Ticket worker solves the problem. At finally error free application provided to the client and then friendly relationship is maintained.

[1] The main motive behind this technology is to reduce and secure the parking issues in metro cities. Their technology is designed to solve the daily problems of parking in many places like malls, multiplex, award functions, auditorium etc. The aim and purpose is to scale down an efficient working model of a vehicle parking system for accommodating vehicles within a limited area. The goal is to provide better security and reliability with an advanced vehicle parking system. Architects and designers are finding various possible solutions to reduce this person’s problem.

[2] The latest threat report reveals that business is struggling to keep up with rapid changes in techniques by cyber criminals as they switch to increasingly malicious campaigns. Vulnerability is a flaw in a system that allows an attacker to suppress a system's information assurance in computer security. Hence vulnerability assessment is performed for defining, identifying, and classifying the vulnerabilities or security loopholes in a computer, network, or other communications infrastructure. Nexus is a comprehensive proprietary vulnerability scanning tool which scans a computer network and raises an alert if it detects any vulnerability. This deals with bug tracking with Nessus, where Nessus doesn’t have a bug tracking with blocker, critical and major report to cyber security team in any organizations. This proposed system is a novel approach which gives an established idea that all organizations needs an automated bug/issue/defect tracking system and significance of a bug tracking system in cyber security team. This proposed tool is a bug tracking tool integrated with Nessus and Bugzilla which can used for automated ticketing in many organizations.

[3] E-ticketing Systems for Public Transportation (ESPT) are an integral part of Intelligent Transportation System (ITS) which shape the urban environment of the future. The wide deployment of ESPT around the world has proven its success and demonstrated large potential. However, till now the majority of such systems being in operation adhere to the specific standards which are often closed. We aim at suggesting an open and secure e-ticketing architecture which provides an alternative to the conventional paper-based ticketing and is customizable for different schemes. Solution is based on NFC which should facilitate its interoperability with value-added services and pave the way to its convergence with other applications comprising an Internet of Things (IoT) ecosystem in the urban environment. The suggested architecture covers the main processes specific to the ESPT scenario and provides for e-ticket unclonability, unforgeability as well as for protection from tapping and replay attacks. Finally, the results of practical validation using real hardware are presented to demonstrate the real-world pertinence of our solution.

[4] Dynamic pricing is a real-time strategy where corporations adjust prices “on the fly” in response to changing market demand. The hospitality industry has widely adopted this strategy to maximize expected revenue by raising prices in the summer or weekends when demand for hotel rooms is at a premium. In recent years, the sports industry has started to catch on to this trend. As the strategies revolutionized baseball ticketing, more hockey teams have started to experiment with dynamic pricing. The purpose of this is to explore the methodology of applying dynamic pricing to the hockey ticketing arena.

[5] The Mobile Ticket Dispenser System (MTDS) allows customers to remotely draw tickets for service orders anywhere through a mobile handset. The MTD applied to the scenario of post office , in which the customers are patient, but the clerk is important since the original ticket drawn by MT customer may be invalid if he/she does not arrive at the post office before his/her turn. In this case, the behavior of the MT customer is the same as the so called In-Person Ticketing (IPT) customer, who needs to draw a ticket in person when he/she arrive at the service counter. The analytical model to derive the probability that an MT, customer misses his turn when he arrives at the post office. A discrete event simulation model is developed to investigate the performance of the predicted time adjustment mechanism for the MTDS. It use real data collected at a post office to observe the queuing behavior. Study provides guidelines for arranging the time for an MT customer to arrive at the MTDS server.

[6] The ticketing in public transport system, prevailing in the megacities introduces severe malfunction in the system, malicious arguments among public, corruption and most of all traffic jams. This actually suggests a much more public friendly, automated system of ticketing as well as the credit transaction with the use of RFID based tickets. The total system mainly acts to bring out the consistency in the public transport system that will conclude, in uniform access of passengers in daily rides through an automated server being updated every single time the passengers travel by carrying the RFID based tickets.

