

Suraksha Sathi: An Innovative SOS Application for Personal Safety

Prathmesh Krishna Aghao^{1,*}, Savani Mahendra Balkawade¹,
Kartik Sandeep Bende¹, Tejas Thakare¹, Pratap Nair²

Abstract

Even in this increasingly connected age, personal safety is still of utmost importance. A smartphone software called Suraksha Sathi makes use of cutting-edge technology to improve user safety. Its SOS function uses geolocation to determine the precise location of users, facilitating prompt replies from people and emergency agencies. It is built using flutter and dart and includes REST APIs for data sharing, location services APIs, and Google Maps. The development environment is Visual Studio, while Firebase securely maintains user accounts and data. Suraksha Sathi, an iOS and Android compatible app, offers comfort and support during emergencies. A section on mental health, preventing suicide, and help-seeking behavior is included in the study. In an emergency, the software can notify contacts and law enforcement by text message (SMS) about the user's position, guaranteeing timely aid. Personal safety has emerged as a top priority in today's world, particularly in light of the rising frequency of emergencies and erratic circumstances. Mobile technology has developed in response to this, providing people in need of quick assistance with real-time solutions. An inventive SOS app called "Suraksha Sathi" gives users a quick and simple way to notify contacts and emergency services when they are in danger, hence improving personal safety. This study discusses how Suraksha Sathi intends to transform personal safety through its clever design and accessibility, outlining its salient features, benefits, and technological foundation.

Keywords: Save our souls (SOS), safety, mobile application, android, alert message, personal safety

INTRODUCTION

SOS-Suraksha Sathi, the groundbreaking emergency response application is designed to revolutionize how individuals seek help during critical situations. With features like SOS messaging, emergency numbers directory, and direct call permission, users can swiftly reach out for assistance when every

second counts. Our platform prioritizes prompt and efficient communication, recognizing that swift action can be the difference between life and death [1]. We ensure accurate location information with address details in SOS messages, enabling responders to reach individuals swiftly. Moreover, our commitment extends to user-friendliness with login/sign-up via One-Time Password (OTP), global compatibility, and an article section for valuable survival information. Embracing decentralization, users can add trusted contacts, utilize map support for precise location sharing, and access vital survival resources. SOS-Suraksha Sathi is more than just an application; it is a lifeline during times of crisis, empowering communities to navigate emergencies with confidence.

*Author for Correspondence

Prathmesh Krishna Aghao
E-mail: prathmeshaghao@kccemsr.edu.in

¹Student, Department of Computer Science and Engineering,
KC College of Engineering and Management studies and
Research, Thane, Maharashtra, India

²Mentor, Department of Computer Science and Engineering,
KC College of Engineering and Management studies and
Research, Thane, Maharashtra, India

Received Date: August 10, 2024
Accepted Date: August 25, 2024
Published Date: September 18, 2024

Citation: Prathmesh Krishna Aghao, Savani Mahendra Balkawade, Kartik Sandeep Bende, Tejas Thakare, Pratap Nair. Suraksha Sathi: An Innovative SOS Application for Personal Safety. Recent Trends in Electronics & Communication Systems. 2024; 11(3): 27–40p.

LITERATURE SURVEY

Literature survey is shown in Table 1 and comparative analysis is shown in Table 2.

PROPOSED TECHNIQUE

Approach Details

The SOS app Suraksha Sathi is designed to operate on Android devices, mostly smartphones. Flow Chart for Suraksha Sathi is shown in Figure 1. Our primary objective is to provide emergency assistance, connecting users with relevant services, integrating emergency contact, location tracking, user authentication, support multiple languages, and ensuring rapid response in critical situations [2].

Table 1. Literature Survey of related articles.

Sr. No.	Project related	Area/Domain	Highlight	Remark
1.	Mobile application to locate and track person's whereabouts and send SOS messages	Personal safety and location tracking technology	The mobile app enables users to share their location in real-time and send distress signals (SOS messages) if needed.	Provides a simple and effective solution for personal safety, particularly in emergency situations.
2.	A mobile application for women's safety: WoSApp	Women's safety technology.	WoSApp is a mobile application designed specifically for enhancing women's safety, offering features like emergency alerts and real-time location tracking.	WoSApp addresses a critical need for improved safety measures tailored to women, utilizing accessible smartphone technology for empowerment and safety
3.	Women Safety-Savior Android Application	Women's safety technology.	The Women Safety-Savior Android Application is designed to provide safety features tailored for women, such as emergency alerts, location tracking, and quick access to help.	This app aims to address safety concerns faced by women by leveraging smartphone technology for immediate assistance and peace of mind.
4.	Play It Safe, a personal security application on Android Platform	Personal security technology.	"Play It Safe" is a personal security application available on the Android platform, offering features for enhancing user safety, such as emergency alerts and location tracking.	This app provides a convenient and accessible solution for individuals to improve their personal security using their Android devices.

Table 2. Comparative analysis.

Feature	Suraksha Sathi	SOS Alert	SOS App	Disha SOS App	112 SOS Mobile App
SOS Message	✓	✓	✓	✓	✓
Emergency Numbers	✓	✓	✓	✓	✓
SOS Contacts	✓	✓	✓	✓	✓
Direct Call Permission	✓	✓	✓	✓	✓
Address in SOS Message	✓	✗	✓	✓	✓
Login/Sign-up with OTP	✓	✓	✓	✓	✓
Global Compatibility	✓	✓	✓	✓	✓
Article Section	✓	✗	✗	✗	✗
Add trusted people	✓	✓	✓	✓	✗
Map Support	✓	✗	✗	✗	✗
Survival information	✓	✗	✗	✗	✗

In India, the existing SOS Application lacks certain functionalities, so we are making sure that we choose the best technology to make our app work well and keep the information secure [3]. Comparative Analysis shown in Table 1 and Table 2. We are connecting users to local emergency services to establish strong connections and ensure they respond quickly [4].

Dependencies

In order to protect confidential information, prevent unauthorized users from accessing reports, and maintain data accuracy, user registration and authentication are essential (Appendix-1) [5].

