Industry Internship Report
on
Open-source Intelligence Data Mining System

Submitted for the Partial Fulfillment of the Requirements for the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Emmadi Sumith Kumar

Roll No.: UI20CS21

Under the guidance of

Dr Pradeep Kumar Roy

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

April, 2024

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SURAT-394190

Indian Institute of Information Technology Surat
Computer Science and Engineering Department

CERTIFICATE

This is to certify that candidate **Emmadi Sumith Kumar** bearing Roll No: **UI20CS21** of B.TECH. IV, 8th Semester has successfully carried out the work on "**Open-source Intelligence Data Mining System**" for the partial fulfillment of the degree of Bachelor of Technology (B.Tech.) in **April, 2024**.

Faculty Supervisor: *Dr. Pradeep Kumar Roy* Sign:.....

1. Examiner 1: *Dr. Pradeep Kumar Roy* Sign:.....

2. Examiner 2: *Dr. Vipul Kumar kania* Sign:.....

3. Examiner 3: *Ms. Jiby* T C Sign:.....

(Seal of the Institute)

DECLARATION

This is to certify that
(i) This report comprises my original work towards the degree of Bachelor of Technology in Computer Science and Engineering at Indian Institute of Information Technology (IIIT) Surat and has not been submitted elsewhere for a degree,
(ii) Due acknowledgement has been made in the text to all other material used.

Signature of Student
(Emmadi Sumith Kumar)

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to all those who have helped me during my internship at **C-Trace Soft Solutions Private Limited** Company.

First and foremost, I would like to thank Prof. **J. S. Bhat**, Director, IIIT Surat for giving me this opportunity. I would also like to thank my college supervisor, **Dr. Pradeep Kumar Roy**, for their constant support and encouragement during the internship. Their insightful comments helped me to improve my work and gave me the confidence to take on new challenges. I am deeply grateful to my company mentor, **Mr. Venkata Rami Reddy Kasu**, who provided me with guidance, support, and valuable feedback throughout my internship. Their expertise, patience, and kindness made a significant difference in my learning and professional growth.

I am grateful to the IIIT Surat and Training and Placement Cell of my college for providing me with the opportunity to pursue this internship.I feel privileged to have had the opportunity to work with such supportive and knowledgeable individuals, and I am grateful for their contribution to my professional development.

ABSTRACT

This **Open-source Intelligence Data Mining System** is software is used to assist law enforcement agencies, such as police departments, in crime investigations by generating a report based on call data records (CDRs) and tower dump data. The software is designed to provide a rapid and efficient report, enabling investigators to identify suspects.

The primary objective of this software is to provide various information about individuals, including vehicle details, location data, IMEI numbers, phone numbers, PAN numbers, MNP details, IP information, etc. The software generates a PDF report containing all relevant information about the person.

The software is designed to be scalable, allowing for future enhancements and additions to accommodate evolving data collection requirements. Its success is dependent on technical infrastructure, legal compliance, user adoption, interoperability, and training and support for successful implementation.

The software is exclusively used by police department and authorized users for crime investigation purposes. It is designed to be secure and reliable, with data validation and verification processes to ensure the accuracy and reliability of collected information. The software serves as a valuable tool for law enforcement agencies, providing them with the necessary information to conduct successful criminal investigations.

The proposed software is already in use by several law enforcement agencies, This software has some pending features under development. It is continuously updated to meet the evolving needs of law enforcement agencies and to enhance its functionality and usefulness.

Contents

List of Principal Symbols and Acronyms

VS	**V**isual **S**tudio
OSINT	**O**pen **S**ource **I**ntelligence
RC	**R**egistration **C**ertificate
CRC	**C**hasis to **R**egistration **C**ertificate
SMS	**S**hort **M**essage **S**ervice
PAN	**P**ermanent **A**ccount **N**umber
IP	**I**nternet **P**rotocol
GPS	**G**lobal **P**ositioning **S**ystem
IPL	**I**nternet **P**rotocol **L**ogger
VN	**V**irtual **N**umber
IMEI	**I**nternational **M**obile **E**quipment **I**dentity
PNR	**P**assenger **N**ame **R**ecord
IFSC	**I**ndian **F**inancial **S**ystem **C**ode
UPI	**U**nified **P**ayments **I**nterface
CC	**C**ourt **C**ase
BTS	**B**ase **T**ransceiver **S**tation
MNP	**M**obile **N**umber **P**ortability
AES	**A**dvanced **E**ncryption **S**tandard
VCS	**V**ersion **C**ontrol **S**ystem
DBMS	**D**atabase **M**anagement **S**ystem
IDE	**I**ntegrated **D**evelopment **E**nvironment
CI/CD	**C**ontinous **I**ntegration and **C**ontinous **D**eployment

List of Figures

Format-3

OBJECTIVES/GUIDELINES/AGREEMENT: INTERNSHIP SYNOPSIS

(THIS WILL BE PREPARED IN CONSULTATION WITH SUPERVISOR)

An internship is a unique learning experience that integrates studies with practical work. This agreement is written by the student in consultation with the faculty Mentor and Industrial supervisor. It shall serve to clarify the educational purpose of the internship and to ensure an understanding of the total learning experience among the principal parties involved.