[7] One of the major challenges in the present ticketing provision is “QUEUE” in buying suburban railway tickets. In this rapidly growing world of technology user buy with oyster and octopus cards for suburban tickets, this is frustrating at times to stand in the queue or if the user forgot his cards. Android Suburban Railway (ASR) ticketing are to buy the suburban tickets which is the most difficult when compared to booking the long journey tickets through ticket. ASR ticket can be bought with the smart phone application, where users can carry his railway tickets in his smart phone as a QR-code. It uses the smart phone’s “GPS” facility to authenticate and erase ticket automatically after a specific interval of time when the user reaches to the destination. User’s ticket information is stored in a cloud database for security, which is not included in the present suburban system. Also the ticket checker is provided with a checker application to look for user’s ticket with the ticket number in the cloud database for checking purposes.

III. EXISTING SYSTEM

Various ticketing system are available in the society based on various needs. In previous days by using existing system only hardware faults are recovered. Before the application development, client and admin or company communication is important. In existing system, there was no secure communication between the client and admin in the company. The communication is not an efficient one. During an application development, client requirements are not satisfied by the existing system, because the client wants to change their requirements whenever they need. Another important problem in the existing system is, budget of the project is not initially fixed. The client cannot pay the entire amount initially without seeing their final output. Premium plans are not available in the existing system. Admin does not assign the ticket to the staff members quickly.

Disadvantages

- It does not provide secure communication.
- Response time is less.
- Different version of the same project does not enhance.
- Standard of their application is not good.

IV. PROPOSED SYSTEM

The global linked ticket raising system is a computer software package that manages and maintains lists of issues as needed by an organization. GLTRS are commonly used in an organization support center to create, update, and resolve reported customer issues, or even issues reported by that organization's other employees or clients. A support center should include vital information for the account involved and the issue encountered. The GLTRS often also contain a knowledge base containing information on each customer, resolutions to common problems, and other such data. GLTRS is similar to a "bug tracker". Consistent use of a bug trading system is considered one of the "hallmarks of a good software team".
A ticket element within a GLTRS is a running report on a particular problem along with its status, and other relevant data. They are commonly created in the support center environment and almost always have a unique ticket ID, also known as an issue

number which is used to allow the admin to quickly locate, add to or communicate the status of the user's issue or request.The Client can send problems, issues and friendly suggestion to the admin by using GLTRS. Ticket is then passed to the ticketing system which is again forwarded to the technical support manager. Technical support manager takes the personal details of the user from the database. If any bugs available in that application that was tracked by the bug trading system, it was recovered. The bug recovered application is then sent to the client.

