American International University-Bangladesh (AIUB)

Faculty of Science & Technology (FST)

Department of Computer Science

MOBDOG: Android Security

A Software Requirement Engineering Project Submitted
By

Semester: Spring_22_23		**Section: F**	**Group Number: 7**	
SN	Student Name	Student ID	Contribution (CO1+CO2)	Individual Marks
08	AFTAB, RAKIN SAD	20-41991-1	25%	
09	AHMED, MIR MARUF	20-42082-1	25%	
11	MOLLAH, MD RAZIB	20-42153-1	25%	
37	RAJIB, AREFIN RAHMAN	20-43713-2	25%	

The following course outcomes will be considered in evaluating the project

Evaluation Criteria	Total Marks (50)	
Introduction, Format, Submission, Defense	[10 Marks]	
System Overall Description & Functional Requirements	[10 Marks]	
System Quality Attributes and Project Requirements	[10 Marks]	
UML and E-R Diagram with Data Dictionary	[10 Marks]	
UI/UX Prototyping	[10 Marks]	

Software Requirements Specification

for

<MOBDOG: Android Security>

Version 4.0 approved

Prepared by **<AFTAB, RAKIN SAD>**
<AHMED, MIR MARUF>
<MOLLAH, MD RAZIB>
<RAJIB, AREFIN RAHMAN >

< American International University-Bangladesh >

< March 26, 2023>

Table of Contents

Table of Figures

Revision History

Name	Date	Reason for Changes	Version
First Review	15-03-2023	New Feature Implementation	V1.0
Second Review	19-03-2023	Design Implementation	V2.0
Third Review	26-04-2023	Document Update	V3.0
Final Review	30-04-2023	Error Fixing	V4.0

1. Introduction

1.1 Purpose

Development team, stakeholders, & customers may clearly understand the requirements for the software product thanks to the software requirement specification (SRS). The primary purpose of an SRS is to act as a blueprint for the development team, which outlines the software product's requirements and the features it must have. The SRS should be clear, concise, and unambiguous so that the software team knows what it needs to do and what features it needs. To ensure that the stakeholders and clients are on the same page as the development team, the SRS is also a crucial tool for communication between the development team and the stakeholders or clients. It provides a clear understanding of the software product's scope, functionalities, and anticipated deliverables. The SRS is also a reference document that can be used to check how well the software product works. It describes the testing methods and acceptance criteria that will be used to judge how well the software works. Lastly, the SRS can also be a legal document that outlines the software product's expectations and requirements. This protects both the development team and the clients or stakeholders.
In summary, an SRS's primary goal is to ensure that the development team, stakeholders, and clients understand the software product's requirements and functions. This ensures that everyone is on the same page and that the software product meets the expected deliverables.

1.2 Document Conventions

1.2.1 Abbreviations List

SRS	Software Requirement Engineering
GDPR	General Data Protection Regulation
PCI DSS	Payment Card Industry Data Security Standard
HIPPAA	Health Insurance Portability and Accountability Act
OTP	One Time Password
IEEE	Institute of Electrical and Electronics Engineers
SDD	System Design Documents
SMS	Short Message Service
CPU	Central Processing unit
RAM	Random Access Memory
OS	Operating System
UML	Unified Modeling Language
UI	User Interface
UX	User Experience

IMEI	International Mobile Equipment Identity
IT	Information Technology
PDF	Portable Document Format
HTML	Hyper Text Markup Language
USB	Universal Serial Bus
GPS	Global Positioning System
PIN	Personal Identification Number
SIM	Subscriber Identity Module

1.2.2 SRS Document Writing Format

The SRS document follows the IEEE standard for documentation. The document is structured to provide clear and concise information about the requirements for developing an Anti-Theft Android Application.

- **Section Heading and Subheadings:** The document follows a clear and consistent hierarchy of section headings and subheadings. The main sections are numbered and titled in bold, with subsections following a standard formatting convention.
- **Font and Formatting:** The font used in this document is standard Times New Roman, size 12. The headings are bold, and the subheadings are italicized. The requirements are presented in bullet point format to clarify and improve readability.

Overall, these document conventions are designed to provide clarity and consistency throughout the document.

1.3 Target Audience and Reading Recommendations

1.3.1 Targeted Audience

This Software Requirement Specification document is meant to be viewed by a variety of people, including

- **Developers:** who will be responsible for designing, implementing, and testing the application.
- **Project Managers:** Who will manage the project, ensure and establish the requirements, and monitor the development team's progress.
- **Marketing Staff:** Who will promote the application to potential users.
- **Users:** Who will be using the application to prevent mobile device theft.
- **Testers:** Who will be in charge of testing the application to make sure it complies with the specifications listed in this document.
- **Documentation Writers:** Who will be responsible for creating user manuals, installation guides, and other documentation for the application.

1.3.2 Document Content and Organization

The organization of the SRS document is as follows:

- **Introduction:** This section gives an overview of the Anti-Theft Android Application, including its goal, its range, and the meanings of some key terms.
- **Overall Description:** This section describes the application in detail, including any requirements or restrictions that must be considered when designing and implementing the application.
- **Specific Requirements:** This part specifies the software's both functional and non-functional needs.
- **Interface Requirements:** Includes UML diagrams and UI/UX Design Specification, which may be helpful to the reader, as well as more information and possible scenarios.

1.3.3 Suggested Reading Sequence

- Everyone should read the "Introduction" section to get a general idea of the application's focus.
- Developers and project managers should review the "Overall Description" section to learn about the application's needs and limits.
- Marketing staff and users should focus on the overall description section and the specific requirements section to understand the features and functionality of the application.
- Testers should focus on the "Specific Requirements" section to determine what test cases they must make and run.
- Writers of documentation should pay attention to all parts of the document so they can understand the needs and limits of the application and write clear and concise documentation.

1.4 References

The following documents and web addresses are referred to in this SRS:

Title: Anti-theft application for android based devices
Author: Azeem Ush Shan Khan
Version Number: 1.0
Date: February 2014
Source:https://www.researchgate.net/publication/271482528_Anti-theft_application_for_android_based_devices

Title: iGuard A Real-Time Anti-Theft System for Smartphones
Author: Meng Jin & Yuan He
Version Number: 1.0
Date: January 25, 2018
Source: https://ieeexplore.ieee.org/document/8269825/authors#authors

Title: Fraud and Security Group
Author: GSMA
Version Number: 1.0
Date: December 01, 2022,

Source: https://www.gsma.com/aboutus/workinggroups/fraud-security-group

Title: Android User Interface Guidelines
Author: Google Inc.
Version Number: 1.0
Date: [Date of publication]
Source: https://developer.android.com/design

Title: Android Development Documentation
Author: Google Inc.
Version Number: [Latest Version]
Date: [Date of publication]
Source: https://developer.android.com/docs

These references provide additional information that is relevant to the development of the Anti-Theft Android application. The Vision and Scope Document and the Use Case Document explain what the application is for and how it works. The System Requirements Specification goes into more detail about the application's needs. The Android User Interface Guidelines guide the application's user-friendly interface design. At the same time, the Android Development Documentation gives technical information about making the app for the Android platform.

2. Overall Description

2.1 Product Perspective

This SRS document aims to define and describe the requirements for developing and implementing a mobile security threat and countermeasure application that provides users with a comprehensive suite of tools and resources to protect their mobile devices from potential security breaches and attacks. Mobile devices, including smartphones and tablets, have become integral to our daily lives. They store a vast amount of personal and confidential data, such as banking details, passwords, and personal identification information. With this surge of mobile technology, there has been an increase in mobile security threats, such as malware, phishing attacks, identity theft, and other cyber-attacks. This mobile security application aims to protect against mobile security threats like hacking or malware attacks, theft, and loss, which can take control of a mobile device and extract data and other confidential information. This application's primary goal is to protect the personal and confidential data stored on a mobile device from malware, phishing attacks, identity theft, and other forms of cyber-attacks.

All mobile devices running the Android or iOS operating systems will be able to install a comprehensive mobile security application. The software should provide an easy-to-use and customizable environment, allowing users to configure their security preferences and settings quickly and easily. This SRS document also lists the requirements for a mobile security product that aims to protect users of smartphones and tablets from the growing number of threats they face. The product is a new, self-contained product that is designed to provide.

Comprehensive security measures to protect against a range of potential security threats. The product's origin is the growing reliance on mobile devices for personal and business use, increasing the number and sophistication of mobile security threats. These threats include malware, phishing attacks, unauthorized access, and theft or loss of devices.

The product intends to solve these threats, offering a range of countermeasures to help users protect their devices and data. This may include features such as antivirus software, intrusion detection and prevention, encryption tools, and remote wiping capabilities. While the product is standalone, it may need to interface with other components of a more extensive system, such as existing IT infrastructure or cloud-based services. A diagram showing the major features of the overall design, including external interfaces and subsystem interconnections, will be included in the document to provide a clear understanding of how the product fits into the broader system.

2.2 Product Functions

- Identify and analyze mobile security threats.
- Provide countermeasures and best practices to mitigate mobile security threats.
- Educate users on mobile security risks and how to protect themselves.
- Implement a secure mobile environment for users.
- Detect and prevent unauthorized access to mobile devices and data.
- Protect against malware and other malicious attacks on mobile devices.
- Secure mobile data storage and transmission.

- Provide remote wipes and other security features for lost or stolen devices.
- Monitor and report on mobile security incidents and threats.
- Continuously update and enhance mobile security measures to stay ahead of emerging threats.

2.3 User Classes and Characteristics

- **General users** use mobile devices for personal use, including internet browsing, social media, emails, chats, and casual gaming. They may need to gain technical expertise or knowledge of mobile security threats.

- **Business or corporate** users use mobile devices for work-related tasks such as emailing, video conferencing, document sharing, and accessing company networks. They have technical expertise and are aware of security risks that can impact confidential information and organizational operations.

- **IT administrators** are users responsible for managing an organization's mobile devices and network infrastructure. They have advanced technical expertise and knowledge of mobile devices and networks' security risks and vulnerabilities.

- **Application developers** are users who develop mobile applications for various purposes. They have technical expertise in coding and application development and are responsible for ensuring that applications are secure and free from vulnerabilities.

Business and corporate users and IT administrators are the most essential user classes of these products, as they handle confidential and sensitive information and require high-level security measures. The general public and application developers are also important, but their overall impact on mobile security threats and countermeasures is relatively lower compared to the former two.

2.4 Operating Environment

As mobile technology has improved, mobile devices have become an important part of our everyday lives. We store sensitive information and make financial transactions on our phones and tablets, which makes mobile security a very important issue. This SRS document describes the system requirements for a mobile security app that protects mobile devices from different security threats. A mobile security app needs to have antivirus protection, anti-phishing protection, firewall protection, anti-theft protection, and updates for the operating system. The application should also meet non-functional requirements like performance, compatibility, usability, security, and dependability. By meeting these requirements, the app will be able to protect mobile devices from all kinds of security threats. The software will operate in a mobile environment, specifically on smartphones and tablets running on widely used operating systems such as iOS and Android. The software must be compatible with the latest versions of operating systems to ensure that it can effectively address emerging mobile security threats. It must also be designed to work seamlessly with other applications and software components that are

commonly installed on mobile devices, such as email clients, social media apps, and messaging services.

To make sure the software works well with other parts and apps on mobile devices and runs at its best, it will be thoroughly tested and optimised. The software must also be made so that it uses resources as efficiently as possible and doesn't hurt the performance or battery life of mobile devices. To ensure that the software can effectively protect against mobile security threats, it must operate in a secure environment with appropriate permissions and access controls in place. The software must be designed to operate in a sandboxed environment, isolating it from other applications on the mobile device, thus preventing hackers and other malicious actors from exploiting vulnerabilities and gaining access to sensitive information. Overall, the software must be designed to operate in a highly secure and controlled environment that is optimized to work with mobile hardware, operating systems, and other software components and is capable of effectively counteracting and preventing mobile security threats.

2.5 Design and Implementation Constraints

There are a number of things or problems that could make it harder for developers to come up with and implement solutions to mobile security threats that includes:

- **Regulatory and Compliance Policies:** Developers need to ensure that their mobile security solutions comply with various regulatory frameworks and industry standards, such as GDPR, HIPAA, PCI DSS, etc. This limits the options available as developers need to adhere to regulatory requirements in terms of data privacy, security, and confidentiality.

- **Hardware Limitations:** Mobile devices have limited hardware capacity, such as limited memory, battery life, and processing power. This could limit the options available to developers in terms of designing security solutions that consume less memory, generate less network traffic, and perform efficiently to minimize the impact on the device's battery life.

- **Interfaces to Other Applications:** Mobile devices can talk to other apps and services, such as third-party libraries, cloud services, and social media platforms. These interfaces could pose security risks and may limit what developers can do to reduce those risks, such as by setting permissions, encrypting data, and authenticating users.

- **Specific Technologies, Tools, and Databases:** Developers may be restricted to using particular technologies, tools, and databases that are mandated by their organization or the client. These restrictions could limit the options available to developers, and they may not be able to use the latest advancements in technology to enhance security.

- **Language and Protocol Requirements:** Developers may have to use a certain programming language and communication protocol to make sure that their system, app, or service works with other systems, apps, or services. In such cases, developers may not have the flexibility to use other languages or protocols that offer better security.

- **Security Considerations:** When creating security solutions, developers need to make sure they address security issues like network-based attacks, malware, data theft, and

unauthorized access. Because of these security concerns, developers may not have as many options for designing, testing, and deploying secure apps.

- **Programming Standards and Design Conventions:** Developers may have to follow certain programming standards and design conventions to ensure that their software maintenance is easy, consistent and also works with other software. This could make it harder for developers to put in place effective security measures that go against the established norms.

In order to ensure that their products satisfy their clients' expectations and adhere to industry standards, developers must keep these constraints in mind while creating and implementing mobile security solutions.

2.6 User Documentation

The following user documentation elements will be included with the software:

User Manual: This document will show you how to use the software step-by-step, including how to set up and configure security settings, how to install updates and patches, and how to fix common problems.

Live Chat: The software will have an online chatting system that shows users how to use the program based on their current situation using.

Tutorials: The software will include video tutorials and interactive walkthroughs to help users learn how to use different features and functions.

The known user documentation delivery formats or standards are:

PDF: The user manual will be delivered in PDF format, which makes it easy for users to view and print the document.

HTML: The online help system will be delivered in HTML format, which lets users access the documentation from any device with a web browser.

Video: The tutorials will be delivered as video files, which can be viewed on any device that supports video playback.

3. System Requirements

3.1 System Features

1. Software Access Permission's

Functional Requirements (FRs)

1.1 The application shall prompt the user to allow background access to the location.
1.2 When using a mobile network, the software shall prompt the user to turn on or off their data connection.
1.3 The software shall request access to the camera from the user.
1.4 The user shall allow the SMS and storage access requests by the system.

Priority Level: High
Precondition: The system has the necessary software capabilities to request, grant, and revoke permissions for accessing sensitive data or hardware features. The system should also have a secure method for storing and managing consent for different individuals. The user should have control over the permissions granted to individual applications and be able to customize permission settings according to their preferences.
Cross-references: N/A

2. Software Access Permission's

Functional Requirements (FRs)

2.1 The software shall ask the user to access the location in the background.
2.2 The software shall have an IMEI viewer button on the registration page.
2.3 The registration credentials will be verified by OTP services and IMEI checker, and verified credentials will be stored in the central database.
2.4 The number of registrations with the same email and IMEI can't exceed its limit (1 times).
2.5 If the registration is successful, the login page will be displayed.

Priority Level: High
Precondition: The user has downloaded and installed the software on their Android device and has an active internet connection. The user knows the IMEI number of their device and has a valid email address. The user is not already registered with the same email address and IMEI number.
Cross-references: N/A

3. IMEI Viewer

Functional Requirements (FRs)

3.1 The software shall allow users to view their device IMEI by a button.
3.2 If the button is pressed an instructions page will appear.

Priority Level: Low

Precondition: The user must be on the registration page of the software where the IMEI viewer button is available and the user must have an android device.
Cross-references: 2.2

4. Forget Password

Functional Requirements (FRs)

4.1 The system shall provide a user-friendly interface for initiating the password reset process.
4.2 The system shall clearly indicate the required steps for resetting the password.
4.3 The system shall verify the user's identity before initiating the password reset process.
4.4 To start the reset procedure, the system shall ask the user for their email address or phone number linked to their account.
4.5 The system shall provide a secure method for resetting the password, such as sending a reset OTP to the user's email address.
4.6 The system shall generate a unique reset OTP for each password reset request to ensure security.
4.7 The system shall provide a secure method for resetting the password, such as sending a reset OTP to the user's email address.
4.8 The system shall inform the user if the new password does not match the requirements.
4.9 The system shall confirm that the password is successfully reset.

Priority Level: High
Precondition: The user must have previously created an account with the system and have access to the email address associated with the account to receive the reset OTP. Users must be able to verify their identity before initiating the password reset process. The system must have a secure method for resetting the password, such as sending a reset link to the user's email address or phone number. The system must enforce password policy requirements, such as minimum length and complexity, for the new password.
Cross-references: N/A

5. OTP Verification

Functional Requirements (FRs)

5.1 The system shall provide multiple delivery methods for sending OPTs, such as SMS, and email.
5.2 The system shall set a specific validity period for the OTP to ensure that it is valid for a limited time only.
5.3 The system shall notify the user if the OTP has expired and prompt them to initiate the verification process again.
5.4 The system shall verify the OTP entered by the user and allow access to the account if the OTP is correct.
5.5 The software shall warn if the user enters an incorrect OTP and resend it to enter a new one.

5.6 The system shall prevent multiple failed OTP attempts and initiate security measures, such as temporarily disabling the account or notifying the user if there are suspicious activities.

Priority Level: High
Precondition: The user has access to the contact information associated with their account, such as their mobile email address, and the system has a secure method for sending OTPs.
Cross-references: 4.5, 4.6

6. Software Settings

Functional Requirements (FRs)

6.1 The software shall convey a user-friendly interface for accessing & modifying the software settings.
6.2 The application shall make it simple for the user to search for and access the options they wish to change.
6.3 The software shall enable user customization of application settings in accordance with preferences.
6.4 The system shall provide the user with a range of options for customization.
6.5 The system shall save any changes made to the software settings.
6.6 The system shall ensure that the saved settings are applied consistently across all instances of the software.
6.7 The application shall include default settings and allow the user to restore the default settings if desired.

Priority Level: High
Precondition: The system has the necessary software capabilities to provide a user-friendly interface for accessing and modifying settings.
Cross-references: 1, 7

7. Biometric

Functional Requirements (FRs)

7.1 The system shall provide a user-friendly interface for initiating biometric authentication.
7.2 The system shall clearly indicate the required steps for completing biometric authentication.
7.3 The system shall collect and store biometric data, such as fingerprints or facial recognition data, from the user and ensure that the biometric data is stored securely.
7.4 The system shall authenticate the user's biometric data before allowing access to the system or sensitive data.
7.5 The system shall allow the user to retry (2 times) biometric authentication if the first attempt is unsuccessful.
7.6 The software shall prompt the user to enter a password after multiple unsuccessful biometric attempts.
7.7 The system shall prevent multiple failed biometric attempts and initiate security measures, such as temporarily disabling the account or notifying the user if there are suspicious activities.

Priority Level: High
Precondition: The system has the necessary hardware and software capabilities to collect and authenticate biometric data, such as a fingerprint scanner or facial recognition technology. The user should have previously registered their biometric data with the system and have access to the device or hardware required for biometric authentication.
Cross-references: N/A

8. Mobile Notification Bar Disabling

Functional Requirements (FRs)

8.1 The application shall convey a function to enable or disable the notification bar.
8.2 The system shall ensure that disabling the notification bar does not affect the performance or functionality of other system features.
8.3 The system shall ensure that enabling the notification bar does not affect the performance or functionality of other system features.
8.4 The system shall display a notification to the user when the notification bar is disabled or enabled.
8.5 The system shall prevent unauthorized access to the notification bar disabling feature by implementing permission-based access control.

