Journal of Engineering, Electrical and Informatics

E-ISSN .: 2809-8706
P-ISSN .: 2810-0557

Research Article

Utilization of *Voice Recognition* in the Development of Android-Based Medical Emergency Response Applications

Suryaningsih Patandung[1] *, Yohanis Padallingan[2], Srivan Palelleng[3], Plesmi[4], Lisna Junita Pairunan[5]

1. Program Studi Teknik Informatika, Universitas Kristen Indonesia Toraja, Indonesia; email: suryaningsihpatandung@gmail.com
2. Program Studi Teknik Informatika, Universitas Kristen Indonesia Toraja, Indonesia; email: padallinganyohanis@gmail.com
3. Program Studi Teknik Informatika, Universitas Kristen Indonesia Toraja, Indonesia; email: srivanpalelleng2@gmail.com
4. Program Studi Teknik Informatika, Universitas Kristen Indonesia Toraja, Indonesia; email: isakplesmi@gmail.com
5. Program Studi Teknik Informatika, Universitas Kristen Indonesia Toraja, Indonesia; email: lisnajunita0020@gmail.com

* Corresponding Author: suryaningsihpatandung@gmail.com[1]

Abstract. Medical emergencies can occur at any time and often without warning. Conditions such as heart attacks, strokes, seizures, or severe accidents require immediate assistance from others. However, many victims are unable to request help due to physical limitations and delays in accessing emergency services, especially among the elderly and people with disabilities. Along with technological advances, medical emergency response systems can be developed through Android-based applications. Limited mobility in emergency situations makes voice recognition technology a suitable solution, using common emergency voice commands such as "help" or "pain." This study aims to design and implement an application capable of rapidly sending emergency alerts through an emergency button or voice command activation, while also displaying the user's real-time location using Google Maps. The system was developed using the Waterfall method, which includes requirements analysis, system design, implementation, testing, and maintenance. Application testing was conducted using black-box testing to evaluate functionality and a User Acceptance Test (UAT) involving 15 respondents based on a Likert scale. The results indicate that all application features function properly, with a user acceptance level of 91.44%, categorized as very good. Therefore, the Tolong Kini application is considered effective and beneficial in supporting faster medical emergency response.

Keywords: Android; Emergency Application; Medical Emergencies; Voice Command; Voice Recognition.

Received: September 12, 2025
Revised: September 26, 2025
Accepted: October 11, 2025
Published: October 31, 2025
Curr. Ver.: October 31, 2025

1. Background

Everyone, regardless of age or health condition, has the potential to experience a medical emergency at any time and in any place. A medical emergency is a sudden condition that can be life-threatening or cause permanent disability if not treated immediately. This condition is one of those events that cannot be avoided or predicted, but often has a significant impact on a person's safety. Examples of common medical emergencies include cardiac arrest/cardiac arrest, seizures, strokes , poisoning, and even serious accidents that require assistance from others, so a quick response is key to saving lives. These emergencies can occur suddenly, often without warning. Unfortunately, many victims fail to receive prompt assistance due to ignorance or delays by those closest to them in receiving emergency information. In a 2022 study by Chantika Septidianti, it was stated that, according to World Health Organization (WHO) data, heart disease is a dangerous disease that claims the lives of 17.9 million people annually, or 31% of deaths worldwide. In emergency situations like this, physical difficulties and limited access often delay the process of seeking help. People who have difficulty calling for help during emergencies, particularly those with physical limitations or situations that prevent them from manually using a device, particularly the elderly or people

DOI : https://doi.org/10.55606/jeei.v5i3.6672

with disabilities, are often unable to directly operate a mobile device to contact emergency services or family. Furthermore, delays in communicating location information can prolong the response time of those expected to provide assistance.