Part I: Contact Information

Student

Name: Emmadi Sumith Kumar Student ID # UI20CS21 Class Year: 4th

Campus Address: IIIT Surat , Kholvad Campus , Kamrej, Surat

City, State: Kamrej, Gujarat

Phone: +91 99128 57147 Email: sumithemmadi@gmail.com

Industrial Supervisor

Name: Kasu Venkata Rami Reddy Title: Senior Developer

Company/Organization: C-TRACE SOFT SOLUTIONS PVT. LTD.

Internship Address: #402, Mallik Chambers, Hyderguda, Himayath Nagar, Hyderabad

City, State, Pin: Hyderabad, Telangana - 500029

Phone: +91 84658 02838 Email: ram.kasu@gmail.com

Faculty Mentor

Name: Dr. Pradeep Kumar Roy Phone: +91 85390 35222

Campus Address: IIIT Surat , Kholvad Campus , Kamrej, Surat

Academic Credit Information

Internship Title: Full Stack Developer Intern Department: CSE

Course #: CS801 Credits: 24

Grading Option: Credit/Non-Credit

Beginning Date: 8 - JAN - 2024 Ending Date: 30 - JUNE - 2024

Hours per Week: 40 Internship is: ✓ Paid Unpaid

Part II: Internship Objectives/Learning Activities

Internship Objectives: What do you intend to learn, acquire and clarify through this internship? Try to use concrete, measurable terms in listing your learning objectives under each of the following categories:

- Knowledge and Understanding

  During my internship, I aim to master my knowledge on server-side scripting languages , particularly focusing on technologies like Node.js and python. I also plan to gain a deep understanding of the React Native framework for cross-platform mobile app development. Additionally, I intend to explore and implement native modules in Kotlin to boost the functionality and performance of the application.

- Skills

  During my internship, I aim to develop skills in backend development, API design, MongoDB database management, and node-addon-apis, python c++ extentions. I'm eager to understand and work with React Native for Android app development. Additionally, I plan to design native modules using Kotlin. My goal is to contribute effectively to the Internship project and gain practical experience in these areas.

Part III: The Internship

Job Description: Describe in as much detail as possible your role and responsibilities while on your internship. List duties, projects to be completed, deadlines, etc. How can you contribute to the organisation/site of internship. It must contain the assurance of carrying real/live project during the internship.

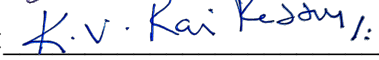
As a Full Stack Developer Intern, my primary role involves designing and implementing APIs in Node.js and Python for our Open Source Intelligence (OSINT) Report system. Specifically, I focus on creating social media APIs to extract user information based on phone numbers, emails, and usernames. These APIs are critical tools used by law enforcement, integrated into WhatsApp bots for efficient data access.

Additionally, I am responsible for developing an Android application using React Native, integrating APIs with some extra APIs, and adding features to enhance user experience and security. This internship offers hands-on experience with live projects, contributing directly to our organisation's goal of providing advanced OSINT tools to law enforcement for effective information gathering and analysis.

Part IV: Agreement

This contract may be terminated or amended by student, faculty coordinator or work supervisor at any time upon written notice, which is received and agreed to by the other two parties.

Student:  Date: 31/01/2024

Industry Supervisor:  Date: 01/02/2024

Faculty Supervisor: _____ Date: _____

Chapter 1

Introduction

The Open-source Intelligence Data Mining System (OSINT), as depicted in Figure 1.1, is an advanced software specifically designed to assist law enforcement agencies, such as the police department, in crime investigations by effectively analyzing call data records (CDRs) and tower dump data. It offers a comprehensive range of features crucial for conducting successful criminal investigations.

The primary objective of this application is to provide law enforcement agencies with a rapid and efficient means of analyzing call data records (CDRs) and tower dump data. C-Trace OSINT software acts as a force multiplier, empowering investigators to make informed decisions and progress their investigations more effectively.

1.1 What is OSINT Data Mining System ?

The developed application is specifically designed for use by police departments only in solving crime cases by analyzing call data records (CDRs) and tower dump data. This software provides a detailed report on individuals using their phone number, IMEI number, PAN number, etc. It can extract tower data to identify all phone users within a specific tower, using azimuth ID. The report includes information such as the person's vehicle details, location data, IMEI numbers, phone numbers, PAN numbers, MNP details, IP information, and more. The software generates a PDF report that encompasses all relevant information about the individual.