Priority Level: High
Precondition: The system has the necessary software capabilities to provide a feature that allows the user to disable or enable the notification bar. Additionally, the system should ensure that disabling or enabling the notification bar does not affect the performance or functionality of other system features.
Cross-references: N/A

9. Mobile Volume & Power Button Disabling

Functional Requirements (FRs)

9.1 The software shall offer an option that disables the stolen mobile device's volume and power buttons.
9.2 The system shall ensure that disabling the volume and power buttons does not affect the performance or functionality of other system features.
9.3 The software shall let the user enable the stolen mobile device's volume and power buttons at any time.
9.4 The system shall display to the user whether the volume and power buttons are disabled or enabled.

Priority Level: High
Precondition: The system has the necessary software capabilities to provide a feature that allows the user to disable or enable the volume and power buttons of the mobile device. Security measures should be implemented to prevent unauthorized access to the volume and power button disabling feature, and permission-based access control should be in place.
Cross-references: 8

10. Emergency Data Transfer & Enable Mobile Data

Functional Requirements (FRs)

10.1 The system shall provide a feature to transfer emergency mobile data to a designated emergency contact if needed.
10.2 The software shall activate or disable mobile data at any moment for the user.
10.3 The system shall notify the user when emergency data is being transferred.
10.4 The system shall display a notification to the user when mobile data is enabled or disabled.

Priority Level: High
Precondition: The system has the necessary software capabilities to provide a feature that allows the user to transfer emergency data to a designated emergency contact and enable or disable mobile data. For doing this the system must have an agreement, that is signed with the SIM companies to provide a data pack when this operation is performed.
Cross-references: N/A

11. Device Tracking

Functional Requirements (FRs)

11.1 The program shall enable users to monitor the GPS position of their Android devices.
11.2 The location of the device shall be displayed on a map within the application.
11.3 The user shall have the ability to remotely track their device using a web-based interface.
11.4 The application shall periodically update the location of the device and display the current location on the map.
11.5 The location of the user's device shall be seen on the map.

Priority Level: High
Precondition: The user's Android device grants the location access permissions.
Cross-references: 8

12. Remote Locking

Functional Requirements (FRs)

12.1 In accordance with the system's permissions, the user shall remotely lock their Android device in the event of loss or theft.
12.2 The device shall be locked with a secure PIN or password that is set by the user.
12.3 A message shall be displayed in the system’s lock screen indicating that the device has been locked and instructions for unlocking.
12.4 The user shall be able to remotely unlock their device using the application or web-based interface.
12.5 The application shall provide a backup PIN or password that can be used to unlock the device if the user forgets their original PIN or password.

Priority Level: High
Precondition: The user has a successful login.

Cross-references: N/A

13. Remote Wiping

Functional Requirements (FRs)

13.1 If an Android device is lost or stolen, the app shall enable remote device wiping for the user.
13.2 The device shall be wiped clean of all data and personal information after biometric confirmation.
13.3 The application shall provide a confirmation prompt before wiping the device.
13.4 The wiping process using the application or web-based interface shall be initiated remotely by the user.
13.5 A backup option shall be provided by the application to restore data on the device if it is found or recovered.

Priority Level: High
Precondition: The user has set up a backup option to restore data.
Cross-references: 7

14. Alarm Triggering

Functional Requirements (FRs)

14.1 The application shall provide an option for the user to remotely trigger an alarm on their Android device if it is lost or stolen.
14.2 The alarm shall be loud and audible even if the device is in silent mode.
14.3 User shall be able to turn off the alarm remotely using application or web-based interface.
14.4 The application shall send a notification to the user when the alarm is triggered or turned off.

Priority Level: Medium
Precondition: The user grants the necessary permissions for triggering the alarm.
Cross-references: N/A

15. Tamper Detection

Functional Requirements (FRs)

15.1 The application shall notify the user if someone tries to tamper with their Android device.
15.2 The application shall capture a photo using the device's camera and send it to the user's email address when tampering is detected.
15.3 The user shall be able to remotely trigger a photo capture using the application or web-based interface.
15.4 The system shall include a backup option for cloud-based tamper-detection data storage.

Priority Level: Medium

Precondition: The user grants the necessary permissions for capturing photos and sending emails.
Cross-references: N/A

16. USB Port Disabling

Functional Requirements (FRs)

16.1 The software shall have the ability to disable the USB port of the device.
16.2 The user shall be able to enable the USB port after disabling it.
16.3 Disabling the USB port shall be possible only after the user has enabled the Anti-Theft mode.
16.4 When the USB port is disabled, the device shall not be able to charge or transfer data via the USB port.
16.5 The software shall display a message to the user when the USB port is disabled or enabled.
16.6 The software shall log all actions related to disabling/enabling the USB port.
16.7 If the device is connected to a power source when the USB port is disabled, the software shall prevent the device from charging through any other port.
16.8 The software shall have the ability to re-enable the USB port automatically after a certain time period (configurable by the user).

Priority Level: High
Preconditions: The user has enabled the Anti-Theft mode in the software and the necessary permissions to disable/enable the USB port also the device should support the functionality of disabling/enabling the USB port.
Cross-reference: 7

17. Device Blocking for a certain period via Different-Sign Face Expression

Functional Requirements (FRs)

17.1 The software shall have a facial recognition system to recognize the user's facial expression.
17.2 The user should be able to set up their sign facial expression as the key for blocking the device.
17.3 If the user's facial expression is different-singed than the default, the device shall be blocked for a certain period.
17.4 The user should have the option to switch between different-sign facial expressions and the default facial expression as the key for blocking and unblocking the device.
17.5 The software shall log all actions related to device blocking and unblocking.

Priority Level: High
Preconditions: The device has given the necessary permissions to the software to access the device's camera and facial recognition system and has a front-facing camera with a minimum resolution and quality that is suitable for facial recognition and has set up a different-sign facial expression as the key for blocking and unblocking device. The user's different-sign facial expression can be reliably recognized by the software's facial recognition system under normal lighting conditions and specified duration of the device blocking.

Cross-reference: 1.3

18. System Support – Live Chat

Functional Requirements (FRs)

18.1 The system shall provide a chat interface that allows users to communicate with support personnel in real time.
18.2 The chat interface shall support sending and receiving messages, multimedia files, and attachments.
18.3 The system shall provide notifications to alert users when support personnel is available to chat.
18.4 The system shall authenticate the support personnel before allowing them to communicate with the user.
18.5 The system shall log all chat conversations for auditing and quality assurance purposes.

Priority Level: Medium
Preconditions: The availability of a system administrator or trained support staff to provide live chat support, integration of the live chat system with the overall system architecture, and the availability of necessary hardware and software resources to support the live chat feature.
Cross-reference: N/A

19. Search

Functional Requirements (FRs)

19.1 The system shall allow users to enter keywords or phrases to search for specific information or content within the system.
19.2 The system shall provide users with search suggestions or auto-complete options to assist them in formulating search queries and finding the desired content quickly.

Priority Level: Medium
Preconditions: The availability of a database or index containing the information to be searched. The database or index should be accessible by the search function through appropriate interfaces. The software should also have permission to access the database or index. Additionally, the search function should be designed to handle different types of input, including text and numerical values, and should provide appropriate feedback to the user in case of invalid inputs or errors.
Cross-reference: N/A

3.2 Non-Functional/Quality Requirements

QA1: Performance:

- Within two seconds of receiving user input, the application shall respond.
- Within one minute, the application shall be able to find the device.
- The application shall not cause a significant impact on the device's battery life.
- The application shall not consume excessive data usage.

Priority Level: High
Preconditions: The device should have sufficient hardware capabilities to meet the performance requirements.
Cross-reference: N/A

QA2: Usability:

- The application shall have a user-friendly interface with clear navigation and instructions.
- The application shall provide notifications and alerts in a clear and concise manner.
- The application shall be accessible to users with disabilities.

Priority Level: High
Preconditions: To achieve the usability requirements, the application must be designed and developed with the consideration of the target user group, their needs, and limitations. The application should be tested with actual users to ensure that it is easy to use, and instructions and notifications are clear and concise. The application must also comply with the relevant accessibility guidelines to ensure that it can be used by users with disabilities. The application developers and designers should work closely with the stakeholders to identify the usability requirements and continuously evaluate the usability of the application throughout the development process.
Cross-reference: QA1

QA3: Security:

- The application shall encrypt user data stored on the device and during transmission.
- The application shall have a secure login and registration process.
- The application shall prevent unauthorized access to the device by blocking all ports and interfaces.
- The application shall have a specific remote wipe feature to erase all user data on the device.

Priority Level: High
Preconditions: To achieve the security requirements, the application must be designed and developed with the consideration of the potential threats and vulnerabilities. To safeguard user data from unauthorised access, theft, and abuse, the application must adhere to the necessary security standards and guidelines. The application should be tested for vulnerabilities regularly, and any identified vulnerabilities should be addressed promptly. The application developers and designers should work closely with the

stakeholders to identify the security requirements and continuously evaluate the security of the application throughout the development process.
Cross-reference: QA2

QA4: Compatibility:

- The application shall be compatible with Android OS versions 5.0 and above.
- The application shall be compatible with a wide range of device models.
- The application shall be compatible with a variety of network types and carriers.

Priority Level: High
Preconditions: To achieve the compatibility requirements, the application must be developed using a flexible and modular architecture that can support multiple device models and network types. The application developers should conduct rigorous testing on various device models, Android versions, and network types to ensure compatibility. The developers should also be aware of the latest Android features and guidelines to ensure that the application is up-to-date with the latest compatibility standards. The application should be designed to minimize compatibility issues that may arise due to differences in device hardware and software configurations.
Cross-reference: N/A

3.3 Project Requirements

- **Development Platform:** The application should be developed on Android Studio with Java/Kotlin programming language.
- **Hardware and software requirements:** The development and testing environment should have appropriate hardware and software requirements, such as a compatible operating system, CPU, RAM, and storage space.
- **Development timeline:** A project timeline with milestones, delivery dates, and review periods should be established.
- **User acceptance testing:** To make sure that the application satisfies user expectations and needs, user acceptability testing should be carried out.
- **Documentation:** Documentation such as Software Requirement Specifications (SRS), System Design Documents (SDD), User Manuals, and Technical Manuals should be prepared.
- **Compliance:** Regulations governing data protection, for example, should be followed by the application.
- **Maintenance & support:** Maintenance and support of the application should be provided to ensure its functionality and usability over time.
- **Budget:** A budget for development, testing, and maintenance of the application should be established.
- **Resources:** Appropriate resources such as developers, testers, project managers, and infrastructure should be allocated for the development and maintenance of the application.

4. Design and Interface Requirements

4.1 UML Diagrams

- **ER Diagram**

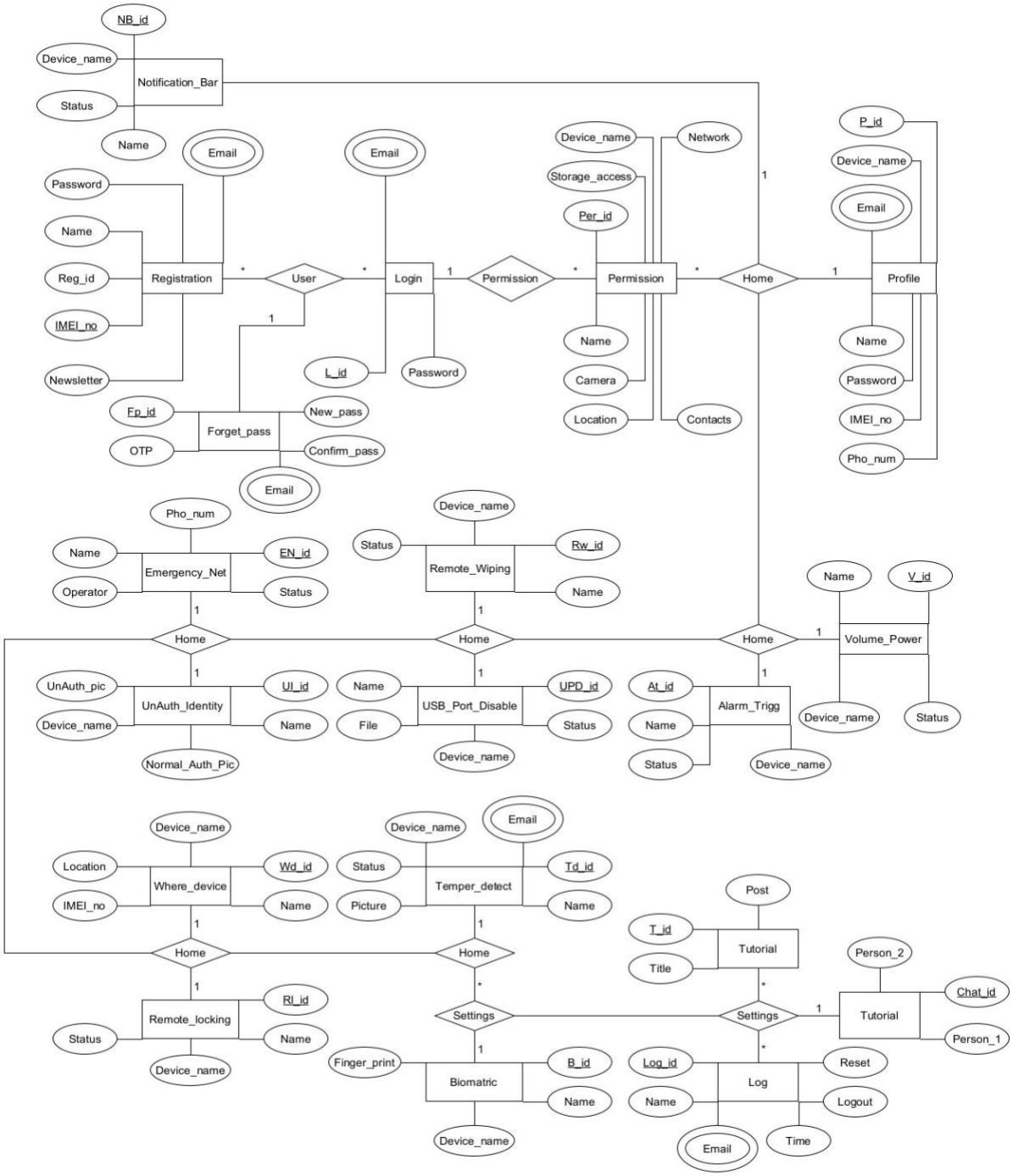

Figure 1: MOBDOG - ER Diagram

- **Use Case Diagram**

The user can see the login interface after installing and launching the app. If the user has an account, they can log in directly with their email, password, or fingerprint (if the device supports fingerprints). New users must create a new account with their name, IMEI, email, and password. If the IMEI is unknown, click "check IMEI" to find the method. If the registered user forgets the password, an OTP will be sent to the registered email after providing the email by clicking on "Forget Password." A new password can be reset using the correct OTP. After logging in correctly, the new user needs to provide some necessary permissions to run the application properly, and those who have already given permission can see the features directly. The application comes with nine new features. We have included two sections: a search tab, a profile page, and settings.

The main features are notification bar deactivation, volume +/- and power button disable, remote data wiping, alarm trigger, USB port disable, remote locking, unauthorised identity, where my device is, and emergency network. Users can protect their privacy by disabling notifications on the notification bar whenever their device is locked. When our phone is locked, it still allows anyone to silence or turn off the device using the power or volume buttons. This feature allows users to disable the volume +/- and power buttons. If it comes to a situation where the device is lost, and it is no longer possible to recover it, then the user can delete all the information on his device by turning on this feature. Using the alarm-triggering feature, the user is notified of their device by playing a loud sound whenever it is nearby. The device's charging port allows users to charge or transfer files to their computers or other devices. Sometimes it can be used to gain illegal access to the phone's data by brute force or bypassing the password. So, the feature disabling the USB port is provided so the user can disable the port during an emergency. To experience the feature's full potential, users must download an additional file on their phones. If the user forgets to lock their device, or if it is lost or stolen, and anyone with physical access to it can easily unlock it, the user will be able to lock his phone using the remote locking feature. In certain situations where a user has been physically harmed, the proprietor has access to their face, and biometrics is bound to unlock their phone for the proprietor. Therefore, the user can make a unique face on the phone's camera, such as "winking his left eye three times," which will tell the phone that the user is in danger, and the phone will lock itself for an amount of time or unlock using only a series of unique vital numbers. The feature "Where is my device?" can be used to determine the current status of the lost device. If the user is out of mobile data (internet), the emergency network feature of the app will provide them with access to the internet or phone calls for a limited time. The search tab or bar can be used to search for a feature on the app. It will be the time needed to use the app. The "View Profile" tab allows the user to edit, change, and save the account information he uses.

Settings have seven sections: permission, biometric, tutorial, support, reset, logout, and uninstall. Permission: If you want to change any permission, you can do it now. Biometric: new users can add fingerprints here as per their wish. Tutorial: videos and documents needed to operate the app will be presented. Support: if the users face problems managing the app, they can directly communicate here through live chat. Reset: This option is given to restart the app. Log out: the account will be signed and taken to the login interface. Uninstall: since this is an admin app, use this option to uninstall it. During an emergency, the user can visit mobdog.com/login in the browser and log in to use several features he can access through the app. Those are emergency

alarm on, USB port disabling, device data wipe, device location, remote lock, and camera movement. Apart from the camera movement feature, other features were described earlier. Camera movement can capture scenes by turning on the front and rear cameras. This was the complete scenario for the use case.

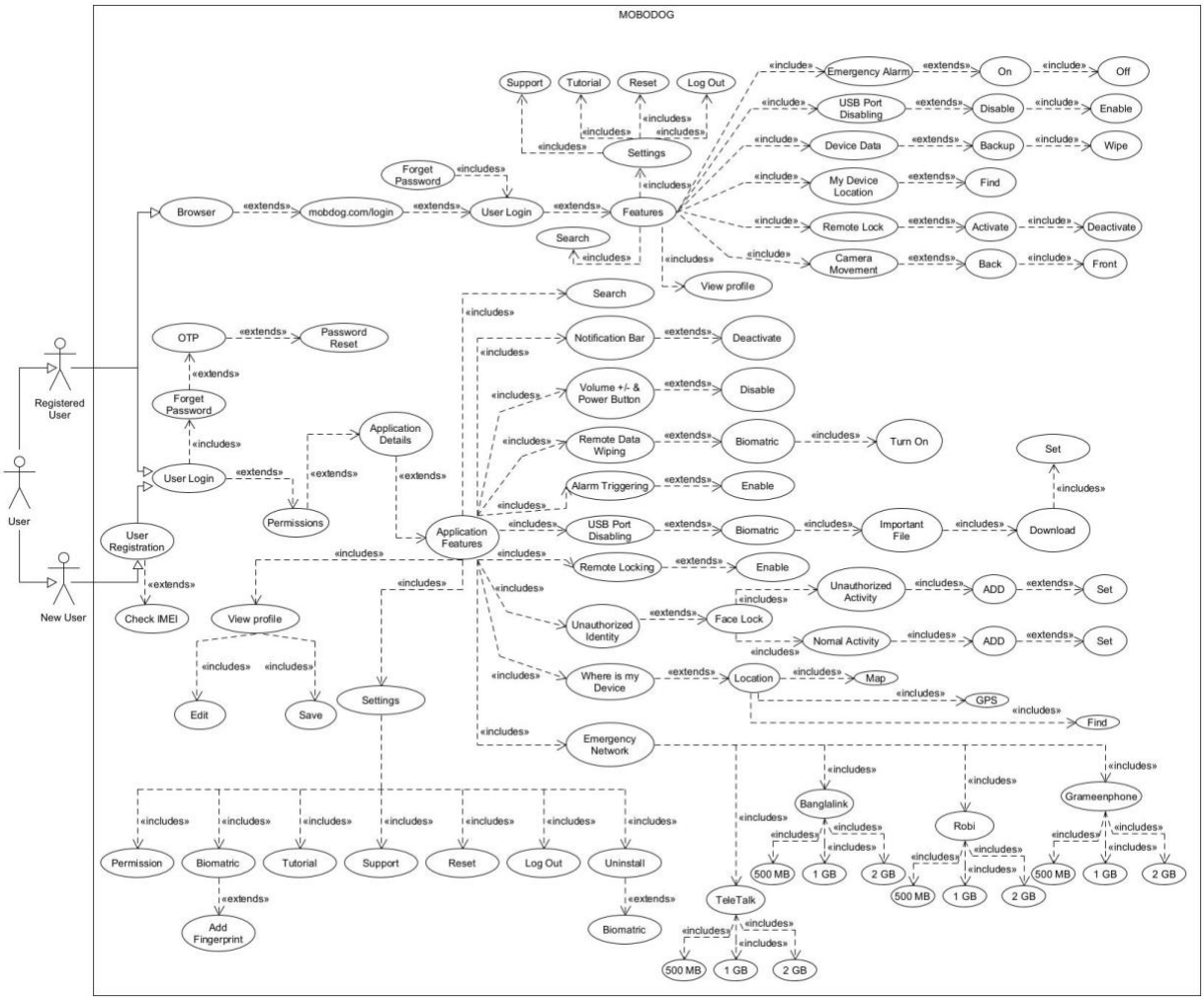

Figure 2: MOBDOG - Use Case

- **Activity Diagram**

Depending on the situation, the user can either open the app directly or use the browser. To use the app, a login process must be followed. New users need to register and re-login after a failed or unsuccessful login. Registered users can directly view the features or search through the search option. Users can access nine different features, including changing profiles and settings. Users can use the features as per their choice and needs. The features must be deactivated beforehand if the user wants to stop using the Notification Bar, Volume +/-, and Power Button during an emergency. Alarm Triggering, Remote Wiping, USB Port Disabling, and Emergency Network should be kept on. Identification should be added to detect unauthorised identities. Users can reset, log out, and uninstall apps with support and a tutorial through settings. Permission and new fingerprints must be given to change or add biometric identification.