- **Development:** Visual Studio provides an integrated environment for building the application.
- **Programming:** Flutter enables cross-platform development, and Dart provides object-oriented language for app logic.
- **Location:** Google Maps and APIs provide accurate location data for SOS and emergency service display.
- **Data Exchange:** REST APIs facilitate secure communication between the app and its backend systems.
- **User Management:** Firebase handles user authentication and data storage, while Firestore provides a flexible NoSQL database solution. Technology used in development is shown in Figure 2.

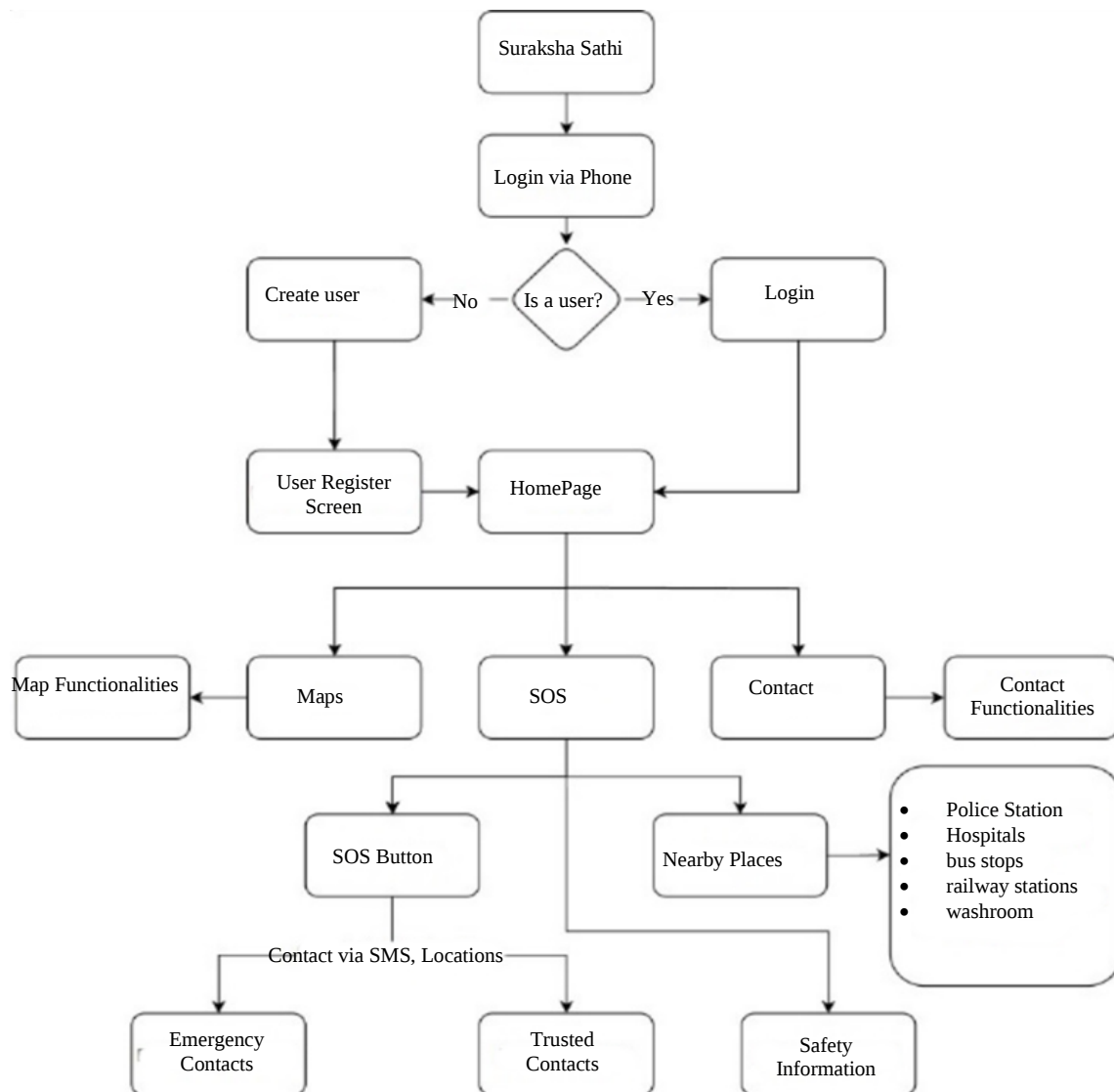


Figure 1. Flow chart illustrating the process for developing the Suraksha Sathi application.

Technology Stack

- An SOS app for Android can be expanded with the use of a variety of technologies [6].
- *Flutter and Dart*: This effective aggregate supports pass-platform mobile app development with expressive user interfaces and overall performance.
- *Mapbox*: a versatile platform for area-primarily based services, imparting customization maps, navigation, and vicinity search features.
- *Google API*: These APIs permit the mixing of various powerful functionalities, from authentication to machine learning, inside programs.
- *Google Maps*: This device offers region-primarily based services, navigation, and rich map reports for apps.
- *Visual Studio*: This integrated improvement environment (IDE) supports more than one language and framework, enhancing each productivity and first-class code [10].
- *Firebase*: A comprehensive improvement platform that gives authentication, statistics storage, analytics, and more, all supported by Google's infrastructure.
- *Firestore*: A scalable, bendy NoSQL database via Firebase that offers real-time synchronization and offline abilities for a smooth app experience [11].

RESULTS

Sign-Up

The intuitive design components and user-friendly interface of the Suraksha Sathi sign-up page provide a smooth onboarding process for an improved user experience (Appendix-1). Sign-up page is shown in Figure 3.

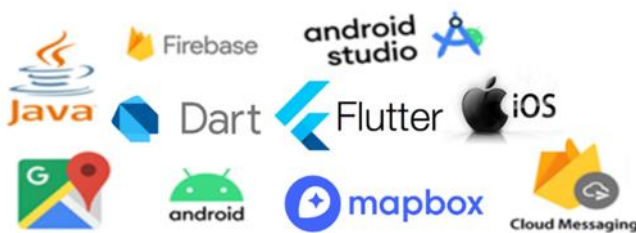


Figure 2. Technology used in development.

