Pothole Application – Data Storage, Processing and Reporting

Nanda Kumar Gunalan

Advisor: Leon Reznik
Contents

1. Introduction.............................................................................................................1
2. Goal..........................................................................................................................1
3. Use Case Diagram..................................................................................................1
4. Architecture...........................................................................................................2
5. Requirement Gathering..........................................................................................3
6. Implementation.......................................................................................................4
7. Database Design....................................................................................................5
8. Database queries – CRUD Operations....................................................................8
9. Technologies used..................................................................................................10
10. User Authorizations............................................................................................11
11. Key features of Application................................................................................12
12. Security of Web Application................................................................................14
13. Backup of Data.....................................................................................................15
14. Set up of triggers-Notification............................................................................16
15. Testing of the Application....................................................................................16
16. Screen shot of the Application............................................................................17
17. Conclusion............................................................................................................19
1. Introduction

A pothole is an unwanted deformation in the road which is caused due to wear and tear, changes in weather conditions, vehicular movement and other various reasons. Potholes are dangerous and can cause serious accidents, sometimes the accidents are fatal. Potholes also cause serious vehicular damage; reports from insurance claims indicate that potholes are the major causes for vehicle damage and accidents. There are a number of ways in delaying the onset of potholes on roads, but it is not possible to completely eliminate the formation of potholes. Authorities cannot be sure when potholes will be formed on the paved roads. To solve this pothole issue, it is necessary that there is a system where Citizens can report the potholes they see or face difficulties with. Once reported the authorities are intimated about the potholes and have all the necessary information regarding the potholes so that they can take timely action and prevent all major untoward incidents caused due to potholes.

2. Goal

The goal for my project is to have a web application where all the data of a pothole reported by a citizen is recorded into a database. This stored data regarding the real time potholes can be used by the public works manager to generate reports on the potholes, assign work to the road works manager to repair the potholes, keep a check on the status of the potholes. A system which can recognize a high priority pothole, and immediately report to concerned authorities. Also these recorded data can be used to analyze trends, find root causes of the potholes and will be helpful to extract other vital information to take necessary actions and minimize the number of potholes.

3. Use Case Diagram

The diagram below shows the Use Case Diagram of the Pothole Application. The pothole application has three users - citizen, road works manager and the public works manager. The citizen reports a pothole by using an android application. All the recorded information from the android application is stored in the MySQL database. The MySQL database is accessible by the road works manager and the public works manager with a web application user interface provided to both of these users. The role of a road works manager is to update pothole status and update pothole data. The role of a public works manager is to assign work on reported pothole to repair it, manage reported potholes and analyze the pothole data.
4. Architecture

**Fig 1.1 – Pothole Application Use Case Diagram**

**Android Application for Citizens to report the Pothole**

**Web Application for authorities to access records and report generation**
From the above Architecture diagram, we can see two major components the android application and the web application both having access to a common MySQL database. The citizens using the android application answer several questions regarding the pothole they have seen, and report the problem. All the submitted information is captured and stored into the MySQL database, this information is available for access and modifications through the Web Application interface. The web server architecture was achieved by using the XAMPP environment, which consists of the Apache Tomcat Server, the MySQL database and also PHP integrated into it. This is basically a client server architecture, all the information submitted by a client is sent as a HTTP POST request. Complete backend of this software was written using the PHP programming language. This allowed me to write the PHP scripts to receive data from the external android application and to store the data in the MySQL database.

5. Requirement Gathering

After understanding the use cases of the entire platform, lot of research was done in accumulating all the necessary fields that is essential for generating a substantial report for authorities, and collecting data that would be essential in understanding the nature of potholes and the data which will be helpful in identifying the root causes for potholes. These data can be later used for data analysis to mine critical information which will help in preventing untoward incidents caused due to potholes. Some of the key fields which play a significant role in the application are latitude and longitude of the pothole, priority, the road material used in constructing the road, the general weather conditions in the locality, the average speed of vehicles travelling on the affected road and the status of the pothole. The status of the pothole can have three input values open, closed and repaired. There is also this last repair performed field, this input is provided by the road works manager. If we closely analyze this we get to predict the longevity of the roads. All the associated fields of this record will indicate how the other factors are determining the longevity of maintaining a good to use road without any potholes.
6. Implementation