The user can log in from the browser and use the previously enabled features during emergencies—emergency alarm on, USB port disabled (file transfer), device data wipe, and remote lock. The user is capable of controlling functional features from a distance. Device location and camera movements can be perceived directly. Users can view profiles and control Support, Tutorial, Reset, and Log Out using Settings.

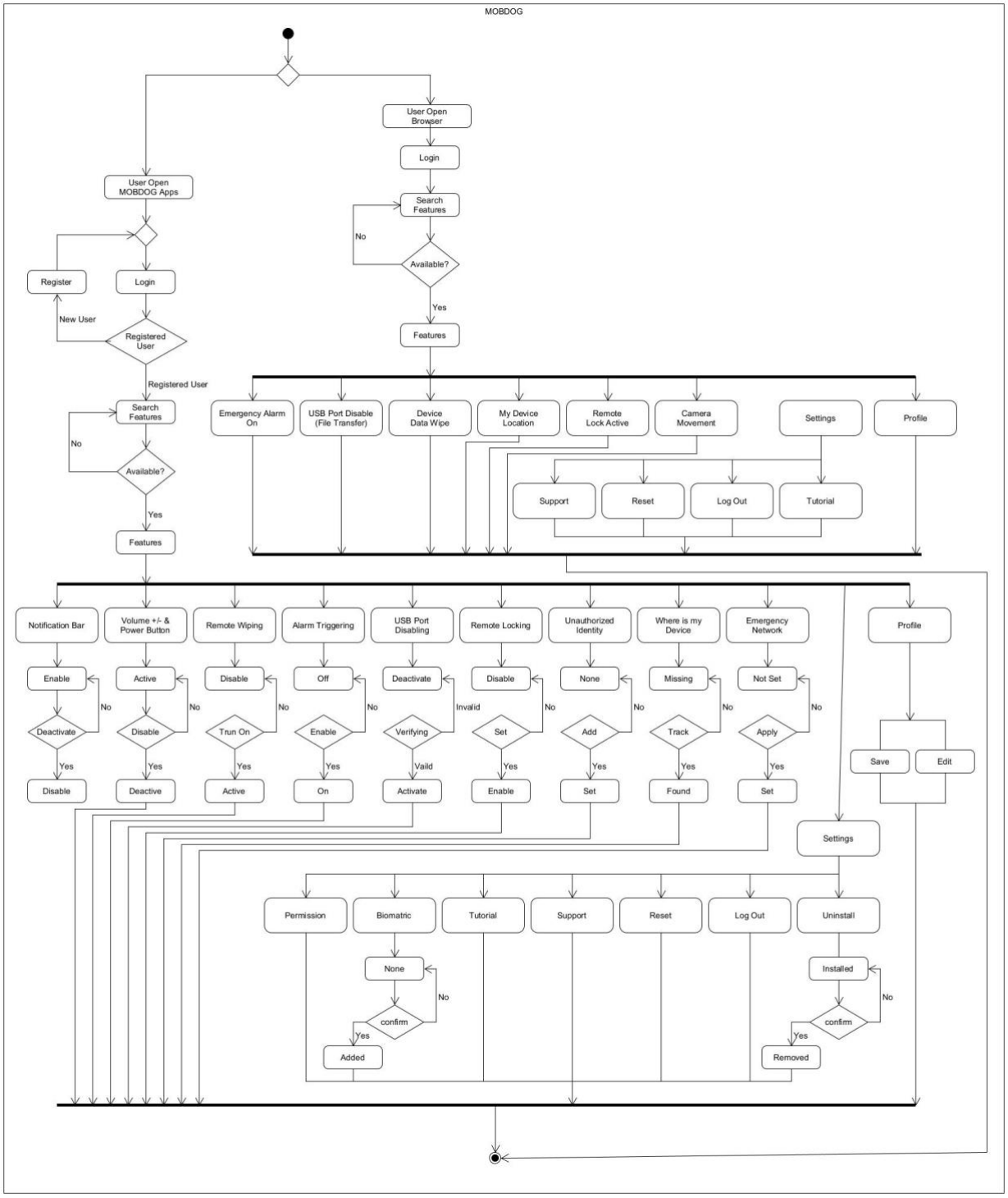

Figure 3: MOBDOG - Activity

4.2 Data Dictionary

Entity	Attribute	Type/Size	Validation	Key
Registration	Reg_id	Number (5)	10000-99999	
Registration	Name	Text (15)	Required	
Registration	IMEI_no	Number (20)	1-20	Primary
Registration	Email	Text (15)	Required	
Registration	Password	Text (20)	Required	
Registration	Newsletter	Text (10)	Required	

Entity	Attribute	Type/Size	Validation	Key
Login	L_id	Number (5)	10000-99999	Primary
Login	Email	Text (15)	Required	
Login	Password	Text (20)	Required	

Entity	Attribute	Type/Size	Validation	Key
Forget_password	Fp_id	Number (5)	10000-99999	Primary
Forget_Password	Email	Text (15)		Foreign - Login
Forget_Password	OTP	Number (6)	Required	
Forget_Password	New_Pass	Text (20)	Required	
Forget_Password	Confirm_Pass	Text (20)	Required	

Entity	Attribute	Type/Size	Validation	Key
Profile	P_id	Number (5)	10000-99999	Primary
Profile	Name	Text (15)	Required	
Profile	Device_name	Text (15)	Required	
Profile	Email	Text (15)	Required	
Profile	Password	Text (20)	Required	
Profile	IMEI_no	Number (20)	1-20	
Profile	Phone_num	Number (20)	1-15	

Entity	Attribute	Type/Size	Validation	Key
Notification_Bar	NB_id	Number (5)	10000-99999	Primary
Notification_Bar	Name	Text (15)	Required	
Notification_Bar	Device_name	Text (15)	Required	
Notification_Bar	Status	Text (15)	Required	

Entity	Attribute	Type/Size	Validation	Key
Volume_Power	V_id	Number (5)	10000-99999	Primary
Volume_Power	Name	Text (15)	Required	
Volume_Power	Device_name	Text (15)	Required	
Volume_Power	Status	Text (15)	Required	

Entity	Attribute	Type/Size	Validation	Key
Remote_Wiping	Rw_id	Number (5)	10000-99999	Primary
Remote_Wiping	Name	Text (15)	Required	
Remote_Wiping	Device_name	Text (15)	Required	
Remote_Wiping	Status	Text (15)	Required	

Entity	Attribute	Type/Size	Validation	Key
Alarm_Trigg	At_id	Number (5)	10000-99999	Primary
Alarm_Trigg	Device_name	Text (15)	Required	
Alarm_Trigg	Status	Text (15)	Required	

Entity	Attribute	Type/Size	Validation	Key
USB_Port_Disable	UPD_id	Number (5)	10000-99999	Primary
USB_Port_Disable	Name	Text (15)	Required	
USB_Port_Disable	file	Text (50)	Required	
USB_Port_Disable	Device_name	Text (15)	Required	
USB_Port_Disable	Status	Text (15)	Required	

Entity	Attribute	Type/Size	Validation	Key
Emergency_Net	EN_id	Number (5)	10000-99999	Primary
Emergency_Net	Name	Text (15)	Required	
Emergency_Net	Pho_num	Text (15)	Required	
Emergency_Net	Device_name	Text (15)	Required	
Emergency_Net	Status	Text (15)	Required	
Emergency_Net	Operator	Text (15)	Required	

Entity	Attribute	Type/Size	Validation	Key
UnAuth_Identity	UI_id	Number (5)	10000-99999	Primary
UnAuth_Identity	Name	Text (15)	Required	
UnAuth_Identity	Device_name	Text (15)	Required	
UnAuth_Identity	Normal_Auth_Pic	blob	Required	
UnAuth_Identity	UnAuth_pic	blob	Required	

Entity	Attribute	Type/Size	Validation	Key
Where_device	Wd_id	Number (5)	10000-99999	Primary
Where_device	Name	Text (15)	Required	
Where_device	IMEI_no	Text (20)	1-20	
Where_device	Device_name	Text (15)	Required	
Where_device	Location	Text (15)	Required	

Entity	Attribute	Type/Size	Validation	Key
Remote_locking	Rl_id	Number (5)	10000-99999	Primary
Remote_locking	Name	Text (15)	Required	
Remote_locking	Device_name	Text (15)	Required	
Remote_locking	Status	Text (15)	Required	

Entity	Attribute	Type/Size	Validation	Key
Temper_detect	Td_id	Number (5)	10000-99999	Primary
Temper_detect	Name	Text (15)	Required	
Temper_detect	Email	Text (15)	Required	
Temper_detect	Device_name	Text (15)	Required	
Temper_detect	Status	Text (15)	Required	
Temper_detect	Pic	blob	Required	

Entity	Attribute	Type/Size	Validation	Key
Permissions	Per_id	Number (5)	10000-99999	Primary
Permissions	Name	Text (15)	Required	
Permissions	Device_name	Text (15)	Required	
Permissions	Location	Text (15)	Required	
Permissions	network	Text (15)	Required	
Permissions	camera	Text (15)	Required	
Permissions	contacts	Text (15)	Required	
Permissions	Storage_access	Text (15)	Required	

Entity	Attribute	Type/Size	Validation	Key
Biomatric	B_id	Number (5)	10000-99999	Primary
Biomatric	Name	Text (15)	Required	
Biomatric	Device_name	Text (15)	Required	
Biomatric	Finger_print	Text (15)	Required	

Entity	Attribute	Type/Size	Validation	Key
Tutorial	T_id	Number (5)	10000-99999	Primary
Tutorial	Title	Text (50)	Required	
Tutorial	Post	Text (5000)	Required	

Entity	Attribute	Type/Size	Validation	Key
Log	Log_id	Number (5)	10000-99999	Primary
Log	Name	Text (15)	Required	
Log	Email	Text (15)	Required	
Log	Time	Date (8)	Valid Date	
Log	reset	Text (15)	Required	
Log	Logout	Text (15)	Required	

Entity	Attribute	Type/Size	Validation	Key
Live_chat	Chat_id	Number (5)	10000-99999	Primary
Live_chat	Person_1	Text (200)	Required	
Live_chat	Person_2	Text (200)	Required	

4.3 UI/UX Design Specification

Figure 4: Phone - Lock Screen (User Hand)

