Design and Development of a Pothole Tracking System (PotAlert) using Web Progressive Application (WPA)

Aerina Sofea Rosli, Maryam Roslan, Azlin Nordin
Department of Computer Science, Kulliyyah of Information and Communication Technology (KICT),
International Islamic University Malaysia (IIUM), 53100 Gombak Kuala Lumpur, Malaysia.
azlinnordin@iium.edu.my.

Abstract— Potholes and deteriorating road conditions are seen as a public disturbance. This paper presents the design and development of PotAlert, which is a progressive web application (PWA)-based pothole tracking system. The application was developed to encourage road users and the authorities to communicate and exchange information concerning the presence of potholes on the road. The purpose of the application is to give road users a way to help the authority enhance public safety. This is to ensure that problems regarding potholes in Malaysia's road infrastructure are effectively addressed. From the perspective of the authorities, the system shall aid in the automation of management reports, ensuring that they are arranged and kept properly in order to efficiently handle work orders and contact road workers. As an outcome of the project, the PotAlert system was deployed and tested within its constraints setting. Due to the difficulty in acquiring actual data during the pandemic, the system was evaluated using dummy pothole data. The system needs to be further improved and evaluated in the future using actual data and settings.

Keywords—pothole detector, pothole tracker, potholes

I. Introduction

The development of any application is usually based on the need to solve the problems that humans face in their daily lives. In this age of the digitized world, a lot of the ways of human's lives have changed. With how rapidly the world is progressing, almost every aspect in exchanging, communicating, and storing data has shifted dependencies towards the internet. Moreover, humans have become increasingly reliant on services that allow them to convey their needs at their fingertips. This includes the need to fix one of road users' most persistent problems on the road, which are potholes.

Potholes are major indicators of structural flaws in asphalt roads, and diagnosing them effectively is one of the most critical challenges in determining proper asphalt-surfaced pavement maintenance and repair procedures [10]. Malaysia, as a rapidly developing country, relies on the road as its primary mode of transportation. It is stated that an average of more than 70 motorcyclists are killed on the road every week due to bad road conditions [1]. To support the statement, a study had also shown that the majority of fatal crashes of motorcycles occur along straight road sections as stated in [2].

Poorly maintained roads can cause accidents mostly due to the fact that they create hazards to drivers. In many instances, a driver may attempt to avoid a certain situation, like a pothole or pooling water, which could cause serious accidents. The amount of damage caused by potholes can be decreased if they are spotted and reported in a timely way [18].

A key reason for the presence of unpatched potholes on the road is due to the information gap between the public, who travel on bad roads, and relevant authorities, which are in charge of road repairs. PotAlert, a pothole tracking and reporting system was created with the goal to ease communication and information exchange between road users and authorities regarding the presence of potholes on the road. The primary objective of the design and development project is to derive a proof-of-concept to demonstrate the viability of a pothole detection system that adopts Progressive Web Application (PWA) technology to make use of smartphone features.

The PotAlert utilizes the support of PWA for the development. PWA is among the latest innovations in Web development, which attempts at making the user experience the same as in the native mobile app [15]-[17]. One of the characteristics is that the client-side of the PWA employs web service calls to load and render content, exactly like native applications [15]. WPA also allows the utilization of all of the device's functions including the GPS, camera, and sound, as well as the device's notification system.

PotAlert uses the public's road experience to alert the relevant authorities about potholes conditions to flow seamlessly between the two parties. The system also allows the road users to mark the pothole location by providing the pothole image, or simply by providing the pothole location.

To ensure that the road users are able to provide accurate and useful information, they are also given the ability to make a remark and assign a severity level to the potholes. The road drivers then can check and display the report status and any updates by the authorities. The authority, on the other hand, may generate a dynamic report with visualization and update the pothole report status based on the pothole maintenance activity. The authority has the ability to designate any specific road teams for road repairs.

This paper is structured as follows. The related work on the pothole tracking system is presented in Section 2, and the PotAlert system development methodology is detailed in Section 3. The technologies employed to develop PotAlert are described in Section 4, and the stakeholders involved in PotAlert are described in Section 5. In Sections 6 and 7, the system's characteristics, as well as its limitations and constraints, are outlined. In the subsequent Section 8, future work for PotAlert is included, and finally, the conclusion is highlighted in Section 9.

II. Related Work

Based on Kim and Ryu [10], the existing pothole detection methods in the literature can be categorized into (1) vibration-based methods – where accelerometers are used to detect potholes, (2) 3D reconstruction-based methods- which can either be using a 3D laser scanner, stereo vision, or visualization using Microsoft Kinect sensor and (3) vision-based methods – which can include 2D image-based or video-based approaches.

For this project, the first category of the classification i.e. vibration-based method was adopted due to the project scope and constraints. An investigation of earlier existing systems that are comparable to the proposed system was carried out in order to determine the strengths and shortcomings of the current system, as well as areas that might be enhanced and improved further.

