5th International Seminar of Research Month 2020
Volume 2021
http://dx.doi.org/10.11594/nstp.2021.0909

NST PROCEEDINGS

OPEN ACCESS

Conference Paper

Analysis and Design of a Crowdsourcing-Based Flooded Road Reporting System Indonesia (FROSTID) Application

Eka Prakarsa Mandyartha*[1], Asif Faroqi[2]

[1]Informatics Department, Faculty of Computer Science, Universitas Pembangunan Nasional "Veteran" Jawa Timur, Indonesia
[2]Information System Department, Faculty of Computer Science, Universitas Pembangunan Nasional "Veteran" Jawa Timur, Indonesia

*Corresponding author:
E-mail: eka_prakarsa.fik@upnjatim.ac.id

ABSTRACT

The problem of flooded road infrastructure mainly occurs in urban areas, due to low soil bearing capacity, high rainfall intensity, and failure of rainwater flow to the drainage system. Flooded roads, when traversed by motorized vehicles, will be very detrimental to the rider. Flood water can damage engines, transmissions, electrical components, and motor vehicle electronic computer systems. Water is also very dangerous to car safety devices, such as airbags and braking systems that use ABS (Antilock Braking System). Also, there may be health problems involving fungi and bacteria, due to damp vehicles, especially on cars. Other problems related to water may not be immediately apparent, such as rust and corrosion, which will develop over time. The flooded road reporting system is an early warning system that provides information to road users so they can take anticipatory steps. This research proposes a crowdsourcing-based flood road reporting system, called Frostid (Flooded Road Reporting System Indonesia), that involves local road users in collecting flood information. Frostid application users can not only access geospatial information for flooded roads but also can collect location data using the Global Positioning System (GPS) and upload the location points of flooded roads using their smartphones. The Frostid application uses a navigation map to describe the location of flooded roads. Flood information stored in the Frostid application is built entirely by local road users. The Frostid application is built on a mobile Android platform.

Keywords: Flood, reporting system, crowdsourcing, road, local road user

Introduction

Flood is the main problem experienced during the rainy season every year in Indonesia, especially in big cities (Ali et al., 2019). Flooding occurs when the river level rises due to high rainfall, reservoirs that are unable to accommodate the volume of rainwater or a poor drainage system. The water level that exceeds the riverbank, can cause the area around the river to be flooded. Floods can occur due to low soil bearing capacity. It will increase in groundwater, potentially causing large-scale flooding (See, 2019).

Floods can also be caused by hurricanes or typhoons, or rising sea levels due to climate change or from tsunami waves. Besides, flooding can be caused by rainwater flows failing to flow into the drainage system or ditch due to the full capacity of the gutters, previously submerged roads, and road structures that prevent water from flowing into the drainage system. The problem of flooded road infrastructure mainly occurs in urban areas, for example, Jakarta and Surabaya, due to the high number of impermeable surfaces (the land cannot absorb water), thus preventing water infiltration into the ground.

One type of infrastructure network is roads. Road infrastructure is built to support the welfare of the people, especially in an economy, which is highly dependent on the mobility of people and goods (Pregnolato et al., 2017). Changes in extreme weather, climate, frequency, and severity of natural disasters

How to cite:
Mandyaetha, E. P., & Faroqi, A. (2021). Analysis and design of a crowdsourcing-based Flooded Road Reporting System Indonesia (FROSTID) application. *5th International Seminar of Research Month 2020*. NST Proceedings. pages 53-63. doi: 10.11594/ nstp.2021.0909

related to flooding can affect the quality of road infrastructure available to the community (Degrossi et al., 2014).

Flooded roads will be very detrimental to its users, such as motorbikes, cars, and trucks. Flood water can damage engines, transmissions, electrical components, and motor vehicle electronic computer systems. Floodwaters also seriously endanger car safety devices, such as airbags and braking systems based on the Anti-lock Braking System (ABS). Also, there may be health problems involving fungi and bacteria, due to damp vehicles, especially on cars. Other problems related to water may not be immediately apparent, such as rust and corrosion, which will develop over time.

The flood reporting system is an early warning system that provides information to residents so they can take anticipatory steps. The involvement of residents in gathering flood information is recommended (Cools et al., 2016) so that the information can be used as an early warning. Citizen involvement in information gathering for environmental monitoring is becoming increasingly common and is often referred to as crowdsourcing (Bonney et al., 2009). In general, crowdsourcing is applied to conservation and biodiversity, but recently, it is also being applied in various fields in geosciences (Harrison & Johnson, 2016). The crowdsourcing approach empowers the "crowd" to collect sufficient data into the knowledge base through the internet (Minh et al., 2017; Lease & Yilmaz, 2012; See et al., 2016). As its main purpose, the data or knowledge provided by the crowd is then shared with other users.