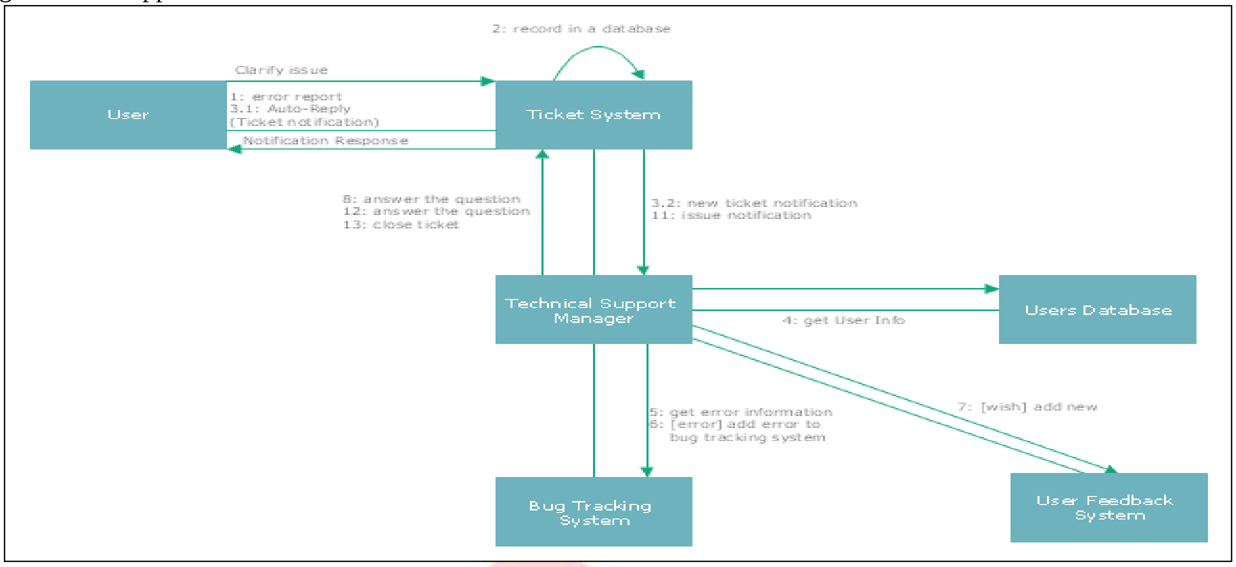


Fig.1 System Architecture for Ticket Creation and Problem Solving

Entire global linked ticket raising system shown in this Fig.1.Client and admin communication is necessary for applications developers. If any bug or any friendly suggestion about the application was developed by the company, It is forwarded to the ticketing system. Ticket system records all the details about the ticket in the database of the user. If any new ticket arrived, technical support manager collects the user details from the database. It also takes the friendly suggestion about the application, that increase the standard of the company or application. It then sends the ticket to the ticket manager and ticket worker. Ticket worker solves the problem. Finally error free application is provided to the client and then friendly relationship is maintained.

Advantages

- Secure communication provided by using MD5 algorithm.
- Response time is more.
- Different versions of the same application will be available.
- Standard applications are developed by the company.

V. ALGORITHMS AND TECHNIQUES

In existing system, there is no secure conversation, so privacy issues occurred. To overcome the above problem, MD5 algorithm and decision tree classification algorithm are used. It is used to provide the secure way of communication. SQL injection, misconfiguration and then crosses site scripting are avoided.

MD5 Algorithm

The MD5 algorithm is widely used hash function, producing a 128-bit hash value. Although MD5 was initially designed to be used as a cryptographic hash function, it has been found to suffer from extensive vulnerabilities. It can still be used as a checksum to verify data integrity, but only against unintentional corruption. A ‘b’ bit message is taken as input and its message digest need to be identified. Here b is an arbitrary nonnegative integer; b may be zero and it may be arbitrarily large. The bit of the message is written down as follows:

m_0 m_1 ... m_ {b-1}

The following five steps are performed to compute the message digest of the message.

- ***Append Padding Bits***

The message is "padded" (extended) so that its length (in bits) is congruent to 448, modulo 512. That is, the message is extended so that it is just 64 bits shy of being a multiple of 512 bits long. Padding is performed as follows: a single "1" bit is appended to the message, and then "0" bits are appended so that the length in bit of the padded message becomes congruent to 448, modulo 512. In all, at least one bit and at most 512 bits are appended.

- ***Append Length***

A 64-bit representation of b is appended to the result of the previous step. In the unlikely event that b is greater than 2^64, then only the low-order 64 bits of b are used. At this point the resulting message has a length that is an exact multiple of 512 bits. Equivalently, this message has a length that is an exact multiple of 16 (32-bit) words. Let M [0 ... N-1] denote the words of the resulting message, where N is a multiple of 16.