Advances in voice recognition technology have had a significant impact on increasing the accessibility and ease of use of devices in various emergency situations. Several previous studies have highlighted the potential of voice recognition-based applications to facilitate requests for assistance in emergency situations. In a study conducted by Inggrit et al. in 2022 entitled Design and Development of an Android-Based " Panic Button " Application in Kendari City, where this study allows users to send an emergency signal with one simple button, demonstrating the importance of a quick response in an emergency, the use of voice command recognition to control home appliances, highlights how this technology can be adapted to the needs of user safety and comfort. Based on this study, voice recognition technology can be implemented in the development of emergency applications that require a quick response without physical interaction. From the Statcounter Global Stats page , as of May 2025, the number of Android users in Indonesia reached 90.81% of the total population. This shows the potential for developing an Android-based medical emergency response application. This application is an implementation of a medical emergency warning system using Android-based voice recognition technology . Through this application, users will be able to automatically send notifications to emergency contacts without requiring physical interaction, but with a voice code that is very helpful in situations where every second is precious to save lives.

2. Theoretical Study

Medical Emergency

A medical emergency is a critical situation that requires immediate treatment to prevent death, disability, or further damage to a person's health.

Android

open-source software platform designed for *mobile devices* . It includes an operating system, *middleware* , and basic applications. Android allows developers to create applications through an open platform. The Android operating system is based on the Linux 2.6 kernel, which has been customized for *mobile devices.*

Model Waterfall

Software development methods are systematic approaches used to organize, design, and manage the process of creating information systems. The *waterfall model* is also commonly defined as a classic approach to software development that is carried out systematically and sequentially. This model is often known as *the classic life cycle* because it follows structured development stages from start to finish.

Database

A database is a system consisting of important elements, namely databases, files, entities , and records , which are interconnected to store and manage information.

Figma

Figma is a design tool often used to design mobile applications , desktop applications , websites , and more. It can be accessed through Windows, Linux , or Mac operating systems as long as they are connected to the internet. One of Figma's strengths is its collaboration capabilities, allowing multiple people to work on the same project simultaneously, even if they are in different locations. This enables efficient teamwork, making Figma the preferred choice for many UI/UX designers to quickly and effectively prototype websites or applications.

SQLite

SQLite is a library that functions as a self-contained, serverless, zero-configuration , and transactional database engine . Self-contained means SQLite requires little support from external libraries or the operating system.

Native Framework

A native application is a type of application designed specifically for a specific operating system on a mobile device. Its main characteristics include the ability to directly access the hardware and support the user interface features and interactions provided by the device's operating system environment.

Android Studio

Android Studio is the official integrated development environment (IDE) for Android development released by Google and developed based on IntelliJ IDEA (Integrated Development Environment Application).

XML

XML (Extensible Markup Language) is a markup language used to store and transport data. XML allows users to define their own tags and document structures, which are easily adapted to represent various types of data. The primary purpose of XML is to provide a structured format for data exchange between disparate systems, ensuring data integrity and consistency, and simplifying information processing.

Java Programming Language

Java is a high-level programming language designed to run on multiple platforms and its simplicity allows developers to easily implement applications using clear and easy-to-understand syntax.

3. Research Methods

waterfall model software development method, as in Figure 1.

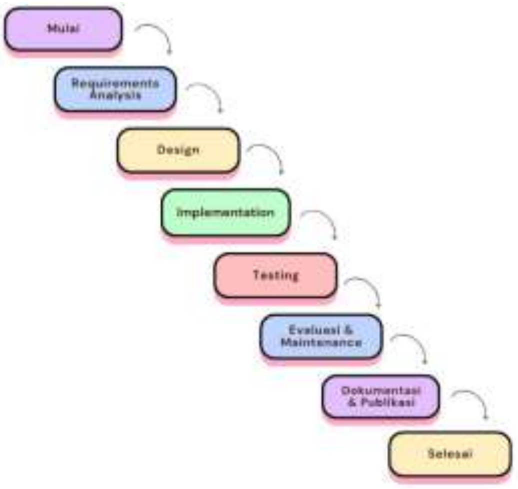


Figure 1. Research Flow.

This research begins with a requirements analysis phase. The first phase consists of three techniques: data collection, problem identification, and system requirements analysis (software and hardware). Data collection activities were conducted through literature studies, observations of the needs of medical emergency response applications, and interviews with people who have experienced medical emergencies and experts. Based on the problem identification, an analysis of software and hardware requirements was conducted, which will later be used in application development. The software and hardware analysis is expected to integrate voice recognition technology and GPS features for speedy access and response to medical emergency situations.