A. Waze

The first such current system examined is Waze, an app that allows users to report potholes on the road. Waze is accessible at https://www.waze.com/ and also in Google Play Store and Apple Store. Waze is a GPS navigation mobile application that assists road users in getting from one point to another. In addition, the application also has a feature that allows users to report potholes that they encounter along the road.

The state government of Selangor saw an opportunity and potential with the application and proceeded to collaborate with the traffic and navigation application, Waze, in order to help drivers and motorists to alert other road users on the location of potholes. Currently, a database system on local roads is maintained by the Public Works Department (JKR) and local councils. To emphasize the criticality, Waze users in Selangor reported 52,295 potholes between January 2019 and September 2020 as reported in [11].

To report a pothole in Waze, users need to choose a pothole and submit it to the app. The system matches the report to the road owner, who can either be the local council or JKR. In order to avoid duplicated reports, the system tracks the reports and sends the information to the respective road owner. When four to five reports of the same pothole are received, it is considered a valid report. The strength of the app is it mainly works as a navigation app, which is also integrated with pothole reporting features. The app provides voice command integration features, which allows hands-free reporting. However, the app needs to be downloaded to be used and Waze only has a one-way communication as the complainant does not get any response [3].

B. *i-Tegur*

The following system under consideration is the *i-Tegur* application, which is also accessible in the Google Play Store and the Apple Store. The application was developed under the purview of the Malaysian Ministry of Housing and Local Government. The *i-Tegur* is a mobile application that connects the public with local governments, particularly to make complaints or raise concerns that impact their residential areas. Users may also make complaints about parking, reckless rubbish disposal, drain obstruction, open burning, and other issues, in addition to potholes and broken lighting. To make a complaint, the user must first select a category and then take a photo of the issue. After that, the user provides a short description of the complaint. The user location is automatically detected by the application. Once the complaint is submitted, the user receives a notification informing them that the complaint has been sent to the relevant city council or authority to review. The details of the complaint can also be viewed from their homepage.

However, there is a lack of feedback status from the relevant authorities for the complaints that had been submitted to the system. Furthermore, according to the analysis of the users' reviews of the application in the Play Store, some users reported that the application took too long to launch. It becomes a hassle when they are driving and have to launch the application. Some users also complained that the application was difficult to use and that t re is no option to retrieve photos from their phone gallery. This complicates things because the users cannot directly upload the picture while driving. Aside from that, the review comments indicated that there were too many functions in the application, yet most of them did not perform well [4].

C. Real-Time Arduino-Based Depth Sensing for Road Condition Monitoring

Another current approach is Real-Time Arduino-Based Depth Sensing for Road Condition Monitoring, which combines a device with an Android-based application [5]. The device uses vibration-based-road condition detection, which consists of an Arduino-based sensing module. First, the Arduino-based sensing module is designed to evaluate the road conditions in real-time and the evaluation result is sent to the smartphone via a wireless medium. While the smartphone receives the data from the sensing module, the designed android-based application marks the position with abnormal road conditions on Google Maps by utilizing the Global Positioning System (GPS) of the smartphone. As a consequence, drivers may alter their commute routes, and the government can use the pothole locational data for future road repair and development.

The advantage of this approach is the automatic sensing of potholes without having to interact with smartphones while driving. Potholes can be marked automatically on the Google Map and displayed along the map route. However, the drawback to this method is the drivers need to reach the pothole in order to detect it [5]. The approach also necessitates the use of an extra device, which must be linked to the smartphone in order for the functionalities to be completely employed, hence increasing the cost of the solution.

D. Roadway State Monitoring System (RSMS)

In this work [12], a method for detecting and localizing road potholes is presented, with the goal of collecting data on road potholes by integrating the use of the accelerometer, GPS, and compass contained in smartphones running the Android operating system. The information about the prospective potholes detected by the application is analyzed, categorized, and stored in the system. For pothole detection, the pothole information comprises latitude, longitude, GPS accuracy, vehicle speed, accelerometer acceleration, and compass azimuth. The pothole information also includes the system's date and time, the analyst who categorized the pothole, the rating according to the specified scale, the kind of pothole, and the detection analysis.

The method enables smartphone-based assessments of the road's physical condition and infrastructure. Nonetheless, when detecting potholes, the device must be positioned in a restricted manner in the direction in which the vehicle is going. If this is not done, the accuracy of the pothole detection may be affected.

E. Android-based Pothole Tracking System

The work [13] presents the development of an Android-based application for detecting and reporting road surface conditions, which aids drivers in avoiding potholes on the road. The system analyzes data from users' photos and locations to determine road surface conditions. Potholes and other significant road surface irregularities are discovered from the photos using a geotagging approach and the GPS.