1.2 Features of OSINT Software

The developed software provides a range of powerful search functionalities based query. Below are some of the key features offered:

i. **Vehicle information** : RC MH02ZX1234

ii. **IP Lookup** : IP 192.168.01.01

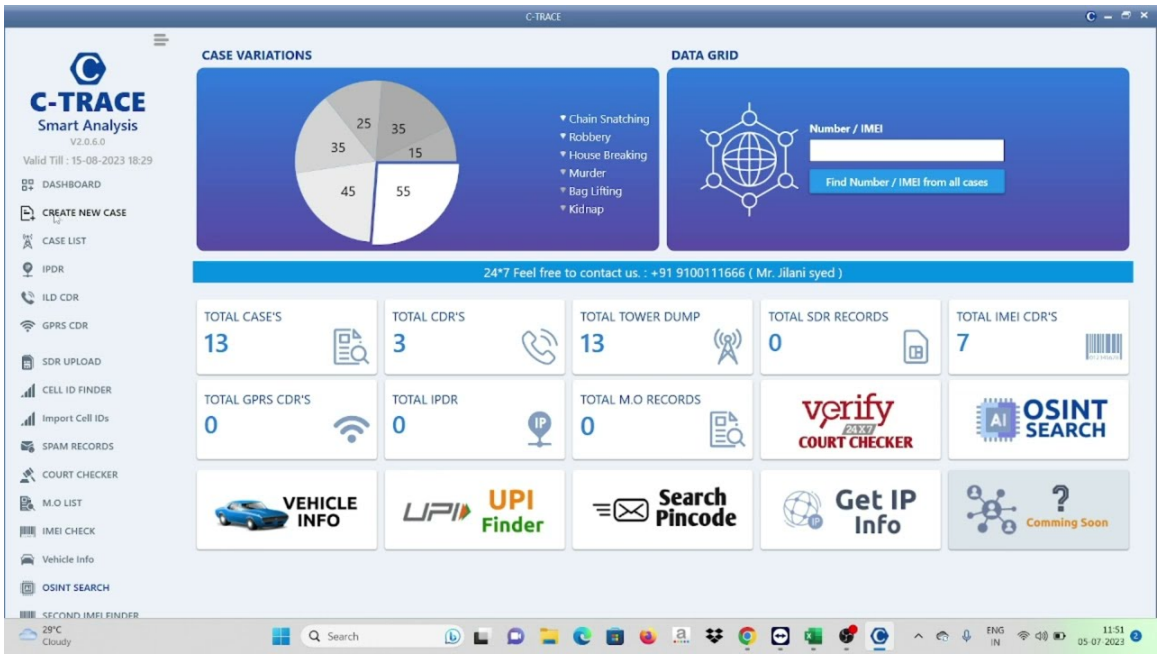


Figure 1.1: C-Trace OSINT Software

iii. **Pin code search** : PIN 524455

iv. **IMEI Search** : IMEI 45671234765823
You can Get Device Model details

v. **PNR Search** : PNR 456789101
Check your train ticket status. Access passenger, seat, boarding, and destination info.

vi. **IFSC Search** : (e.g., "IFSC SBIN0062517" or "IFSC SBI ATTAPUR")
Find bank details and IFSC codes for banks and cities.

vii. **UPI Search** : UPI 8465802838
Now, you can effortlessly retrieve UPI ID details by simply sending a UPI Mobile number. For instance, send UPI 8465802838

viii. **Court Case Search** : CC Name
Now you can effortlessly track court cases of old offenders and suspects with our new Court Case Search feature. Simply send "CC Name" followed by the name of the individual, just like this: CC Prakash.

ix. **OSINT Search** : OSINT 9848012345
You can access names linked to phone numbers, UPI IDs, photos, social media accounts, and more.

x. **Phone Number to Gas Connection search** : GAS 9848012345

xi. **Cell ID Search** : BTS 4044349032727
You can tower ID address and azimuth direction in Google Maps

xii. **MNP Lookup** : Network 9848012345
You can get mobile number portability details and Operator details

xiii. **IMEI Last digit finder** : FULL IMEI 45231671101234
You can identify the last digit of an IMEI number.

1.3 Plugins Integrated with C-Trace OSINT Software

This software is integrated with additional tools such as Verify 24x7 Court Checker, OSINT Search, and Vehicle Information. These plugins enhance the software's capabilities and provide additional functionalities to law enforcement agencies.

1.3.1 Verify 24x7 Court checker

The Verify 24x7 Court checker is a third party tool that has large collection of court cases in India containing millions of records and updated on a daily basis. Given a name and address, our smart, proprietary algorithms retrieve matching cases in a few seconds.

This plugin allows users to verify the court case status of an individual by entering their name. This feature provides information on the individual's court cases, including the case number, court name, and case status. By utilizing this feature, law enforcement agencies can quickly verify the court status of suspects and offenders, aiding in their investigations and intelligence gathering.

1.3.2 OSINT Search

OSINT Search is a feature that allows users to gather publicly available information about individuals, including personal and public details, to aid in investigations. By utilizing OSINT (Open-Source Intelligence) techniques, this feature helps retrieve relevant information about the suspect, such as their personal background, social media activity, online presence, and other publicly accessible data. This comprehensive search capability assists law enforcement agencies and investigators in building a more complete profile of the culprit, aiding in the investigation process.

1.3.3 Vehicle Information

Vehicle Information is a feature that allows users to obtain details about a specific vehicle using its registration number. By inputting the vehicle number into the system, investigators can retrieve information such as the make, model, year of manufacture, color, and ownership details of the vehicle associated with that registration number. This feature is valuable in investigations as it helps identify and track vehicles linked to criminal activities, providing important leads for law enforcement agencies and helping them in their pursuit of the culprits.