The initial steps of the implementation involved setting up the XAMPP. The XAMPP environment provides the infrastructure for the Apache tomcat server which acts as the web server, the MySQL database and a PHP platform to write the server side code. Both the Apache and the MySQL are started which occupies the ports 80,443 and 3306. Now the system the running to establish a connection with the Android Application and receives the data. Json data was sent from the Android Application, using the PHP script receivedata.php I separated the data into individual fields. The next step was establishing the connection with the database which was achieved using below code.

```php
//Connect to server
mysql_connect("localhost", "root", ") or die (mysql_error());
//Connect to database
mysql_select_db("capstonedatabase") or die ("Cannot connect to database");
```

Finally all the fields are inserted into the table with respective columns.
7. Database Design

This was one of the most important stages in my project. I designed the database keeping in mind that there is a lot of reporting activities going to be performed through my web application. To achieve efficient retrieval operations I used the right mix of normalized and de-normalized tables to reduce duplication of data and reducing the number of join operations performed during data retrieval.

The name of the database is a capstone database which has three main tables. The three tables are pothole table, contact table and user table.
The above picture shows the fields present in contact table. This application is designed for the Rochester city, New York. The Rochester city has six county namely Livingston, Monroe, Ontario, Orleans, Wayne and Yates. The field contact_id is the primary key field of this table which gets auto increment as more records get inserted. Each of the six county is associated with the email id, when a report needs to be sent to authorities, the location is identified and sent to the respective emails with a complete report detailing about the pothole.
Fig 1.9 – Pothole table Structure
The above picture shows the structure of the users table which has four fields ID, username, password and accesslevel. Access level determines the functionalities each user is entitled to use in the web application.

8. Database Queries – CRUD Operations

a. Insert

```php
mysql_connect("localhost","root",""") or die (mysql_error());
mysql_select_db("capstonedatabase") or die ("Cannot connect to database");
$query = mysql_query("Select * from users");
while($row = mysql_fetch_array($query))
{
    $table_users = $row['username'];
    if($username == $table_users)
    {
        $bool = false;
        Print ' <script> alert("Username has been taken!");</script>';
        Print ' <script> window.location.assign("adduser.php");</script>';  
    }
}

if($bool)
{
    mysql_query("INSERT INTO users(username, password, accesslevel)
VALUES ('$username', '$password', '$accesslevel')");
}  
```
b. Delete

Delete operation is needed when unwanted records need to be deleted.

c. Retrieve

Many retrieve queries is used in my project for a report generation. Below is an example to retrieve four fields: data, location, priority and status of the records from the database and generating a report.

```php
if($_SERVER['REQUEST_METHOD'] == "POST" ){

$startdate = mysql_real_escape_string($_POST['startdate']);
$enddate = mysql_real_escape_string($_POST['enddate']);
$location = mysql_real_escape_string($_POST['location']);

// $date1 = date("m-d-Y",strtotime($date));
$startdate1 = date("Y-m-d",strtotime($startdate));
$enddate1 = date("Y-m-d",strtotime($enddate));

mysql_connect("localhost","root","") or die (mysql_error());
mysql_select_db("capstonedatabase") or die ("Cannot connect to database");

$query = mysql_query("Select * from pothole where date1 between '$startdate1' and '$enddate1' and location = '$location'");
$counter = 0;
while($row = mysql_fetch_array($query)) 
{
    $counter++; 
    Print "<tr>
    Print '<td align = "center">' . $counter . "</td>
    Print '<td align = "center">' . $row['date1'] . "</td>
    Print '<td align = "center">' . $row['location'] . "</td>
    Print '<td align = "center">' . $row['priority'] . "</td>
    Print '<td align = "center">' . $row['status'] . "</td>
    Print "</tr>
    }
```

Fig 1.12 – Example Retreive Query
d. Update

Frequently, records in the database needs to be updated. This is done by using update queries. Some of the common fields that is updated regularly in my project is the status field. The status field usually gets updated to either closed or repaired from open, after a repair has been performed on the reported pothole.

9. Technologies Used

Front end – HTML, CSS, Javascript

HTML (Hyper text markup language) – HTML is a markup language used to create websites. HTML is interpreted by a web browser and the content like the texts and images are displayed on the World Wide Web page. The language syntax contains tags, all the content is written within the tags.