The authors also mentioned in the research that a simulator was utilized to detect the potholes using pothole video data recorded by a black-box camera. The width, height, and variation of the pothole may be determined using the simulator. The report, on the other hand, does not specify how these recordings were utilized to discover the potholes.

F. Automatic Pothole and Speed Breaker Detection using Android System

This study describes a vibration-based method for automatic detection of potholes. This study describes the monitoring of road conditions using Android's built-in accelerometer, which captures 3-axis acceleration [14]. The accelerometer data is analyzed to recognize the pothole type and severity level. The pothole location was captured using Android's built-in GPS. This approach is used by Rishiwal and Khan [14] to locate potholes by monitoring the z-axis acceleration, which changes rapidly when a pothole is encountered.

The technique for detecting potholes and speed breakers has a high accuracy of 93.7%, which is a benefit of this study. The results, however, were compared to a manual inspection of potholes and speed breakers on the same route.

G. Gap Trap

Gap Trap detects and reports potholes automatically using sensors on smartphones. Gap Trap uses standard JSON format to post detected potholes to a publicly available interface [18]. The android smartphone utilized is equipped with an accelerometer sensor that continually delivers the force of acceleration felt on its three dimensions in meters per second squared in the form of x, y, and z variables. The accelerometer is continually polled to detect the device's acceleration. When a pothole detection is triggered, the GPS sensor is queried to identify the pothole's position [18]. Google MAP API is also used to display the map location of the detected potholes. However, the search for pothole data is currently not particularly straightforward or difficult.

H. Real-Time Pothole Detection

The research provides a mobile sensing system for pothole detection that makes use of the accelerometer in Android OS devices [19]. This research compared four pothole detection methods based on accelerometer data for several Android OS-based devices. Algorithm testing revealed the optimal setup for each selected algorithm. Their evaluation with a true positive rate of up to 90% employing real-world data was presented. Nonetheless, in terms of application development, the study [19] did not provide sufficient information on the implementation phase.

III. Methodology

For this project development, the spiral model was adopted as it is most suitable for the project that will require continuous enhancement of requirements (see Fig. 1). The spiral model defines iterations, which comprise a set of repeating activities. In each iteration, four main activities are conducted, which include planning, design, development, and testing.

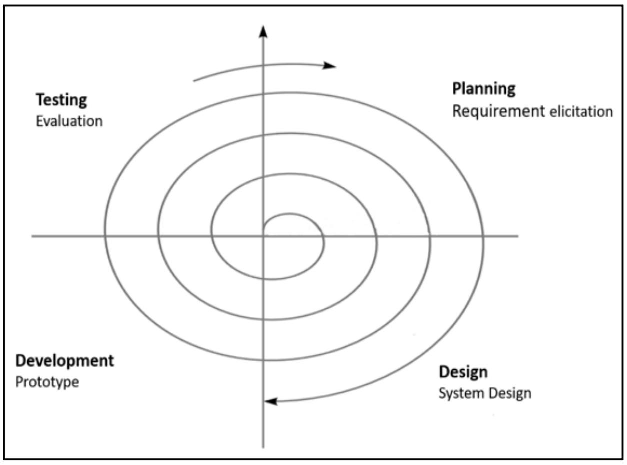

Fig. 1 Spiral model methodology [6]

The same set of activities is then repeated for all the iterations until the completed software is built. By using this model, more features can be added systematically as the project develops. This way, there will be a control mechanism in place for the development phases.

In this project, three iterations were planned and executed. Each of these iterations will be elaborated on in the following paragraphs.

A. *Iteration 1: Preliminary stage*

In the first iteration of the spiral methodology i.e. the preliminary stage, the time and effort estimation for this project were defined and the tasks were distributed amongst the developers. In addition, the requirements for the system were gathered through requirement sources including stakeholders, existing systems, and documents. Initial requirements were derived by analyzing the similar solutions and reviewing system-specific documents such as technical reports of similar systems. The scope and objectives of the system were also identified. It was decided that PotAlert shall be focusing on leveraging the PWA to utilize the features on the smartphone in order to detect potholes.

1) ***Planning:*** Requirements elicitation was performed to identify the functional requirements and quality requirements of the system. The technique that has been used was document analysis. The evaluation was done through existing documents and journal articles to identify the strengths and weaknesses of the existing systems. The collected data was then organized to be listed as the potential new system requirements. Refer to Section 6.0 for the finalized features obtained from the requirements elicitation process.

2) ***Design:*** The initial design for the system has been determined in the first iteration. This crucial step was done to ensure that all the system requirements are achievable to reach the project goals. The use case diagram was designed first to visualize the interaction of the actors with the system. From the use case diagram, the actors involved were identified. Refer to Section 5.0 for the detailed elaboration of the actors involved and Section 6.0 for the use case diagram. Moreover, the functions of the system were elaborated further in the form of use case specifications. In addition, the activity diagram of the system was then made for the road user and authority. Finally, the user interface has also been designed for the road user and the authority to visualize the system better.