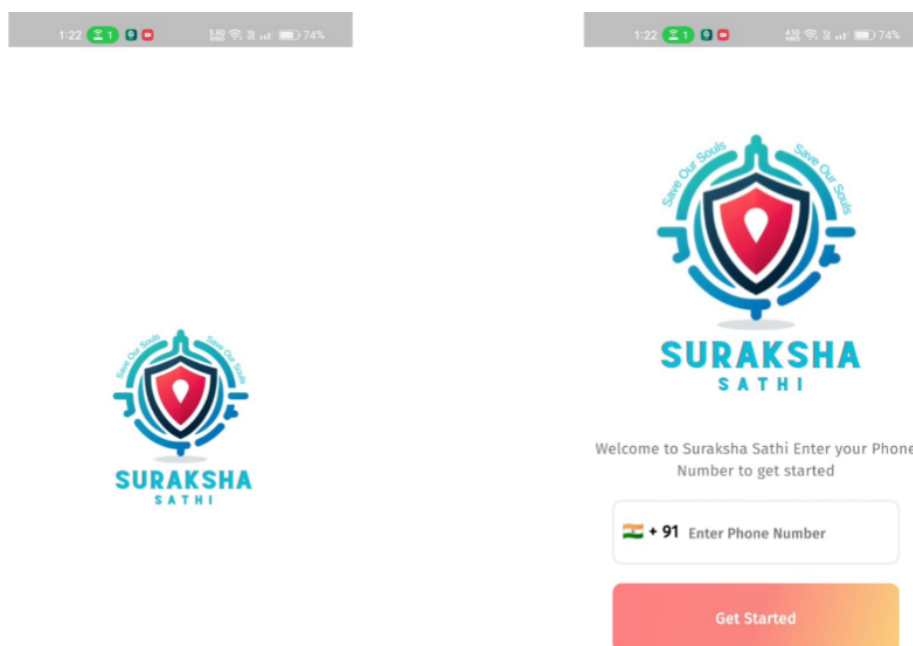


Figure 3. Sign-up page.

OTP verification

By using a one-time password to provide safe user authentication, the Suraksha Sathi OTP Verification page improves account security [12]. Verifying and getting the OTP is shown in Figure 4.

Home

The “Suraksha Sathi” home page functions as All India Alerts' central location and a thorough survival guide, offering crucial details at a glance. Creating Account/Home Page is shown in Figure 5. Creating an account in an application is a fundamental feature that allows users to establish a unique identity within the app. This identity can be used for personalized experiences, data storage, access control, and more [13].

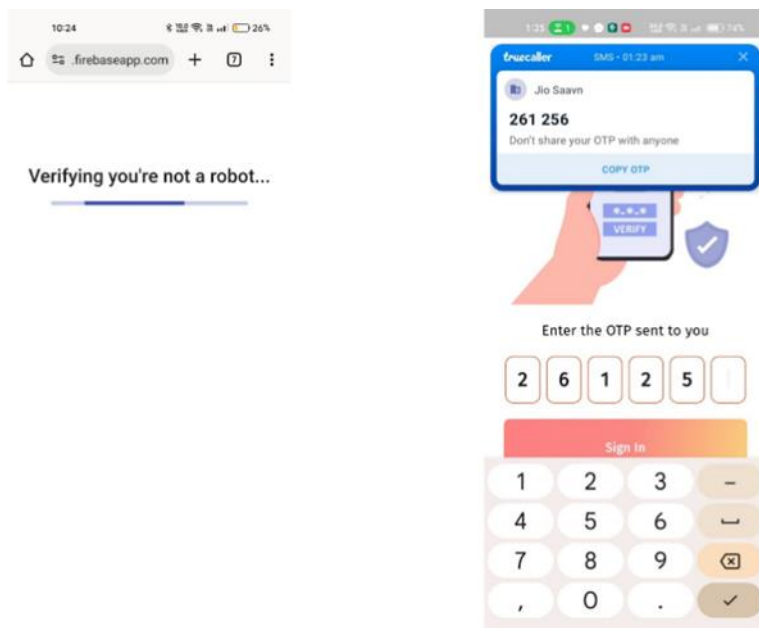


Figure 4. Verifying and getting the OTP.

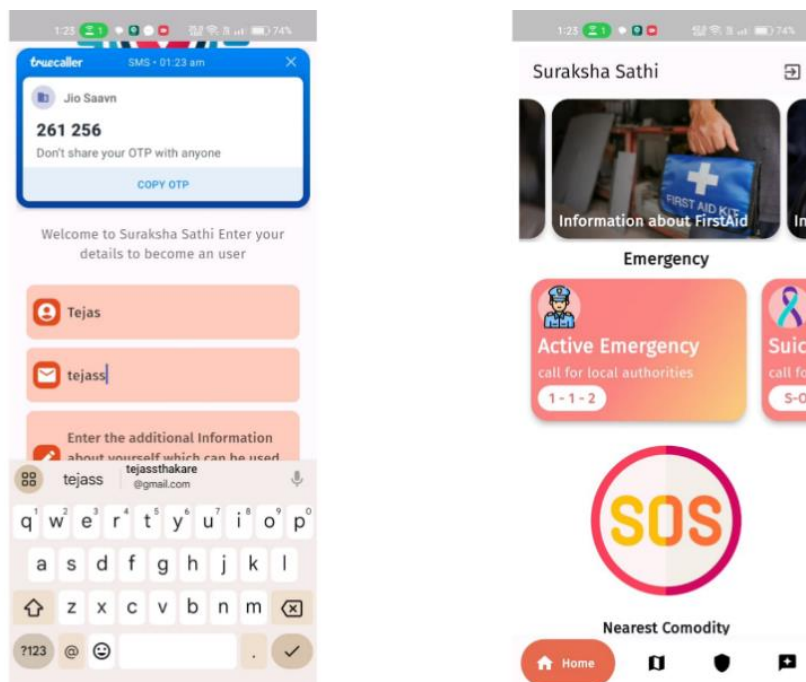


Figure 5. Creating Account/Home Page.

Maps

By swiping within the emergency section, users gain immediate access to critical information like nearby police stations, hospitals, and other relevant emergency services [14]. Locating a place on map is very essential for the further process as shown in Figure 6.