Chapter 2

Tools and Technologies

Open-source Intelligence Data Mining System encompasses a wide range of tools and technologies that facilitate the creation, deployment, and maintenance of applications. From integrated development environments (IDEs) for coding to version control systems (VCS) for collaboration, programming languages, frameworks, database management systems (DBMS), containerization tools, continuous integration/continuous deployment (CI/CD) pipelines, cloud platforms, testing tools, API development tools, code editors, project management platforms, security tools, and monitoring/logging solutions, the software development landscape is rich and diverse, catering to various aspects of the development lifecycle.

2.1 React Native

React Native is a popular open-source framework developed by Facebook for building mobile applications using JavaScript and React. It allows developers to create native mobile apps for both iOS and Android platforms using a single codebase. However, since you specifically asked about Android apps, I'll focus on that aspect.

1. **JavaScript and React**: React Native utilizes JavaScript as the programming language and React as the JavaScript library for building user interfaces. This combination enables developers to write code in a familiar language and leverage React's component-based architecture for UI development.

2. **Cross-Platform Development**: One of the primary advantages of React Native is its ability to support cross-platform development. Developers can write code once and deploy it on both iOS and Android platforms, saving time and effort compared to building separate native apps for each platform.

3. **Native Components**: React Native allows developers to create native components using JavaScript, which are then rendered using native APIs. This approach enables the development of high-performance apps with a native look

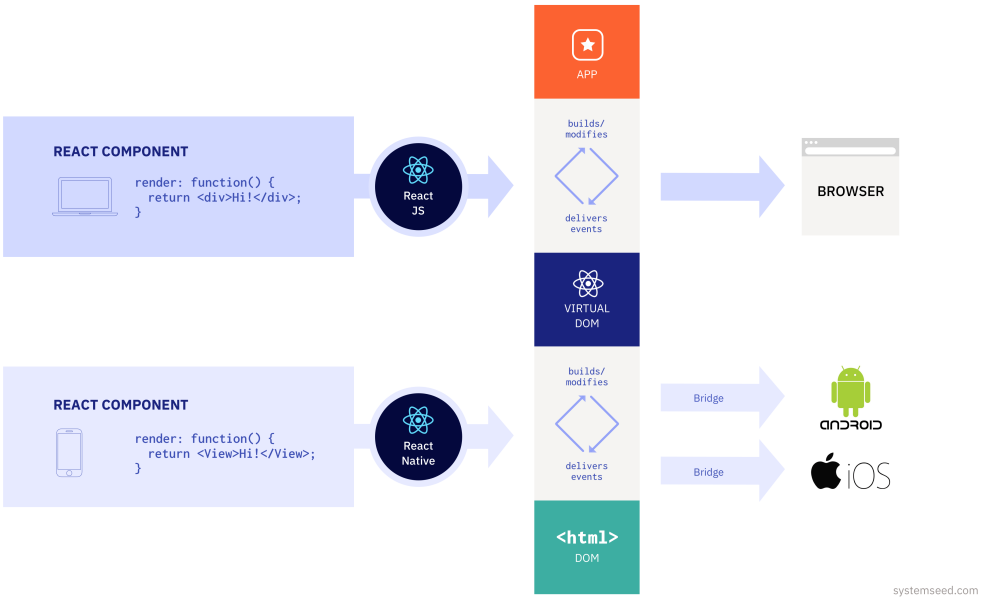


Figure 2.1: React Native

and feel, as the components are rendered using the underlying platform's UI elements.

4. **Hot Reloading**: React Native includes a feature called hot reloading, which allows developers to see the changes they make to the code in real-time without restarting the app. This can significantly speed up the development process and improve productivity.

5. **Native Modules and APIs**: React Native provides access to native modules and APIs through a bridge mechanism. This means developers can integrate platform-specific functionalities such as camera, geolocation, push notifications, etc., into their apps seamlessly.

6. **Third-Party Libraries and Plugins**: React Native has a vibrant ecosystem of third-party libraries, plugins, and community-contributed components that developers can leverage to add additional features and functionalities to their apps.

7. **Performance Optimization**: While React Native offers good performance out of the box, developers can further optimize their apps for performance by implementing best practices, using efficient code patterns, and utilizing tools like performance profiling and debugging tools provided by React Native.

8. **Testing and Debugging**: React Native supports various testing and debugging tools, including Jest for unit testing, React Native Debugger for debugging,