3) ***Development:*** The development phase of the first iteration was focused on the road user. The high-fidelity prototype was developed using Figma as shown in Fig. 2 and Fig. 3.

4) ***Testing:*** In the first iteration of testing, the scope of testing was identified. The testing was intended to cover the black box functional testing of the system. The objective of the testing is to ensure that the requirements are satisfied and the system is made for its intended purpose.

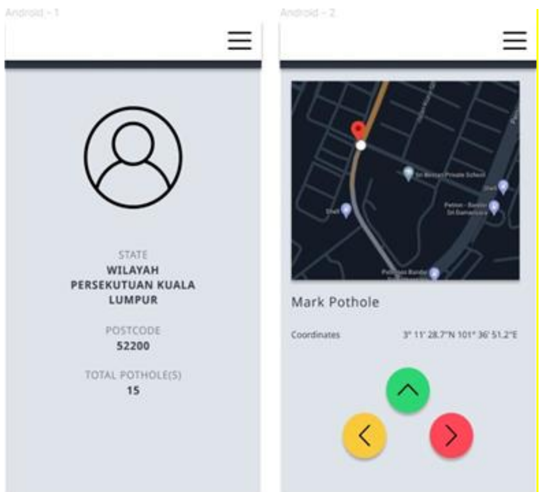

Fig. 2 Figma prototype design-1

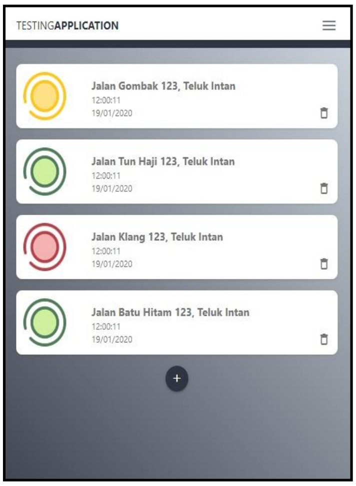

Fig. 3 Figma prototype design-2

B. Iteration 2: Refinement stage

The second iteration of the spiral methodology involved additional requirements elicitation techniques. Here, the tools and languages used for the initial system development were identified. In addition, the database design and the data dictionary were created during the design phase. Aside from that, the user interface for the authority was designed.

1) ***Planning****:* The requirements were further refined from the first phase by using questionnaires. The requirements elicitation that has been applied was observation and brainstorming. The observation was done to collect information on the processes, inputs, and outputs of the system. On the other hand, brainstorming was done by analyzing existing systems like Waze and *i-Tegur* to adopt relevant requirements for PotAlert. Finally, the tools and languages that are required to build the system were finalized accordingly as listed in Tables I and II.

2) ***Design***: In this iteration, an Entity-Relationship Diagram (ERD) was created to cover all possible entities that were relevant for this phase of the system. A total of seven entities have been identified for the PotAlert database including road crew, authority, pothole, road user, progress, and authority report. In addition to the ERD, a complete data dictionary representing each entity was produced. The attribute names and their types were recorded in the data dictionary.

3) ***Development:*** The development phase of the second iteration was focused on the authority perspective. The high-fidelity prototype was developed using Figma.

4) ***Testing****:* In this phase, a preliminary test case was prepared for the authentication module to validate the user login credential.

C. Iteration 3: Final refinement stage

The spiral methodology's third iteration featured the final refinement for all phases. The requirements were validated against the developed system according to the latest requirements list. All final changes for the system were conducted in this iteration. Finally, the completed system is tested.

1) ***Planning****:* The system's scope has been defined to cover primarily two stakeholders in this iteration: the road user and the authority. After significant deliberation and preparation, this was chosen.

2) ***Design****:* Based on the user interface and the high-fidelity prototypes that were built during the first and second iterations of the design phase, the front-end parts for the road user and the authority were developed using HTML and CSS.

3) ***Development****:* In this iteration, the PotAlert system was developed for both the road users and the authority. The systems were developed separately using the tools and languages shown in Tables I and II. The system was developed based on the requirements that have been defined throughout iterations one and two. The prototype that was made in iteration one and two was also used as a guideline during the development.

4) ***Testing****:* In this iteration, test case specifications are prepared accordingly. The system was tested based on the requirements that have been defined in the requirements list. However, the focus of the test was prioritized by the criticality of the requirements. Dynamic unit testing was done throughout the development between the developers using stubs and drivers. This was done to test several functions during the development process. It was helpful when other parts of the system are not completed yet. The results of the test were recorded, and any defects were fixed accordingly. As stated in the previous iterations, the testing only covered functional requirements.

IV. Actors in PotAlert Tracking System

This section describes the actors involved in the PotAlert system that are (1) the road users, and (2) the authority. Their roles will be elaborated on in the subsequent paragraphs.