Several studies have developed the application of the crowdsourcing approach to flood management for disaster management systems and flood mitigation (Eilander et al., 2016; Ushahidi or 'testimony': Web 2.0 tools for crowdsourcing crisis information (PLA 59), 2009; Koswatte et al., 2018; Pánek et al., 2017; Ludwig et al., 2015; Middelhoff et al., 2016; Hendricks et al., 2018). However, this research proposes an online flood road reporting application that focuses on traffic navigation maps for road users. This proposed application involves active citizen participation (crowdsourcing) where the collection and dissemination of flood information are carried out by the crowd of residents. Our application is called the Flooded Road Reporting System Indonesia (Frostid). Flood road reports were visualized using the Mapbox Maps SDK (Mapbox, 2020). The Mapbox Maps SDK is a collection of open-source libraries that can be called from a web page or mobile application to build various interactive map elements. Frostid application users not only have access to geospatial information for flood roads but also can collect location data using the Global Positioning System (GPS) and upload flood road location points using their smartphones. The flood information stored in the Frostid application is built entirely by residents as users. The Frostid application is built on the Android mobile platform. Using the Frostid application, anyone can click on a street location point on the map. If they know the road is flooded, they can report it, then the map will update the flood road information for everyone in real time. Users can zoom in and out of the map with a pinch-to-zoom gesture to visualize the roads which are being reported as floods, and decide what to do next.

Material and Methods

The Frostid application (Flooded Road Report System Indonesia) provides geospatial information on flood roads. Geospatial data collection through crowdsourcing takes advantage of citizen participation. Frostid is built within a Software Development Life Cycle (SDLC) framework that adopts the Waterfall model shown in Figure 1. The waterfall model is a sequential software development method, consisting of four interrelated stages including system analysis and design. The four stages include system planning, system analysis, system design, and system implementation. System planning is a description of the estimated physical requirements, the requirements used to support system development and operation after the system is implemented.

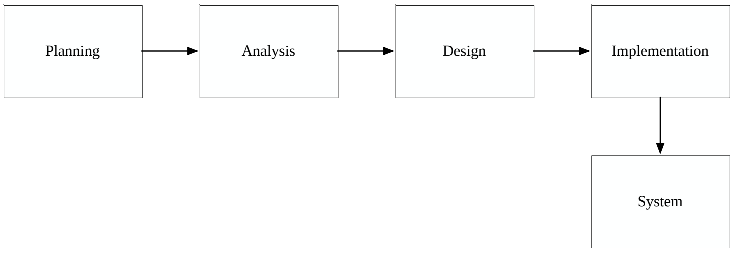


Figure 1. The waterfall model

System analysis

System analysis is the decomposition of a complete system into its parts to identify the expected requirements. The system analysis stage generates a workflow system to analyze system requirements. The system workflow describes how the system functions are performed. It determines the work procedures in each system function. A system function is an activity in the business workflow that involves software. Meanwhile, work procedures are a series of activities based on a time sequence involving several components, especially the role of system users. The Frostid application has system functions such as user registration, user login, adding flood reports, deleting flood reports, viewing flood report lists, viewing flood location maps, viewing flood report details, resetting passwords, and reporting bugs. Besides, the Frostid System Manager as a support system should also be able to view flood reports, deactivate users, change flood level master data, and view bug reports.

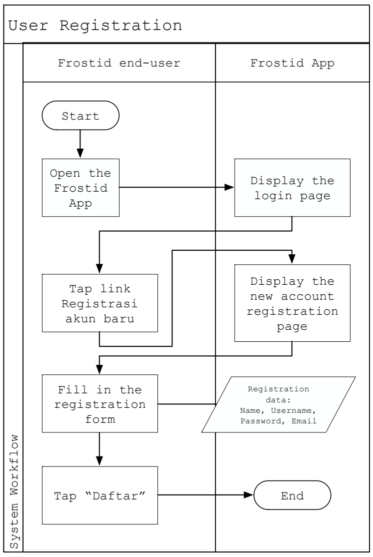

User Registration
Frostid end-user
Frostid App
System Workflow
Start
Open the Frostid App
Display the login page
Tap link Registrasi akun baru
Display the new account registration page
Fill in the registration form
Registration data: Name, Username, Password, Email
Tap "Daftar"
End