Emergency Services

Adding a contact to a contact list involves interfacing with a device's native contact management system [15]. Although the steps may differ slightly depending on the platform, adding new contacts usually entails gaining access to the device's contacts database. Adding a contact to close list is shown in Figure 7. In a flutter context, this typically requires permissions and the use of a plugin to interact with the native functionality [16]. SOS functionality is shown in Figure 8.

The SOS functionality in applications is designed to provide quick and efficient access to emergency services, or contacts in critical situations. During emergency situations, SOS functionality is meant to provide both safety and prompt communication [17].

Comparative Analysis

The Suraksha Sathi system distinguishes itself through its comprehensive approach to safety, addressing the well-being of not only women but also all individuals [7]. While some researchers focus solely on women's safety concerns, Suraksha Sathi encompasses a broader spectrum of safety considerations. Through features such as advanced geolocation, real-time communication, and access to mental health resources, this proposed system ensures a holistic safety solution for all user demographics, irrespective of gender, thereby fostering a secure environment for everyone [1–17].

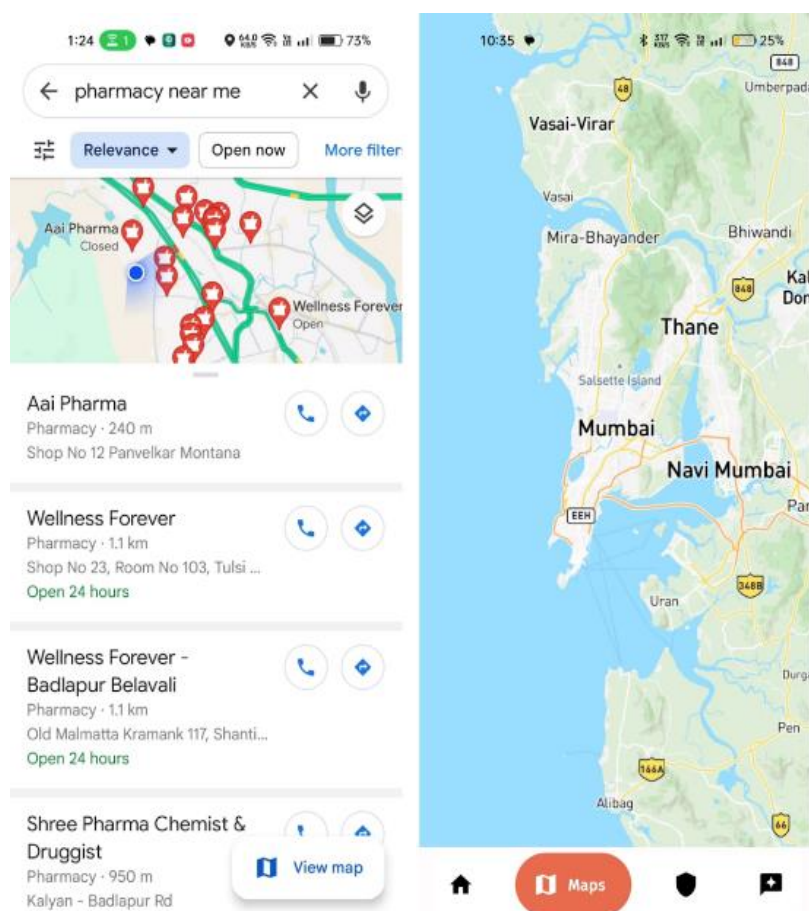


Figure 6. Locating a place on map.

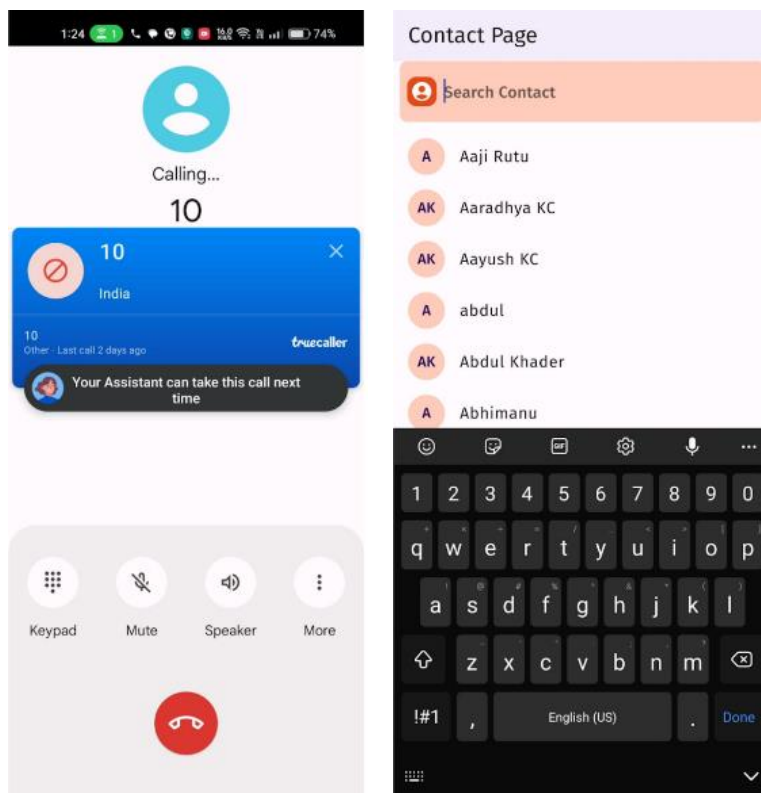


Figure 7. Adding a contact to close list.