A. *Road Users*

The road users can either be a driver or a passenger of a vehicle. The road user can report potholes to the authorities when they encounter one on the road, in which they can capture the image or the scene of the pothole or simply give the location of the pothole, and submit it through the system just by using their smartphones. Road users can then view the submitted using their browsers regardless of whether it is on a smartphone, tablet, or laptop. This is possible as PWA may be accessed from any browser. Not only that, the road users can also view the feedback of the reported pothole status.

B. *Authority*

PotAlert allows the authority to view a detailed report of potholes in the form of visualizations and tables. The details in the report are date, location, status, severity, road crew, image, and pothole status. PotAlert also provides the authority with the ability to print the report. The authority is also allowed to update the status of the pothole and assign potholes to the responsible road crew for maintenance purposes.

V. POTALERT FEATURES

The system offers essential features (see Fig. 4), which support the interactions between the road users and authority with the system.

A. *Login and Logout*

The road users and the authority are the main system's stakeholders. To engage with the system, users must first log in with their registered username and password.

B. *Report Pothole*

This use case is triggered by the road user. The road user is able to make a report of a pothole by taking a picture or attaching a picture of the pothole from their photo gallery. The road user needs to provide the location, severity of the pothole, and description to complete the report submission. After the submission, the road user will then receive a report status.

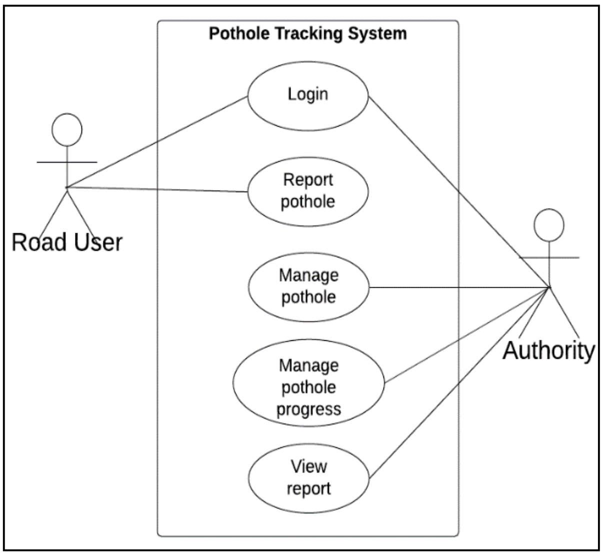

Fig. 4 PotAlert use case diagram

C. *Manage Pothole*

The authority interacts with this manage pothole use case. Pothole complaints filed by road users can be viewed by the authorities. The authority can utilize this report to update the status of the report and offer feedback to the road user. Any road crew can be assigned to the pothole.

D. *Manage Pothole Progress*

For this use case, the authority can view a list of potholes to be fixed and their locations. Once the pothole has been patched, the authority is able to update the pothole progress status.

E. *View Report*

The authority can view the overall list of reported potholes and generate a report from the list.

F. *Visualize Report Data*

The report data visualization includes categories of reports based on (1) weekly reports, (2) pothole severity, and (3) progress in order for the authority to have a better knowledge of the data's patterns, trends, and groupings. See Fig. 13. This information can be used by the authority to make decisions about any future strategic planning or improvements.

VI. POTALERT TECHNOLOGIES

In the development process of the PotAlert, several development environments and programming languages were used to build the system. PWA, Firebase, and Google Map API are some of the few important core elements of the PotAlert development.

A. *Progressive Web App (PWA)*

A PWA is the set of mobile web application development techniques that entails building apps that feel and look like native apps. Using a web stack (JS, HTML, and CSS), a PWA combines various functionalities and a smooth user experience that is associated with native apps [15]-[17]. In other words, PWA is a web app with native-app behaviors. After the installation, a user clicks on its icon on a device home screen and gets straight to the website as shown in Fig. 5 and Fig. 6.

Fig. 7 is a representation of the architecture diagram for PotAlert. The diagram presents a comprehensive view of how the system functions and how each tool and framework were utilized for the development of the system. PotAlert is accessible by two types of stakeholders, which are the authority and the road user. Users may use any browser on any device including PCs, tablets, and smartphones to access the system

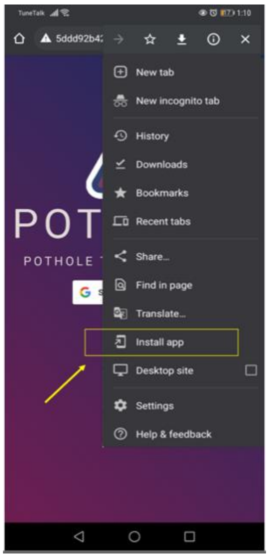

Fig. 5 PotAlert installation

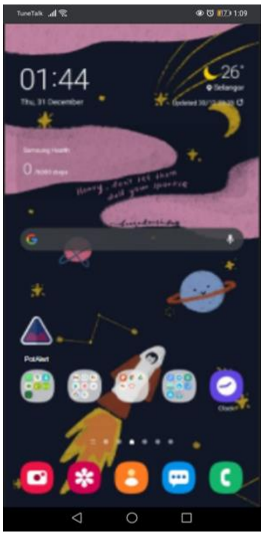

Fig. 6 PotAlert icon on home screen

B. *PotAlert System Architecture*

The system adopted Svelte, which is a front-end Javascript framework that provides client start-up and run-time performance. Node JS handles the runtime environment and acts as the mediator between the framework and the Firebase database.