The next stage is the design stage. Information obtained from the requirements analysis stage serves as an indicator for the implementation of the design process. First, create a structure or navigation flow within the application for each actor. Second, design the system's functionality in the form of a use case diagram to illustrate the interactions between actors within the system. One interaction that will be visualized in the use case diagram is the sender, as a person experiencing an emergency situation, can send an emergency signal via voice code. Then, the system will perform voice recognition and send the emergency signal along with the sender's location point to the emergency contact stored in the application. Third, create an activity diagram using the draw.io tool to illustrate the activities that occur in the application. Fourth, design the application's UI by determining and arranging emergency-friendly elements or widgets for buttons, text, images, and certain icons in the correct layout, so that they are easily understood by users using the Figma application. The UI design considers the principles of simple, consistent, and responsive design in emergency situations. Indicators of achieving this stage are the suitability of the system flow design, system functionality, and emergency-friendly display to user needs.

The process that will be carried out in the implementation stage is translating the design that has been prepared in the design stage into a program script. Members of the developer section will create a series of program codes with Java and XML language structures with the help of the Android Studio text editor because the framework that will be used is Native; build the application UI based on predetermined and arranged widget elements; and carry out integration between widgets, features, and pages so that they can interact. The indicator of achieving this stage is the availability of all software and hardware requirements to implement voice recognition technology and GPS features in the application.

After the implementation process is complete, the testing phase begins. This phase ensures all program scripts are running as planned. Testing is conducted on both the emergency signal sender and the emergency signal receiver. Each feature in the application will be tested using black box testing, while user acceptance or satisfaction will be tested using User Acceptance Testing (UAT). Once completed, the application is compiled into an APK file for distribution to the general public.

During the app distribution process, evaluation and maintenance stages are still ongoing. Evaluations are conducted to obtain direct feedback from users—those who have experienced or may face medical emergencies—regarding their experiences using the app. Furthermore, user feedback and findings are identified, refined, and improved to ensure the research's performance indicators are truly effective.

This section contains the research plan, including the research design, population/sample, data collection techniques and instruments, data analysis tools, and the research model used. Common methods do not need to be described in detail; references to the referenced formulas (e.g., F-test, t-test, etc.) are sufficient. Validity and reliability testing of the research instrument does not need to be described in detail; the test results and their interpretation are sufficient. Explanations for symbols in the model are written in sentences.

4. Results and Discussion

Results

Requirements Analysis

Data collection in this study aims to gather voice codes that can be recognized by the system when a user is experiencing a medical emergency. These voice codes were collected through a literature review of existing related research, observations of the needs of medical emergency response applications, and interviews with people who have experienced medical emergencies and experts. The voice codes frequently used in medical emergencies that can be recognized by the system using voice recognition technology are as follows:

a. "Sick"
b. "Help me"
c. "Now"
d. “please now”

From the collected data, several issues with the current system were identified, including the reliance on manual interactions, such as pressing buttons or typing messages, which are difficult to perform in an emergency. This identification revealed key features needed by users in a medical emergency, along with the application's workflow from start to finish, and identified software and hardware requirements .

Design

After the data is collected and identified, the next stage is the design stage . First , the system's functionality is designed in the form of a use case diagram to illustrate the interactions between actors within the system, as seen in Figure 2. Drawio, a web-based software tool , is used to illustrate the use case diagram .