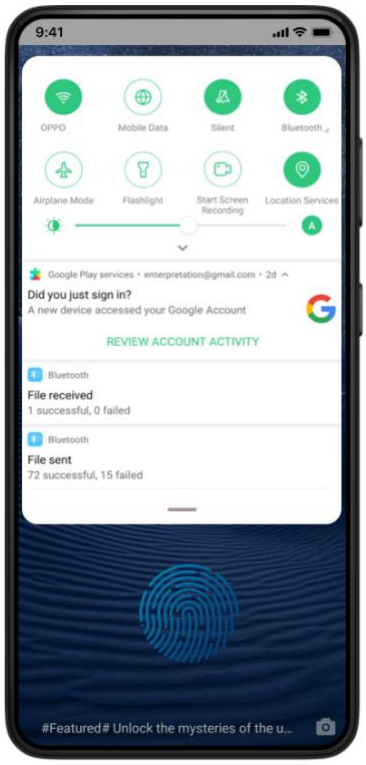

Figure 5: Phone - Enabled Notification Bar

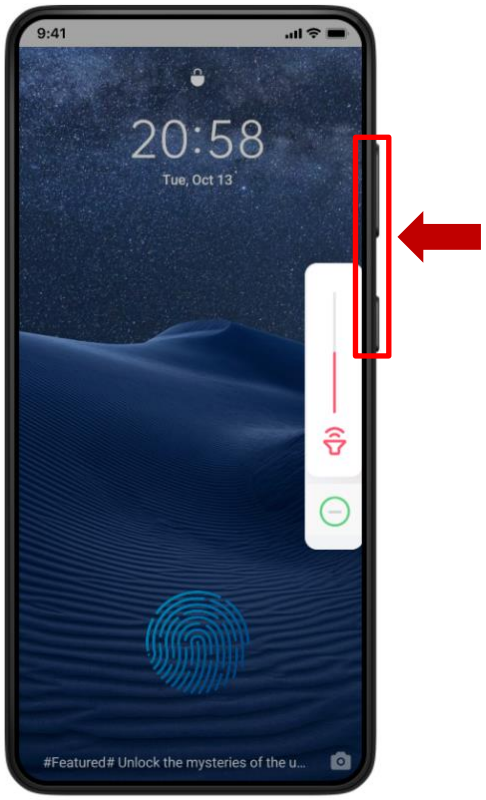

Figure 6: Phone - Enabled Volume and Power

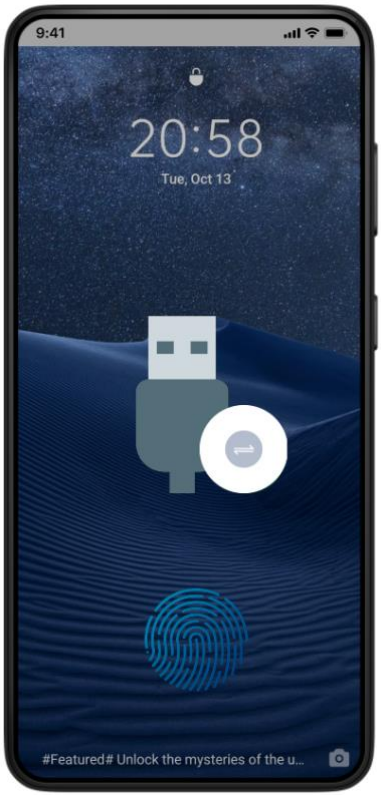

Figure 7: Phone - Charging Enabled

Figure 8: Phone – Open MOBDOG Apps

Figure 9: MOBDOG – Application

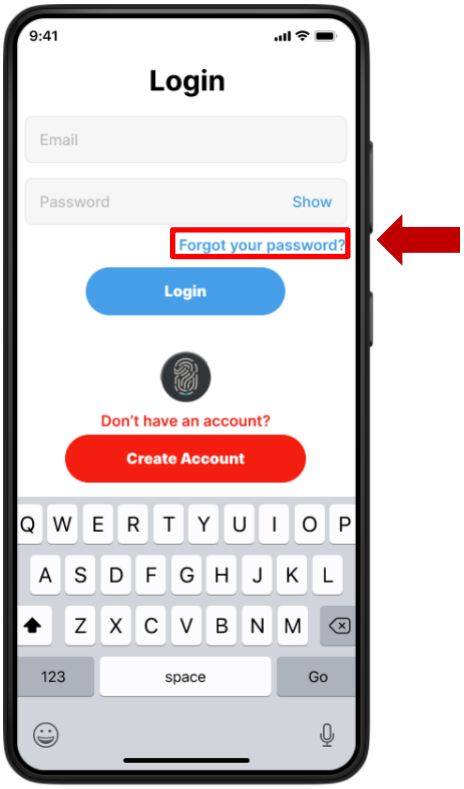

Figure 10: MOBDOG – Forget Password

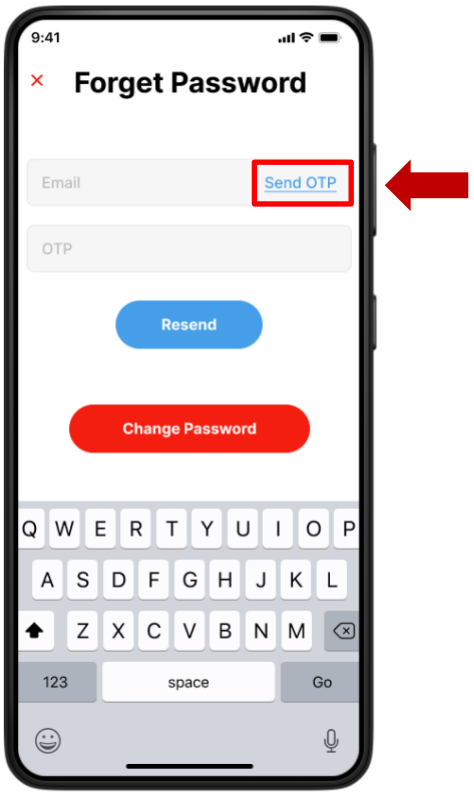

Figure 11: MOBDOG – Send OTP

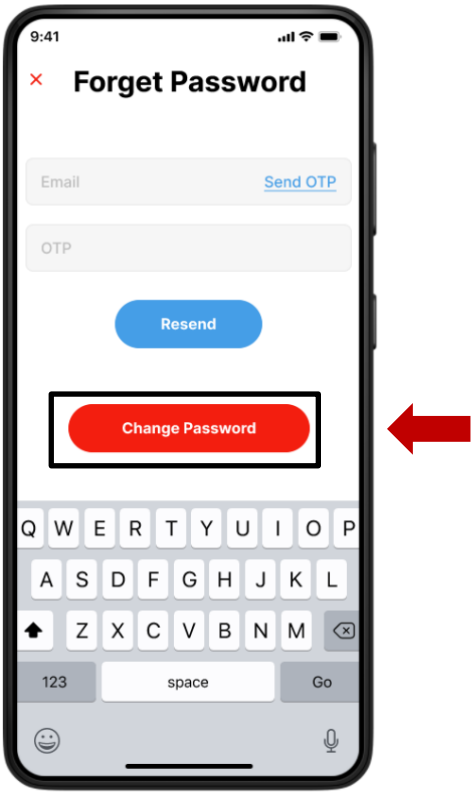

Figure 12: MOBDOG - Change Password

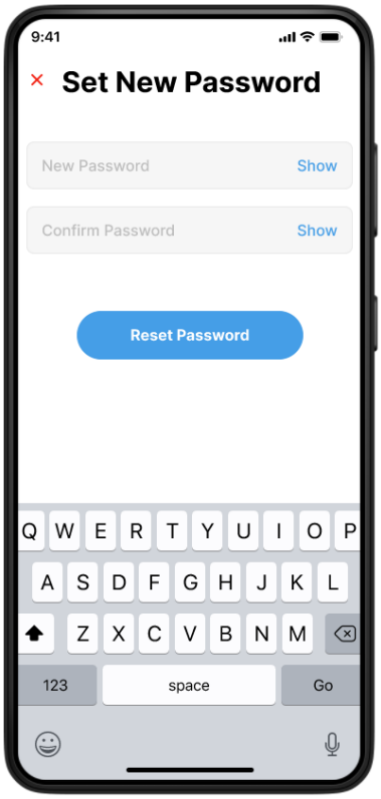

Figure 13: MOBDOG - Set New Password

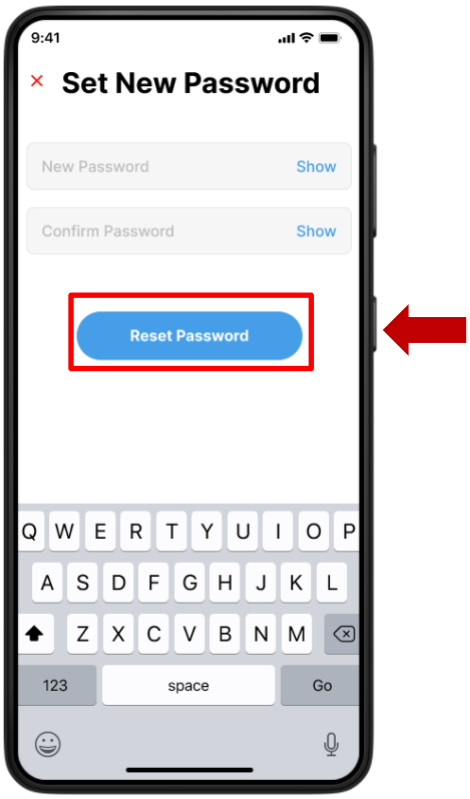

Figure 14: MOBDOG – Reset Password

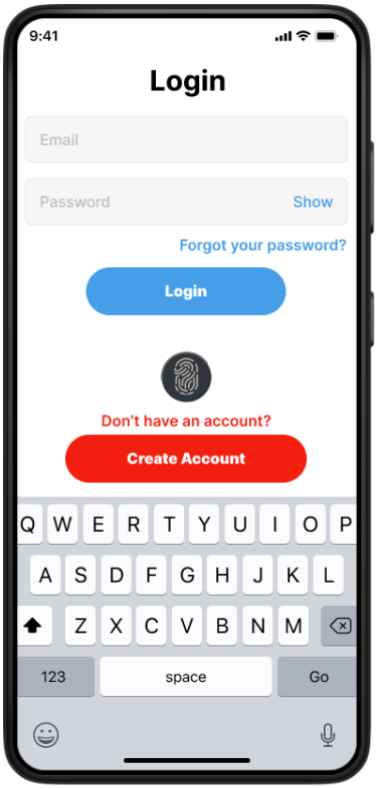

Figure 15: MOBDOG - Login

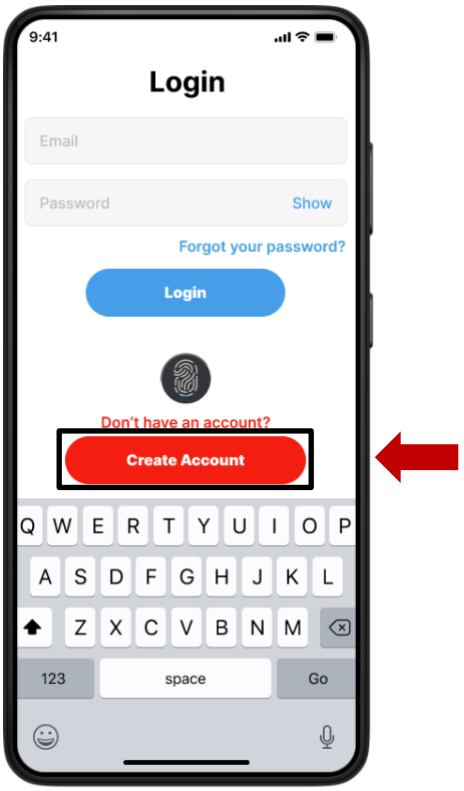

Figure 16: MOBDOG – Create Account

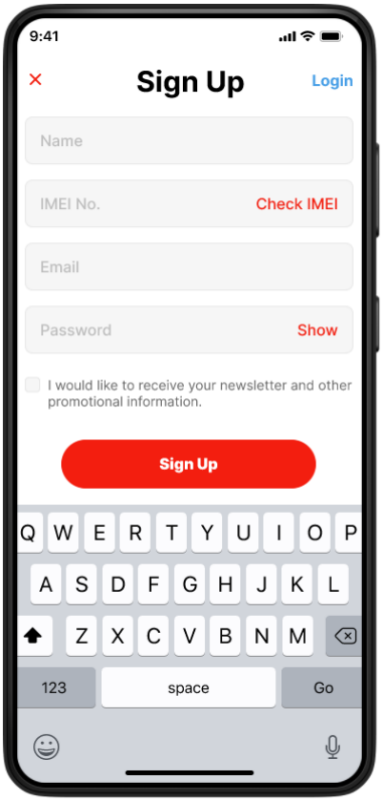

Figure 17: MOBDOG - Sign Up

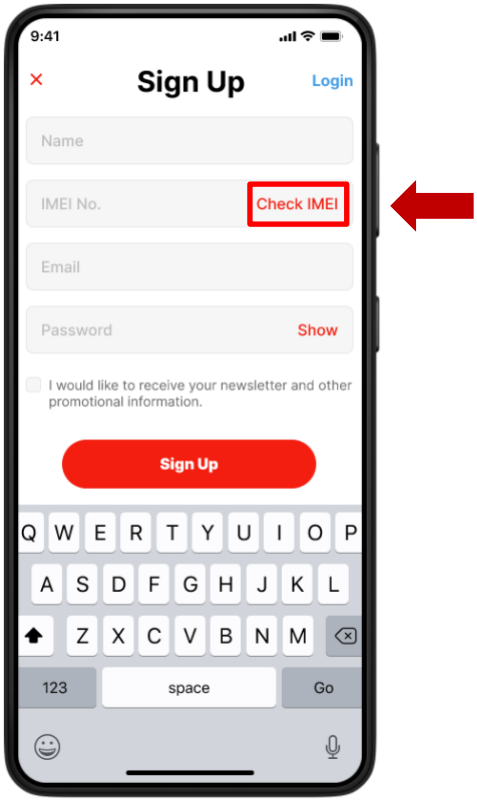

Figure 18: MOBDOG – Check IMEI

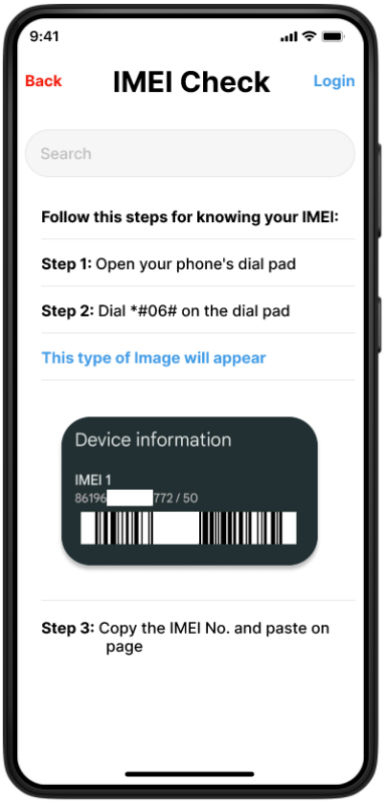

Figure 19: MOBDOG - IMEI Check

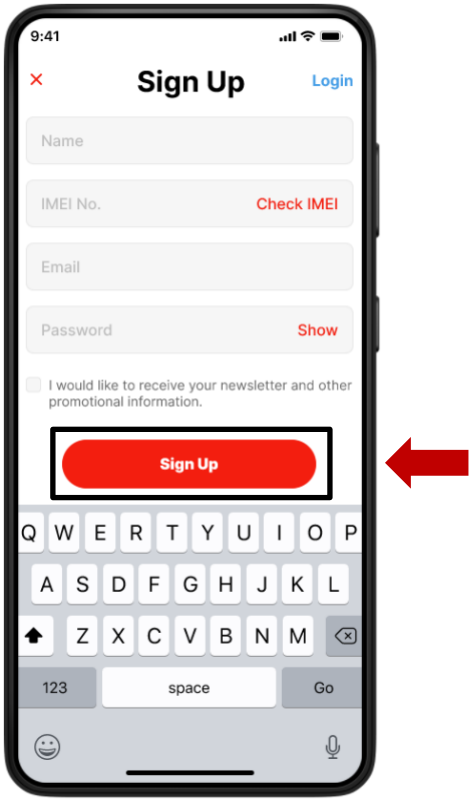

Figure 20: MOBDOG - Sign UP

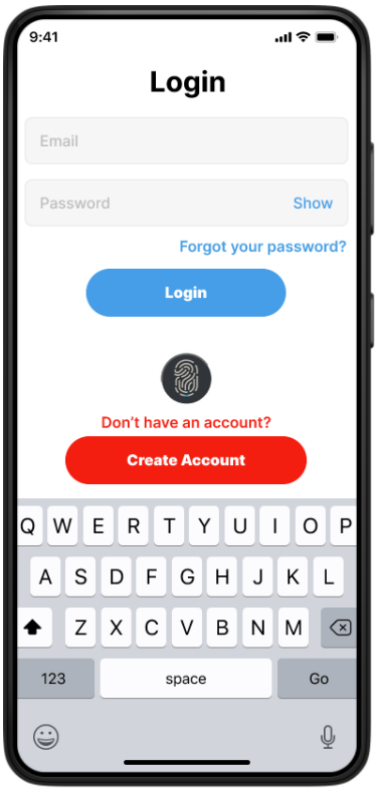

Figure 21: MOBDOG - Login

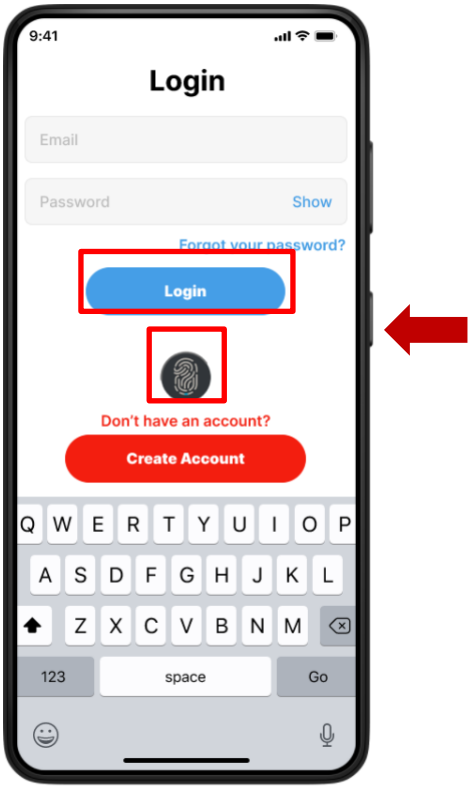

Figure 22: MOBDOG - Login

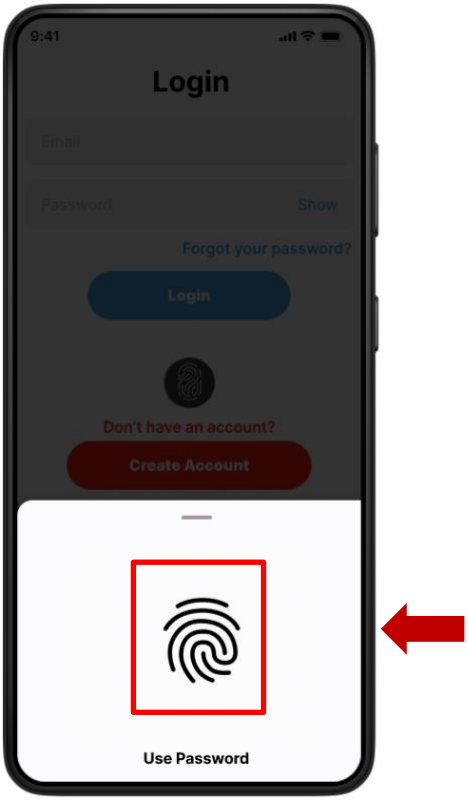

Figure 23: MOBDOG - Biometric Login

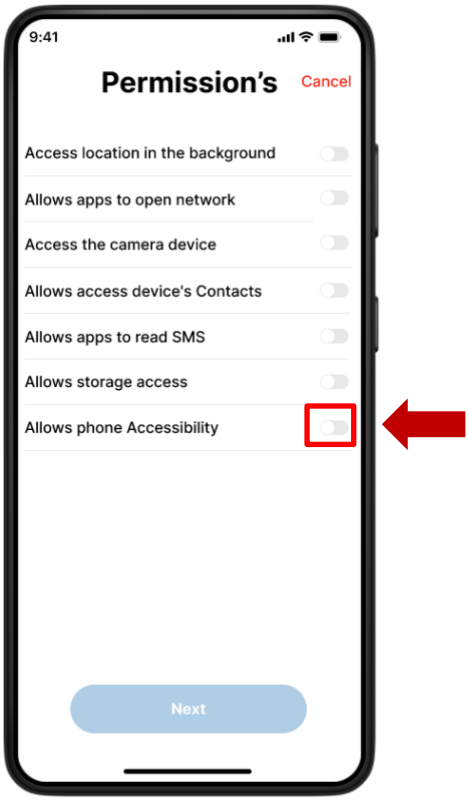

Figure 24: MOBDOG – Permissions

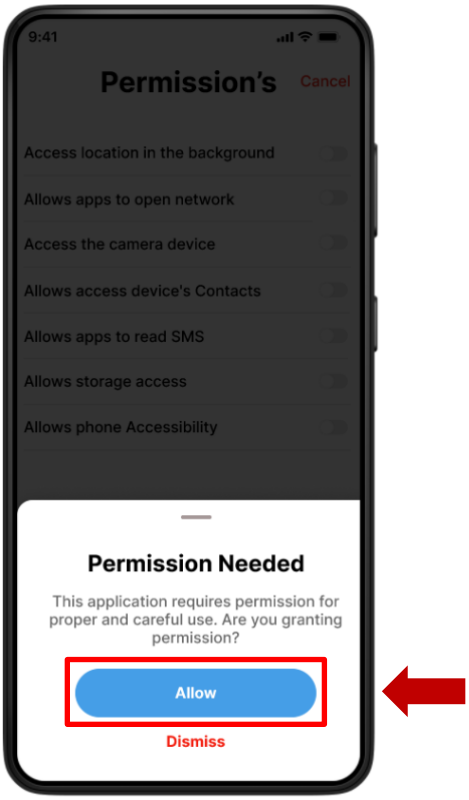