C. *Programming Languages, Frameworks, Libraries, and Tools Used*

The PotAlert system was developed using the languages and tools as listed in Tables I and II. For each language and tool, a brief description is included in the respective tables.

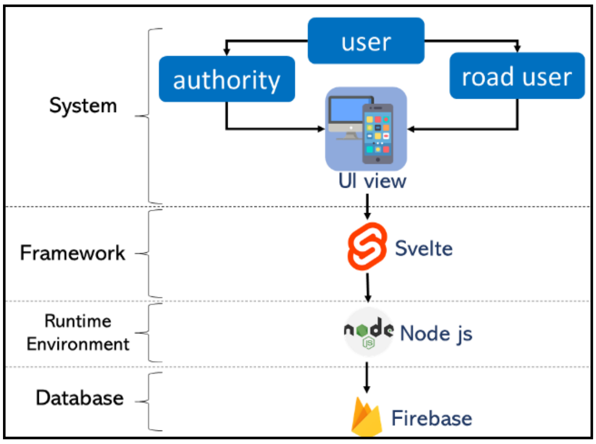

Fig. 7 PotAlert Architecture diagram

TABLE I
PROGRAMMING LANGUAGES, FRAMEWORKS, AND LIBRARIES USED TO DEVELOP POTALERT

Languages/ Framework/ Libraries	**Description**
HTML, CSS, JavaScript	Used in web development to organize the content of web pages and provide the necessary building blocks for the web app. Apart from that, CSS aids in the look and the feel of the page, and Javascript assists in its programming capabilities.
Svelte	Javascript framework for interactive web page building for the system.
Node.js	Server-side Javascript run-time environment used for web app development and serves as the backend of the system.
D3.js	A JavaScript library is used to manipulate documents using data. D3.js uses HTML, SVG, and CSS to visualize data in the form of bar charts and donut charts effectively for this system.

TABLE II
TOOLS USED IN DEVELOPMENT

Tools	**Description**
Figma	Used for prototyping and user interface (UI) design.
Lucidchart	The tool is used to sketch out early visualization and collaborative ideas with the use of a flowchart. It also aids in the early stages of system design.
Ngrok	Used to run the system web server running on the local machine to the internet to test the real-time web UI where HTTP traffic running can be observed.
Visual Studio Code	Integrated development environment for coding and implementation.
Firebase	A cloud-hosted database. The reporting data made by the road users are stored as JSON on Firebase. The data is also synchronized in real-time if it is added, edited, or deleted.
Google Maps API	Embedded in the road user system to detect the location of the pothole.

D. *User Interfaces for Road Users*

The main user interfaces for the road user are the login page, the user profile, and then create report page as shown in Figures 8-11.

Fig. 8 PotAlert Login page

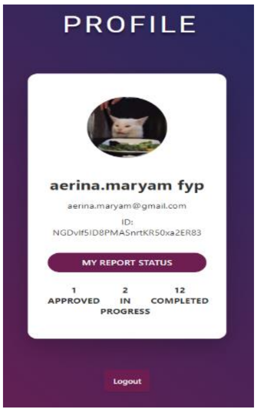

Fig. 9 PotAlert Profile page

E. *User Interfaces for Authority*

The user interfaces for the authority are the pothole report list and the visualization page as shown in Figures 12 and 13. As shown in Fig. 12, the authority is able to view, update and delete the pothole status.

