CENTRE FOR SPONSORED RESEARCH AND CONSULTANCY

Student Innovative Project 2025 Report

Smart Device Locking & Monitoring for Child Safety

SIP ID - 2425S0022

Submitted By

P. Somasundharam
M. Seshavarshan
M. Navin surgith

Guided By

Dr . M R. Sumalatha

Sponsored by

Centre for Sponsored Research and Consultancy (CSRC)

Anna University, Chennai – 600 044

DEPARTMENT OF INFORMATION TECHNOLOGY
MADRAS INSTITUTE OF TECHNOLOGY, CHROMPET
ANNA UNIVERSITY

BONAFIDE CERTIFICATE

Certified that this project Report titled '**Smart Device Locking & Monitoring for Child Safety'** was submitted by Mr. Somasundharam P, Mr. Seshavarshan M, Mr Navin surgith M, who carried out the work under our supervision. Certified that to the best of my knowledge the work reported herein all the guidelines prescribed by the University was followed and after the implementation of the report.

Dr . M R. Sumalatha

Head of Department,

Department of Information Technology

Madras Institute Of Technology, Chrompet

Anna University, Chennai – 600 044

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to everyone who supported and encouraged us throughout the successful completion of this project.

First and foremost, we sincerely thank the **Centre for Sponsored Research and Consultancy (CRC)** for selecting our proposal and providing us the opportunity to work on this project.

We extend our deepest gratitude to **Dr. M.R. Sumalatha**, **Professor and Head of the Department, Department of Information Technology, Anna University, Chennai**, for her continuous encouragement, guidance, and valuable support as both our **Head of the Department** and **Project Guide**.
We also thank all the faculty members of the Department of Information Technology for their valuable insights and support throughout the course of this project.

Finally, we are deeply grateful to our families and friends for their unwavering encouragement and motivation, without which this project would not have been successfully completed.

SOMUSUNDHARAM P

SESHAVARSHAN M

NAVIN SURGITH M

ABSTRACT

In today's digital era, ensuring the safety and well-being of children in both online and offline environments is a critical responsibility for parents. This project, titled Smart Device Locking & Monitoring for Child Safety, presents a comprehensive mobile application built using Flutter and Android Studio to empower parents with remote control and monitoring of their child's device usage.

Developed using Flutter and Android Studio, the application offers comprehensive features including remote device locking and unlocking, screen time control, real-time app usage monitoring, content filtering by website blocking, SOS alert system, and live location sharing. The system empowers parents with full control through a dedicated dashboard, allowing them to manage restrictions and receive instant alerts based on their child's behavior.

The application uses Firebase as a secure and scalable backend, supporting cloud data storage, real-time database communication, and user authentication. Both parent and child modules are integrated seamlessly to ensure consistent synchronization and user experience.

This solution not only promotes digital safety and healthy device usage habits among children but also gives parents peace of mind through real-time insights and intervention tools. The system is designed to be intuitive, secure, and scalable—offering a significant step forward in smart parenting technology.

TABLE OF CONTENT

CHAPTER NO.	TITLE	PAGE NO.
	Abstract	1
	List of tables	2
	List of figures	3
	List of symbols & abbreviations	5
1	**Introduction**	7
1.1	Background and Motivation	7
1.2	Objective of the Project	7
1.3	Scope of the System	8
1.4	Technologies Used	8
2	**Literature survey**	10
2.1	Parental Control Technologies	10
2.2	Content Filtering and AI Approaches	10
2.3	GPS Tracking and Geolocation Technologies	10
2.4	Real-Time Communication in Mobile Apps	11
2.5	Privacy Considerations in Child Monitoring	11
3	**System analysis**	12
3.1	Existing System Overview	12
3.2	Proposed System	12

CHAPTER NO.	TITLE	PAGE NO.
3.3	Functional Requirements	13
3.4	Non-Functional Requirements	13
3.5	System Modules	14
4	**Design and Architecture**	16
4.1	System Architecture	16
4.2	User Interface Flow	16
4.3	Use Case Diagram	17
4.4	Sequence Diagram	18
4.5	Class Diagram	19
5	**Implementation**	20
5.1	Tools and Technologies	20
5.2	Firebase Schema and Data Flow	21
5.3	Core Features Implemented	22
6	**Testing and results**	23
6.1	Functional Testing	23
6.2	Result Screenshots	24
6.3	Summary of Outcomes	27
7	**Limitations and Future enhancements**	28
7.1	Proposed Improvements	28
8	**Conclusion**	29

LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE

Symbol / Abbreviation	Description
UI	User Interface
UX	User Experience
GPS	Global Positioning System
API	Application Programming Interface
DB	Database
RT	Real-Time
OTP	One-Time Password
HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation
SDK	Software Development Kit
IDE	Integrated Development Environment
FCM	Firebase Cloud Messaging
Firebase	Backend-as-a-Service (BaaS) platform by Google
Flutter	Cross-platform mobile app development framework by Google
Android Studio	Official IDE for Android development
Dart	Programming language used with Flutter
CRUD	Create, Read, Update, Delete (database operations)
UI/UX	User Interface / User Experience
Auth	Authentication

Symbol / Abbreviation	Description
ISP	Internet Service Provider
SOS	Emergency alert system (Save Our Souls)

1.Introduction

1.1 Background and Motivation

The widespread adoption of smartphones and mobile devices among children has raised significant concerns regarding screen addiction, exposure to inappropriate content, cyber threats, and lack of parental oversight. While technology offers educational and recreational benefits, it also poses risks when usage is unsupervised or unrestricted. In today's digital landscape, parents need intelligent tools to monitor and control their children's digital behavior without being overly invasive.

Motivated by the increasing need for proactive parental control, this project proposes the development of a mobile-based solution that enables parents to remotely lock devices, monitor screen time, track app usage, and block harmful content. The idea is to combine essential child protection mechanisms into one seamless, user-friendly mobile application that supports both child-side and parent-side interfaces.

1.2 Objective of the Project

The primary goal of this project is to design and implement a smart, scalable mobile application that allows parents to safeguard their children's digital experience. The objectives include:

1. To develop a child monitoring system using a cross-platform mobile framework (Flutter).
2. To enable remote locking/unlocking of the child's device through the parent app.
3. To log and display app usage statistics in real-time.
4. To filter websites and apps by providing a custom block list.
5. To send instant notifications (alerts, SOS messages, screen time violations) to the parent's app.
6. To allow children to send emergency location data (SOS) directly to their parents.

This project aims to create a balance between allowing children access to digital resources and ensuring their well-being through responsible usage.

1.3 Scope of the System

The **Smart Device Locking & Monitoring for Child Safety** system is designed as a mobile-based application with Firebase cloud integration. The scope includes:

1. **Real-Time Monitoring**: Logs the usage pattern of applications on the child's device.
2. **Screen Time Restriction**: Automatically locks the device when limits are exceeded.
3. **Content Filtering**: Allows parents to block specific websites by URL.
4. **Emergency Alerts**: Child can send SOS alerts with location to the parent app.
5. **Live Location Sharing**: The child app shares device location through Firebase.
6. **Parent Dashboard**: A dedicated dashboard interface to view reports, logs, and send commands.

The system currently targets Android smartphones, with plans for future expansion into iOS and web-based dashboards.

1.4 Technologies Used

To ensure efficient, scalable, and real-time operation, the following technologies are used:

1. **Flutter SDK**: Cross-platform framework for building both parent and child apps using a single codebase.
2. **Dart Language**: The programming language used with Flutter for UI and logic implementation.
3. **Android Studio**: The Integrated Development Environment (IDE) for testing and debugging Android apps.
4. **Firebase Firestore**: NoSQL cloud database for storing app logs, user data, block lists, and location information.
5. **Firebase Authentication**: For secure sign-up and login functionalities.

6. **Firebase Cloud Messaging (FCM)**: For real-time alerts and push notifications to the parent app.
7. **Google Maps API**: Integrated for location sharing and SOS-based updates from the child's device.

2.Literature survey

2.1 Parental Control Technologies

Parental control technologies have evolved from basic device locking features to intelligent systems that allow for real-time intervention and granular supervision. Early systems were limited to static website filters or fixed app blocking, but the emergence of smart mobile devices has made it possible to dynamically monitor and control device behavior.