CSS (Cascading Style Sheet) – CSS has been used in my web application to design and style HTML elements in web pages, this has enabled in achieving a well-defined structure and a good look and feel for the entire web application.

Javascript – It is essential that web pages are interactive and respond to user actions.

Server side scripting – PHP

Databases – SQL programming

Design Pattern – MVC

I have used Model View Controller Architecture for developing my web application. This has enabled in separating the application logic from the front end. Each layer is isolated to ensure changes in one layer do not affect the other. In my application, the MySQL database acts as the Model which is responsible for managing the data. The View in my application is the front end user interface (HTML) that is visible to users, based on the actions of the user requests is sent to make the necessary changes in the model. The controller is the HTTP request, requests is made using the view. The controller handles the connection between the model and view, it receives
requests from the users, which is sent to the model for processing. The processed data from the model is again handled by the controller and sent back to the view.

10. User Authorizations – Different Level of access based on the role Of User

Unauthorized users should be prevented from accessing the application and having access to critical data in the database. I have implemented this by using session variables. When a user is logged in the accesslevel is recorded in the session variable. Using this session variable there is a check performed before a permission is given to access the web page. If access is denied immediately a message is shown and redirected to the original page. Below is the screen shot

![Access denied - you do not have access to this page](image)

You will be redirected in 3 seconds

Fig 1.13 – Screen shot preventing access for unauthorized users.

```php
<?php
    session_start(); // starts the session
    if($_SESSION['user']){ // checks if user is logged in
        
    } else {
        header("location:index.php"); //redirects if user is not logged in
    }
    $user = $_SESSION['user']; //assigns user value
?>
```

Fig 1.14 – Check for User Login
<?php

session_start();
// If no session variable exists, or unauthorized user_level, redirect the user:
if (!isset($_SESSION['user'])) {
    // if a valid user session is found then the user level is checked, if the
    // user has level 3 access they will be granted access if not a access denied
    // message be displayed and the user will be redirected.
    if ($_SESSION['accesslevel'] == 3) {}

    else {
        header("Refresh: 3; url=./home.php");
        echo '<h3>Access denied - you do not have access to this page</h3>';
        echo 'You will be redirected in 3 seconds';
        // include ('includes/footer.html');
        exit(); // Quit the script.
    }
}
// if no valid session is found then the user is not logged in and will
// receive a access denied message and will be redirected to the login page.
else if (!isset($_SESSION['user'])) {
    header("Refresh: 3; url=login.php");
    echo '<h3>Access denied - you do not have access to this page</h3>';
    echo '<p>You will be redirected in 3 seconds</p>';
    include ('includes/footer.html');
    exit(); // Quit the script.
}

Fig 1.15 – Check for User Authorizations

The above image shows the code to implement the user authorization in the web application.

11. Functionalities- Key Features

- Access to the reported pothole records.
- Ability for authorized users to create, retrieve, update and insert pothole records.
- Application is secure to prevent common attacks like SQL Injection and Cross Site Scripting.
- Different levels of user authorizations.
• Well defined and easy to use User Interface.

• Secure login for the application.

Fig 1.16 – Screen shot of a Secure Login Screen
Fig 1.17 – Check if Username and Password match in the database records

- The generation of a report of key data.
- Triggers email to concerned authorities on arrival of high priority potholes and periodically updates potholes information via email.
- Regular backup of the database to prevent loss of critical data.
- Important critical real time data is available which can be used for data analysis and mining key information.

12. Security of Web Application

I have used prepared statements in my code whenever there was a need to establish a connection to the database and accessing fields in the database, this will help in preventing some of the most common web attacks like the SQL Injection and Cross site scripting. Filtering input
fields in the application that is concerned to be a threat, before they are passed as an input to the SQL query.

**SQL Injection** – In this type of attack an attacker executes SQL statements to take control over the database of a web application. This is one of the most prevalent type of web application vulnerability. If an attacker is successful, he gains the mechanisms to retrieve and modify the contents of a database. To perform SQL injection an attacker looks for ways to input information, which will be directly used to query a database. In this web application there are several input fields, let us take an example of the login screen. The login screen has two input fields username and password. For any user to enter the application, the user has to be an authorized user. This check is performed by checking the username and password fields in the database, if there is a match then the user gains to the web application. SQL query is used to perform this check. There are many other ways an attacker provide values in the input fields such that the SQL query returns true and he is able to gain access to the database. An example would be to enter the following input in the password field password OR 1=1, this would result in bypassing the authentication. There are two necessary conditions for a SQL injection to take place, the web application using a SQL database and having user input fields. Below are some of the procedures I have followed to mitigate the SQL injection attack

- Filtering the input – Removal of dangerous code, special characters from input fields which are likely to be a threat.
- Prepared Statements – Prepared statements works by creating placeholders, this ensures the user input is considered as a content of a parameter instead of being directly supplied as input to the SQL query. Efficiency of SQL queries has been improved by using prepared statements.