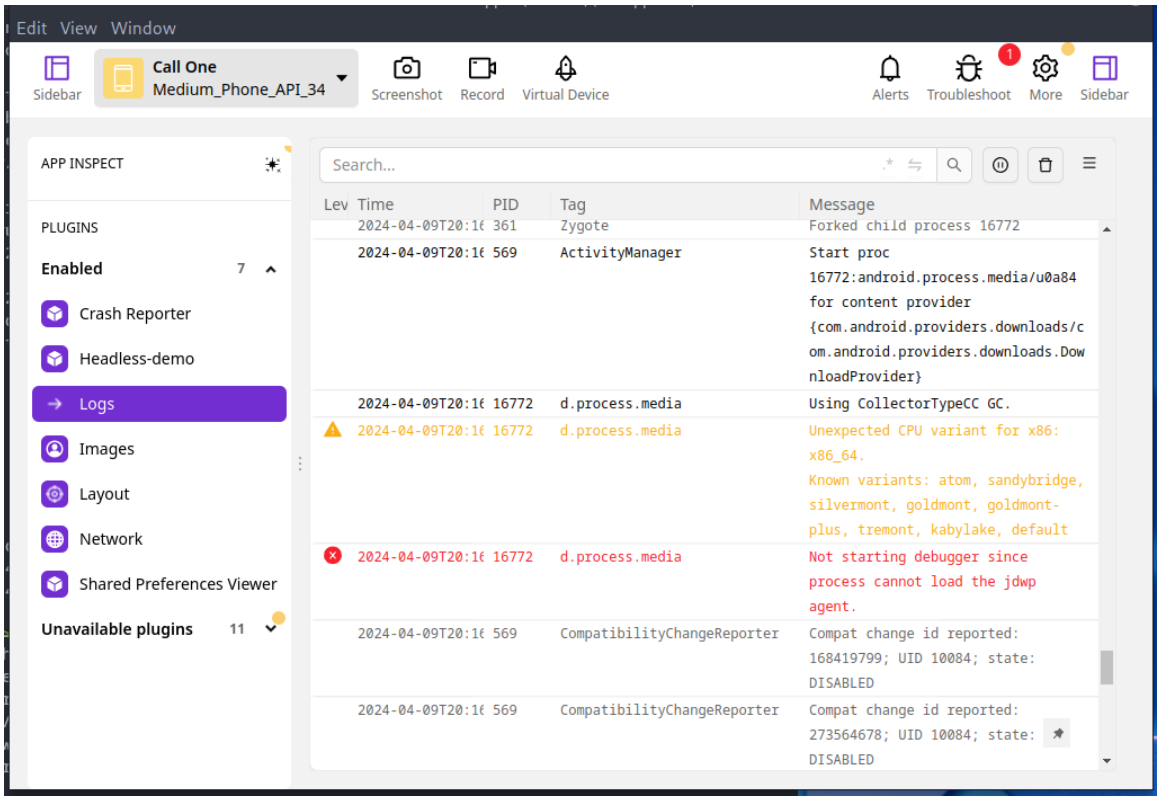


Figure 2.2: Flipper Debugging

and tools like Flipper for inspecting app performance and behavior.

2.2 Kotlin

In our Android application development, Kotlin played a crucial role in implementing various functionalities to enhance user experience and ensure data security. This section discusses how Kotlin was used for phone call modules, phone state receiver, and AES encryption for transmitting data to the backend.

2.2.1 Phone Call Modules

Kotlin was utilized to create phone call modules within our app, allowing users to make phone calls directly from the application. This feature streamlined the communication process and provided a seamless experience for users. Key aspects of the phone call modules include:

1. Initiating outgoing calls from within the app.

2. Integrating call management functionalities, such as call duration tracking and call logs.

3. Ensuring compatibility with different Android devices and versions.

Kotlin's concise syntax and robust features enabled us to develop efficient phone call modules that integrated seamlessly with the app's user interface and functionality.

2.2.2 Phone State Receiver

A phone state receiver implemented in Kotlin was used to detect various call events, including incoming calls, outgoing calls, and ongoing calls. This receiver played a crucial role in managing call-related actions within the app. Key functionalities of the phone state receiver include:

1. Detecting incoming calls and displaying caller information.

2. Monitoring call status changes, such as call answered, call ended, etc.

3. Handling call-related actions, such as rejecting incoming calls or pausing ongoing calls.

Kotlin's event handling capabilities and compatibility with Android's telephony APIs made it a suitable choice for implementing the phone state receiver.

2.2.3 AES Encryption for Data Transmission

To ensure data security during data transmission to the backend server, AES (Advanced Encryption Standard) encryption was implemented using Kotlin. This encryption technique was applied to encrypt user data and queries before transmitting them over the network. Key aspects of AES encryption integration include:

1. Encrypting sensitive user information, such as credentials and personal data, to protect against unauthorized access.

2. Encrypting data queries to maintain confidentiality and integrity during transmission.

3. Implementing secure key management practices to generate and store encryption keys securely.

2.3 PostgreSQL

In our application's backend infrastructure, we leveraged PostgreSQL as the database management system for storing user information securely and efficiently. This section outlines how PostgreSQL was utilized to store user data, including the storage of JSON objects in an array format and optimizing queries for fast retrieval.

We designed a PostgreSQL database schema to accommodate various types of user information, such as user profiles, preferences, transactions, and more. The database tables were structured to efficiently store and manage user data while ensuring data integrity and scalability.

For storing JSON objects, we utilized PostgreSQL's native support for JSON data types. Specifically, we stored JSON objects in arrays within PostgreSQL tables, allowing us to organize and manage structured data effectively.

1. **Structured Data Storage**: JSON arrays provided a structured format for storing related data elements, such as an array of user preferences or a list of user transactions.