Figure 25: MOBDOG - Individual Permission Allowing

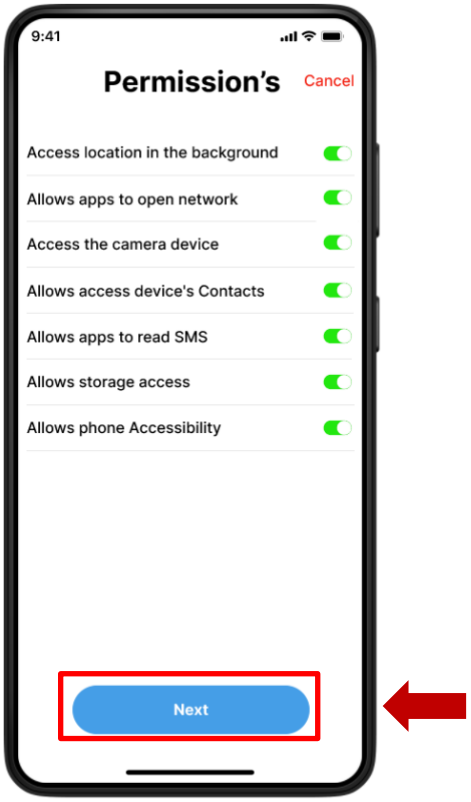

Figure 26: MOBDOG - Next Button Enabled

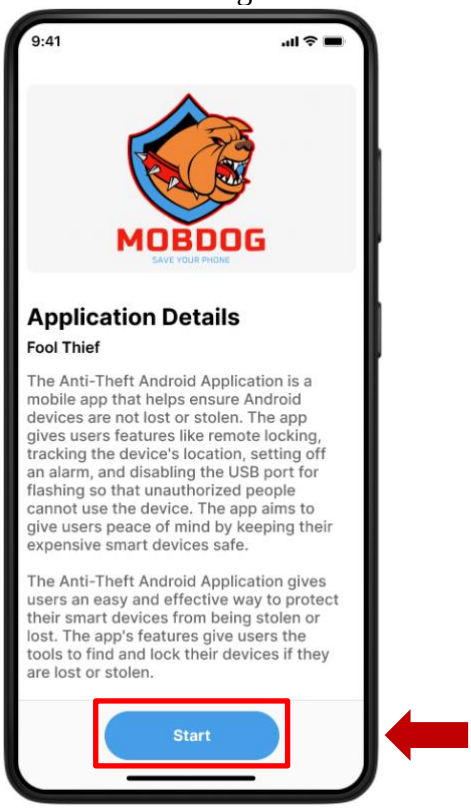

Figure 27: MOBDOG - Application Details

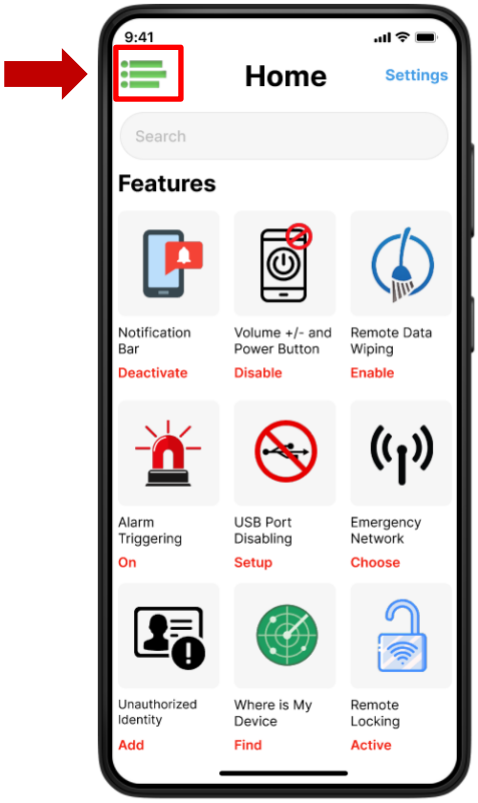

Figure 28: MOBDOG - Application Home

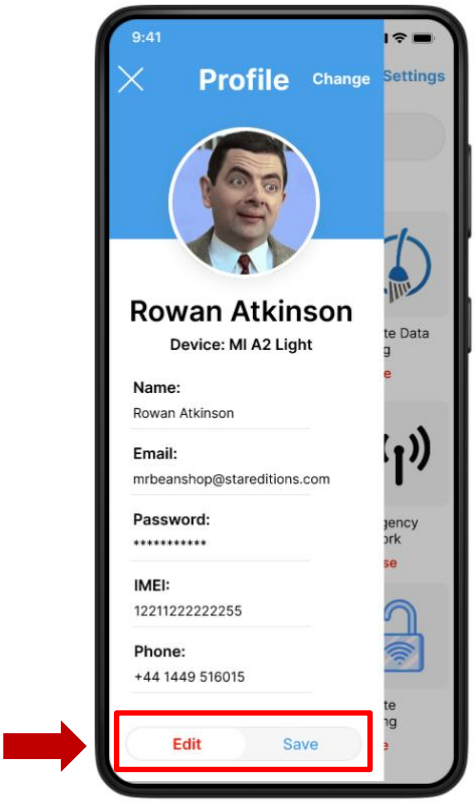

Figure 29: MOBDOG – User Profile

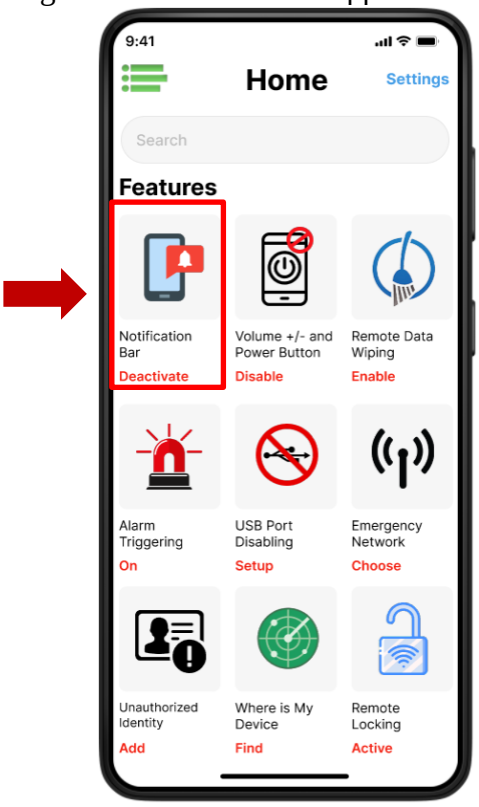

Figure 30: MOBDOG - Notification Bar

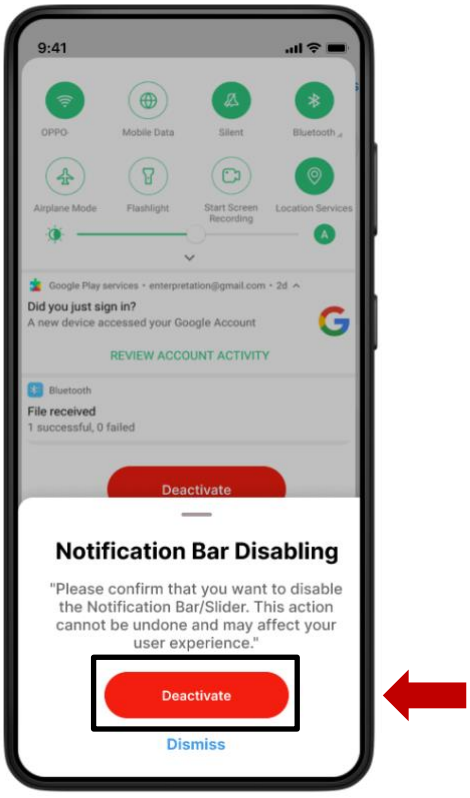

Figure 31: MOBDOG – Notification Bar Deactivation

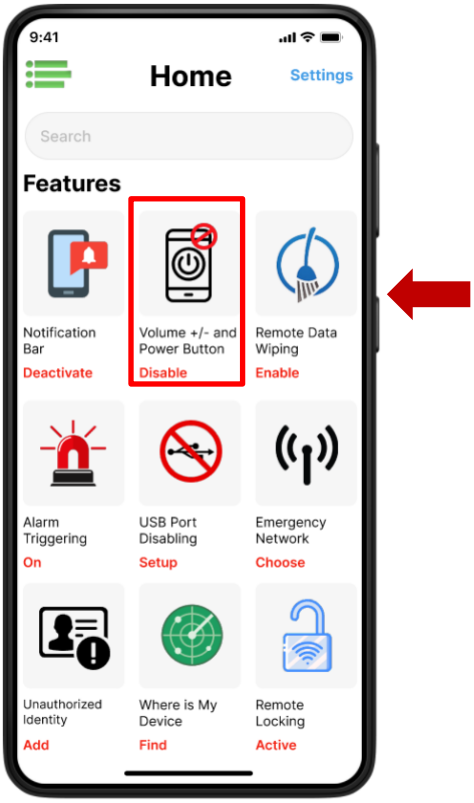

Figure 32: MOBDOG – Volume & Power Button

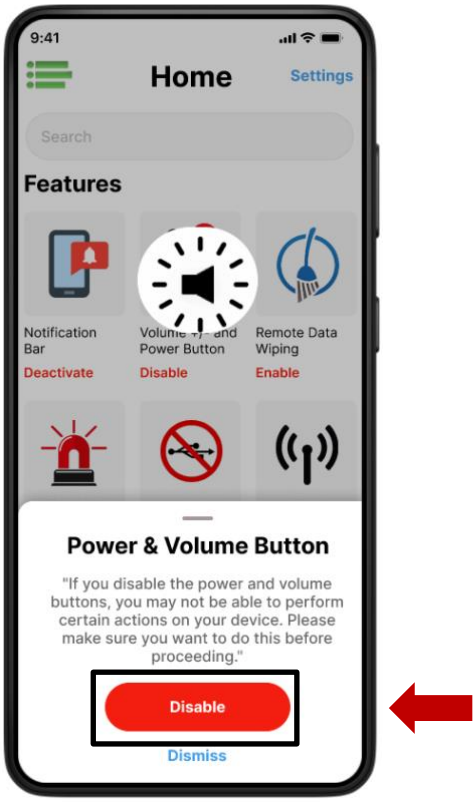

Figure 33: MOBDOG - Volume & Power Button Disabling

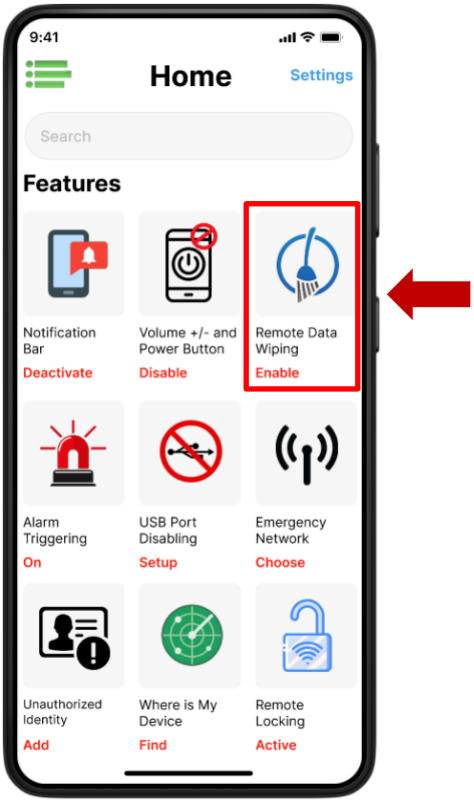

Figure 34: MOBDOG – Remote Data Wiping

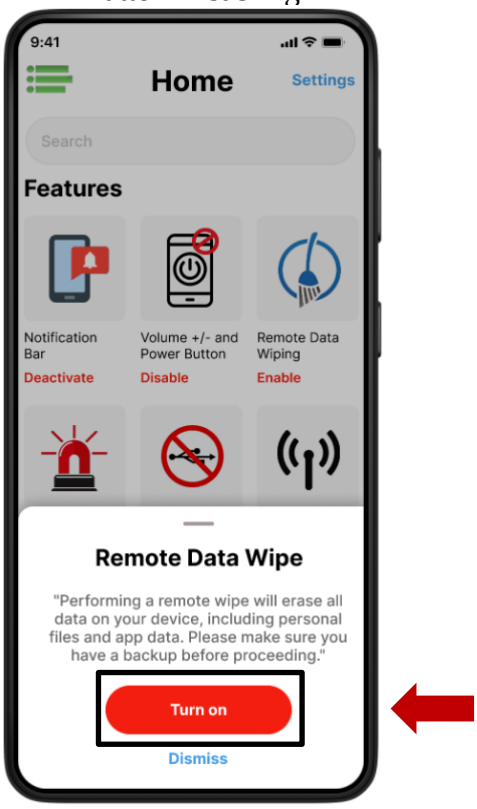

Figure 35: MOBDOG - Data Wiping Trun On

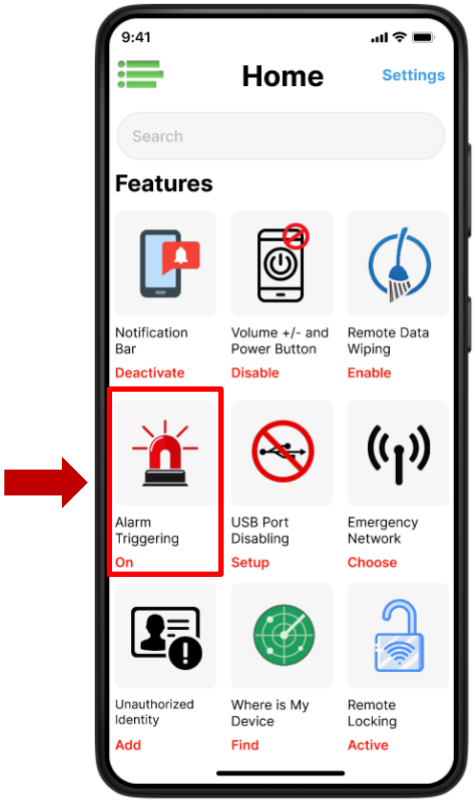

Figure 36: MOBDOG – Alarm Triggering

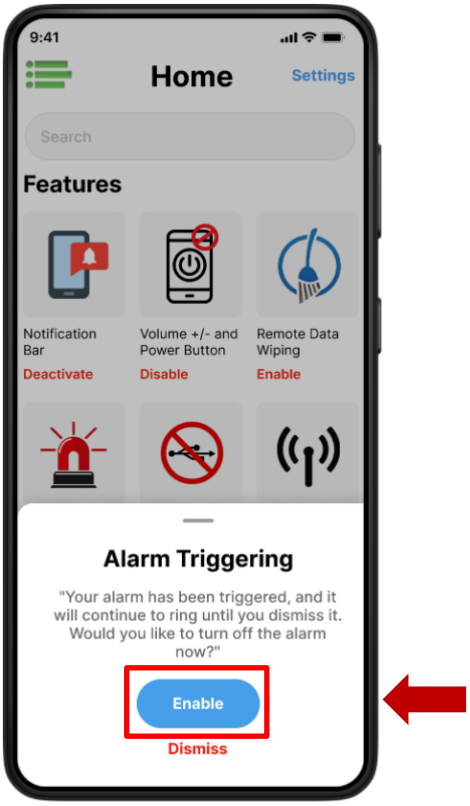

Figure 37: MOBDOG - Alarm Enable

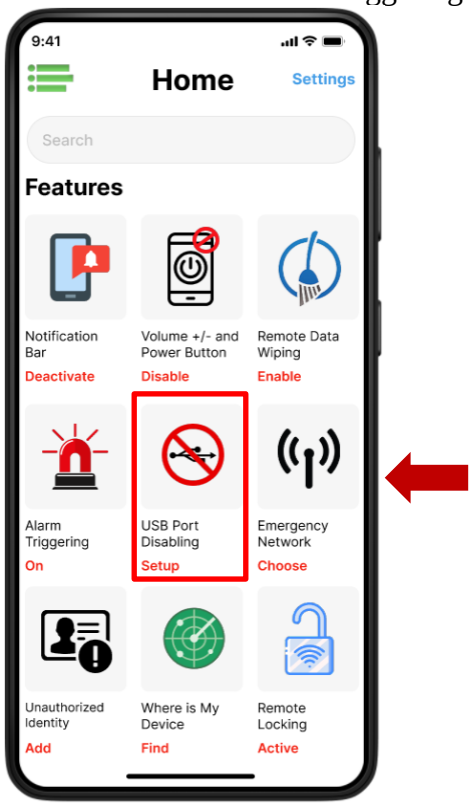

Figure 38: MOBDOG – USB Port Disabling

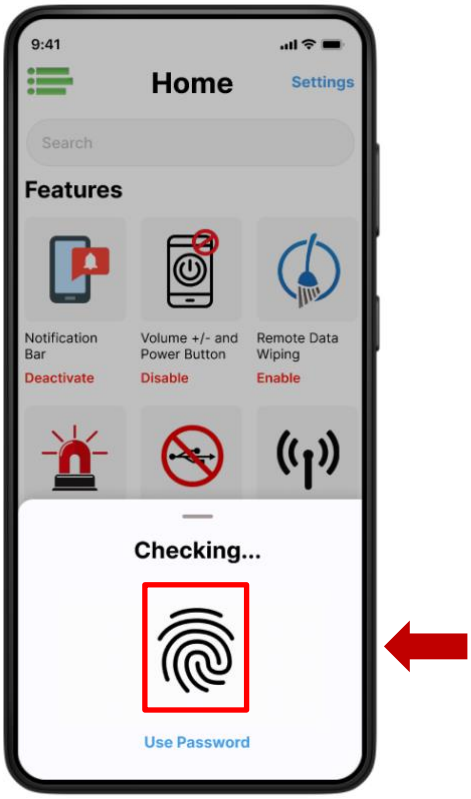

Figure 39: MOBDOG - Port Disabling Verifying

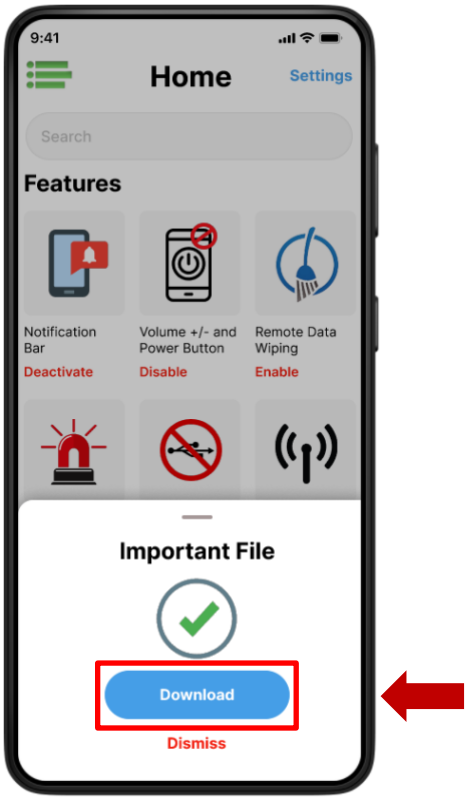

Figure 40: MOBDOG - USB Port Disabling File Download

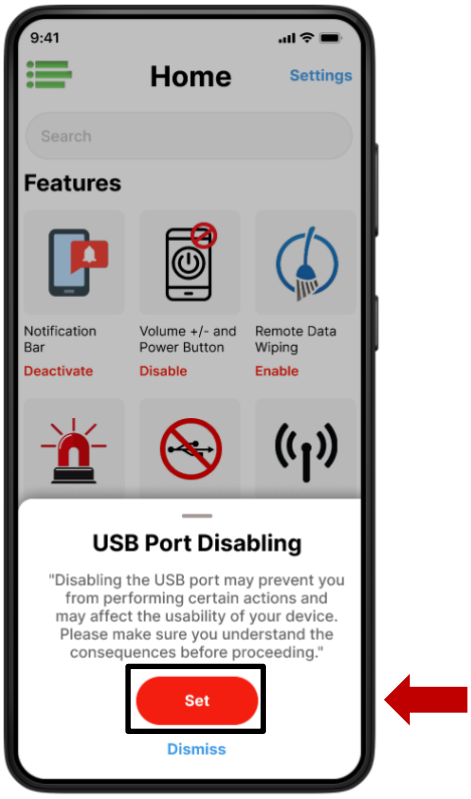

Figure 41: MOBDOG - USB Port Disabling Set

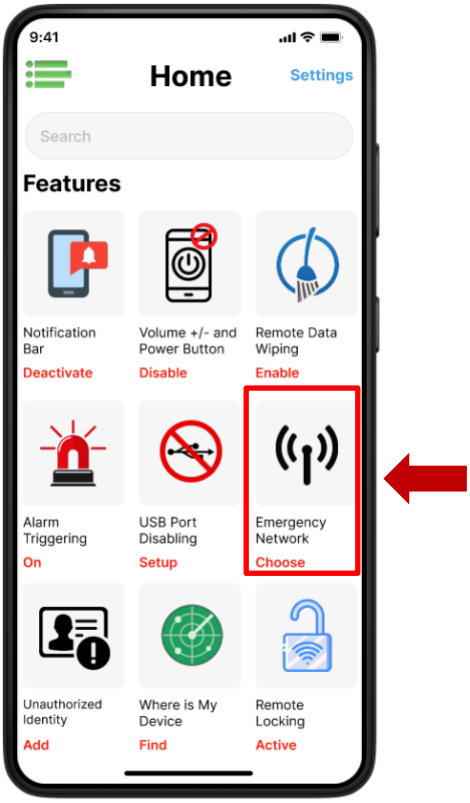

Figure 42: MOBDOG – Emergency Network