- ***Initialize MD Buffer***

A four-word buffer (A, B, C, and D) is used to compute the message digest. Here each of A, B, C and D is a 32-bit register. These registers are initialized to the following values in hexadecimal, low-order bytes first:

Word A: 01 23 45 67
Word B: 89 ab cd ef
Word C: fe dc ba 98
Word D: 76 54 32 10

- ***Process Message in 16-Word Blocks***

The auxiliary functions take as input. Three 32-bit words and produce as output one 32-bit word.

F(X, Y, Z) = XY v not(X) Z
G(X, Y, Z) = XZ v Y not (Z)
H(X, Y, Z) = X xor Y xor Z
I(X, Y, Z) = Y xor (X v not(Z))

In each bit position 'F' acts as a conditional: if X then Y else Z. The function F could have been defined using + instead of v since XY and not (X) Z will never have 1's the same position. It is Interesting to note that if the bits of X, Y, Z are unbiased, the each bit of F (X) will be independently unbiased.The functions 'G', 'H', and 'I' are similar to the function F, in that in "bitwise parallel" to produce their output from the bits of X Y and Z, in such a manner that if the corresponding bits of X, Y and Z are independent and unbiased, then each bit of G (X, Y, Z), H (X, Y, Z), and I (X, Y, Z) will be independent and unbiased. Note that the function H is the bit-wise "xor" or "parity" function of its input.

TECHNIQUE

Uploading Files

First locate the core files in the folder: **core_files/core.zip** and extract them. Upload the files to your main web directory. A sub folder is fine too. Modify the file: **application/config/config.php** and edit the following code with your own URL to the system.

&config ['base_url'] = 'http: // www.example.com/';

```
$db.['default']['hostname'] = 'local host';
$db ['default'] ['username'] = 'your username';
$db ['default'] ['password'] = 'your password';
$db ['default'] ['database'] = 'database name';
```

Modify the file: **application/config/database.php** with your database settings:

- Import the SQL file core_files/database.sql to the database you specified in the above settings.
- Run the install file on your website. This URL is usually: http://www.example.com/install/
- Alternative install URL: http://www.example.com/index.php/install/.

Configuration

When creating libraries for our applications use configuration parameters in order to perform its tasks.

```
$value = $this->config->item ('some item');
```

This is fine, if we have just a couple of configuration parameters, but when we need to extract a dozen configuration parameters.

```
$this->load->config ('config_file', true);
$options    = $this->config->item ('config_file');
$options ['some item'];
```

The single line in our entire configuration file will be placed in the $options array. And to further sweeten the deal, using this technique also makes sure to avoid the name collision problem in configuration files. This way, the parameters can be named anything without worrying about other configuration files.