Figure 2. the flow of event: user registration

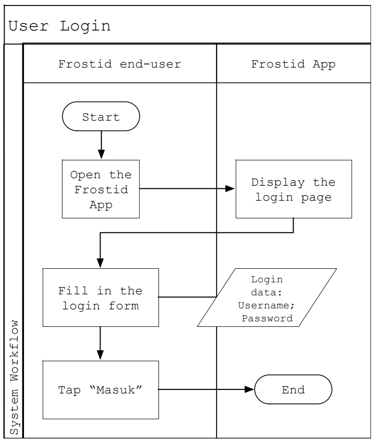


Figure 3. The flow of event: user login

System analysis is the decomposition of a complete system into its parts to identify the

Figure 2 describes the flow of the "user registration" activity. This system function accommodates new Frostid user registrations. To view flood reports and add flood reports, Frostid users are required to log in with their accounts. If a new user does not have an account, the user can register by entering data in the form of their name, username, password, and email. Figure 3 describes the flow of the "user login" activity.

The system function for reporting the location of the flood roads is shown in Figure 4. To report a flood waypoint, the user presses (tap) the "+" button on the map. Then the user fills in the report form, which consists of a photo data field of the flood location, flood level, and a detailed description of the flood conditions on the road. The flood level field is mandatory. Flood level is the severity of the flood. The flood level is classified into three levels. They are mild, moderate, and dangerous. After the report data is saved to the system, the system will update the flood road report data on the navigation map. Figure 5 describes the activity of deleting flood reports. The system function allows the cancellation of a flood road report, for example when there is a correction to the flood report data.

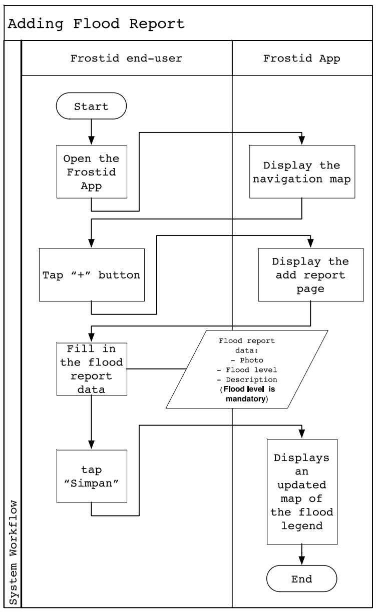


Figure 4. The flow of event: adding flood report

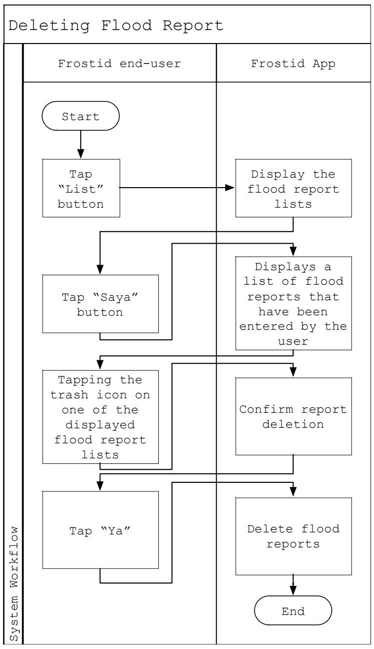

Deleting Flood Report
Frostid end-user
Frostid App
System Workflow
Start
Tap "List" button
Display the flood report lists
Tap "Saya" button
Displays a list of flood reports that have been entered by the user
Tapping the trash icon on one of the displayed flood report lists
Confirm report deletion
Tap "Ya"
Delete flood reports
End

Figure 5. The flow of event: deleting flood report

Figure 6 describes the flow of the event " view flood location maps ". The system function accommodates the user's need to view reports of flood waypoints on a traffic navigation map. The navigation map will update the flood road report data periodically, every second, in real-time. Figure 7 describes the flow of the event "view flood report details". The system function described in Figure 7 accommodates the need for users to view more detailed information from each flood waypoints report, for example, a description of the user-reported flooded road, flood level, and a photo of the flood road location. To see this detailed description, first, the user must tap the "water icon" on the navigation map which is a flood report by other users. Figure 8 shows the flow of the event "view flood report lists". This system function aims to display a list of flood reports around the user's current location.