Modern systems such as **Google Family Link**, **Qustodio**, and **Norton Family** offer parents insight into app usage, screen time, and device location. However, many of these solutions are not customizable for individual family needs or fail to offer real-time controls. This project builds on these concepts by providing a modular, parent-configurable application that addresses common gaps in traditional control systems, such as lack of app usage analytics and limited remote locking capabilities.

2.2 Content Filtering and AI Approaches

The need for filtering harmful or age-inappropriate content is a foundational feature of parental control systems. Traditional keyword-based filters were often unreliable, either failing to block malicious content or over-blocking legitimate websites.

Recent advancements in **AI-based content filtering** have enabled more adaptive systems. According to Davidson & Livingstone (2021), AI techniques can scan content contextually and filter based on behavioral patterns or known threat categories. Although this project uses static URL blocking rather than AI-driven filters, it provides parents with a flexible interface to blacklist websites based on personal criteria. The system is designed to support future AI-based dynamic filtering.

2.3 GPS Tracking and Geolocation Technologies

Geolocation tracking is critical in enhancing children's physical safety, especially in situations where they are away from home or school. The use of **GPS-enabled mobile apps** allows parents to monitor their child's location in real-time.

Research by Li and Goodchild (2020) demonstrated that **geofencing**, when integrated with location tracking, provides an additional layer of safety by notifying guardians when a child enters or exits a designated zone. Although

geofencing has not been fully implemented in the current version of the application, the project includes **live location sharing** and SOS alerts to keep parents informed about the child's whereabouts. This functionality is supported via **Firebase Firestore and Google Maps API** integration.

2.4 Real-Time Communication in Mobile Apps

Real-time responsiveness is essential for parental control systems, especially for features such as emergency alerts, device locking, or screen time warnings. Technologies such as **WebSocket** and **Firebase Cloud Messaging (FCM)** are commonly used to establish low-latency communication channels.

According to Bussone et al. (2019), real-time messaging using FCM can deliver instantaneous notifications with minimal resource overhead, making it ideal for mobile-based parental control systems. In this project, **FCM is used to notify parents about app usage summaries, block violations, and SOS alerts**, ensuring that critical information is delivered without delay.

2.5 Privacy Considerations in Child Monitoring

While monitoring tools are essential, they must also respect the privacy rights of children and avoid excessive surveillance. Over-monitoring can lead to issues of mistrust or dependency, and legal frameworks such as **COPPA (Children's Online Privacy Protection Act)** emphasize the importance of consent and minimal data collection.

Livingstone et al. (2021) argue that ethical parental control solutions should focus on empowering rather than restricting children. In line with this philosophy, the current system uses **Firebase Authentication** to ensure that only verified parents can access sensitive data like usage history or location information. No personal media or conversations are logged, and all data is securely stored and encrypted, following best practices in user privacy.

3.System Analysis

3.1 Existing System Overview

Existing parental control systems vary in scope and effectiveness. Many rely on static control mechanisms, where content is blocked or filtered based on predefined keyword lists or application categories. Tools like **Google Family Link**, **Qustodio**, and **Screen Time** allow parents to set daily limits, monitor app installations, and restrict usage during specific hours.

However, these systems have limitations:

1. Lack of customizable real-time control
2. No emergency alert mechanism
3. Limited granularity in app usage tracking
4. Inconsistent notifications
5. Insufficient flexibility in content filtering

Moreover, these solutions are often tied to specific platforms (e.g., Android only) or require complex setup and subscription costs. There is also a dependency on continuous internet connectivity, and very few support direct real-time interactions between parent and child devices in an efficient, low-latency manner.

3.2 Proposed System

The proposed **Smart Device Locking & Monitoring for Child Safety** system is designed to address the shortcomings of existing systems by introducing an easy-to-use, cross-platform mobile application powered by **Flutter** and **Firebase**. It provides real-time monitoring, device control, and parental alerts through a dual-interface structure — a **child module** and a **parent module**.