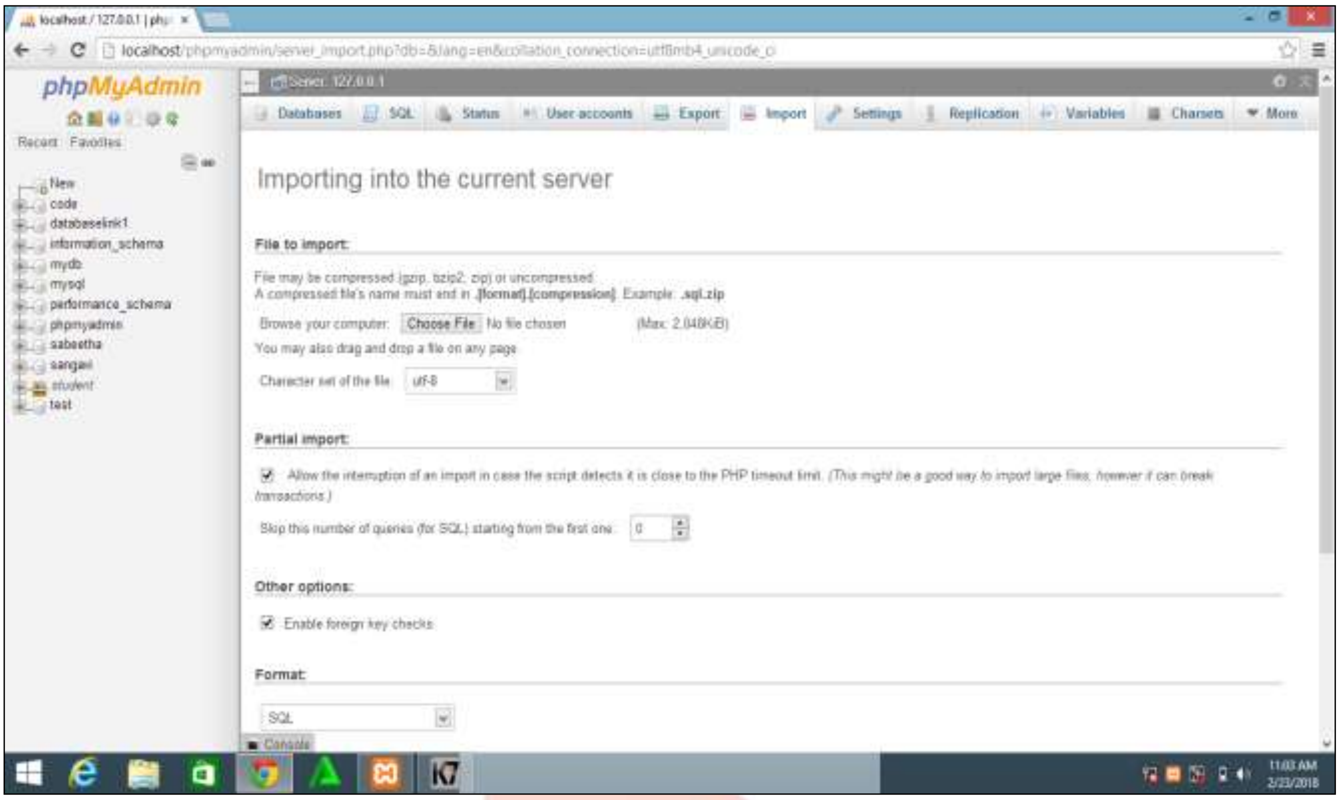
Leveraging HTML Emails

The need to send HTML emails is omnipresent, be it a newsletter or some automatic welcome message.

```
Public function sends mail ()
{
$this->load->library ('email');
$this->email->from ('jdoe@gmail.com', 'John Doe');
$this->email->to ('jane_doe@gmail.com');
$this->email->subject ('some subject');
$this->email->message $this->load->view ('emails/message', $data, true      ));
$this->email->send ();
 }
```

Database

Find the file: application/config/database.php and open it up using a text editor like Notepad. For the most part, host stays the same as the local host unless web host specifies to you otherwise. Following are needed that is database name, database username and database password.Once the file modified, save it and overwrite the existing one on your web server. Next, you will need to execute the SQL file that came with the product download.


localhost/phpmyadmin/server_import.php?db=&lang=en&collation_connection=utf8mb4_unicode_ci

Server: 127.0.0.1

Importing into the current server

File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**

Browse your computer: Choose File No file chosen (Max: 2,048KiB)

You may also drag and drop a file on any page.

Character set of the file: utf-8

Partial import:

[✓] Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. *(This might be a good way to import large files, however it can break transactions.)*

Skip this number of queries (for SQL) starting from the first one: 0

Other options:

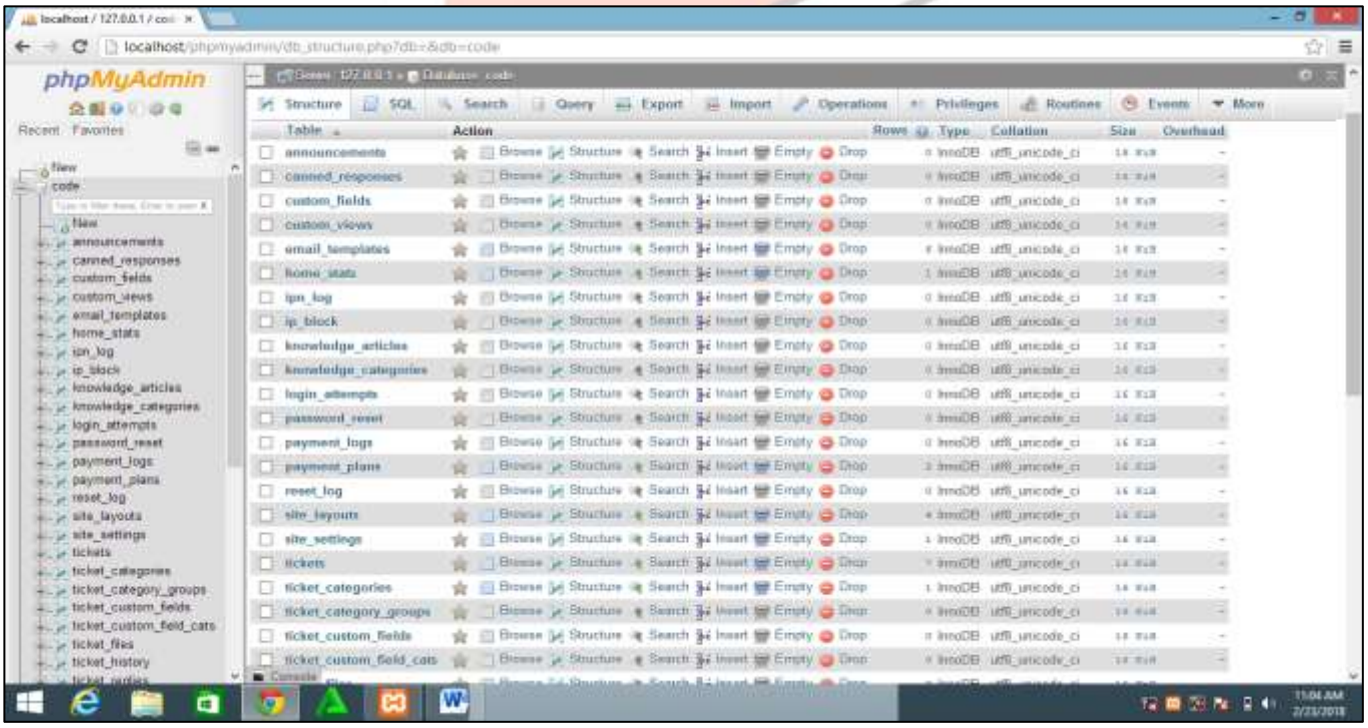
[✓] Enable foreign key checks

Format:

SQL

Fig.2 Database Creation

This file will create the database tables that are used to store the data of your application. Usually you can import the SQL tables by using a database management system like PHPmyAdmin. The SQL file is located in core_files/database.sql in the product download. To Import the file, login to your database management system, such as PHP myAdmin. Make sure the database is named as the database.php config file then locate the Import Tab on PHPmyAdmin.