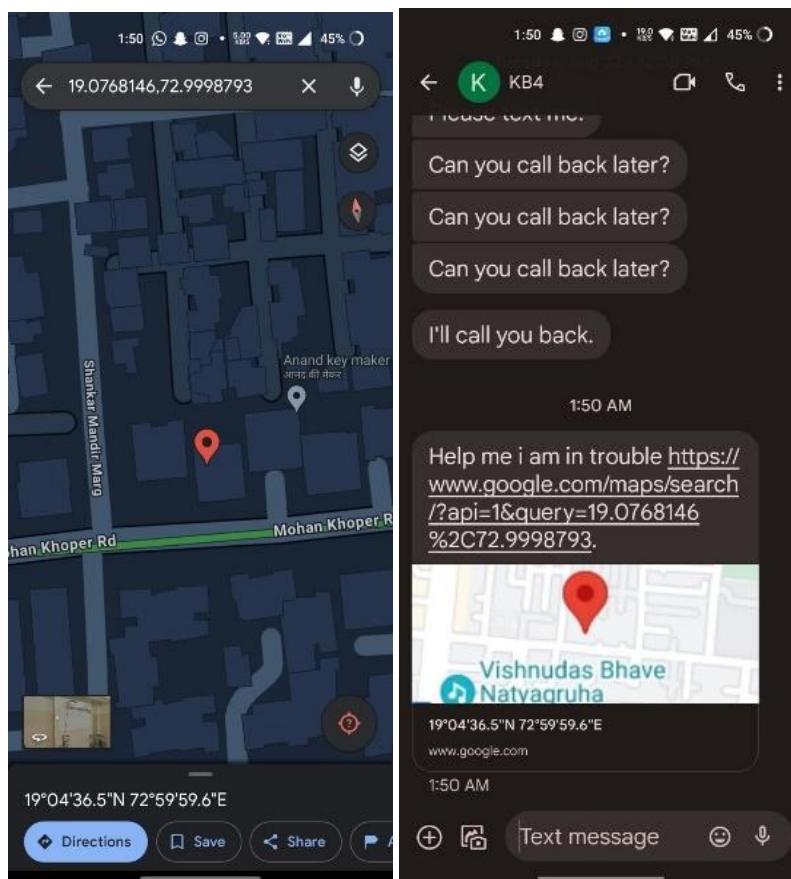


Figure 8. SOS functionality.

Limitations

Although the app can send an SOS message rapidly, the location and other circumstances can affect how quickly emergency services respond [8]. If users are unfamiliar with the features and operations of the app, it may be a limitation for them to know how to use it efficiently in an emergency. Distribution of users by gender is shown in Figure 9. Continuous usage of the device's GPS and network services for messaging and location tracking may quickly deplete its battery, particularly in emergencies where the app may be used for extended periods. User feedback on usability, UI, and nearby location feature of Suraksha Sathi application is shown in Figure 10. User error or app malfunction may unintentionally send out an SOS alert, prompting needless emergency reactions. It may be difficult for users with impairments or disabilities to utilize the app efficiently, particularly in high-stress situations [9]. The application is dependent on the user's Android device, which may cause a problem if the device is misplaced, broken, or experiences malfunctions. Users with data plans that are limited may find that the app uses a lot of their mobile data [1].

Survey

Below is the feedback ratings provided by users for various aspects of the Suraksha Sathi application, along with accompanying graphs illustrating the survey results. Overall user experience rating for Suraksha Sathi application is shown in Figure 11. Abbreviations and nomenclature used in this article is mentioned in Table 3.

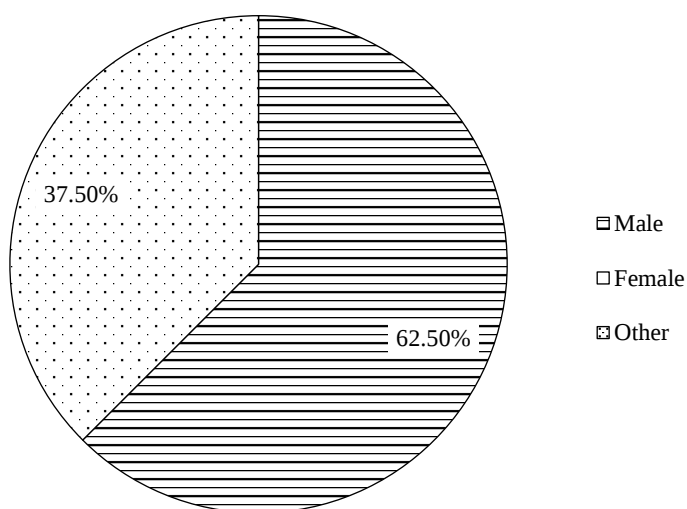


Figure 9. Distribution of users by gender.

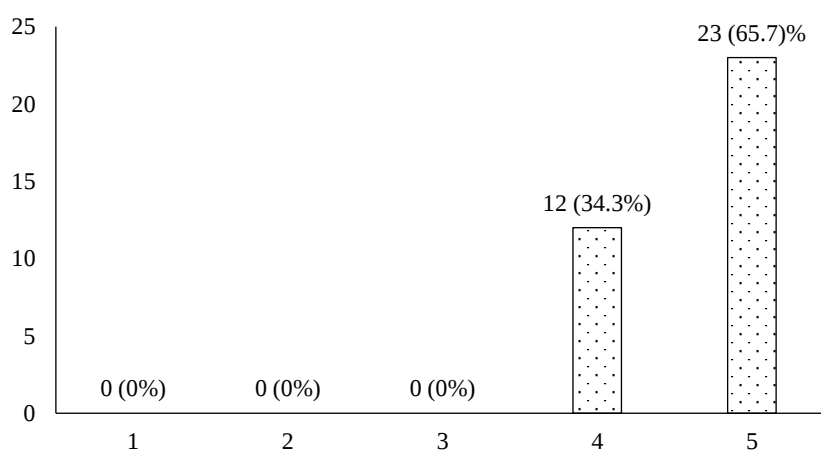


Figure 10. User Feedback on Usability, UI, and Nearby Location Feature of Suraksha Sathi Application.