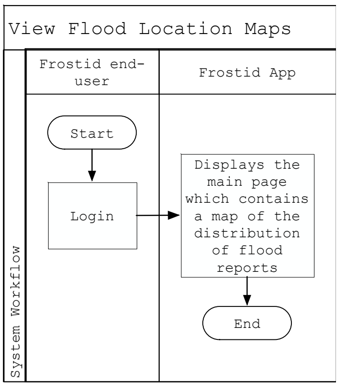

View Flood Location Maps
Frostid end-user
Frostid App
System Workflow
Start
Login
Displays the main page which contains a map of the distribution of flood reports
End

Figure 6. The flow of event: view flood location maps

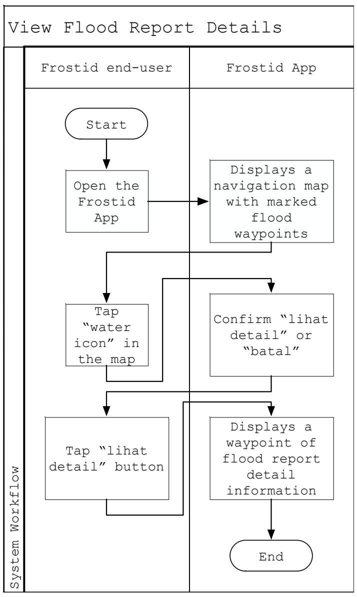

View Flood Report Details
Frostid end-user
Frostid App
System Workflow
Start
Open the Frostid App
Displays a navigation map with marked flood waypoints
Tap "water icon" in the map
Confirm "lihat detail" or "batal"
Tap "lihat detail" button
Displays a waypoint of flood report detail information
End

Figure 7. The flow of event: view flood report details

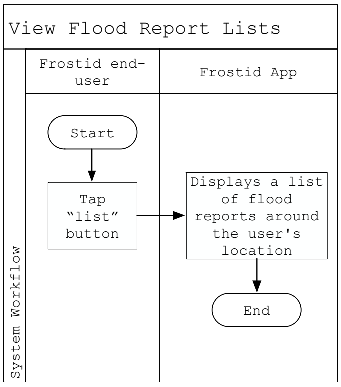


Figure 8. The flow of event: view flood report lists.

Figure 9 describes the system flow for resetting user passwords. This system function allows users change their password, if they are unable to log into Frostid in case of forgetting the account password. The password reset flow starts with the user pressing the "*profil*" menu, then the system displays the user profile page. The user selects the "*ubah password*" menu. The system displays a change password page that contains fields for the old password and new password. The user enters the old password, and the new password in the fields provided then presses the "*ubah*" button. In the end, the system displays information that the password was successfully changed. The system functions for bug reporting are shown in Figure 10. Bug reporting is a feedback feature for users if users want to report bugs found in Frostid application, for example, certain system failures.

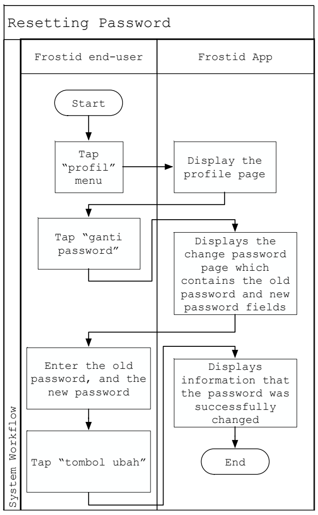

Resetting Password
Frostid end-user
Frostid App
System Workflow
Start
Tap "profil" menu
Display the profile page
Tap "ganti password"
Displays the change password page which contains the old password and new password fields
Enter the old password, and the new password
Displays information that the password was successfully changed
Tap "tombol ubah"
End

Figure 9. The flow of Event: Resetting Password.

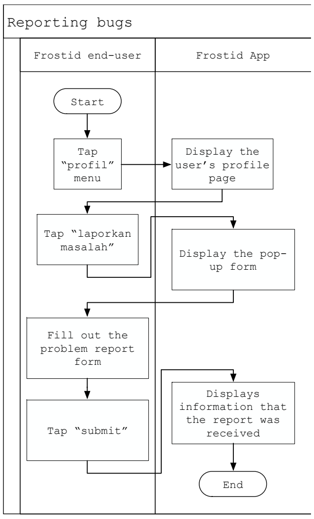


Figure 10. The flow of Event: Reporting bugs.

Figure 11-14 describes the activities in the Frostid System Manager including view flood reports, deactivate users, change flood master data, and view bug reports.

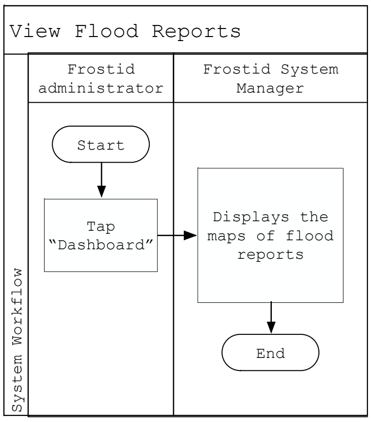


Figure 11. Flow of Event: View Flood Reports.