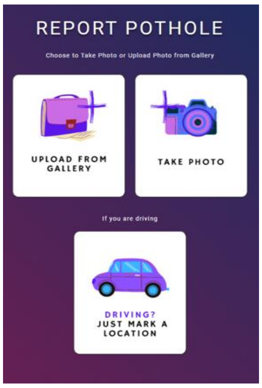

Fig. 10 PotAlert Report page

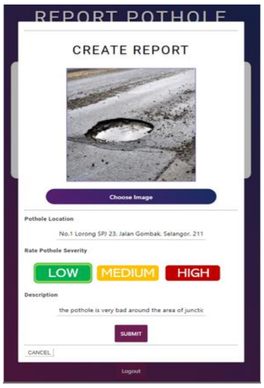

Fig. 11 Create report

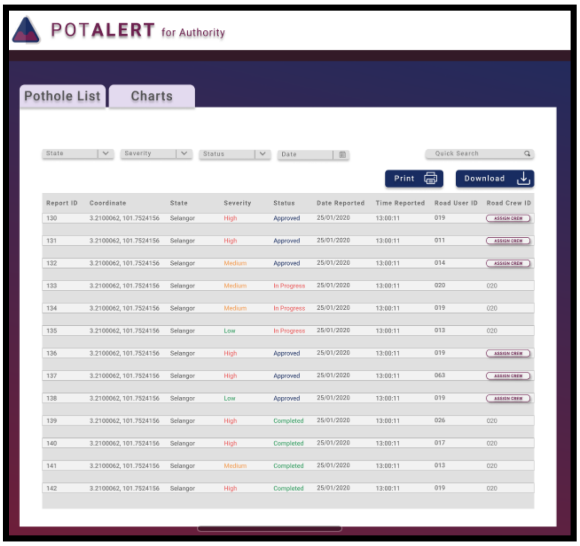

Fig. 12 PotAlert for Authority pothole list status

Meanwhile, Fig. 13 shows the available type of visualizations including visualizations of the number of reported potholes, pothole severity, and pothole maintenance progress. These visualizations can be derived from the existing pothole data in the database.

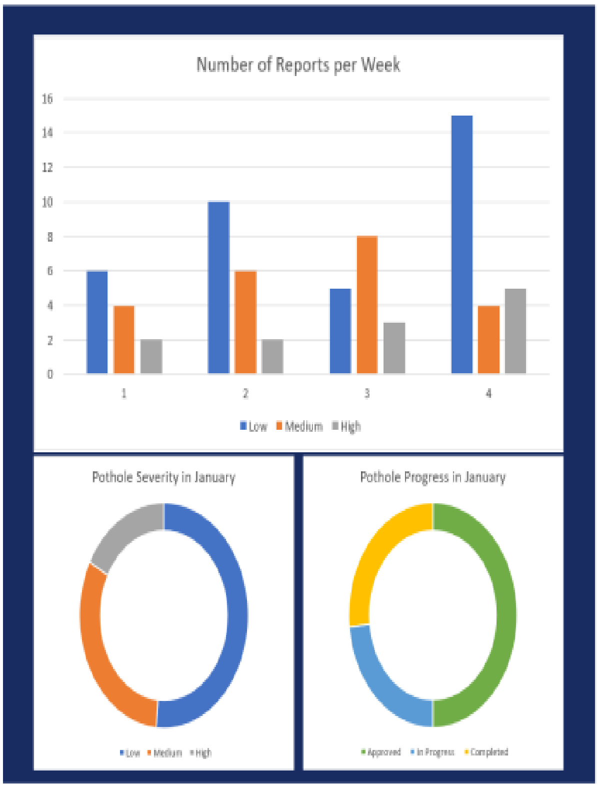

Fig. 13 Visualizations of Number of reports, Pothole severity and Pothole Maintenance Progress

VII. Limitations and Constraints

Throughout the project's development, certain restrictions have been found. To begin with, there is presently no way for a road user to alter their report after it has been created in the system. The authority also lacks the ability to manually enter a new report into the system. Due to time restrictions, the project focuses on the road users' reporting function and the authority's pothole management. As a result, several other functions, such as road crew and administrator, are not yet completely built and may need to be added to the PotAlert system in the future.

Furthermore, because the app is built with Firebase technology, the current administration role can only regulate user access and accounts through the Firebase dashboard, as a complete administrator function would require a lot of time to construct. This specific functionality will be expanded together with the other features in the system's future development.

In addition, it is worth noting that because actual data from real users is difficult to obtain due to the present pandemic and movement control order in the nation due to Covid 19, the system only utilizes dummy data for system implementation in the authority part of data visualization and reporting.

VIII. Future Work

There are a few enhancements that could potentially be added to the system to improve its performance and usefulness. As specified in the limitations, facilities for amending road user reports and removing authority reports should be included in future PotAlert development. This feature is especially desirable as it increases the level of error tolerance in the system.

Apart from that, a feature that automatically detects and collects redundant pothole reports that point to the same location in one group may also be added in the future work of the system. This would allow the authority to rectify if a pothole has received multiple identical reports, which at the same time, could indirectly inform them of the criticality of the pothole condition.

Because our effort for this work primarily focuses on the adoption of PWA, the optimal pothole detecting algorithm to enhance the accuracy of the pothole detection will be addressed in future updates. This could also potentially be done with an addition of automatic pothole detection using the camera, which functions on image recognition and processing capability. Once a pothole is detected with the camera, the location could also be plotted in the same instance. In addition to the image processing, there could also be the determination of pothole severity using data training models.

In addition, a feature to allow the road crews to update the maintenance progress of the reported potholes to the authority could also be added. The road crew should have the ability to be navigated to the pothole location through Google Maps support in the system. With a list of the pothole reports, the road crew would have a smoother experience with communication and interaction with the authorities. Henceforth, it would facilitate the integration of features with the overall system.