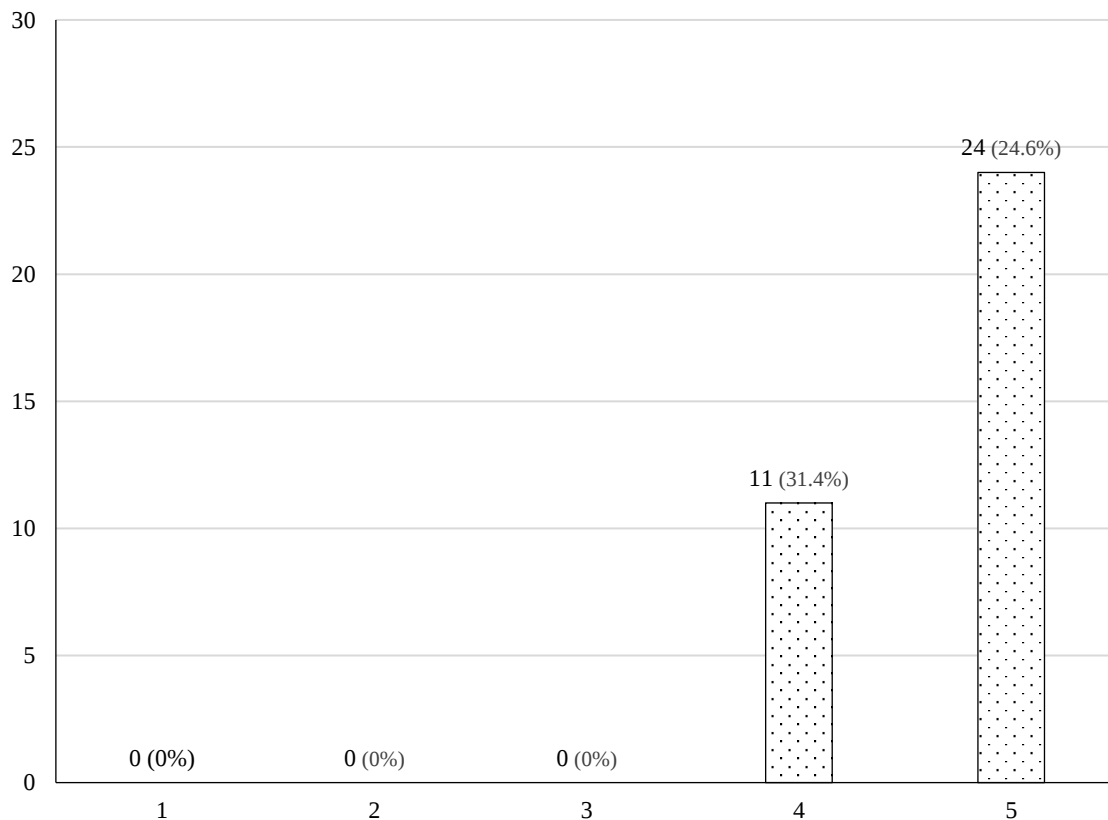


Figure 11. Overall User Experience Rating for Suraksha Sathi Application.

CONCLUSION

This proposed system endeavors to develop a user-friendly emergency communication tool tailored for individuals across India. With fundamental features such as text communication and customizable distress messages, user safety is prioritized. Moreover, integration of advanced functionalities like location tracking and real-time emergency response coordination aims to enhance effectiveness. As the proposed system progresses, expansion into flood-prone regions, integration of drones and sensors, and collaboration with relevant agencies are envisioned. Ultimately, this initiative is poised to significantly enhance the safety and well-being of individuals nationwide, thereby shaping the future landscape of emergency communication technology in India.

Conflicts of Interest

The authors claim that there may be no conflict of interest concerning the guide of this manuscript. All authors have authorized the final version of the manuscript and accept it as true with its submission to the magazine. The development of the Suraksha Sathi (SOS software) was conducted independently, and no monetary or non-public relationships encouraged the consequences and conclusions supplied in this paper.

Acknowledgment

We would love to express our honest gratitude to all those who have supported and guided us for the duration of the improvement of the Suraksha Sathi (SOS utility). Unique thanks to our mentor, Prof. Pratap Nair, for his or her useful steerage and encouragement, and to our colleagues and friends at K.C. college of Engineering and Management Studies and Research, Mumbai, India for their collaborative efforts and constructive discussions. Sooner or later, we renowned our households and friends for his or her unwavering support.

Table 3. Abbreviations and Nomenclature.

Sr. No.	Abbreviation	Expansion
1	SOS	Save Our Souls
2	GPS	Global Positioning System
3	API	Application Programming Interface
4	UI	User Interface
5	SMS	Short Message Service
6	SDK	Software Development Kit
7	IDE	Integrated Development Environment
8	HTTP	Hypertext Transfer Protocol
9	AES	Advanced Encryption Standard
10	OTP	One-Time Password
11	iOS	iPhone Operating System
12	APK	Android Application Package

Future Scope

The application intends to include cutting-edge technologies like drones and sensors, broaden its geographic coverage to include flood-prone areas, and work with government organizations to improve response. Through instructional modules and interactive elements, it will increase public awareness, diversify disaster kinds, and boost community engagement. Global relationships will be forged, Artificial Intelligence (AI) capabilities enhanced, user feedback gathered, business sector collaborations facilitated, policy advocacy undertaken, and multilingual assistance expanded.

REFERENCES

1. Bharathsrinath S, Dhamu M, Dhanush R, Kathiresan A. IoT Based Smart System for Women Safety. *Int Res J Eng Technol*. May 2021; 08(05): 782–790.
2. Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Commun Surv Tutor*. 2015; 17(4): 2347–2376. doi: 10.1109/COMST.2015.2444095
3. D’Orazio D, Leo M, Guaragnella C, Distanto C, Siciliano P. An IoT-aware architecture for smart healthcare systems. *IEEE Internet Things J*. 2015; 2(6): 515–526. doi: 10.1109/JIOT.2015.2417684
4. Banerjee S, Maiti P, Biswas S. Empowering Personal Safety Through Mobile Apps. In: Ponnusamy S, Bora V, Daigavane P, Wazalwar S, editors. *AI Tools and Applications for Women’s Safety*. IGI Global; Pennsylvania, United States. 2024; 91–120. <https://doi.org/10.4018/979-8-3693-1435-7.ch006>
5. Rajkumar N, Viji C, Jayavadeivel R, Prabhu Shankar B, Vetrimani E, Mary Stella J. Women Safety and Monitoring System Using Geo-Fence: Advances in SIoT (Social Internet of Things). CRC Press; Boca Raton, Florida, United States. 2023.
6. Sharma K, Londhe DD. Human Safety Devices Using IoT and Machine Learning: A Review. *2018 3rd International Conference for Convergence in Technology (I2CT)*, Pune, India. 2018; 1–7. doi: 10.1109/I2CT.2018.8529351.
7. Akash Akare, Aditya Gahile, Aniket Vishwas, Vaibhave Fokmare. Women Safety App: A Review of Existing Applications. *Int Res J Mod Eng Technol Sci*. 2023 Apr; 05(04): 7253–7255.
8. Yan He, Jiageng Chen. User location privacy protection mechanism for location-based services. *Digit Commun Netw*. May 2021; 7(2): 264–276.
9. Ahir S, Kapadia S, Chauhan J, Sanghavi N. The Personal Stun-A Smart Device for Women's Safety. *2018 International Conference on Smart City and Emerging Technology (ICSCET)*, Mumbai, India. 2018; 1–3. doi: 10.1109/ICSCET.2018.8537376.
10. Bhardwaj N, Aggarwal N. Design and Development of “Suraksha”-A Women Safety Device *International Journal of Information & Computational Technology (IJICT)*. 2014; 4(8): 787–792.
11. Monisha DG, Monisha M, Pavithra G, Subhashini R. Women Safety Device and Application-FEMME. *Indian J Sci Technol*. 2016; 9(10): 1–6. 10.17485/ijst/2016/v9i10/88898