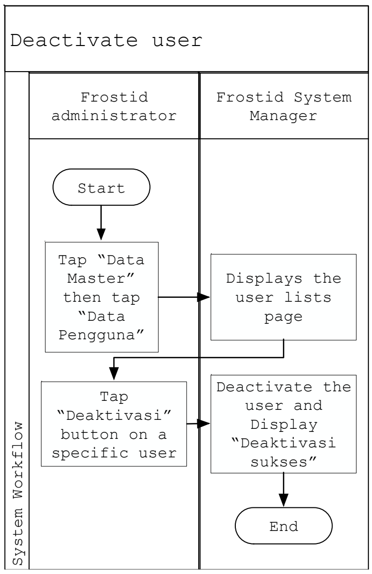

Deactivate user
Frostid administrator
Frostid System Manager
System Workflow
Start
Tap "Data Master" then tap "Data Pengguna"
Displays the user lists page
Tap "Deaktivasi" button on a specific user
Deactivate the user and Display "Deaktivasi sukses"
End

Figure 12. Flow of Event: Deactivate user.

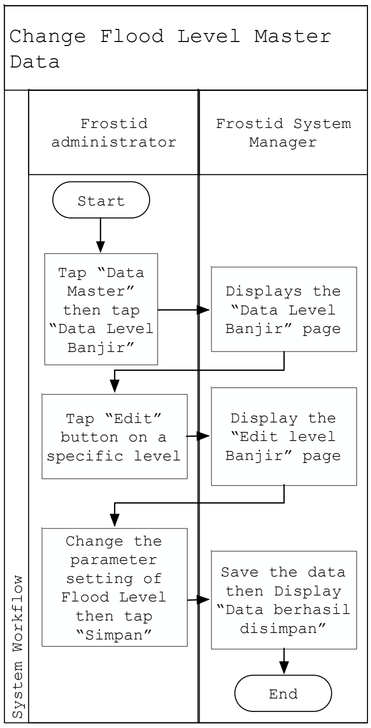

Change Flood Level Master Data
Frostid administrator
Frostid System Manager
System Workflow
Start
Tap "Data Master" then tap "Data Level Banjir"
Displays the "Data Level Banjir" page
Tap "Edit" button on a specific level
Display the "Edit level Banjir" page
Change the parameter setting of Flood Level then tap "Simpan"
Save the data then Display "Data berhasil disimpan"
End

Figure 13. Flow of event: change flood level master data.

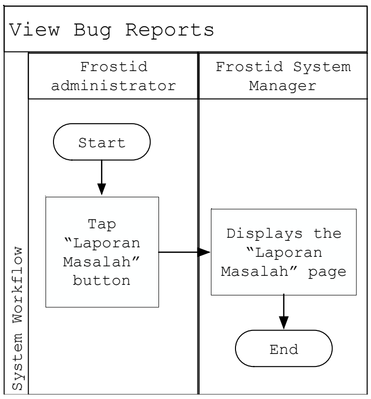


Figure 14. Flow of Event: View Bug Reports.

System design

The stage after the analysis of the systems development cycle is system design. The purpose of this system design is to describe a computer program to be built. The use case diagram of the Frostid system is shown in Figures 15 and 16. The diagram illustrates the interaction of actors with system functions. The use case diagram is divided into two, the use case diagram for mobile applications and the system manager. Both use case diagrams have a login use case, which has a relationship "include" with the main use case. The login use case must be satisfied first, as a condition for the occurrence of another event.

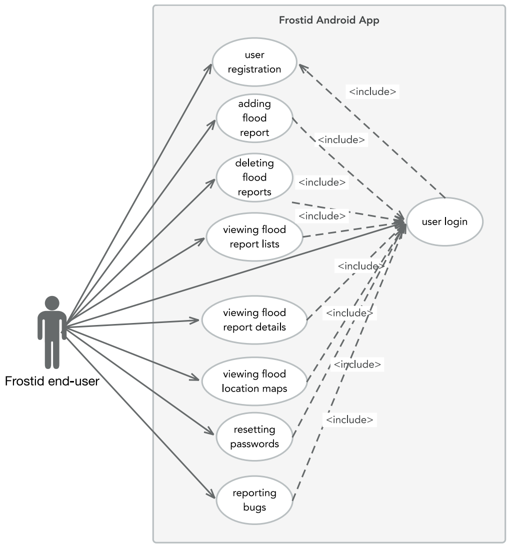


Figure 15. Frostid android app use case diagram

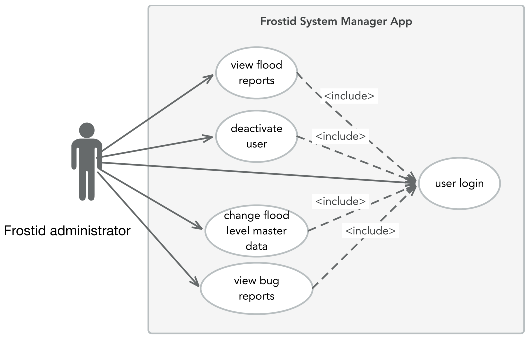


Figure 16. Frostid system manager uses a case diagram

Figure 17 shows the general architecture of the system being built. The system consists of two parts, the Frostid android based mobile application, and the support system.