Key features include:

1. **Remote device locking/unlocking** via the parent app
2. **Live app usage tracking** and screen time summaries
3. **Custom website blocking** via a block list

4. **Instant alert notifications** (for time violations, SOS, website access)
5. **Child-side SOS alert feature** with location sharing
6. **Firebase-based real-time communication and data logging**

This cloud-integrated, mobile-first solution empowers parents to proactively manage both digital habits and safety without depending on third-party subscriptions or invasive tracking.

3.3 Functional Requirements

The functional requirements of the system define what it is intended to do. The system will:

1. Allow parents to **remotely lock/unlock** the child's device
2. Record and display **real-time app usage statistics**
3. Enable **custom screen time settings** for individual apps
4. Allow **URL-level website blocking**
5. Enable children to send **SOS alerts** to the parent's device
6. Provide **live location updates** on request or during SOS
7. Push **real-time alerts** to the parent's app for blocked website access, time limit violations, and emergency signals
8. Provide **login/signup functionality** for parents and children using Firebase Authentication

3.4 Non-Functional Requirements

In addition to core functions, the system must also meet certain quality and operational standards. These include:

1. **Security**: Data must be encrypted and accessible only to authenticated users.
2. **Scalability**: The app should support multiple child profiles and adapt as usage increases.

3. **Platform Independence**: The system should run seamlessly on Android and be scalable to iOS with the same codebase.
4. **Usability**: The user interface should be intuitive for both parents and children.
5. **Maintainability**: The codebase should be modular, clean, and easy to update.
6. **Performance**: Real-time features (e.g., alerts, app logging) must operate with minimal delay.
7. **Availability**: System should be accessible anytime and require minimal downtime.

3.5 System Modules

The application is divided into the following core modules:

1. **Parent Interface Module**
    i. Dashboard for app usage summary
    ii. Screen time settings and remote locking
    iii. Block list management for websites
    iv. Alert notification history and SOS tracking
2. **Child Interface Module**
    i. App usage logger
    ii. SOS alert trigger with optional message
    iii. Location updater (on-demand or via SOS)
3. **Authentication Module**
    i. Firebase Authentication for login and signup
    ii. Secure access to individual dashboards
    iii. Role-based access (parent or child view)
4. **Notification Module**
    i. Firebase Cloud Messaging integration

ii. Push alerts for restricted actions, SOS, or warnings

5. **Database Module**
    i. Firebase Firestore used to log usage, restrictions, block list, alerts
    ii. Optimized for real-time syncing and minimal latency
6. **Location Tracking Module**
    i. Google Maps API used for location logs
    ii. Integrated into SOS and parent-triggered checks

Each module works independently but is interconnected through Firebase services to ensure real-time synchronization and a seamless experience.

4.Design and Architecture

4.1 System Architecture

The system is built on a **client–server architecture**, leveraging Firebase for real-time cloud data management. It comprises two client apps — the **Parent App** and the **Child App** — that communicate with a common backend to store and retrieve data securely.

1. **Parent App (Flutter):**
    i. Acts as a control dashboard for screen time monitoring, blocking websites, viewing usage logs, receiving alerts, and unlocking devices remotely.
2. **Child App (Flutter):**
    i. Records app usage, sends alerts, and shares the child's location upon SOS.
3. **Firebase Backend:**
    i. Handles data synchronization (Firestore), user authentication, and push notifications via FCM.
4. **Communication Model:**
    i. Uses HTTP (via Firebase SDK) for data write/read and Firebase Cloud Messaging for real-time alerting.

Key Components:

1. Firebase Firestore (NoSQL DB)
2. Firebase Authentication
3. Firebase Cloud Messaging
4. Google Maps API (for location)
5. Flutter Framework
6. Android SDK (build & deploy environment)

This architecture supports a **loosely coupled, scalable, and real-time system** ideal for mobile parental control.

4.2 User Interface Flow

The UI is designed to be **simple, responsive, and platform-agnostic**, with clearly separated modules for parent and child roles.