12. Kumbhar Sonali S, Jadhav Sonal K, Nalawade Prajakta A, Mutawalli Tamanna Y. Women Security System using GSM and GPS. *Int Res J Eng Technol*. 2018 Mar; 5(3): 2406–2408.
13. Sogi NR, Chatterjee P, Nethra U, Suma V. SMARISA: A Raspberry Pi Based Smart Ring for Women Safety Using IoT. *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India. 2018; 451–454. doi: 10.1109/ICIRCA.2018.8597424.
14. Vijaylaxmi B, Renuka S, Chennur P, Patil S. Self-defense system for women safety with location tracking and SMS alerting through GSM network. *Int J Res Eng Technol (IJRET)*. 2015 May; 4(5): 57–60.
15. Prakash R, Litoriya R. Pedagogical Transformation of Bloom Taxonomy's LOTs into HOTs: An Investigation in Context with IT Education. *Wirel Pers Commun*. 2022; 122: 725–736. <https://doi.org/10.1007/s11277-021-08921-2>
16. Narasimhan VL, Bhargavi V, Prakash R. Automatic generation of Context Sensitive Dictionary using computational linguistics approach. *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, Bangalore, India. 2016; 509–514. doi: 10.1109/ICATCCT.2016.7912052.
17. Ravi Prakash, Ranjan RK. Role of ICT in Reverse Logistics: An Analytical Approach. *International Journal of Innovations in Engineering and Technology (IJIET)*. 2012; 1(3): 23–27.

APPENDIX

What is Flutter?

Flutter is an open-source framework developed and supported by Google. Front-end and full-stack developers use flutter to build an application's UI for multiple platforms with a single codebase. When flutter was launched in 2018, it mainly supported mobile app development. Flutter now supports application development on six platforms: iOS, Android, the web, Windows, MacOS, and Linux.

Installation of Flutter

To install flutter, begin by ensuring your system meets the requirements for your chosen operating system (MacOS, Windows, or Linux). Next, download the flutter Software Development Kit (SDK) from the official flutter website or clone it from GitHub. Extract the SDK to a suitable location and add the flutter bin directory path to your system's PATH variable. This step enables you to access flutter commands from the terminal or command prompt. Run flutter Doctor to check for missing dependencies like flutter itself, the Dart SDK, Android SDK, and Xcode (for macOS). Flutter download site is shown in Figure 12.

The Creation of the Project

Creating a flutter SOS Application involves several core concepts and technologies. The goal of an SOS app is to send emergency alerts or messages, typically with a user's location, to pre-defined contacts or emergency services. Here, is an overview of the theory behind building such an app: The UI of an SOS app should be simple and easy to use, especially in emergencies. Typically, this involves a single, prominent button that a user can press to initiate an SOS action. The design should be intuitive, minimizing the need for multiple steps or complicated interactions. Visual cues like bright colors and simple text help make the emergency button easily recognizable. When creating an SOS app, the core functionality is sending an alert to one or more emergency contacts. This might involve SMS, initiating a phone call, or sending a notification through other communication channels like email or instant messaging.

A common feature in SOS apps is the ability to send the user's location along with the alert. This requires accessing the device's GPS or location services. We typically use flutter packages like geolocator or location to get the user's current coordinates. Given that location data is sensitive, this also involves asking for appropriate permissions from the user and handling cases where permissions are denied. To send SMS messages or initiate phone calls, you can use flutter packages like url_launcher, telephony, or similar. Sending SMS messages usually requires permission, and you need to ensure your app handles permission requests properly.