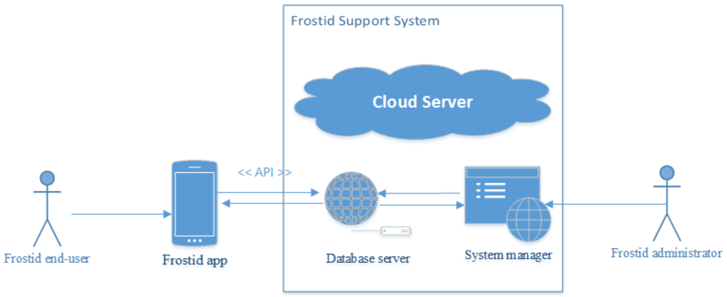


Figure 17. Frostid system architecture

The Frostid system consists of three components including the Frostid mobile application, database server, and system manager. The Frostid mobile application communicates with the support system through a cloud server. The support system serves to support the storage and provision of Frostid application data through a database server, as well as for data management through the system manager. An interface, an Application Programming Interface (API), is required for communication between the mobile application and the database server. The interface is located on the support system side of the cloud server. Another component, the system manager, is used for crowdsourcing data management, flood reports, and user data management. The two actors involved in the Frostid system are end-users and administrators. End-users are Frostid application users, while administrators are responsible for the system manager.

The cloud server has the task of receiving and sending requests (requests) from the mobile application. The request from the mobile application is not received directly by the database server but is forwarded to the database server through the API application. Likewise, data obtained from the query results to the database server, then forwarded by the API to the Frostid mobile application. The API application built is the REST-API using the Lumen framework. The database server used is MySQL.

Figure 18 shows a diagram of the Frostid application database. There are eight tables, namely the reports table, report_levels, report_pictures, users, user_logs, user_levels, problems, and report_logs. The reports table functions to store flood report data. Flood level attributes, such as flood level icon color, and flood level icon description, are defined in the report_levels table. The user's table stores Frostid user registration data that is used to log in. The user_logs table functions to store user activity data while using the Frostid application. The user_levels table defines the user's access rights to the system. Bug report data reported by users is stored in the problems table. Meanwhile, the report_logs table stores flood report data that was deleted by users.

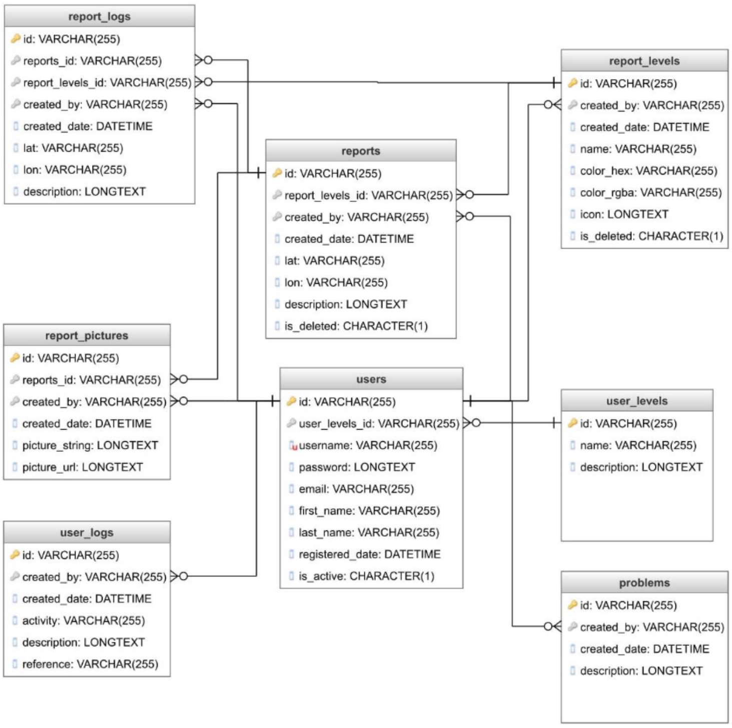


Figure 18. Frostid system entity relational database

Results and Discussion

The Frostid system is implemented with the experimental environment specifications shown in Table 1. The results of the implementation are shown in Figure 19-21. Figure 19 shows the main page. This page contains a traffic navigation map that provides flood waypoint information. The flood waypoint is represented by a water icon. The water icon represents the flood level. The flood level is divided into three, including low level, medium level, and high level. Figure 20 is a page for adding flood report data. Meanwhile, Figure 21 shows the flood report list page around the user's location.