Parent App UI Flow:

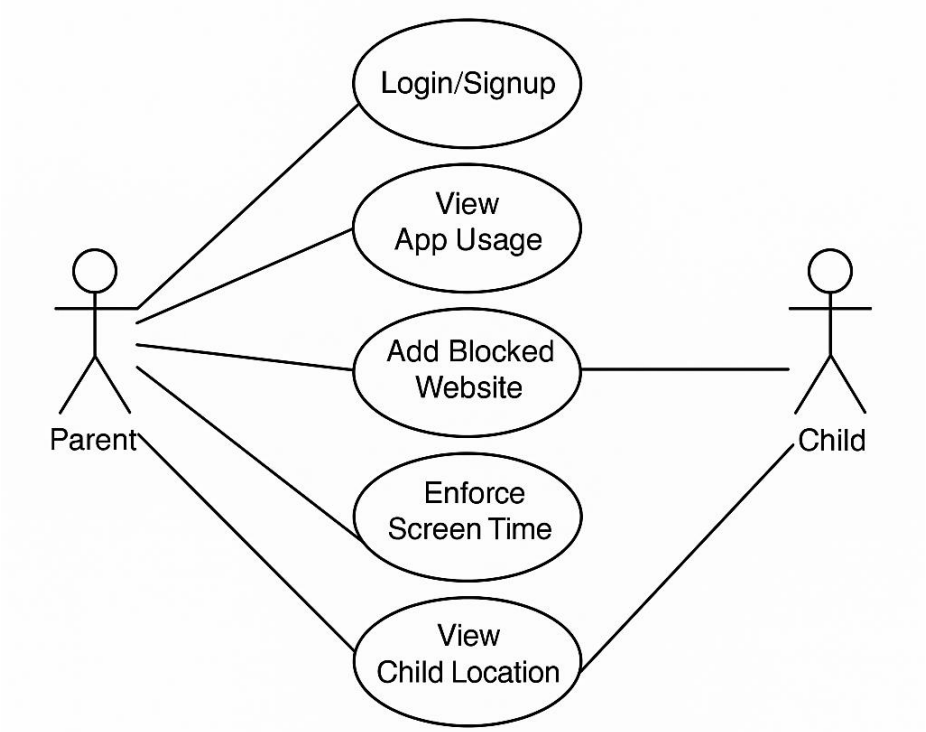
1. Login/Signup (Firebase Auth)
2. Home Dashboard
3. View App Usage → Daily Logs / Filter by App
4. Set Screen Time Limits → App-wise / Global
5. Block Website (via URL)
6. Receive Alerts (Block Violation / SOS / Usage Time Crossed)
7. Unlock/Lock Device Remotely
8. View Location on Map

Child App UI Flow:

1. Login with Device ID
2. App Usage Logger (runs in background)
3. SOS Button → Sends Location & Notification
4. Screen Lock Trigger (as per time policy)
5. App usage monitoring view

4.3 Use Case Diagram

A typical use case diagram includes the following actors and functionalities:



Actors:

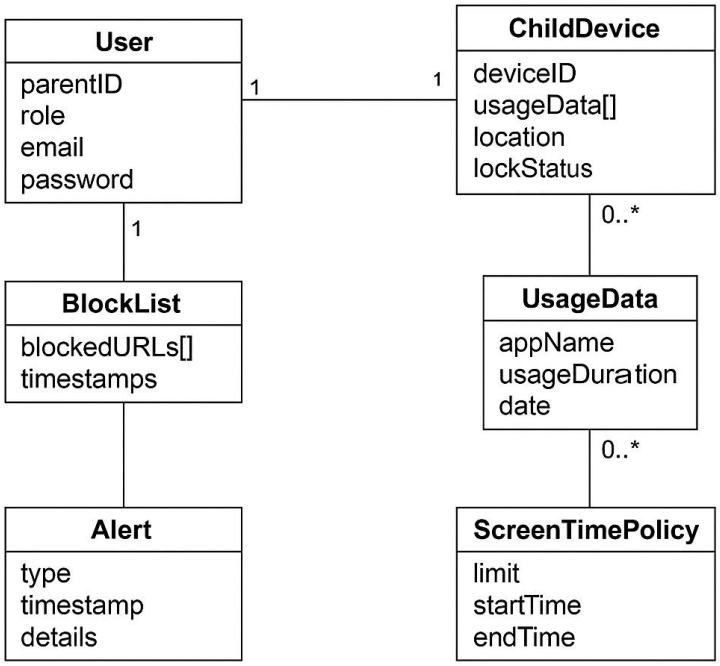
1. Parent (Primary User)
2. Child (Monitored User)
3. Firebase (System Backend)