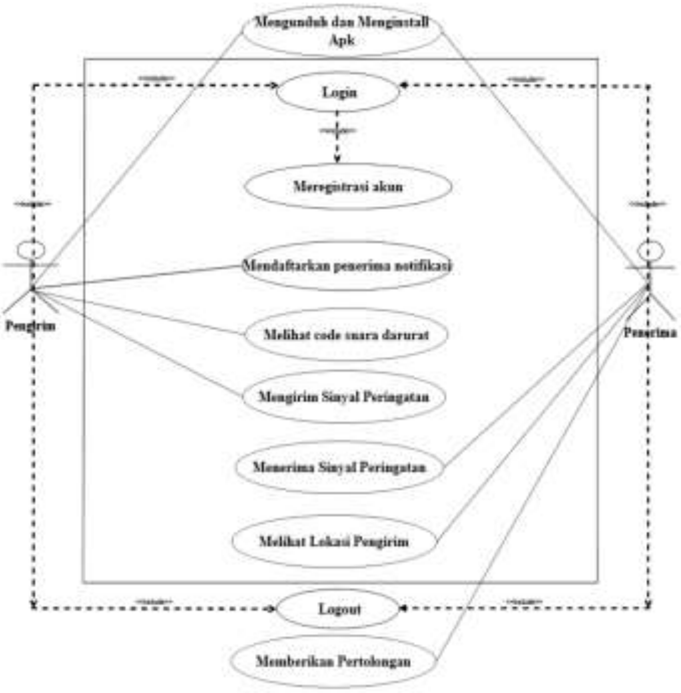

Mengunduh dan Menginstall Apk
Login
Meregistrasi akun
Mendaftarkan penerima notifikasi
Melihat code suara darurat
Mengirim Sinyal Peringatan
Menerima Sinyal Peringatan
Melihat Lokasi Pengirim
Logout
Memberikan Pertolongan
Pengirim
Penerima

Figure 2. *Use Case Diagram.*

Second , create an activity diagram to illustrate the flow of activities within the system, namely presenting the sequence of processes from when the user provides input until the system responds by sending an emergency alert signal. Activity diagrams help in understanding how the application works sequentially and how the system handles each emergency scenario. This diagram explains the steps that occur in the system, including downloading and installing the application; logging in and creating an account; the notification sender registering the notification recipient; sending an alert signal; sending, receiving, displaying an emergency signal, and viewing the location.

Third, we designed the user interface (UI) using Figma. This stage aimed to create an interface that was easy to understand, responsive, and simple to use in emergency situations. The design consisted of eight main pages, each with a specific role in supporting smooth emergency communication between the sender and receiver.

Implementation

At this stage, all previously designed interface designs are compiled into program scripts. The implementation process includes creating application pages along with Messaging logic. Supporting user interactions, such as data input, navigation between pages, and integration with third-party services such as Firebase and Google Maps. In this study, the implementation was carried out using the Native framework with Java and XML programming languages, a local database using SQLite, and online service integration through *Firebase Authentication* and *Firebase. Cloud.* This stage aims to transform the design into a functionally executable system that meets user needs.

a. Splash Screen Page Implementation

The Splash Screen is the first screen that appears when the application is run. Its main function is to provide a pleasant initial experience for the user, displaying the application logo or name, as well as a short slogan "Speak, Act, Congratulations". In terms of coding implementation, as seen in Figure 3. This screen acts as the initial gateway for the application's navigation logic. After a short delay of 3 seconds, the Splash Screen then determines the next page to be displayed to the user. This is done by checking the user's login status (isLoggedIn) and user role (*userRole*) stored in *SharedPreferences* .

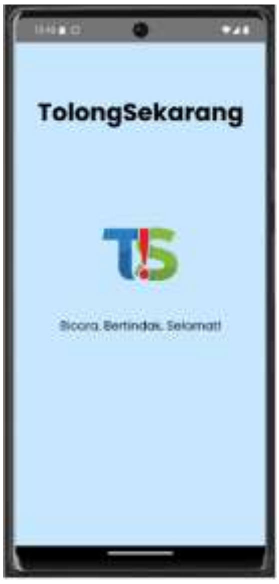


Figure 3. *Splash Screen Page.*

b. List Page Implementation

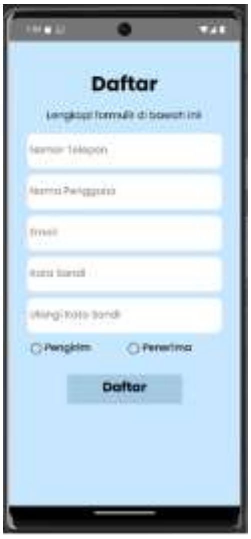


Figure 4. List Page.