Table 1. The Experimental Environment Specifications

Mobile application programming language	Android Kotlin
Database server	MySQL
API programming language	PHP with Lumen Framework
Traffic Navigation Maps	MapBox
System Manager programming language	PHP with Laravel Framework

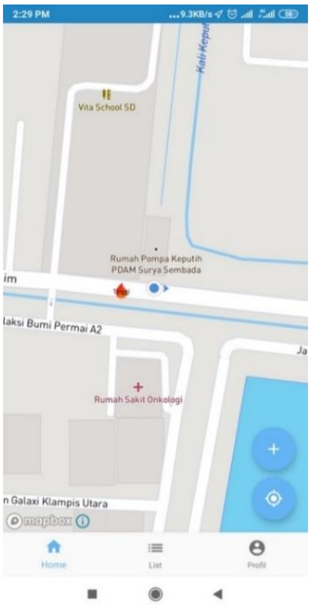
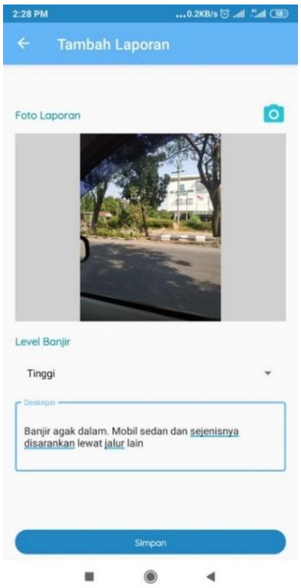
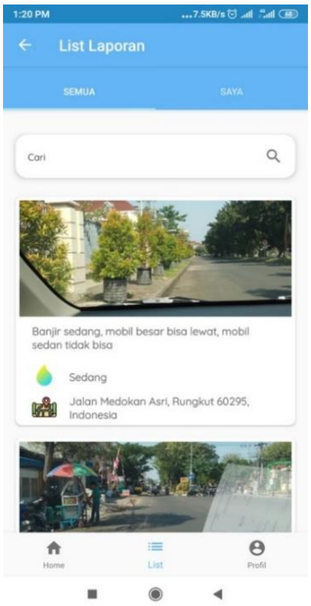


Figure 19. Frostid Main Page



Banjir agak dalam. Mobil sedan dan sejenisnya disarankan lewat jalur lain

Figure 20. Frostid Add Flood Report Page



Banjir sedang, mobil besar bisa lewat, mobil sedan tidak bisa

Jalan Medokan Asri, Rungkut 60295, Indonesia

Figure 21. Frostid flood report lists page

Experimental testing aims to ensure the application meets predetermined functional requirements. In this study, the black box testing method was employed. Table 2 describes the results of the Frostid functional requirements testing.

Table 2. The Experimental Testing Results

	No.	Functional Requirement Testing	Testing Result Status
Frostid Mobile Application	U1.1	User registration	OK
	U1.2	User login	OK
	U1.3	Adding flood report	OK
	U1.4	Deleting flood report	OK
	U1.5	Viewing flood report lists	OK
	U1.6	Viewing flood report details	OK
	U1.7	Viewing flood location maps	OK
	U1.8	Resetting password	OK
	U1.9	Reporting bugs	OK
Frostid System Manager	U2.1	View flood reports	OK
	U2.2	Deactivate user	OK
	U2.3	Change Flood Level Master Data	OK
	U2.4	View bug reports	OK
	U2.5	User login	OK

The application test results are shown in Table 2. It says that the Frostid application has met the functional requirements that have been defined.

Conclusion

In this study, an application that uses a crowdsourcing approach was developed to provide road flood information to local road users, especially motorists in Indonesia. All flood road information is accommodated in the Frostid application, built entirely by user participation. The Frostid application runs on Android mobile devices. The results of the functional testing using the black box method indicate that the Frostid application can run well and meet its functional requirements.

Acknowledgment

The author would like to thank LPPM UPN "Veteran" Jawa Timur for their support by providing funding for this research.

References

Ali, M. S. S., Arsyad, M., Kamaluddin, A., Busthanul, N., & Dirpan, A. (2019). Community based disaster management: Indonesian experience. *IOP Conference Series: Earth and Environmental Science*, 235, 012012.