Figure 43: MOBDOG – Data Limit Set

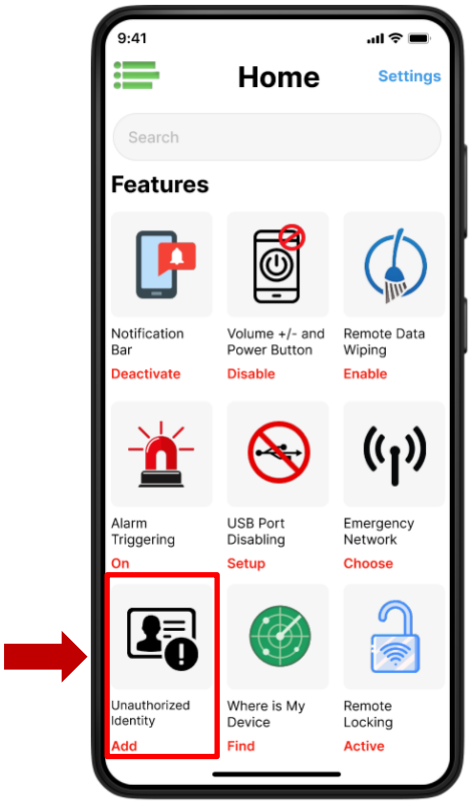

Figure 44: MOBDOG – Unauthorized Identity

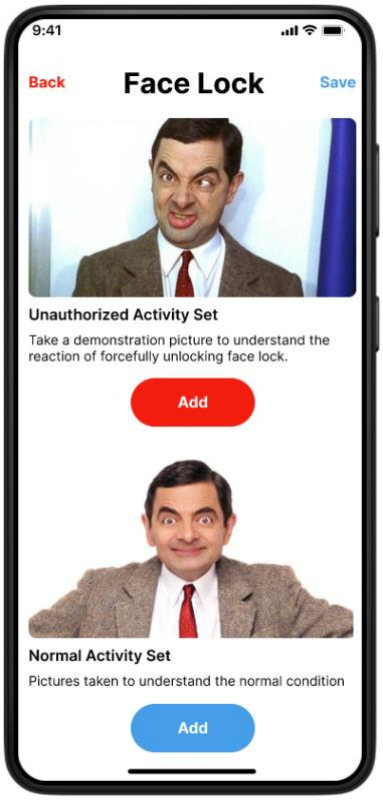

Figure 45: MOBDOG - Unauthorized Identity Add

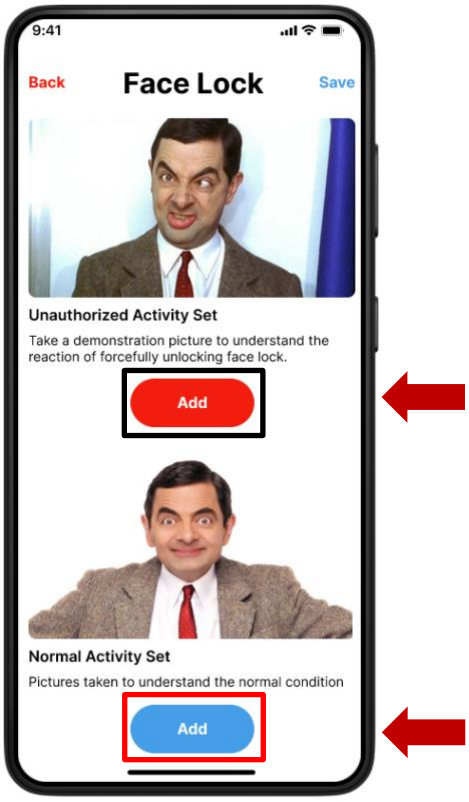

Figure 46: MOBDOG – Activity Add

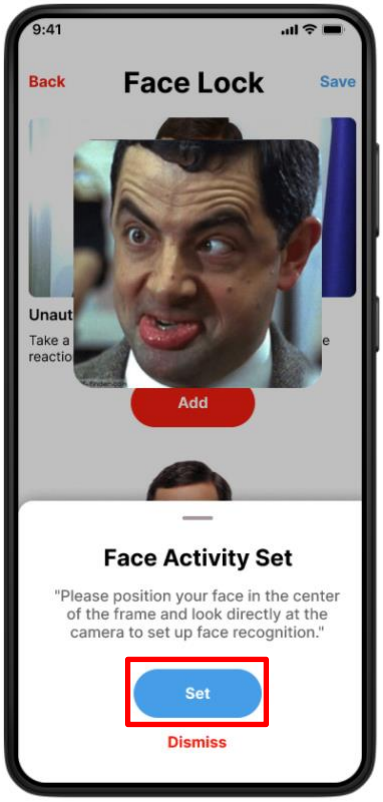

Figure 47: MOBDOG - Activity Set

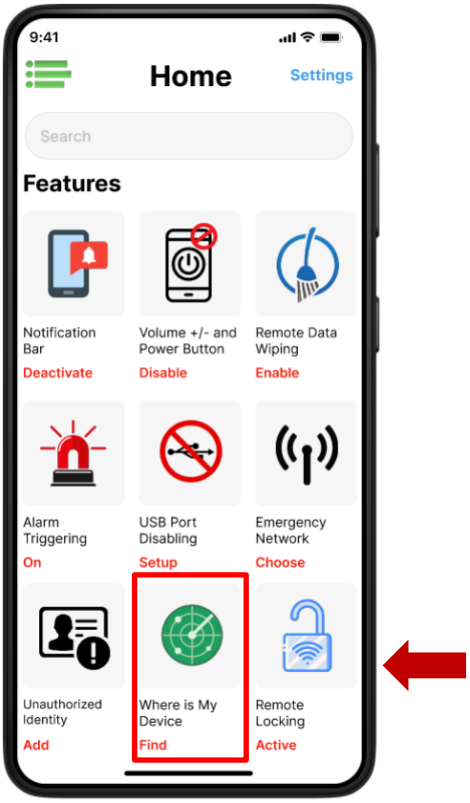

Figure 48: MOBDOG – Where is My Device

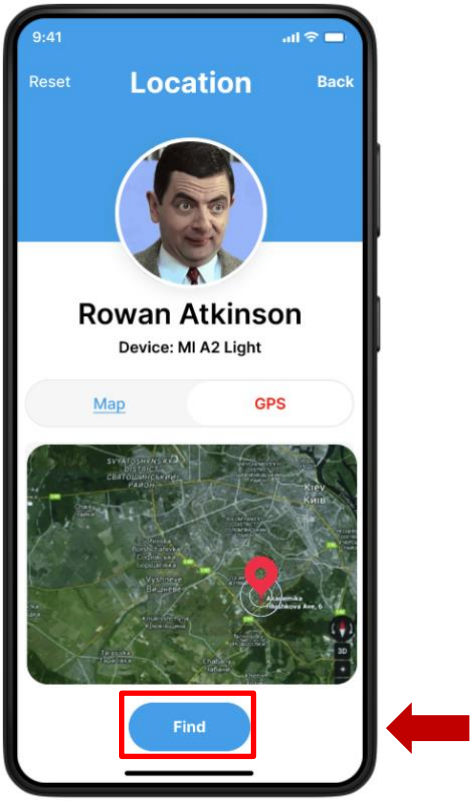

Figure 49: MOBDOG – Device Find

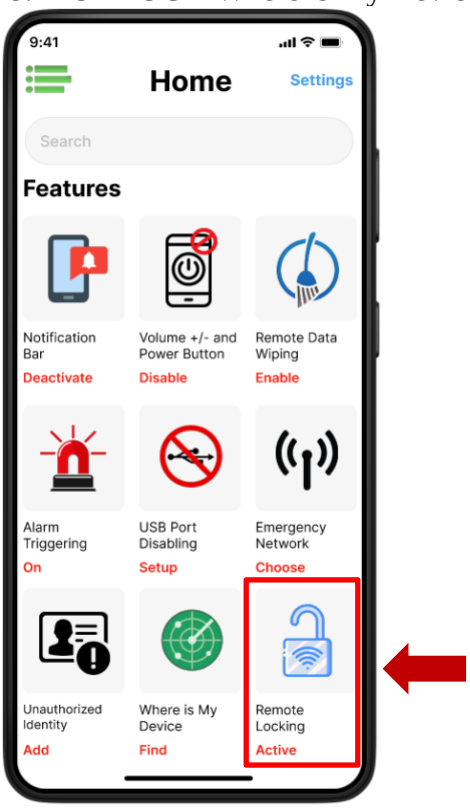

Figure 50: MOBDOG – Remote Locking

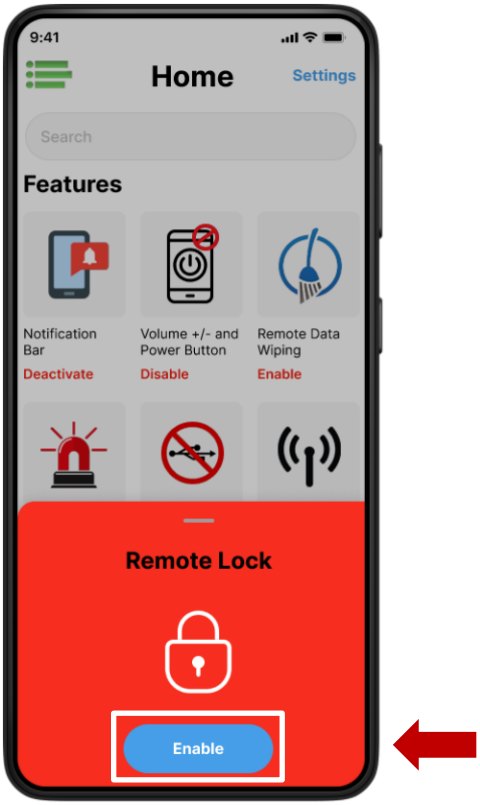

Figure 51: MOBDOG - Remote Locking Enable

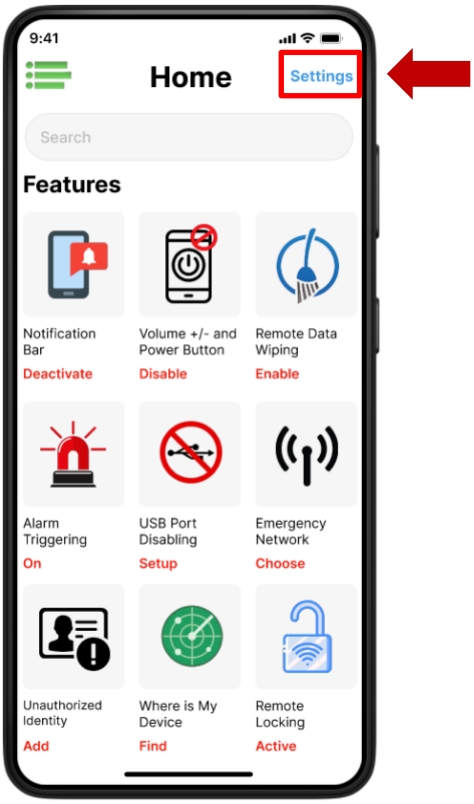

Figure 52: MOBDOG – Settings

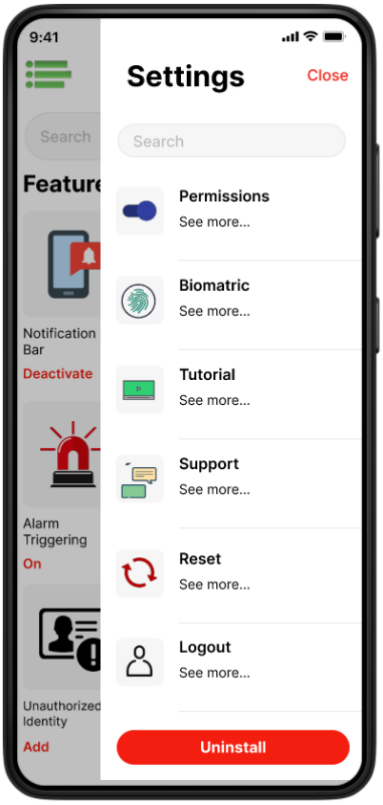

Figure 53: MOBDOG – Settings Menu

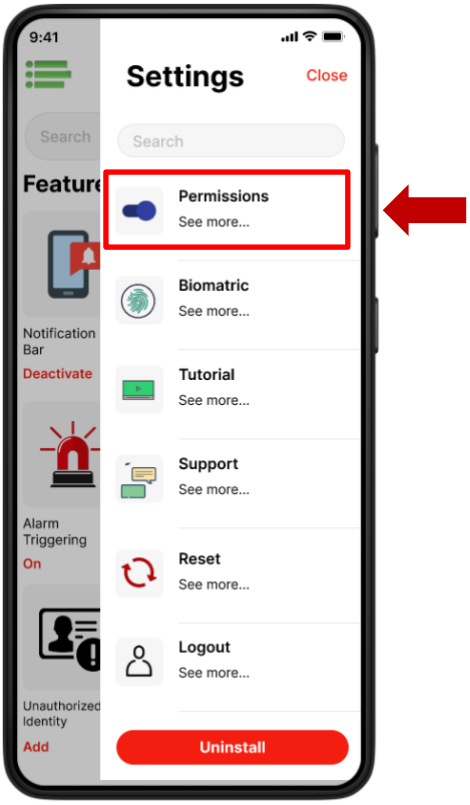

Figure 54: MOBDOG – Settings Permissions

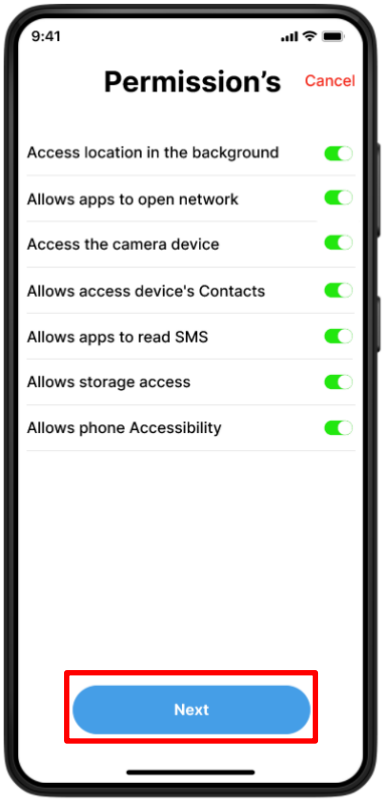

Figure 55: MOBDOG - Update Permissions

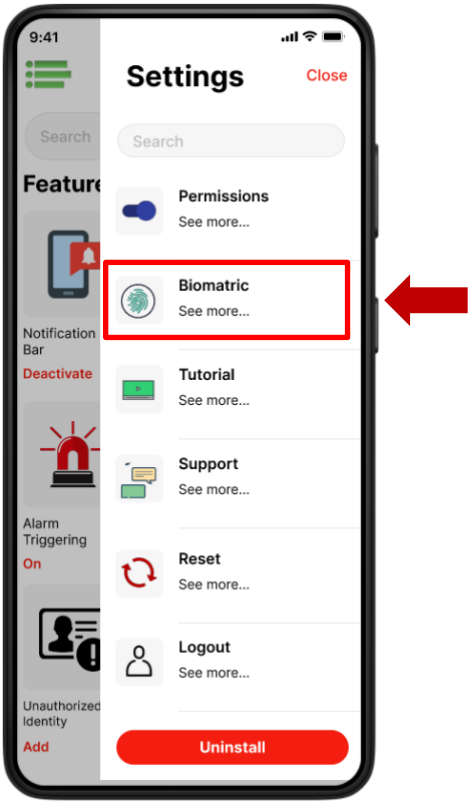

Figure 56: MOBDOG – Settings Biometric

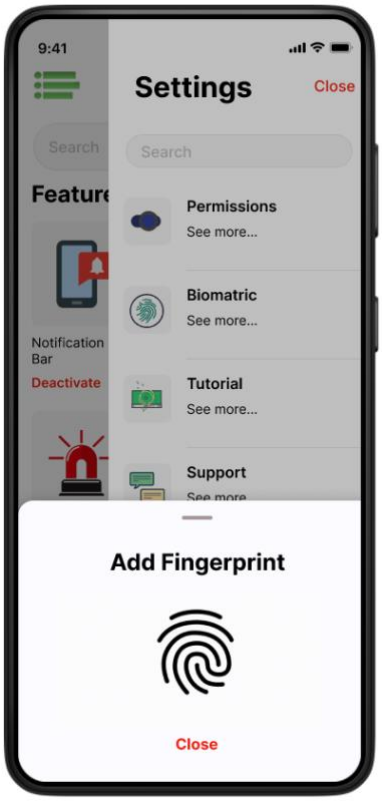

Figure 57: MOBDOG - Biometric Addition

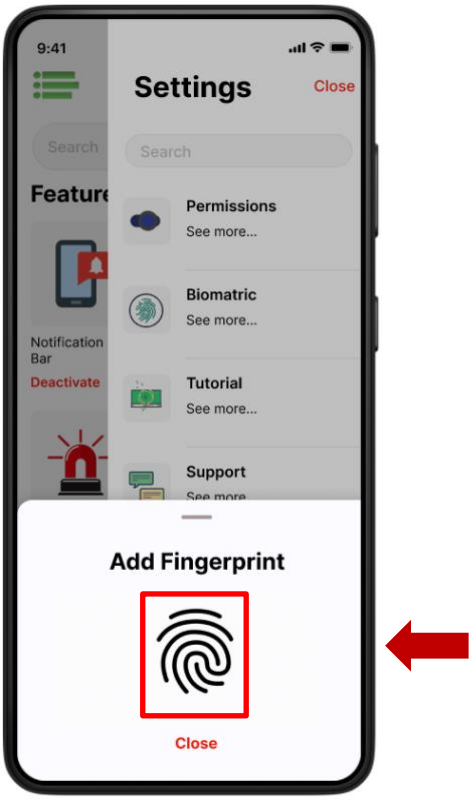

Figure 58: MOBDOG - Biometric Add

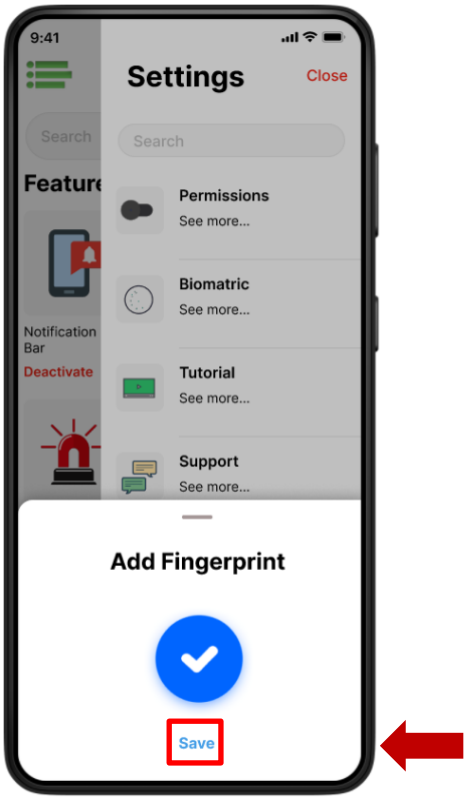

Figure 59: MOBDOG - Biometric Saving

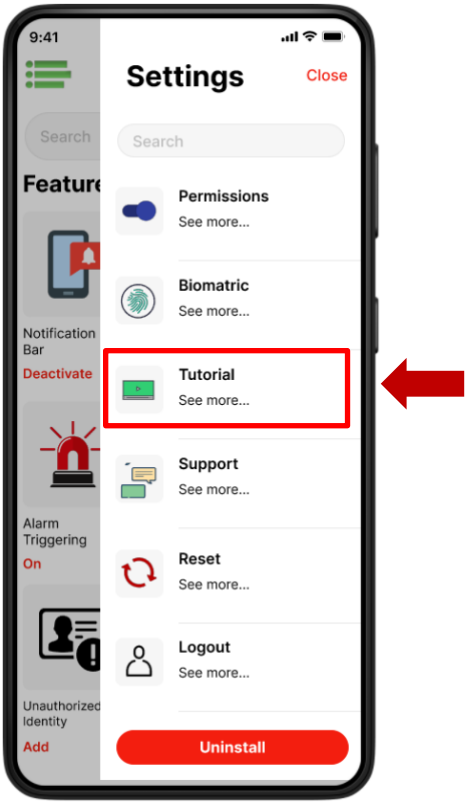

Figure 60: MOBDOG – Settings Tutorial

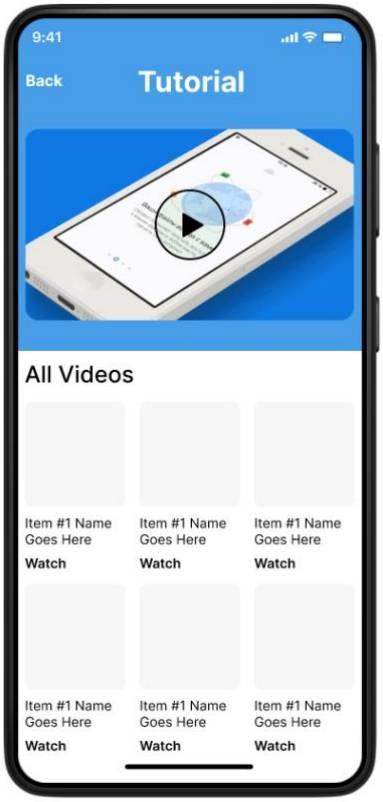

Figure 61: MOBDOG – Tutorial