The registration page is a page specifically designed for new users who want to join the application. The primary function of this page is to facilitate the process of creating a new account by collecting basic information from potential users. As seen in Figure 4, users are asked to fill out a form that includes a Phone Number, Username, Email , Password, and Repeat Password for confirmation.

c. Login Page Implementation

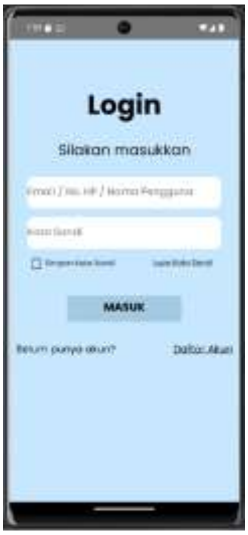


Figure 5. *Login Page.*

Login page is a crucial page that serves as an access gateway for registered users, ensuring access security and facilitating personalized navigation after successful authentication. In terms of coding implementation, as shown in Figure 5, this page allows users to enter their data, namely Email /Phone Number/Username and Password. This data is then processed through FirebaseHelper for authentication using the Firebase Authentication service.

d. Sender Home Page Implementation

Home page is the primary interaction hub for users with the "Dispatcher" role in the app. Its primary function is to provide a quick and easy interface for users to send an emergency signal, either through manual interaction or voice activation. Figure 6 shows that this page displays a greeting (Hello, [sender user's username]) and instructions (Say Emergency Voice Code) to trigger the app's core functions.

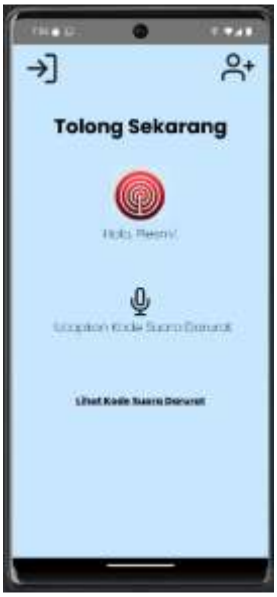
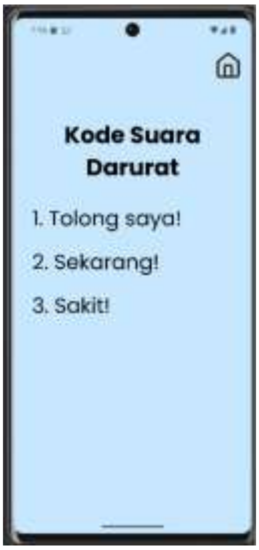


Figure 6. Sender *Home Page.*

e. Emergency Voice Code View Page Implementation

This page aims to allow users to learn emergency words without having to test them directly through the voice recognition feature. Figure 7 displays a list of voice codes such as "Help me," "Now," and "Sick," along with a brief explanation of each code's function.

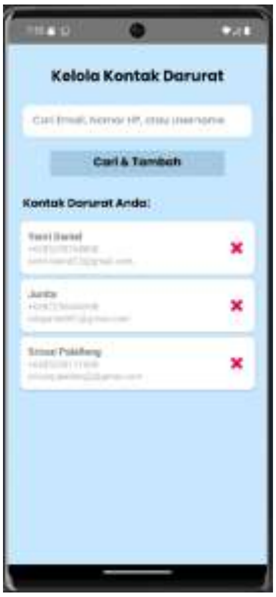


Kode Suara Darurat

1. Tolong saya!
2. Sekarang!
3. Sakit!

Figure 7. View Emergency Voice Codes Page.

f. User List Page



Kelola Kontak Darurat

Cari Email, Nomor HP, atau username

Cari & Tambah

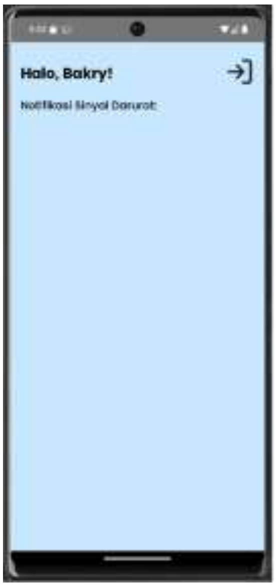
Kontak Darurat Anda:

Figure 8. Recipient/ *User List Page.*

This page serves as a central hub for Sender users to manage their emergency contact list, which is the Recipient accounts that will receive emergency alerts. Its primary function is to allow users to search for and add new contacts, as well as display a list of existing contacts. Figure 8 shows a search field (Search Email , Phone Number, or Username) that allows users to search for Recipient accounts already registered in the Firestore Database.

g. Recipient Home Page Implementation

Home page serves as the main dashboard for users with the "Receiver" role, specifically designed to receive, display, and enable real-time management of emergency signals . Figure 9 shows the implementation of the receiver *home page for listening to real-time* data updates from *Firebase Firestore* on the *emergency_signals collection* , via *the ListenerRegistration* implemented in *ReceiverHome.java* . This mechanism ensures that any emergency signals addressed to the current receiver are immediately displayed on the screen.



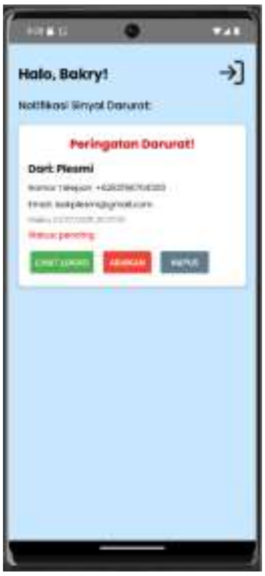
Halo, Bakry!

Notifikasi Sinyal Darurat:

Figure 9. Recipient *Home* Page.

h. Notification Page Implementation

The notification page view is part of the application's " Receiver Home Page ," a key component of the Receiver Home Page specifically dedicated to displaying and managing emergency signals received by users with the "Receiver" role. Its primary function is to provide real-time updates and interactions regarding emergency situations. As seen in Figure 10, the notification page displays a list of notifications in card format (CardView).



Halo, Bakry!

Notifikasi Sinyal Darurat:

Peringatan Darurat!

Dari: Plesmi

Nomor Telepon: +62[illegible]

Email: [illegible]@gmail.com

Waktu: [illegible]

Status: pending

Figure 10. *Notification Page.*

i. Location Page Implementation

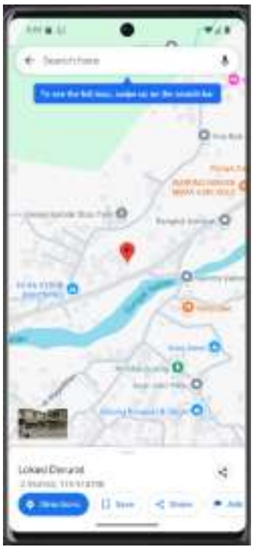


Figure 11. *Location Page.*

Location page displays the geographic location of the user sending the emergency signal on an interactive map. Its primary function is to provide an accurate location visualization, allowing the recipient to pinpoint exactly where assistance is needed. As seen in Figure 11, this page uses the Google Maps API.

Testing

Black box testing is conducted to ensure that all features on the emergency signal sender and receiver sides function as intended. Each test scenario is designed to reflect the actual conditions of an application being used by a user in an emergency. The sending user is the party in an emergency situation and requires assistance. Table 1 presents the black box testing results from the emergency signal sender side.

Table 1. Testing of the Emergency Signal Transmitter Black Box.

Features tested	Testing Scenario	Expectations	Test Results
Splash screen	The application is opened by the user	*Splash screen* appears for 3 seconds, then goes to the *Home page*	In accordance
Login	*Login* with valid mobile number/username/email and password	Go to *Home page*	In accordance
Login	*Login* with invalid data	An error message appears	In accordance
Register	Register a new account with complete data	Account saved successfully, directed to login	In accordance
Emergency Button	Press the emergency button	Signal sent, notification arrives at the recipient	In accordance
Voice Recognition	Say one of the emergency voice codes	Voice recognized, emergency signal sent	In accordance
Recipient List	View a list of available recipient accounts	List appears according to data	In accordance
View emergency voice codes	View the emergency voice codes used	List of emergency codes according to the codes used	In accordance
Logout	Press *the icon logout*	Return to login page	In accordance

In addition to testing the transmitter side, testing was also conducted on the emergency signal receiver side to ensure that the features related to signal reception and handling functioned as expected. Table 2 shows the results of the Black Box testing on the emergency signal receiver side.