Bonney, R., Cooper, C. B., Dickinson, J., Kelling, S., Phillips, T., Rosenberg, K.V., & Shirk, J. (2009). Citizen science: A developing tool for expanding science knowledge and scientific literacy. *BioScience*, *59*(11), 977–984. https://doi.org/10.1525/bio.2009.59.11.9

Cools, J., Innocenti, D., & O'Brien, S. (2016). Lessons from flood early warning systems. *Environmental Science & Policy*, *58,* 117–122. DOI: 10.1016/j.envsci.2016.01.006

Degrossi, L. C., de Albuquerque, J. P., Fava, M. C., & Mendiondo, E. M. (2014). Flood citizen observatory: a crowdsourcing-based approach for flood risk management in Brazil. *In: SEKE,* 570–575.

Eilander, D., Trambauer, P., Wagemaker, J., & Van Loenen, A. (2016). Harvesting social media for generation of near real-time flood maps. *Procedia Engineering*, *154*, 176–183.

Harrison, S. E., & Johnson, P. A. (2016). *Crowdsourcing the disaster management cycle*. [article] International Journal of Information Systems for Crisis Response and Management (IJISCRAM). Available at: <www.igi-global.com/article/crowdsourcing-disaster-management-cycle/185638> [Accessed 22 Oct. 2020].

Hendricks, M. D., Meyer, M. A., Gharaibeh, N. G., Van Zandt, S., Masterson, J., Cooper, J. T., Horney, J. A., & Berke, P. (2018). The development of a participatory assessment technique for infrastructure: Neighborhood-level monitoring towards sustainable infrastructure systems. *Sustainable Cities and Society*, *38,* 265–274. doi: 10.1016/j.scs.2017.12.039

Koswatte, S., McDougall, K., & Liu, X. (2018). VGI and crowdsourced data credibility analysis using spam email detection techniques. *International Journal of Digital Earth*, *11*(5), 520–532. https://doi.org/10.1080/17538947.2017.1341558

Lease, M., & Yilmaz, E. (2012). Crowdsourcing for information retrieval. *ACM SIGIR Forum*, *45*(2), 66–75. DOI: 10.1145/2093346.2093356

Ludwig, T., Siebigteroth, T. and Pipek, V. (2015). CrowdMonitor: Monitoring Physical and Digital Activities of Citizens During Emergencies. In: L.M. Aiello and D. McFarland, eds. *Social Informatics*, Lecture Notes in Computer Science. Cham: Springer International Publishing, 421–428.

Middelhoff, M., Widera, A., van den Berg, R.P., Hellingrath, B., Auferbauer, D., Havlik, D. and Pielorz, J., 2016. Crowdsourcing and crowdtasking in crisis management: Lessons learned from a field experiment simulating a flooding in the city of the Hague. In: *2016 3rd International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*. 2016 3rd International Conference on Information and Communication Technologies for Disaster Management (ICT-DM), 1–8.

Minh, Q.T., Chi, T. N., & Toulouse, M. (2017). Toward a Crowdsourcing-Based Urban Flood Mitigation Platform. In: *Proceedings of the Eighth International Symposium on Information and Communication Technology*, SoICT 2017. [online] New York, NY, USA: Association for Computing Machinery.pp.301–308. Available at: <https://doi.org/10.1145/3155133.3155153.

Pánek, J., Marek, L., Pászto, V., & Valůch, J. (2017). The crisis map of the czech republic: The nationwide deployment of an Ushahidi application for disasters. *Disasters*, *41*(4), 649–671. doi: 10.1111/disa.12221

Pregnolato, M., Ford, A., Wilkinson, S. M., & Dawson, R. J. (2017). The impact of flooding on road transport: A depth-disruption function. *Transportation Research Part D: Transport and Environment*, 55, 67–81. https://doi.org/10.1016/j.trd.2017.06.020

See, L. (2019). A review of citizen science and crowdsourcing in applications of pluvial flooding. *Frontiers in Earth Science*, [online] 7. Available at: <https://www.frontiersin.org/articles/10.3389/feart.2019.00044/full> [Accessed 6 Jul. 2020].

See, L., Mooney, P., Foody, G., Bastin, L., Comber, A., Estima, J., Fritz, S., Kerle, N., Jiang, B., Laakso, M., Liu, H.-Y., Milčinski, G., Nikšič, M., Painho, M., Pődör, A., Olteanu-Raimond, A.-M., & Rutzinger, M. (2016). Crowdsourcing, Citizen Science or Volunteered Geographic Information? The Current State of Crowdsourced Geographic Information. *ISPRS International Journal of Geo-Information*, *5*(5), 55.