Use Cases:

1. Login/Signup
2. View App Usage
3. Lock/Unlock Device
4. Add Website to Blocklist
5. Receive Notification
6. View Child Location
7. Send SOS Alert
8. Enforce Screen Time Limits

4.4 Class Diagram

The class diagram represents the major components and their relationships:



Classes:

1. User (parentID, role, email, password, childIDs[])
2. ChildDevice (deviceID, usageData[], location, lockStatus)
3. UsageData (appName, usageDuration, date)
4. BlockList (blockedURLs[], timestamps)
5. Alert (type, timestamp, details)
6. ScreenTimePolicy (limit, startTime, endTime)

Relationships:

1. A User can control multiple ChildDevices
2. Each ChildDevice maintains multiple UsageData entries
3. Alerts are triggered based on rule violations and tied to parent account

5.Implementation

5.1 Tools and Technologies

To build a scalable, cross-platform, and responsive system, the following tools and technologies were used:

Tool/Technology	Purpose
Flutter SDK	UI framework for building Android (and future iOS) apps
Dart Language	Programming language used to develop Flutter applications
Android Studio	Integrated Development Environment (IDE) for Flutter app development
Firebase Authentication	Handles secure login and signup for parents and child profiles
Firebase Firestore	Cloud NoSQL database to store user profiles, usage logs, and alerts
Firebase Cloud Messaging (FCM)	Push notification service used for sending alerts and status updates
Google Maps API	Enables live location sharing and mapping via SOS alerts
Git/GitHub	Version control and team collaboration

The system was designed to run on **Android 10+ devices**, with future scope to port to iOS and Web platforms using the same Flutter codebase.

5.2 Firebase Schema and Data Flow

Firebase Firestore is used as the central backend service to store and sync real-time data between child and parent devices. Below is the high-level structure of the schema:

Collections:

1. /users
    - Stores parent and child accounts
    - Fields: uid, email, role, deviceID, childIDs[]
2. /devices
    - Represents each child's device
    - Fields: deviceID, locked, currentLocation, screenTimePolicy
3. /usageLogs
    - Stores app usage logs
    - Fields: appName, duration, timestamp, deviceID
4. /alerts
    - Alerts pushed to the parent
    - Types: SOS, Limit Exceeded, Website Blocked
5. /blockedSites
    - List of blocked URLs per parent-child mapping
    - Fields: parentID, deviceID, blockedURL, timestamp

Data Flow:

1. When a child app runs, it logs app usage to /usageLogs.
2. If a screen time policy is violated, an alert is pushed to /alerts, and FCM sends a notification.
3. If the SOS button is pressed, the child's current GPS coordinates are logged to /devices → currentLocation and also pushed to /alerts.

4. Parents retrieve this information via real-time listeners or through periodic requests via their app.

This modular schema ensures scalability and real-time responsiveness with minimal reads/writes.

5.3 Core Features Implemented

The following are the major features successfully developed and tested in the system:

1. Screen Time Management
Parents can define daily screen time limits. The system tracks usage in real-time and automatically locks the device when the time is exceeded. Unlocking requires parental approval through the app.

2. Real-Time App Usage Monitoring
The child app tracks which apps are being used and for how long. This data is pushed to Firebase and displayed in charts or tables inside the parent app.

3. Website Content Filtering
Parents can manually add harmful or unwanted URLs to a block list. If the child tries to access these URLs, a notification is sent and access is denied.

4. Instant Notifications and Alerts
All critical events (e.g., SOS alert, screen time violation, blocked URL access) trigger real-time alerts through Firebase Cloud Messaging. These are shown in the parent dashboard immediately.

5. SOS Alert System
The child app includes an SOS button. When pressed, it sends the device's current location along with a timestamped alert message to the parent.

6. Remote Device Lock/Unlock
The parent app includes a toggle to lock/unlock the child's device in real time.

7. Secure Authentication
Signup and login are handled securely using Firebase Authentication. Separate roles (parent vs. child) are assigned after account creation.