Table 2. *Black-box Testing* of Emergency Signal Receiver.

Features tested	Testing Scenario	Expectations	Test Results
Login	Login as recipient	Go to the notification page or *Home*	In accordance
Receive Notification	Get emergency notification from sender	Notifications appear with emergency code information accompanied by emergency ringtones.	In accordance
View Sender Location	Press the "View Location" button from the notification	Google Maps opens, the sender's location is displayed.	In accordance
Logout	Press *the icon logout*	Return to *login page*	In accordance

Evaluation

The evaluation was conducted to obtain direct feedback from users, namely those who have experienced or may experience medical emergencies, regarding their experiences using the application. Next, user feedback and findings were identified, refined, and improved to ensure the research's performance indicators were truly effective. This stage was used to determine user acceptance or satisfaction with the application using the user acceptance test (UAT) method [10].

The User Acceptance Test (UAT) involved asking several questions to the Toraja community to determine the success of the application implementation. This test involved 10-15 respondents from the Toraja community. After analyzing each statement submitted in the User Acceptance Test (UAT), the total score and average percentage index were calculated for all respondents' answers. Table 3 presents a summary of the UAT scores based on data processing from 15 respondents.

Table 3. Summary of UAT Values.

No	Statement	Percentage Value	Information
1	The application display is easy to understand	96%	Very good
2	The menus in the application are easy to recognize	92%	Very good
3	This application can help users.	90.6%	Very good
4	The appearance of this application is attractive	88%	Very good
5	The application provides sufficient information	84%	Very good
6	The use of each page in this application is in accordance with	86.6%	Very good
7	Button and sound features can be used	96%	Very good
8	The app can send emergency notifications to predetermined contacts.	94.6%	Very good
9	I think this app is easy to use in emergency situations.	93.3%	Very good
10	The application displays the location in *real-time.*	93.3%	Very good
	Total Average	**91.44%**	**Very good**

Based on Table 3 above, it can be concluded that the final result of the UAT (User Acceptance Test) is 91.44% , which means that the majority of the public stated that the "Tolongsekarang" application as a medical emergency warning system using Android-based voice codes is good to use.

Discussion

The "Tolongsekarang" application has been tested using the Black Box method and demonstrated good performance across all key features. The splash screen appears as designed, login and registration respond correctly to both valid and invalid input, and the emergency button functions effectively with just one touch. The voice recognition feature is able to recognize three codes ("Sick", "Help me", "Now", "Help Now") and works well in quiet environments, although it needs improvement in crowded conditions. In addition, sending notifications and locations via Firebase Cloud Messaging and Google Maps is fast and accurate, thus supporting real-time emergency assistance .

5. Conclusion and Suggestions

Conclusion

black box testing conducted on the application, all tested functions from the emergency signal sender and receiver side showed results in accordance with expectations. This indicates that the application functionality is running well. Likewise, the results of the User Acceptance Test (UAT) show that users find this application easy to use in emergency situations. The index value obtained, which is 91.44%, indicates high user acceptance of the application's features and interface. Thus, the results of research and testing on the "Tolongsekarang" application as a medical emergency warning system using Android-based voice codes, it can be concluded that the "TolongSekaran g" application was successfully built using the Waterfall software development method , with the following stages: analysis, design, implementation, testing, and evaluation.

Suggestion

As a suggestion for further research, the "Tolongsekarang" application needs to be developed more optimally with several improvements, such as adding a shortest route feature if Google Maps is less effective, optimizing voice recognition to remain accurate in noisy environments, and providing an alternative SMS-based notification for limited network conditions. In addition, it is necessary to build a new backend so that the application can run automatically with voice code, develop Firebase Cloud Messaging so that notifications are still received even when the application is closed, and add infrastructure for regular location updates. Finally, the application should be supported by a system that allows its use in various regions, so that help remains efficient even though the distance between the sender and recipient is quite far.