Lastly, it is also crucial to ensure that this system works effectively and accurately. Therefore, trial runs should be conducted for future works, to ensure that accuracy is achieved and a seamless experience can be provided to the PotAlert users.

IX. Conclusion

This paper presents an approach for detecting and reporting potholes on the road in a pothole tracking system, PotAlert, and at the same time encouraging communication between road users and the authorities. The authorities are also provided with analytical data visualization of the reports to support decision-making. The main drive in developing this system is to provide convenience and ease of access for both road users and authority in the process of reporting and managing the potholes.

The future work for the system can be further upgraded and broadened with ample features that can accommodate more user experience among road users and authorities. Moreover, after the potholes are repaired, the roads become safer for commuters and their vehicles. Although it would not be a prohibitively expensive or time-consuming strategy, it would encourage individuals to contribute to public safety.

Acknowledgment

We would like to express our sincere gratitude and profound appreciation to Kulliyyah of Information and Communication Technology, International Islamic University Malaysia for all the support.

References

[1] M. M. Abdul Manan, and A. Várhelyi, “Motorcycle fatalities in Malaysia,” *IATSS Research*, vol. 36, no. 1, pp. 30–39, July 2012.

[2] M. A. Elliott, C. J. Baughan, J. Broughton, B. P. Chinn, G. Grayson, J. Knowles, L. Smith, and H. Simpson, “Motorcycle safety: A scoping study,” Transport Research Laboratory (Great Britain), Crowthorne, Berkshire, England, 2003.

[3] D. S. Fernandez. (2017, December 17). How to Report Potholes in Selangor using Waze. *DSF.my*. [Online]. Available: https://www.dsf.my/2017/12/how-to-report-potholes-in-selangor-using-waze/

[4] S. Khor. (2018, December 19). You Can Now Report Potholes and Broken Traffic Lights To Local Authorities Via A Mobile App. SAYS. [Online]. Available: https://says.com/my/tech/govt-launches-i-tegur-official-app-to-file-complaints-about-potholes

[5] A. K. Maurya, A. Yadav, P. Arote, A. Baviskar, and J. Baviskar, “Real-time Arduino-based depth sensing for road condition monitoring,” *International Journal of Recent and Innovation Trends in Computing and Communication (IJRITCC)*, vol. 4, no. 4, pp. 947-952, 2016.

[6] R. J. Leach, Introduction to software engineering. Boca Raton: Chapman&Hall/CRC, 2020.

[7] A. Karnik, “Performance of TCP congestion control with rate feedback: TCP/ABR and rate-adaptive TCP/IP,” M. Eng. Thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.

[8] J. Padhye, V. Firoiu, and D. Towsley, “A stochastic model of TCP Reno congestion avoidance and control,” Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.

[9] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1997.

[10] T. Kim, and S. Ryu, “Review and analysis of pothole detection methods,” *Journal of Emerging Trends in Computing and Information Sciences*, vol. 5, no. 8, pp. 603-608, 2014.

[11] B. Ch’ng, (2020, November 3). More than 50k potholes in Selangor reported on Waze. [Online]. Available: https://www.thestar.com.my/metro/metro-news/2020/11/03/more-than-50k-potholes-in-s039gor-reported-on-waze. *The Star*.

[12] D. A. Casas-Avellaneda, and J. F. Lopez-Parra, “Detection and localization of potholes in roadways using smartphones,” *DYNA*, vol. 83, pp. 156-162, 2016.

[13] A. Bamne, and O. Shinde. “Pothole Tracking System Using Android,” *International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE)*, vol. 5, no. 6, pp. 350-354, 2016.

[14] V. Rishiwal, and H. Khan, "Automatic pothole and speed breaker detection using android system," *in 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2016, pp. 1270-1273.

[15] V. S. Magomadov, "Exploring the role of progressive web applications in modern web development," *Journal of Physics: Conference Series*, vol. 1679, no. 2, pp. 022043, IOP Publishing, 2020.

[16] L. E. Nugroho, A. G. H. Pratama, I. W. Mustika and R. Ferdiana, "Development of monitoring system for smart farming using Progressive Web App," in 9th International Conference on Information Technology and Electrical Engineering (ICITEE), 2017, pp. 1-5.

[17] C. J. Gómez-Sierra, "Design and development of a PWA-Progressive Web Application, to consult the diary and programming of a technological event," *IOP Conference Series: Materials Science and Engineering,* vol. 1154, no. 1, pp. 012047, 2021.

[18] S. Burgart, (2014), Gap Trap: A pothole detection and reporting system utilizing mobile devices. [Online]. Available: https://www.cs.umd.edu/sites/default/files/scholarly_papers/Burgart.pdf

[19] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, and L. Selavo, "Real-Time Pothole Detection Using Android Smartphones with Accelerometers," in 7th IEEE *International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, 2011, pp. 1-6.