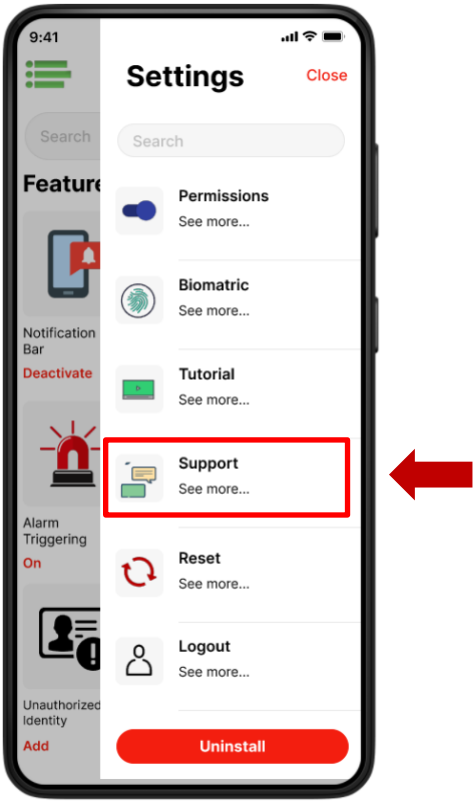

Figure 62: MOBDOG – Settings Support

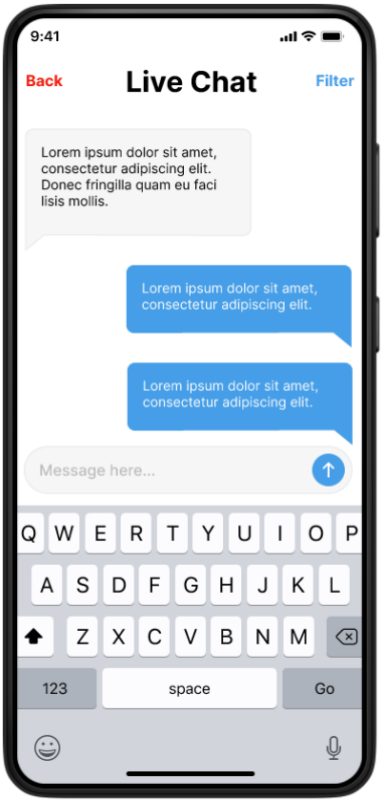

Figure 63: MOBDOG - LIVE Chat

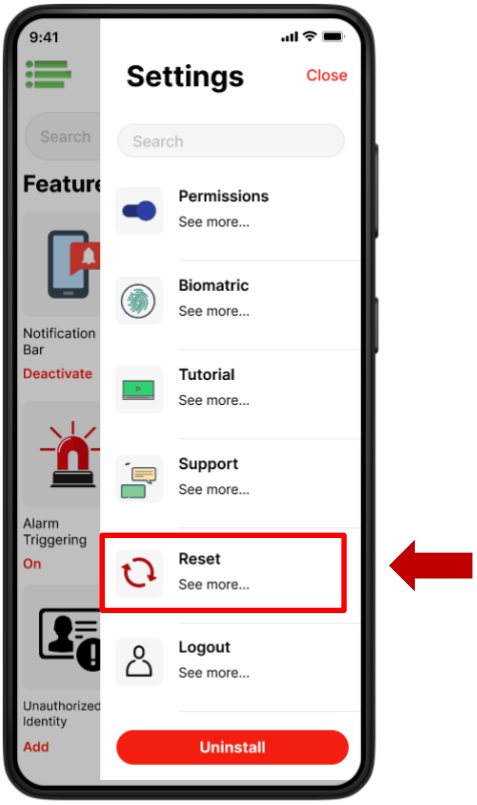

Figure 64: MOBDOG – Settings Reset

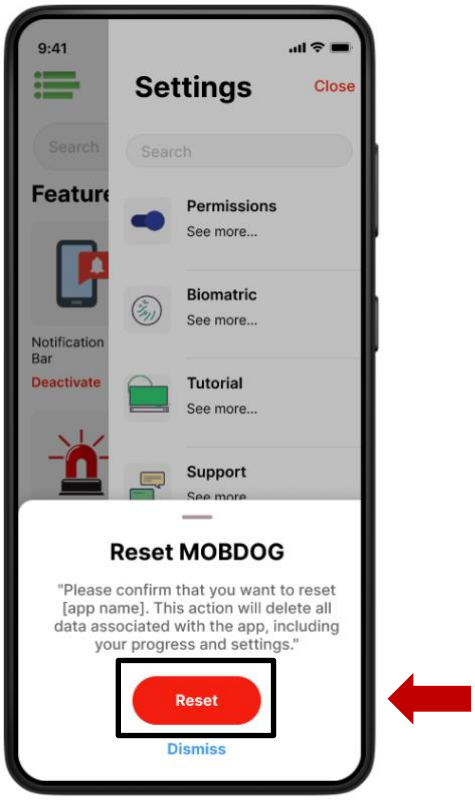

Figure 65: MOBDOG – Reset

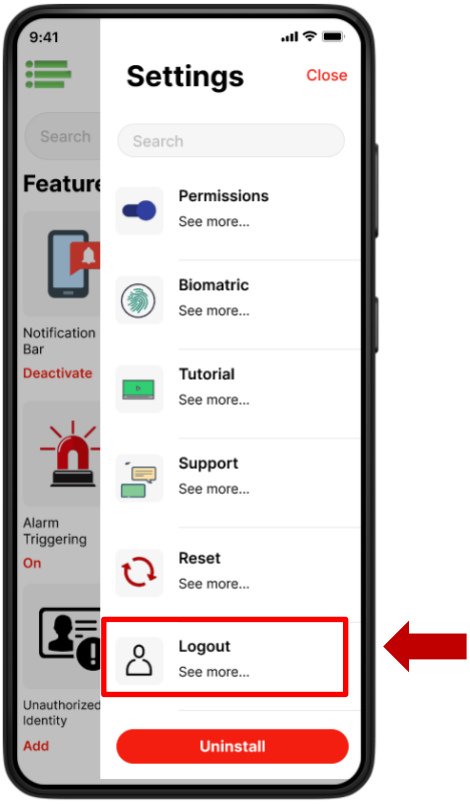

Figure 66: MOBDOG – Settings Logout

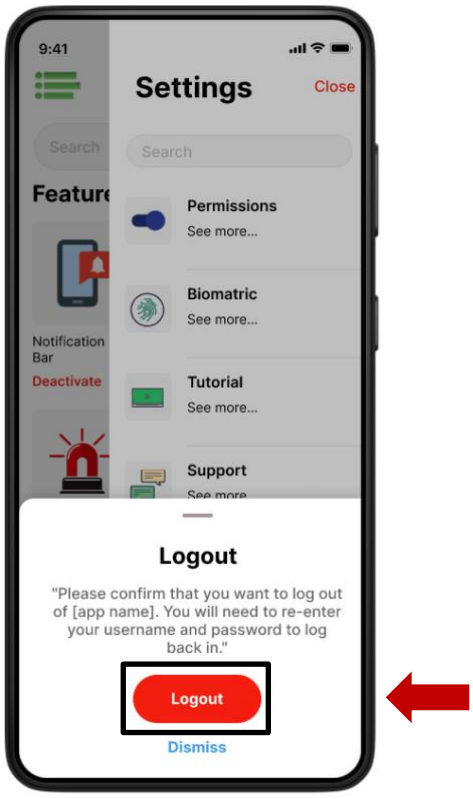

Figure 67: MOBDOG – Logout

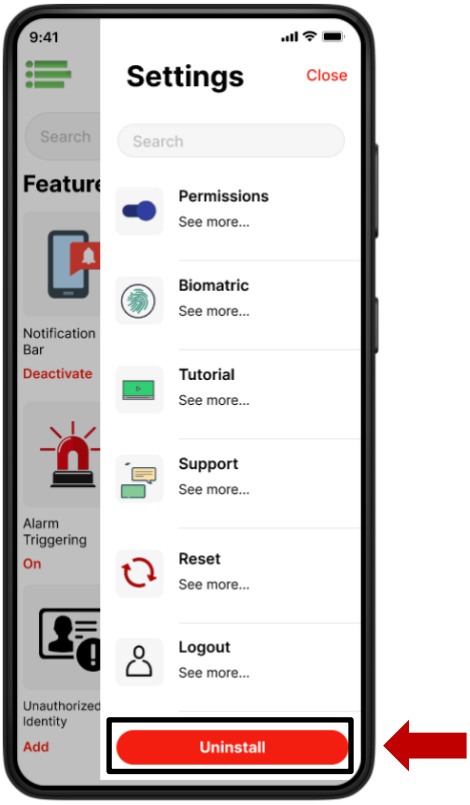

Figure 68: MOBDOG – Settings Uninstall

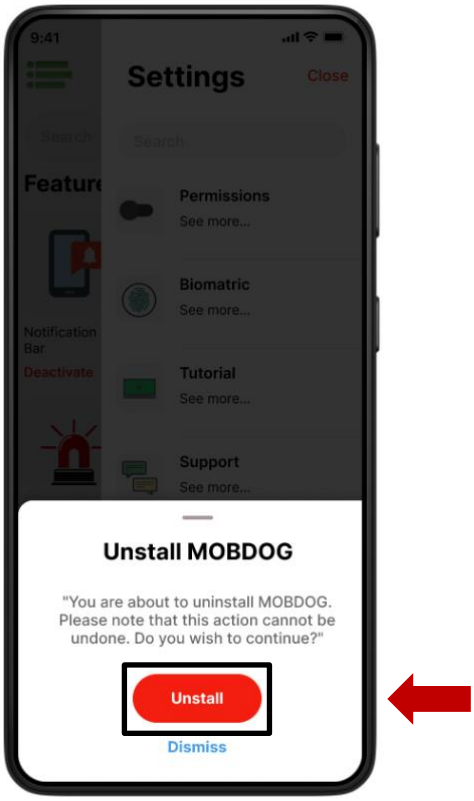

Figure 69: MOBDOG - Uninstall

Figure 70: MOBDOG - Application Uninstalling

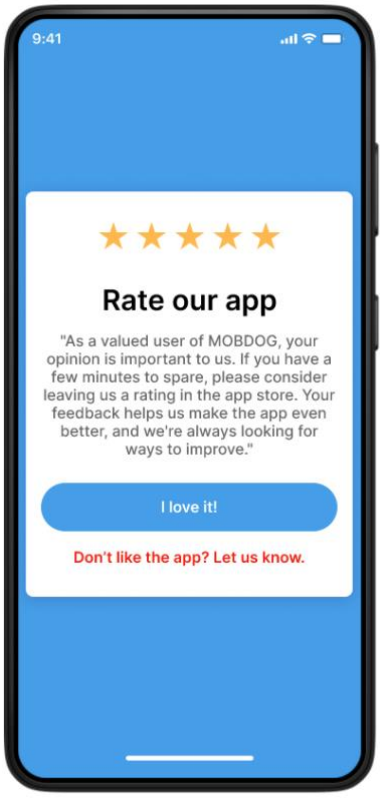

Figure 71: MOBDOG - Application Feedback

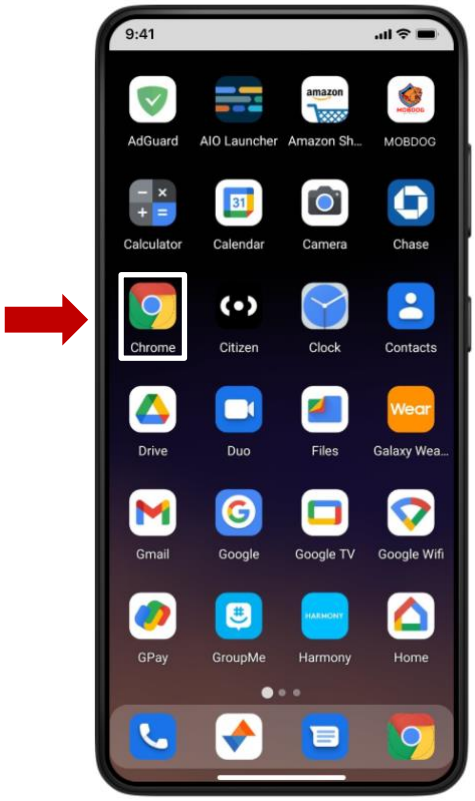

Figure 72: Phone – Opening Browser

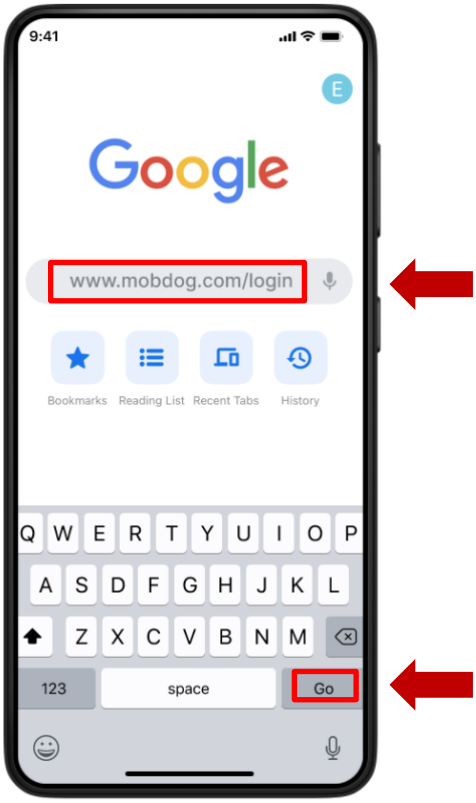

Figure 73: Browser – Home Search

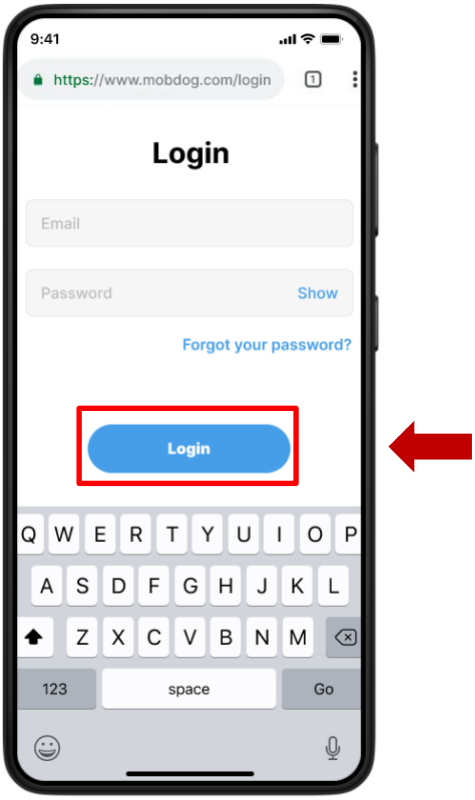

Figure 74: Browser – MOBDOG Web Login

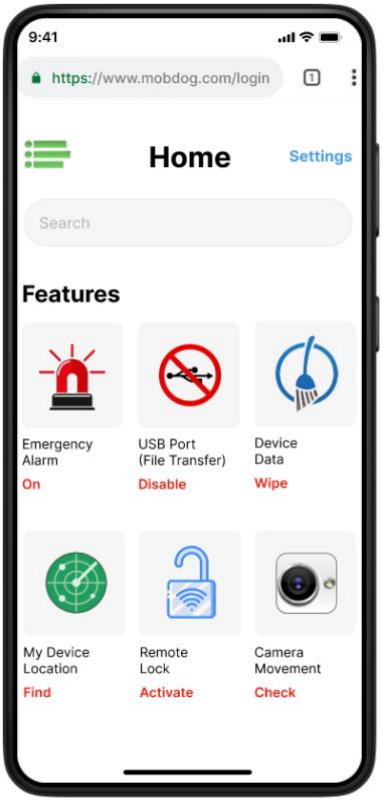

Figure 75: Browser - MOBDOG Home

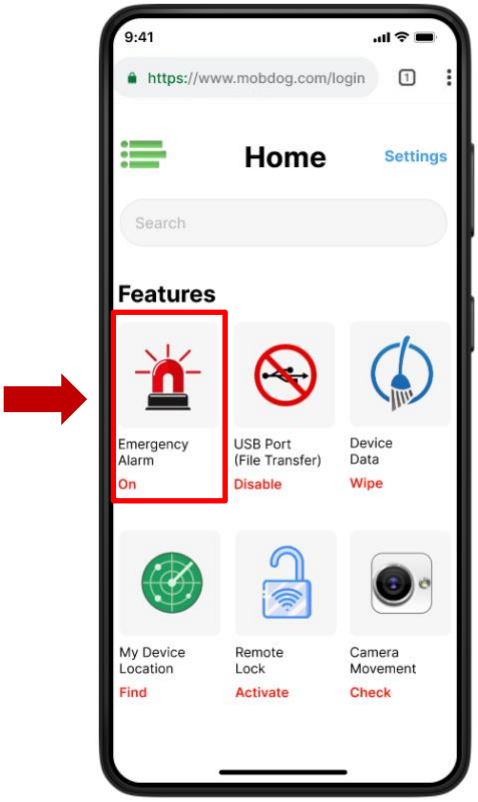

Figure 76: Browser – Emergency Alarm

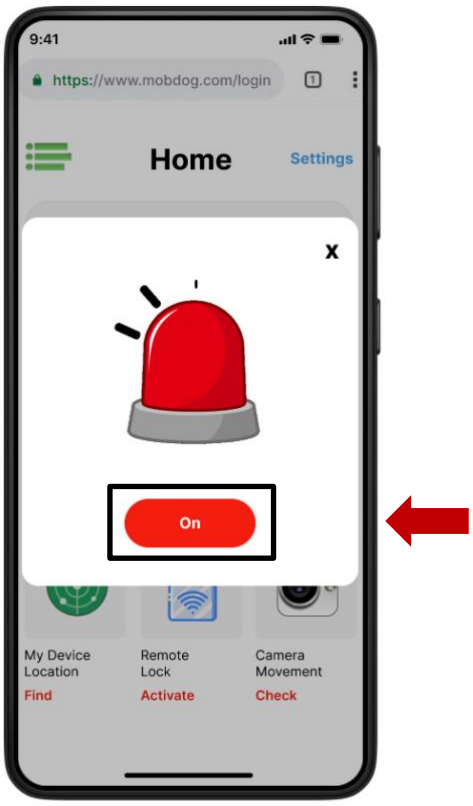

Figure 77: Browser - Emergency Alarm On

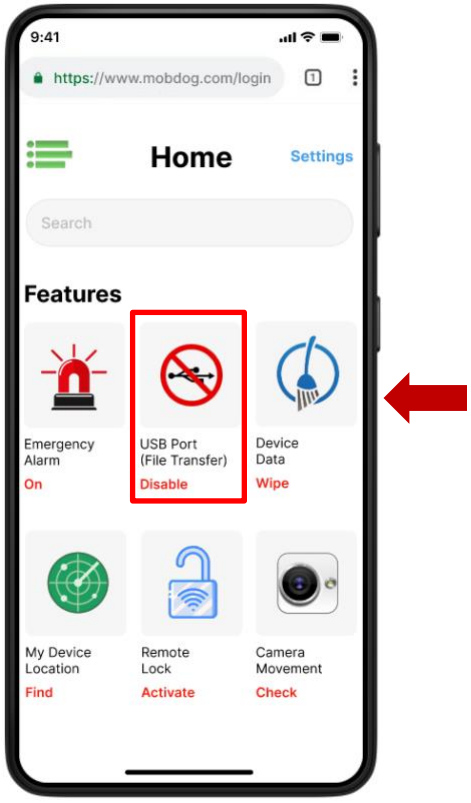

Figure 78: Browser – USP Port

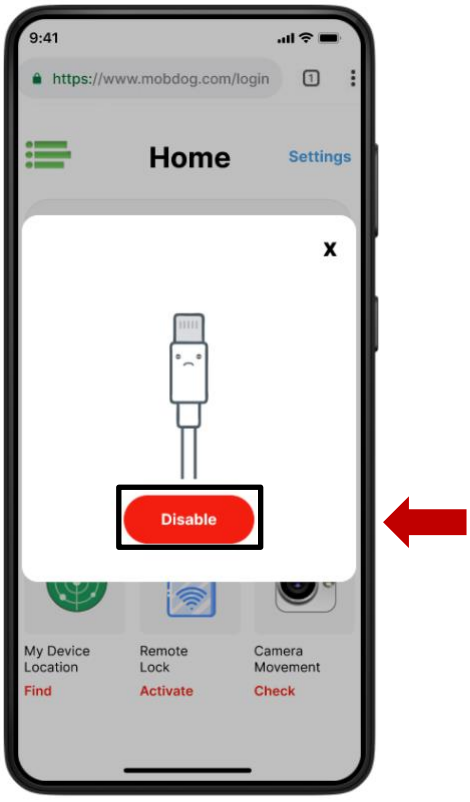

Figure 79: Browser - USP Port Disable

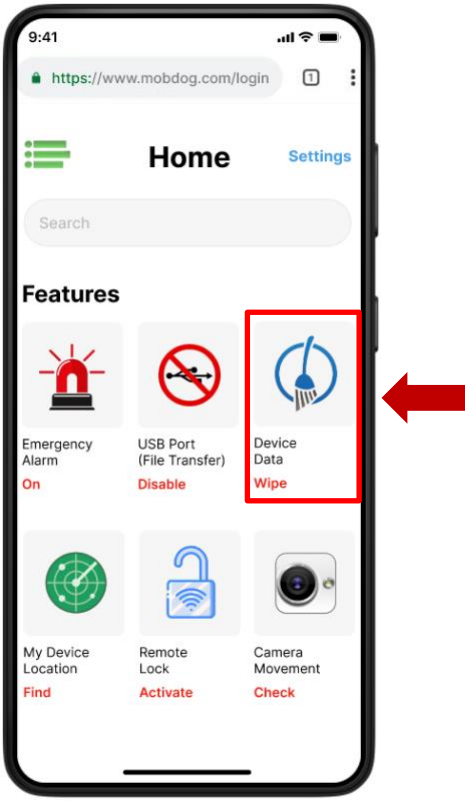

Figure 80: Browser – Device Data