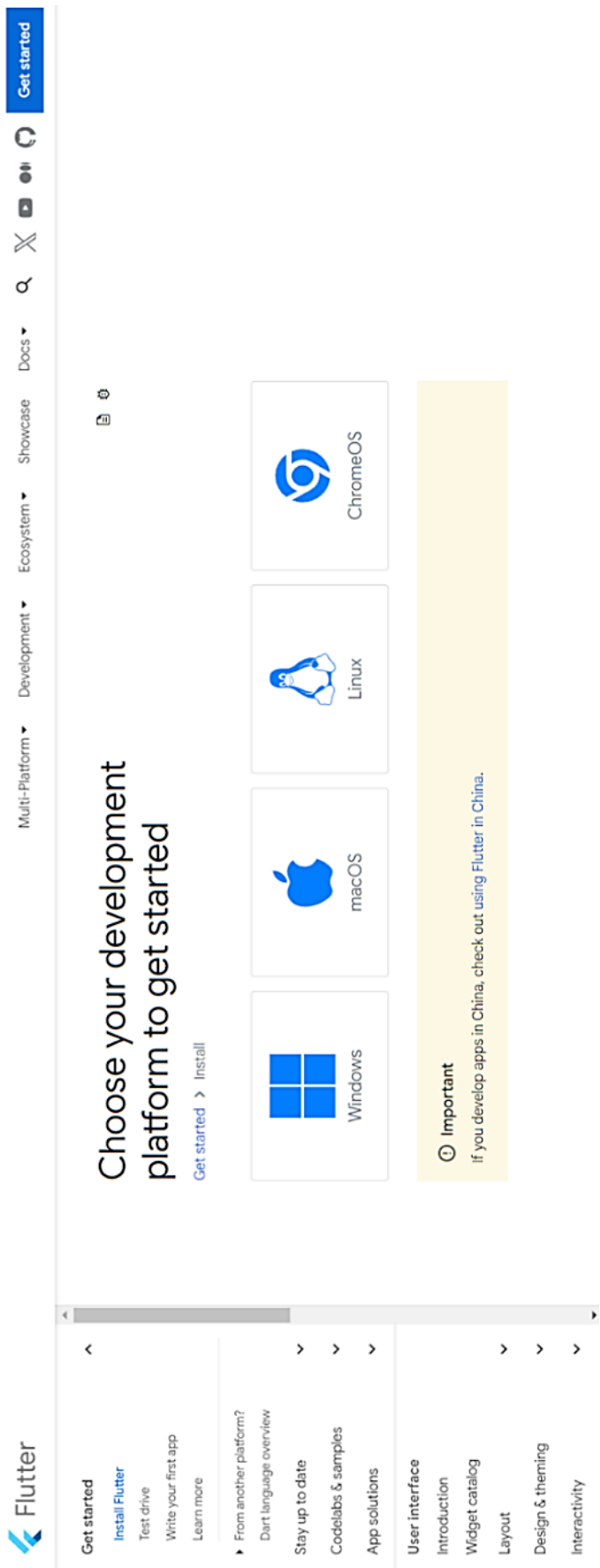


Figure 12. Flutter download site.

Flutter Project

1. *Set Up Flutter:* Install Flutter and Dart and set up an IDE like Visual Studio Code or Android Studio.
2. *Create a New Project:* Use the flutter create <project_name> command to start a new Flutter project.
3. *Structure the Project:* Navigate to your project's directory (cd <project_name>) and explore the default folders (lib, assets, etc.).
4. *Develop the UI:* Build your user interface with flutter widgets inside the lib/main.dart file.
5. *Run Your App:* Use flutter Run to build and run your flutter project on an emulator or a physical device.
6. *Add Packages:* To extend functionality (like HTTP requests, animations, or state management), add dependencies to pubspec.yaml and run flutter pub get.
7. *Configure Platforms:* Adjust platform-specific settings (like permissions) in AndroidManifest.xml (Android) and Info.plist (iOS).
8. *Test and Debug:* Use hot reload and debug tools in your IDE to refine your application.
9. *Build for Release:* Use flutter build to create release builds for different platforms (e.g., flutter build apk for Android).

Plugins

Plugins in flutter enable flutter applications to interact with platform-specific features, like accessing device hardware, making HTTP requests, integrating with third-party services, and more. Plugins consist of Dart code for flutter and native code for specific platforms (like Java/Kotlin for Android and Objective-C/Swift for iOS).

Plugins extend the capabilities of a flutter application by providing a consistent Dart interface for otherwise platform-specific functionality. This abstraction allows developers to write cross-platform applications without having to implement native features from scratch for each platform.

Types of Plugins

- *Platform Plugins:* These interact with native platform features, such as camera, location, sensors, Bluetooth, etc.
- *Utility Plugins:* Provide additional utilities like state management, animations, or data serialization.
- *Service Plugins:* Integrate with external services or APIs like Firebase, Stripe, or Google Maps.

Dependencies

Dependencies are external components that your flutter project relies on for additional functionality. They can be other Dart packages, flutter plugins, or even direct references to code repositories. Dependencies encapsulate common functionalities or services, reducing the need to write and maintain complex code from scratch.

Types of Dependencies

- *Flutter Packages:* These are Dart-based packages designed specifically for flutter applications.
- They can include UI components, plugins for accessing platform features, or utility packages.
- *Dart Packages:* These are general-purpose Dart packages that can be used in both flutter and non-flutter projects. They often provide utility functions, data structures, or general-purpose libraries.
- *Native Platform Dependencies:* These are dependencies on specific platform codes or services, typically used in plugins that bridge Dart with native iOS and Android functionalities.

State Management Bloc

Bloc is a state management pattern where the application's business logic and state transitions are encapsulated in dedicated components called "Blocs". These blocs act as intermediaries between the UI and the business logic, handling all state transitions based on incoming events.

Core Principles of Bloc

1. *Separation of Concerns*: Bloc aims to separate the application's business logic from the UI. This makes the UI components simpler and focuses on data presentation, while the business logic and state transitions are managed by the bloc.
2. *Reactive Programming*: Blocs use a reactive approach to state management, where UI components react to changes in the state emitted by the bloc.
3. *Unidirectional Data Flow*: In Bloc, the data flow is unidirectional. Events flow into the bloc, which processes them and emits new states, to which the UI components respond. Bloc operates with three key components: events, states, and blocs.
4. *Events*: These represent actions or interactions that trigger state changes. Examples include user input, network responses, or lifecycle events.
5. *States*: These represent the various states in which the application can be based on the processed events.
6. *Blocs*: These are the controllers responsible for handling events and managing state transitions. Each bloc defines its own event-handling logic and state management.

How to Publish an Application on Play Store

1. *Set Up Developer Account*: Register on the Google Play Console and pay the one-time fee (\$25). Complete account setup with contact information.
2. *Prepare Flutter App for Release*: Switch to the release build: flutter build apk release. Test the release build on a real device.
3. *Create an App in Play Console*: In the Play Console, click "Create App". Provide the required information (app name, default language, etc.). Select "Free" or "Paid" and agree to the distribution terms.
4. *Complete App Details*: Fill out product details (description, screenshots, app icon, category, etc.). Prepare the Android Application Package (APK) or AAB (Android App Bundle) for upload.
5. *Set App Configuration*: Define app pricing (if paid), countries of distribution, and age rating. Configure app permissions and privacy policies.
6. *Upload the APK/AAB*: Navigate to "App releases". Select "Production" and upload the APK/AAB. Review and confirm the release.
7. *Review and Publish*: Go through the "App Content" section to address content policies and requirements. Review the release summary for any errors or missing information. Click "Start rollout to production" to submit your app for review.
8. *Monitor and Respond*: After publishing, monitor the app's status for approval or feedback. Once approved, your app will be live on the Play store. Keep track of user feedback and update your app as needed.