6.Testing and Results

6.1 Functional Testing

To ensure the reliability and correctness of each module in the system, comprehensive functional testing was conducted. The testing focused on verifying whether each feature performed its intended task under normal and edge-case conditions.

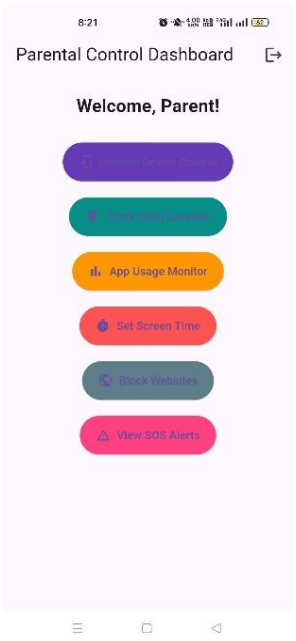
Feature Tested	**Test Case Description**	**Expected Result**
User Authentication	Login with valid and invalid credentials	Accept valid login, reject invalid
Screen Time Restriction	Set daily screen time and exceed the limit	Device should auto-lock
App Usage Logging	Use multiple apps for varying durations	Logs correctly recorded and displayed
Website Blocking	Access a blocked site from child device	Access denied and alert sent
SOS Alert	Tap SOS from child app	Location alert sent to parent device
Location Sharing	Fetch location on-demand or during SOS	Accurate live location shown
Notifications	Trigger blocked site or time violation	Parent receives real-time FCM alert
Remote Lock/Unlock	Toggle lock from parent app	Child’s screen state changes

6.2 Result Screenshots

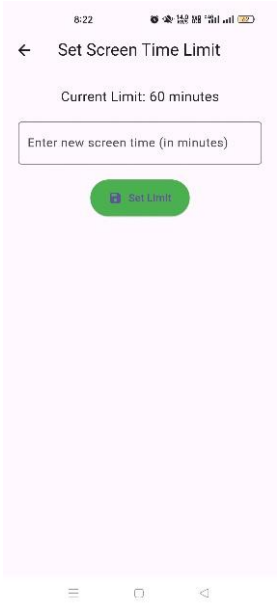
Screenshots were taken from both the **parent** and **child** applications to demonstrate core features and outcomes.

Parent App:

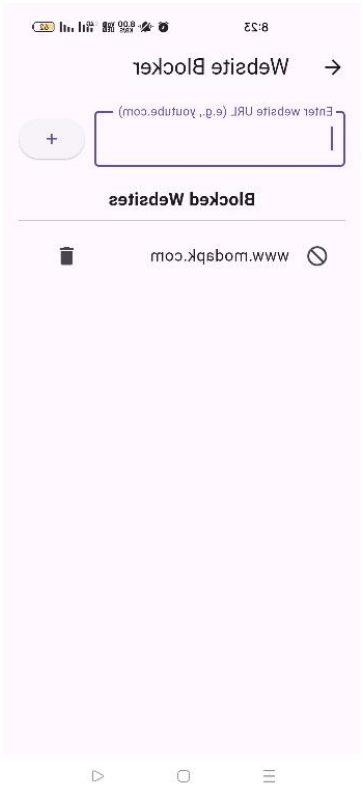
1. **Dashboard Summary** – Real-time app usage statistics (e.g., WhatsApp, YouTube).



2. **Screen Time Settings** – User-defined limits per app and total screen time.



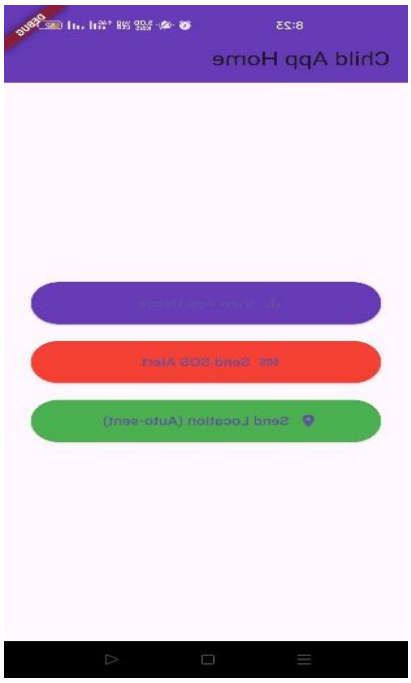
3. **Blocked Website Entry** – Form to add new blocked URLs.