localhost/phpmyadmin/db_structure.php?db=&db=code

Server: 127.0.0.1 » Database: code

Table	Action	Rows	Type	Collation	Size	Overhead
announcements	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_unicode_ci	16 KiB	-
canned_responses	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_unicode_ci	16 KiB	-
custom_fields	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_unicode_ci	16 KiB	-
custom_views	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_unicode_ci	16 KiB	-
email_templates	Browse Structure Search Insert Empty Drop	7	InnoDB	utf8_unicode_ci	16 KiB	-
home_stats	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8_unicode_ci	16 KiB	-
ipn_log	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_unicode_ci	16 KiB	-
ip_block	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_unicode_ci	16 KiB	-
knowledge_articles	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_unicode_ci	16 KiB	-
knowledge_categories	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_unicode_ci	16 KiB	-
login_attempts	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_unicode_ci	16 KiB	-
password_reset	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_unicode_ci	16 KiB	-
payment_logs	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_unicode_ci	16 KiB	-
payment_plans	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8_unicode_ci	16 KiB	-
reset_log	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_unicode_ci	16 KiB	-
site_layouts	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8_unicode_ci	16 KiB	-
site_settings	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8_unicode_ci	16 KiB	-
tickets	Browse Structure Search Insert Empty Drop	7	InnoDB	utf8_unicode_ci	16 KiB	-
ticket_categories	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8_unicode_ci	16 KiB	-
ticket_category_groups	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_unicode_ci	16 KiB	-
ticket_custom_fields	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_unicode_ci	16 KiB	-
ticket_custom_field_cats	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_unicode_ci	16 KiB	-

Fig.3 Database Configuration

Controller

The controller contains two methods

Construct

- Load Pagination library.
- Load Post model.
- Specify a pagination limit (per page).

Index

- Get total rows count using get rows () method of the Post model
- Specify the configuration in $config array for Pagination class
- Initialize the Pagination class with your preferred options using initials () method.
- Define offset by the page URI
- Get rows from posts using get rows () method of the Post model
- Pass the post data to view and load the list view

Model

The Post model is used to retrieve posts data from the database. The get rows () function fetches the records from the posts table based on the limit restriction provided in the $params array and returns posts data as an array.

View

The view folder contains index.php file to list the paginated data. The bootstrap library is used for styling the posts lists and pagination links. So, include the Bootstrap library first.

Mod Rewrite

We use a URL routing trick called mod_rewrite. This is normally installed by default on most web servers and allows for URLS .If you don't have mod_rewrite settings enabled on your web server, you can still use the application by setting your index_page value to the following line in the application/config/config.php file.
$config [‘index_page’] =’ index.php’;
This setting will make it so your URLs have index.php in them. However, the default setting is to remove this and assumes you have mod_rewrite enabled on your webserver.

VI.CONCLUSION AND FUTURE WORK

GLTRS provides secure communication and then client problems are resolved immediately. Response time less between client and admin. By using GLTRS client can change their requirements whenever their needs. Premium plans are available in this concept, that plans reduce the client troubles. These methods establish a friendly relationship between client and admin.Each new thought company develops the new application but in GLTRS, different versions of the same application will be developed. This increases the standards of their application and company also. In future, we add the organization to organization communication in this application. This way increase the standards of their company compared to the foreign companies.

REFERENCES

[1] Azhar Mithani, “A Smart Parking System using Rpi”, Vol.no.6, Issue no.09, September 2017.
[2] Siva Guru. G. S, “Integrating Ticketing System for Vulnerability Management”, International Journal of Engineering and Innovative Technology (IJEIT), Volume 6, Issue 1, July 2016.
[3] Ivan Gudymenko, Felipe Sousa, Stefan Köpsell, “A Simple and Secure E-Ticketing System for Intelligent Transportation based on NFC”, Volume-3, Issue-11, November, 2015.
[4] Sabah Sadiq, Jing Zhao, Christopher Jones Deloitte Consulting LLP, “Puck Pricing: Dynamic Hockey Ticket Price Optimization”, Paper 2863-2015.
[5] Gi-Ren Liu, Phone Lin, Senior Member, Yi-Bing Longfellow, “Modelling Mobile Ticket Dispenser System with Impatient Clerk”, Volume 21 number -2015.
[6] Rutuja Warhade1, Prof. Basha Vankudothu2, ‘‘RFID-Based Ticketing For Public Transport System, International Journal of Advance Research in Computer Science and Management Studies”, Volume 2, Issue 12, December 2014
[7] Sana Khoja, Maithilee Kadam, “Android Sub-urban Railway Ticketing using GPS as Ticket Checker”, Volume 2 Issue 3, May-June 2014.