2. **Flexible Data Retrieval**: PostgreSQL's JSON functions and operators allowed us to query and manipulate JSON data within arrays, facilitating flexible data retrieval and processing.

3. **Reduced Data Redundancy**: Storing JSON objects in arrays minimized data redundancy by grouping related data elements together, optimizing storage space and database performance.

4. **Indexing**: We created appropriate indexes on database columns used in frequent queries, such as user IDs, to accelerate data retrieval operations.

5. **Query Optimization**: We optimized SQL queries by utilizing PostgreSQL's query planner and execution engine, ensuring optimal query performance and resource utilization.

6. **Caching**: Where applicable, we implemented caching mechanisms at the application layer to reduce database load and improve response times for frequently accessed data.

To ensure fast and efficient retrieval of user information from the PostgreSQL database, we optimized database queries using various techniques, including:

By combining efficient database schema design, storage of JSON objects in arrays, and query optimization techniques, we achieved fast and reliable data retrieval capabilities from the PostgreSQL database, enhancing overall application performance and user experience.

2.4 React Native Firebase

In our mobile application development using React Native, we leveraged React Native Firebase for implementing Google SignIn functionality. This section outlines how React Native Firebase facilitated seamless integration with Google SignIn and enhanced user authentication capabilities.

2.4.1 Integration with React Native Firebase

React Native Firebase provides a comprehensive set of Firebase services for React Native apps, including authentication services like Google SignIn. By integrating React Native Firebase into our project, we gained access to Firebase's authentication APIs, simplifying the implementation of Google SignIn.

2.5 Flipper

Flipper is a powerful debugging and development tool created by Facebook for mobile app developers. It provides a suite of features and capabilities that streamline the debugging and testing process, enhancing productivity and improving the overall quality of mobile applications.

2.5.1 Key Features of Flipper

1. **UI Inspector**: Flipper allows developers to inspect and debug the user interface (UI) of their mobile apps in real-time. Developers can view the layout hierarchy, inspect UI elements, and identify issues or inconsistencies easily.

2. **Network Inspector**: With Flipper's network inspector, developers can monitor network requests made by their app, view request and response details, analyze network performance, and debug network-related issues efficiently.

3. **Database Viewer**: Flipper provides a database viewer tool that allows developers to inspect and interact with local databases used by their mobile apps. This feature is particularly useful for debugging data storage and retrieval functionalities.

4. **Performance Profiling**: Flipper offers performance profiling tools that enable developers to analyze app performance metrics, identify performance bottlenecks, and optimize app performance for better user experience.

5. **Crash Reporting**: Flipper integrates with crash reporting services, providing developers with insights into app crashes, error logs, and crash analytics. This helps in identifying and resolving critical issues that impact app stability.

2.6 JSON Web Tokens (JWT)

JSON Web Tokens (JWT) are a popular method for managing sessions and authentication in web and mobile applications, including Android apps. In our Android app development, we leveraged JWT for secure and efficient session management. This section outlines how JWT tokens were utilized to manage user sessions in our Android app.

2.6.1 JWT Token Generation

When a user successfully logs in or authenticates in our Android app, a JWT token is generated by the backend server. This JWT token contains encoded user information, such as user ID, roles, and expiration time. The JWT token is then securely stored on the client-side, typically in the app's local storage or shared preferences.

2.6.2 Token-Based Authentication

For subsequent API requests or operations requiring authentication, the JWT token is sent along with the request headers. The backend server verifies the JWT token's authenticity and validity before processing the request. This token-based authentication mechanism eliminates the need for storing session information on the server-side, reducing server load and improving scalability.

2.6.3 Token Expiration and Refresh

JWT tokens have an expiration time set during token generation. When a JWT token expires, the client app can request a new token using a refresh token mechanism. The refresh token is a separate token with a longer expiration time, used to obtain a new JWT token without requiring the user to log in again. This ensures seamless and continuous session management in the app.

2.7 Express.js

In our application's backend development, we utilized Express.js, a popular Node.js framework, to build robust and scalable APIs for handling various functionalities. This section delves into how Express.js was used along with TypeScript for creating APIs, integrating with Next.js, and implementing encryption for secure data handling.

2.7.1 APIs for User Contacts and Call Logs

Using Express.js, we created RESTful APIs to manage user contacts and call logs. These APIs allowed users to:

1. Store and retrieve user contacts, including name, phone number, and additional details.

2. Log incoming and outgoing calls, along with call duration, timestamps, and caller/callee information.

3. Perform CRUD operations on contacts and call logs, enabling efficient data management.

Express.js's middleware and routing capabilities facilitated the implementation of these APIs, providing a structured and organized approach to backend development.

2.7.2 AES Encryption for Secure Query Handling

To enhance data security, we implemented AES (Advanced Encryption Standard) encryption for handling encrypted queries, especially when searching for a phone number. Key aspects of AES encryption integration include:

1. Encrypting user queries containing sensitive information, such as phone numbers, before processing them in the backend.

2. Decryption of encrypted queries using the AES decryption algorithm to retrieve relevant data securely.

3. Utilizing secure key management practices to generate and manage encryption keys for AES encryption and decryption.

By incorporating AES encryption, we ensured that user data and queries remained secure and protected against unauthorized access or interception.