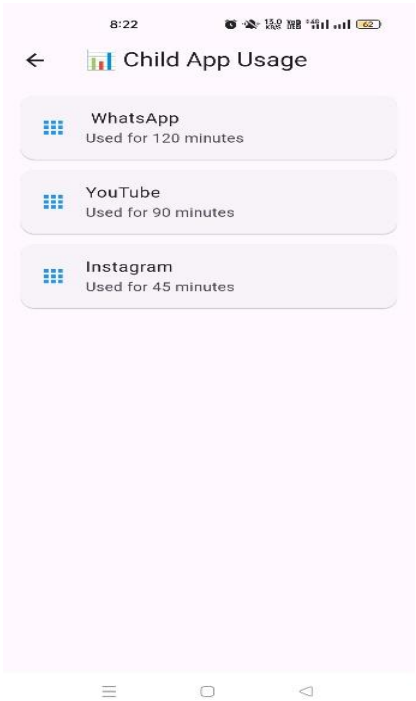
4. **Alerts Section** – Display of notifications such as “Access Denied” or “SOS Received”.

Child App:

1. **SOS Button Interface** – Simple layout with instant alert trigger.



2. **App Usage Logger** – Background service tracking app names and durations.



6.3 Summary of Outcomes

The implemented features were tested successfully and demonstrated the system's effectiveness in managing and monitoring a child's device usage. Below is a summary of key outcomes:

- Accurate tracking of app usage with real-time database logging.
- Successful enforcement of custom screen time policies with auto-lock functionality.
- Real-time push notifications (SOS alerts, time violations, blocked URLs) received within 1–2 seconds.
- Firebase authentication handled secure login and access control with low latency.
- Live location sharing and Google Maps integration verified in multiple test cases.
- Remote control (lock/unlock) performed reliably across network conditions.

The system delivered all intended results with a clean, responsive UI and stable backend performance. Overall, the project has proven to be a practical and deployable solution for smart child safety and parental control.

Proposed Improvements

To enhance the current system, several improvements are planned:

1. **Geofencing**: Implement safe zones (e.g., home, school) with entry/exit alerts.
2. **OAuth 2.0**: Upgrade to industry-standard authentication for enhanced security.
3. **Web Dashboard**: Introduce browser-based access for easier parental control.
4. **iOS Support**: Extend app compatibility to iPhones using Flutter's cross-platform capabilities.
5. **AI-Based Filtering**: Add intelligent content filtering to block harmful material in real time.
6. **Usage Reports**: Generate weekly/monthly reports with usage trends and alerts.

These improvements aim to make the system more comprehensive, secure, and user-friendly.

Conclusion

The "Smart Device Locking & Monitoring for Child Safety" project successfully addresses a growing concern in today's digital world—ensuring the safe, balanced, and monitored use of smartphones by children. Through this mobile application built using Flutter and Firebase, we have enabled a seamless bridge between parents and their child's device, providing effective real-time control and monitoring.

The system's core modules—such as remote screen locking, app usage tracking, content filtering, SOS alert functionality, and real-time notifications—were all implemented and validated through extensive testing. These features empower parents to set boundaries, receive timely alerts, and take necessary actions to protect their child both digitally and physically.

By utilizing Firebase services, the system achieves cloud-based scalability, secure authentication, and low-latency data synchronization. The user-friendly interface ensures that both parents and children can interact with the system efficiently without any steep learning curve.

This project not only serves as a technical achievement but also contributes to digital parenting in a meaningful way. It demonstrates how technology, when used responsibly, can provide a balanced digital lifestyle for children while preserving their independence. With continued enhancements such as geofencing, AI-based content filtering, and biometric access control, the system has the potential to evolve into a comprehensive parental control ecosystem adaptable to various environments.

In conclusion, the project successfully meets its objectives and lays a solid foundation for future innovation in the field of child digital safety and smart device supervision.