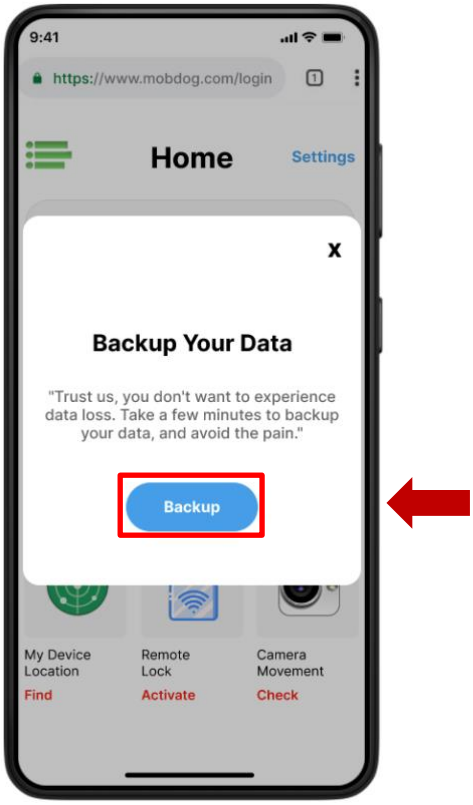

Figure 81: Browser - Device Data Backup

Figure 82: Browser - Data Backup in

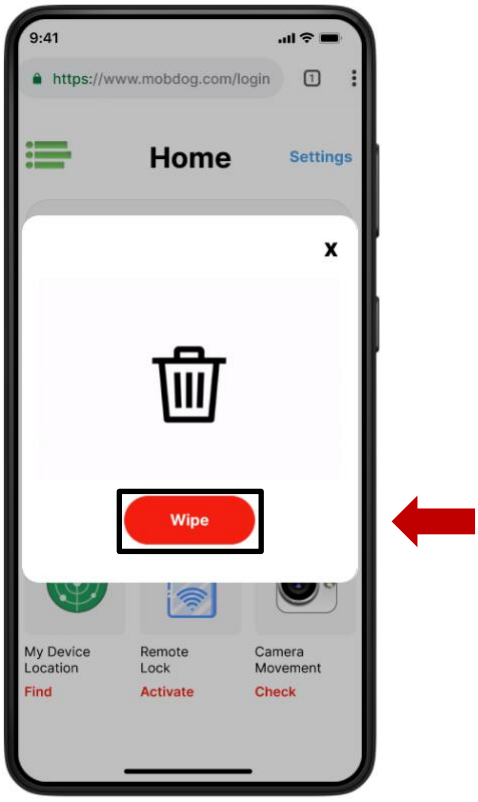

Figure 83: Browser - Device Data Wipe

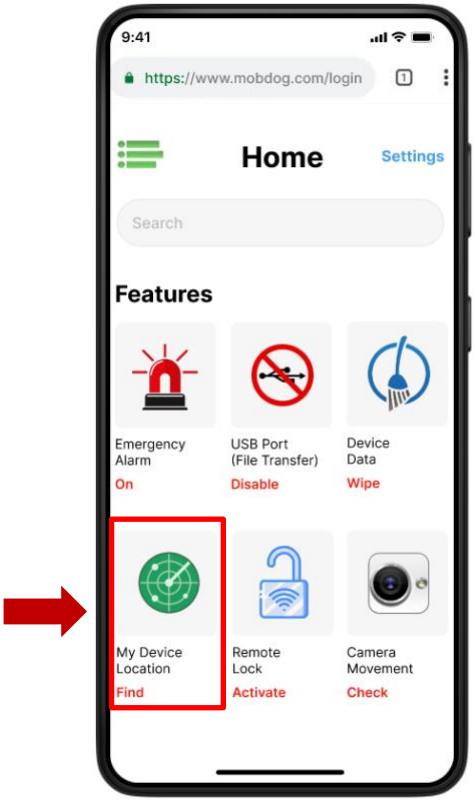

Figure 84: Browser – My Device Location

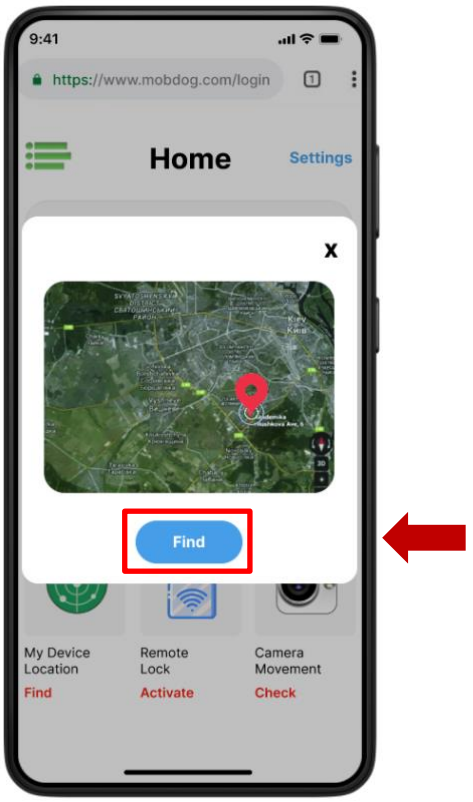

Figure 85: Browser - Device Location Find

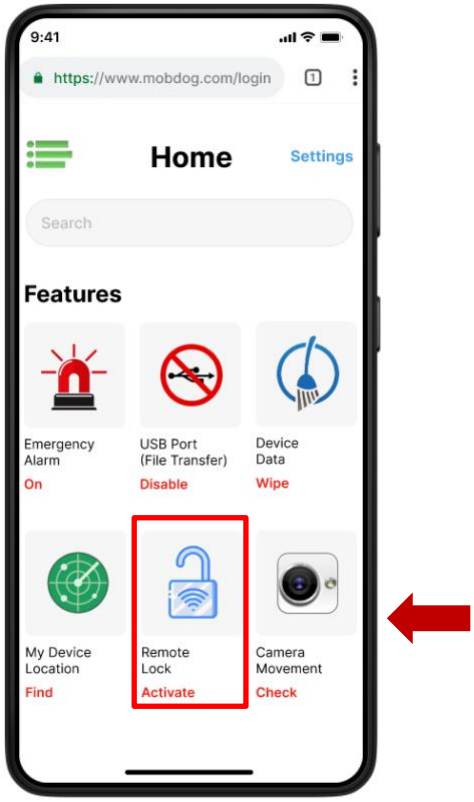

Figure 86: Browser – Remote Lock

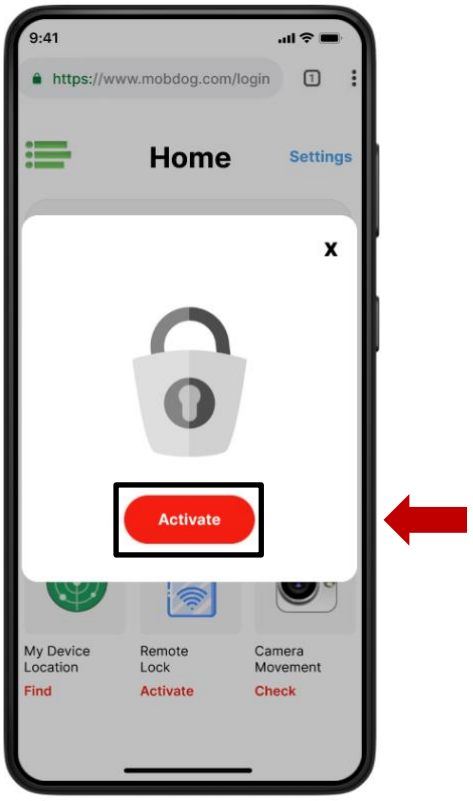

Figure 87: Browser - Remote Lock Activate

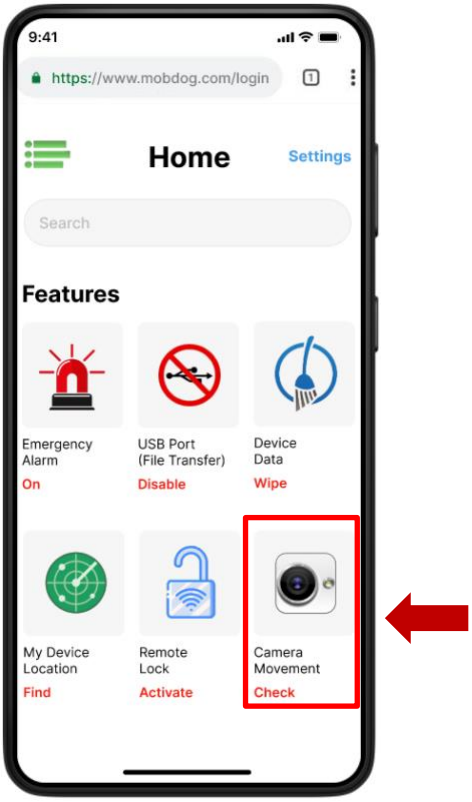

Figure 88: Browser – Camera Movement

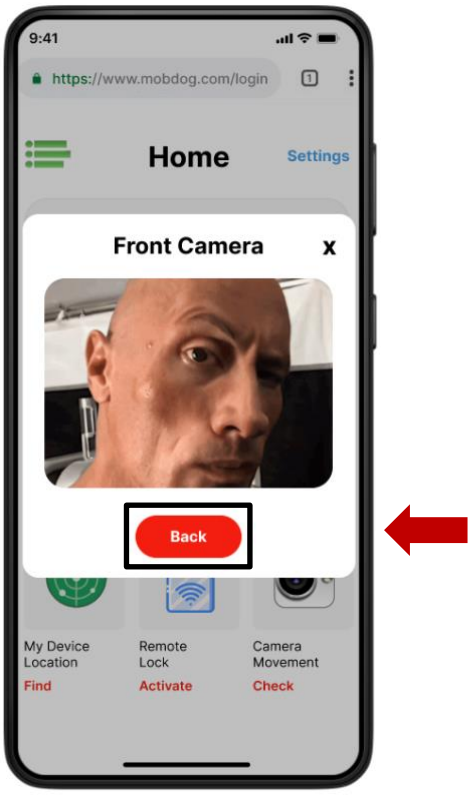

Figure 89: Browser – Front Camera Movement

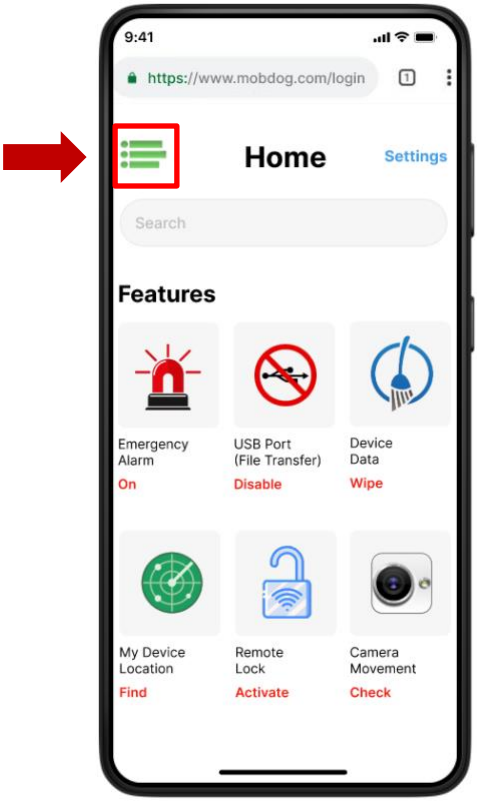

Figure 90: Browser - Profile

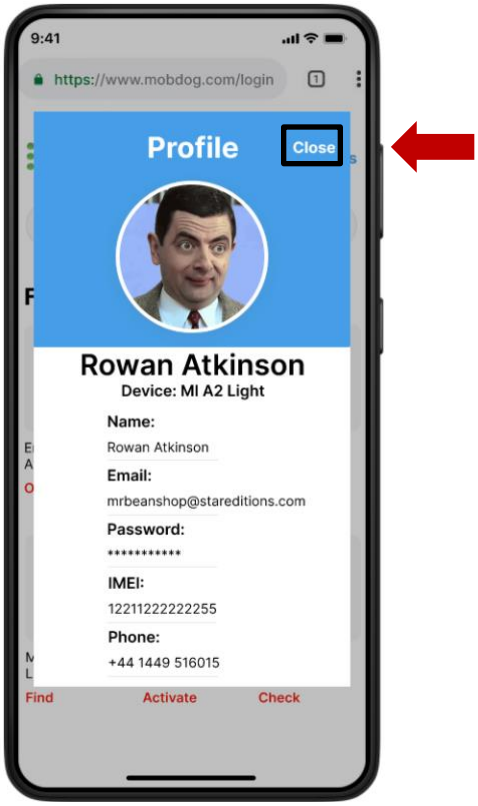

Figure 91: Browser – User Profile

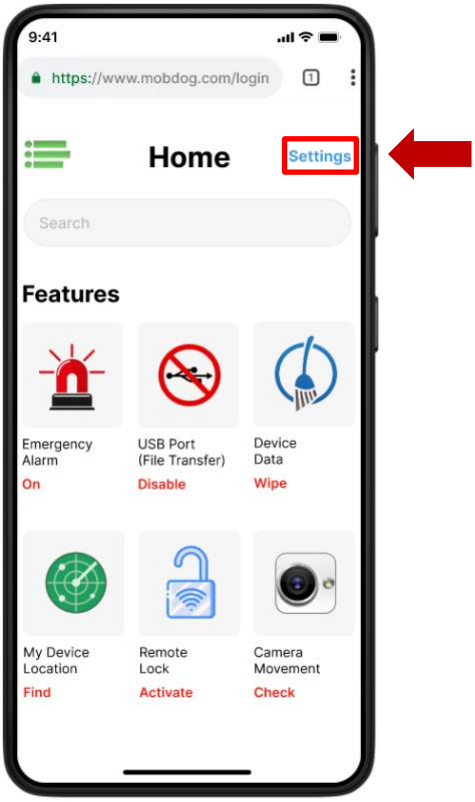

Figure 92: Browser – Settings

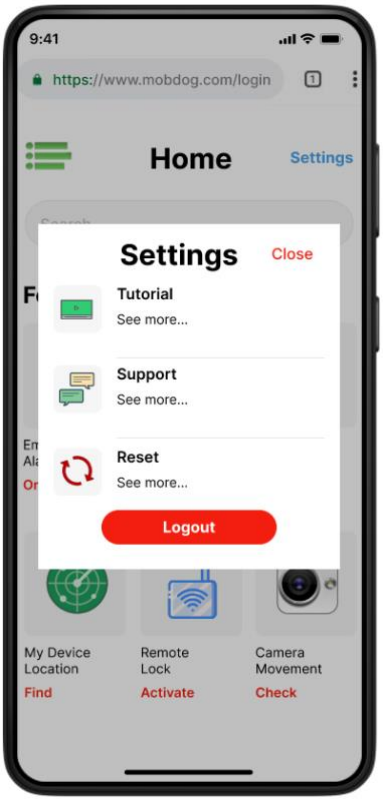

Figure 93: Browser – Settings On

Figure 94: Phone - Lock Screen (Lost/Thief Hand)

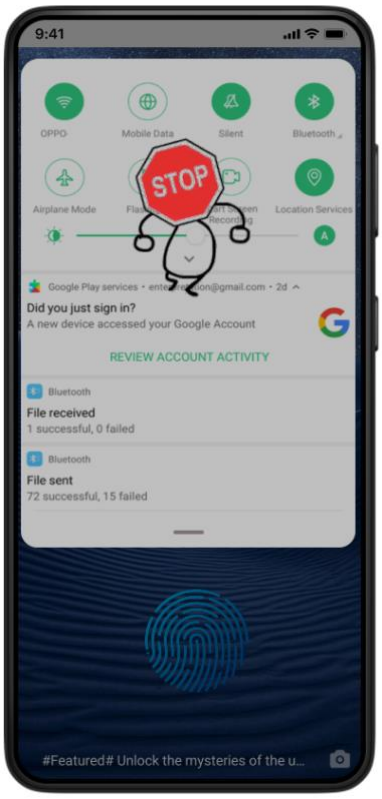

Figure 95: Phone – Notification Bar Disable

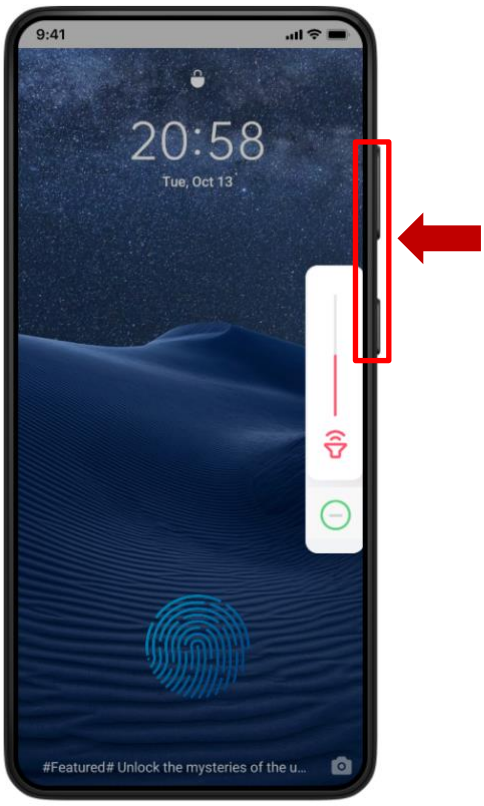

Figure 96: Phone – Volume & Power Button

Figure 97: Phone - Button Disable

Figure 98: Phone – Multiple Unauthorized Attempt

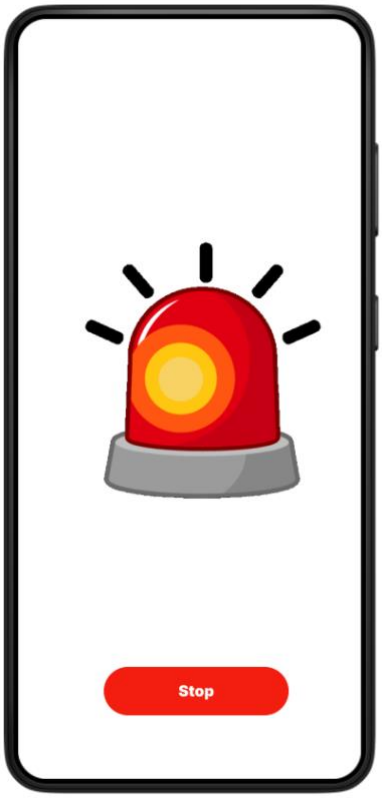

Figure 99: Phone – Emergency Sirens On

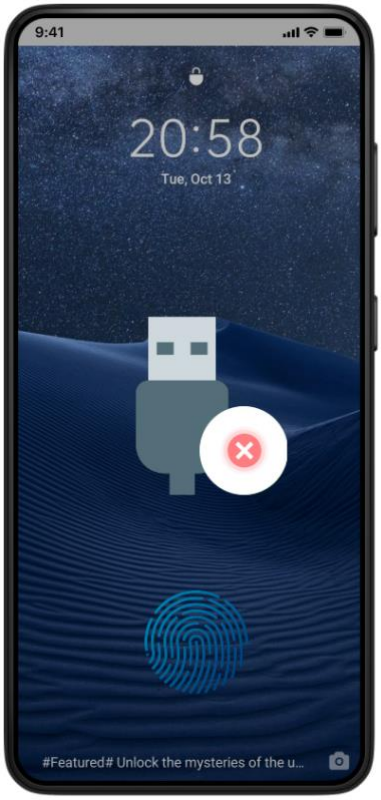

Figure 100: Phone – USB Port Disable

Figure 101: Phone – Remote Wipe