**13. Backup of Data**

I have set the system for a periodic backup of the database to prevent the loss of data due to any untoward incidents. This is implemented by automatically sending the SQL dump file via email to the concerned official or administrator.
14. Set up of Triggers

In databases, triggers is a SQL procedure which initiates or automatically fires an action based on any occurrences of an event. An event is any of the following: INSERT, DELETE, UPDATE, CREATE, ALTER, DROP, SERVERERROR, STARTUP, LOGOFF, LOGON, or SHUTDOWN.

Triggers have been set up to send email to the authorities immediately if a high priority pothole record is inserted into the database. The corresponding email-id is obtained from the contact table in the database based on the location of the pothole, then the email sent with all the details of the pothole.

In this application there are 5 levels of priority ranging between 1 and 5. A pothole with priority of 5 is considered very severe and requires immediate action. The triggers has been set up on two events INSERT and UPDATE. An email with pothole details will be fired when a record with pothole priority 5 is inserted into the database or if an existing record in database has its priority field updated to 5. The email is sent by using the PHP mail() function. The syntax of the mail() function is mail(to, subject, message, headers, parameters). The arguments to, subject and message are mandatory fields specifying the recipient address, the subject of the message and content of the email. The other two arguments, headers and parameters are optional.

15. Testing Of Application

The testing of the web application involved the following tests: Functionality testing, Usability testing, Interface testing, Performance testing and Security testing. All the testing was done manually.

Functionality Testing - Every functionality of the application was tested. All queries used in the application was validated by testing with real time data. Verified that there is no dead pages or broken links.

Usability Testing – Verified the navigation and controls in the application and tested the user experience.
Interface testing – This web application is connected to the external Android Application. After starting the web server, several tests were performed to check the data flow from the Android Application to the database interfaced with the web application. Tested if all requests are handled without any service denial.

Performance testing – One of the main objectives of this project is to achieve high performance data retrieval by having a well-designed database and by using prepared statements.

Security testing – Tested the web application for SQL Injection and Cross site scripting by providing different kinds of inputs to the form fields. This application did well in preventing these attacks for the values tested.

16. Screen Shots of the Application

a. Report Generation

User can generate reports by providing input to the three fields start date, the end date and location. It is possible to generate report even if either one of the start date or end date is missing, but location is a mandatory input field. On the click of submit button all the records are returned which match the below entered criteria in the database.

The reports generated has the following fields

Sl No – indicates the number of the record in the report.

Date – Indicates the date on which the pothole was reported.

Location – The city is divided into different county, and each county is responsible for repairing the pothole. With the location field it gets easier to assign the potholes to the respective county.

Priority – This is the most important field, as it indicates the severity of the pothole. This determines the time frame for officials to act on repairing the pothole.

Status – The status field has mainly two inputs- open and closed. Open indicates that still no action has been taken whereas closed indicates the pothole has been repaired.
b. User Interface to perform CRUD operations

An authorized user will have complete access to the database and make changes to the database tables via a well-defined user interface. In real words, there is always chances of human error in entering information. Also, this provides an alternate interface to android application for enter pothole data into the database.
The above screenshot shows all the quick links available, with a single click the daily data, weekly data, and monthly data report is generated. And also there is single-click links for a user to manually send information about the potholes and have a backup of database.

17. Conclusion

- This report presents a solution to tackle the problem of potholes in roads by providing an efficient way for users to report the problem and also with a good web application for city officials aiding them to quickly identify and repairing potholes.

- User-friendly front end of the application for complete access and making changes to all the tables in the MySQL database.

- Efficient data retrieval operations.
• It is possible to integrate the Android Application and the Web Application both having access to a common MySQL database.

• Future work would involve having reports generated in the form of charts for better understanding and visualization of the data.