Reference

Al-Faruq, M. N. M., Nur'aini, S., & Aufan, M. H. (2022). UI/UX design for Semarang virtual tourism using Figma. *Walisongo Journal of Information Technology, 4*(1), 43–52. https://doi.org/10.21580/wjit.2022.4.1.12079

Andyka, D., & Anwar, M. C. (2017). Design and construction of an Android application for smarthome control using voice commands.

Badrul, M. (2021). Application of the waterfall method for designing an inventory information system at the Bintang Terang Ceramics Store. *Prosisko: Jurnal Pengembangan Riset dan Observasi Sistem Komputer, 8*(2), 57–62. https://doi.org/10.30656/prosisko.v8i2.3852

Hasbid, R., Yusuf, R., & Muharni, S. (2021). Implementation of the waterfall model in the design of service and information systems with the OOAD approach using UML, *5*.

Ilma, N. I., Feoh, A. D. C., & Saifudin, W. A. (2021). Smartphone-based safety application with social media integration as a solution for building a disaster-responsive community. *Citrawira: Journal of Advertising and Visual Communication, 2*(1), 23–33. https://doi.org/10.33153/citrawira.v2i1.3666

Inggrit, I., Maruji, M., & Henny, H. (2022). Design and construction of an Android-based panic button application in Kendari City. *Simtek: Jurnal Sistem Informasi dan Teknik Komputer, 7*(2), 143–151. https://doi.org/10.51876/simtek.v7i2.158

Ismail, M., Prasetyowati, S. A. D., & Sulchan, A. (2020). PKM Android application for vehicle theft detection to improve the security and comfort of Karangtaruna Gajahmada Gebangsari Semarang. *E-Dimas: Jurnal Pengabdian kepada Masyarakat, 11*(3), 394–399. https://doi.org/10.26877/e-dimas.v11i3.4731

Mallisza, D., Hadi, H. S., & Aulia, A. T. (2022). Implementation of the waterfall model in designing a website-based travel order system using the SDLC method. *Journal of Technology, Computer, Agrotechnology and Science, 1*(1), 24–35. https://doi.org/10.56248/marostek.v1i1.9

Muhyidin, M. A., Sulhan, M. A., & Sevtiana, A. (2020). UI/UX design of My CIC application for student academic information services using Figma application. *Jurnal Digit, 10*(2), 208–216. https://doi.org/10.51920/jd.v10i2.171

Noviantoro, A., Silviana, A. B., Fitriani, R. R., & Permatasari, H. P. (2022). Design and implementation of a web-based badminton court rental application in the Depok region. *Journal of Technology and Science, 1*(2), 88–103. https://doi.org/10.56127/jts.v1i2.108

Rahmat, M., Ichsanul, A., Sarini, V. D., & Aulia, S. A. (2023). E-module of Java programming language based on Exe-Learning. *Jurnal Processor, 18*(1). https://doi.org/10.33998/processor.2023.18.1.704

Septidianti, C. (2022). The effect of cardiopulmonary resuscitation practice on improving cardiac arrest management skills in the Payo Lebar Village community, Jelutung District, RT 029. *Jurnal Borneo Holistic Health, 5*(2). https://doi.org/10.35334/borticalth.v5i2.2756

Sibuea, S., Mohammad, I. S., Agie, A., & Yohanes, B. W. (2022). Android-based mobile collection application at PT. Suzuki Finance Indonesia. *Jitek: Jurnal Informatika dan Teknologi Komputer, 2*(1), 31–42. https://doi.org/10.55606/jitek.v2i1.185

Syahrin, M. A., & Mei Prianto, E. (2024). Android & SQLite-based social fundraising application. *Jurnal Gemilang Informatika GIT, 1*(1), 16–20. https://doi.org/10.58369/git.v1i1.88

Wijayanto, H., Raharja, B. D., & Prabowo, I. A. (2021). Designing an Android-based mobile application to improve product promotion at CV Putra Nugraha. *E-Dimas: Jurnal Pengabdian kepada Masyarakat, 12*(3), 473–476. https://doi.org/10.26877/e-dimas.v12i3.7248