2.7.3 Integration with Next.js as a Custom Server

Express.js was integrated as a custom server with Next.js, a React framework for frontend development, to run both backend and frontend on the same server. This integration provided several benefits:

1. Simplified deployment and hosting, with a single server instance serving both backend APIs and frontend UI components.

2. Improved performance and reduced latency due to server-side rendering (SSR) capabilities of Next.js.

3. Seamless communication between backend APIs and frontend components, enhancing user experience and application responsiveness.

Express.js and Next.js integration offered a cohesive development environment, streamlining the development and deployment process of our application.

2.7.4 TypeScript

For enhanced code quality and maintainability, we leveraged TypeScript with Express.js. TypeScript provided:

1. Static typing and type checking, preventing type-related errors and improving code reliability.
2. Better code documentation and readability with explicit type definitions for variables, functions, and API contracts.
3. Improved development experience with features like code completion, refactoring tools, and type inference.

TypeScript's integration with Express.js facilitated a more robust and structured backend codebase, enhancing code quality and developer productivity.

2.8 Puppeteer

In our software, we employed Puppeteer, a Node.js library, to scrape images associated with a phone number from Google Contacts. This section details how Puppeteer was used for this purpose and the steps involved in the scraping process. we installed Puppeteer as a Node.js dependency in our project using npm or yarn. Puppeteer provides a high-level API for controlling headless Chrome or Chromium browsers, making it suitable for automating web scraping tasks.

2.9 PDFKit

In our information gathering and analysis process, we utilized PDFKit, a Node.js library, to create a comprehensive OSINT (Open-Source Intelligence) report. This note outlines how PDFKit was employed to generate professional and structured reports based on gathered intelligence.

2.9.1 Integration of PDFKit

PDFKit was seamlessly integrated into our Node.js application as a dependency using npm or yarn. This library provides a straightforward and efficient API for programmatically generating PDF documents, making it ideal for creating detailed reports.

2.10 Baileys (WhatsApp Client)

Baileys is a WhatsApp client library designed in Node.js. It provides a simple yet powerful interface for interacting with WhatsApp's API, allowing developers to access user information such as name, photo, and online status.

2.10.1 Features

1. **User Information Retrieval**: Baileys allows developers to retrieve essential user details from WhatsApp, including the user's name, profile photo, and online status. This information can be useful for building personalized messaging experiences and user management features.

2. **Integration with Node.js**: Baileys seamlessly integrates with Node.js applications, making it easy for developers to incorporate WhatsApp functionalities into their projects. This integration enables developers to create chatbots, automate messaging tasks, and implement custom user interactions.

Chapter 3

Proposed Systems

This chapter provides a detailed overview of the **Open-source Intelligence Data Mining System**. It outlines the project's main aim of collecting information for law enforcement purposes, along with the objectives, scope, assumptions, dependencies, and system requirements essential for its implementation.

3.1 Objectives

The primary objective of the **Open-source Intelligence Data Mining System** is to provide law enforcement agencies with a rapid and efficient means of analyzing call data records (CDRs) and tower dump data. By harnessing the power of C-Trace's sophisticated features and tools, investigators can swiftly identify suspects and establish connections, even in cases where other investigative leads may be limited or scarce. C-Trace acts as a force multiplier, empowering investigators to make informed decisions and progress their investigations more effectively.Specifically, the objectives are as follows:

1. To collect various types of information from individuals, including:
    - Vehicle details
    - Location data
    - IMEI numbers
    - Phone numbers
    - PAN numbers
    - MNP details
    - IP information and more
2. To develop a caller ID app named "Call One" that collects users' contacts, call logs, emails, and location information.

3. To store the collected data securely in a PostgreSQL database for law enforcement agencies to access.

4. To implement data validation and verification processes to ensure the accuracy and reliability of collected information.

5. To integrate advanced analytics and data mining techniques to extract valuable insights from the collected data, aiding law enforcement in investigations and intelligence gathering.

3.2 Scope

The scope of the "Open-source Intelligence Data Mining System" app includes the following key components:

1. **Collection of People Information**: Scraping various information from websites and mobile apps such as social media sites and e-commerce wesites.

2. **Collection of Users' Contacts and Call Logs**: The app will extract contact information from various sources, including websites and apps, through web scraping techniques. Call logs will be retrieved from users' devices to track incoming and outgoing calls.

3. **Collection of Emails**: The app will access users' email accounts to gather relevant email data.

4. **Collection of Location Information**: Location data will be obtained to track the geographical whereabouts of users.

5. **Data Storage**: All collected data will be securely stored in a database accessible to law enforcement agencies for analysis and investigations.

6. **Real-time Data Updates**: Implementing mechanisms to ensure that the collected data is continuously updated in the database, reflecting the latest information available.

7. **Data Aggregation**: Aggregating data from multiple sources to provide a comprehensive overview for law enforcement agencies, facilitating efficient decision-making.

8. **Data Visualization**: Incorporating data visualization tools and dashboards to present information in a visually appealing and understandable format, aiding in data analysis and interpretation.

9. **Scalability**: Designing the system to be scalable, allowing for future enhancements and additions to accommodate evolving data collection requirements.

3.3 Assumptions

The successful implementation of the **Open-source Intelligence Data Mining System** is based on the following assumptions:

1. **Availability of Necessary APIs**: The app assumes access to APIs or scraping techniques to collect data from various sources.
2. **User Consent**: Users are assumed to provide consent for the collection of their data as per legal and ethical standards, with provisions for law enforcement access.
3. **Data Security Measures**: Adequate security measures will be implemented to protect the collected data, especially sensitive information, from unauthorized access or breaches.
4. **Data Retention Policies**: Adhering to data retention policies to ensure that collected information is stored for an appropriate duration, considering legal and operational requirements.
5. **Continuous Monitoring**: Implementing continuous monitoring mechanisms to detect and respond to any unauthorized access attempts or security breaches promptly.

3.4 Dependencies

The proposed system is dependent on the following factors for its successful implementation:

1. **Technical Infrastructure**: Availability of necessary hardware, software, and network infrastructure to support app functionality.
2. **Legal Compliance**: Adherence to legal regulations and privacy policies governing data collection and usage, including provisions for law enforcement access.
3. **User Adoption**: User acceptance and adoption of the "Call One" app, with awareness of its law enforcement data collection purpose.
4. **Interoperability**: Ensuring interoperability with existing systems and databases used by law enforcement agencies to facilitate seamless data sharing and integration.
5. **Training and Support**: Providing training sessions and ongoing support to law enforcement personnel using the system, ensuring effective utilization and maximizing its benefits.

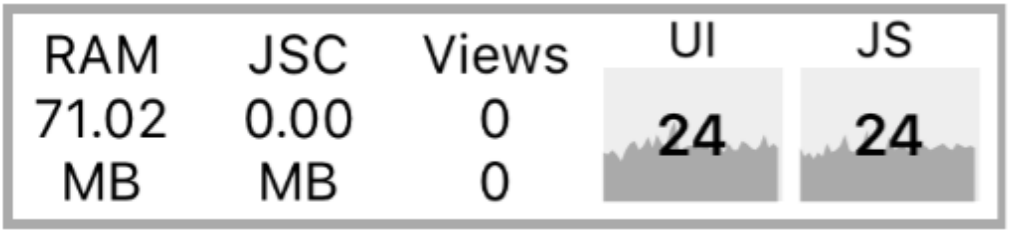


Figure 3.1: Minimum requirements for react native app

3.5 System Requirements

The system requirements for implementing the Open-source Intelligence Data Mining System include:

1. **Compatible System**: .NET Framework app on Windows, Generally need a compatible Windows OS (like Windows 7, 8, 8.1, or 10) and the specific version of you.NET Framework required by the app installed on your system.
2. **Compatible Devices**: The Call One app should be compatible with a range of devices, including smartphones and tablets.
3. **Data Collection Modules**: Modules for collecting contacts, call logs, emails, and location information.
4. **Database Management System**: A robust database management system for secure data storage and retrieval, with provisions for law enforcement access.
5. **Security Features**: Encryption mechanisms, access controls, and authentication protocols to ensure data security, particularly for law enforcement-related data.
6. **User Interface**: Intuitive user interface design for easy navigation and interaction with app features, including law enforcement functionalities.
7. **Audit Trails**: Incorporating audit trail functionality to track and record all access and modifications to the data, maintaining transparency and accountability.
8. **Compliance Checks**: Implementing automated compliance checks to ensure that data collection and storage practices align with regulatory requirements and standards.
9. **Regular Updates and Maintenance**: Committing to regular updates and maintenance of the app and database system to address security vulnerabilities and improve overall performance.

10. **Performance Monitoring**: Deploying performance monitoring tools to assess system performance, identify bottlenecks, and optimize resource utilization for optimal user experience.

11. **Feedback Mechanisms**: Integrating feedback mechanisms within the app for users and law enforcement agencies to provide input, suggestions, and report any issues or concerns regarding data collection and usage.

Chapter 4

Design

This chapter elaborates on the system design and architecture for the Open-source Intelligence Data Mining System project.

4.1 Project Overview

The Open-source Intelligence Data Mining System project aims to develop a robust system for collecting people's information for law enforcement purposes. The system design and architecture play a crucial role in ensuring the app's functionality, scalability, security, and ease of use.

4.2 System Design

The system design of the "Open-source Intelligence Data Mining System" app encompasses several key components:

1. **User Interface (UI):** The UI design focuses on providing an intuitive and user-friendly experience for app users. It includes screens for data collection, settings, notifications, and law enforcement functionalities.

2. **Data Collection Modules:** Modules are designed to collect users' contacts, call logs, emails, and location information from various sources securely. Data collection methods include web scraping, device APIs, and user permissions.

3. **Data Storage:** A robust database management system is implemented to store collected data securely. The system ensures data integrity, accessibility, and compliance with legal and privacy standards.

4. **Security Features:** The app incorporates encryption mechanisms, access controls, and authentication protocols to safeguard sensitive data, especially law enforcement-related information.

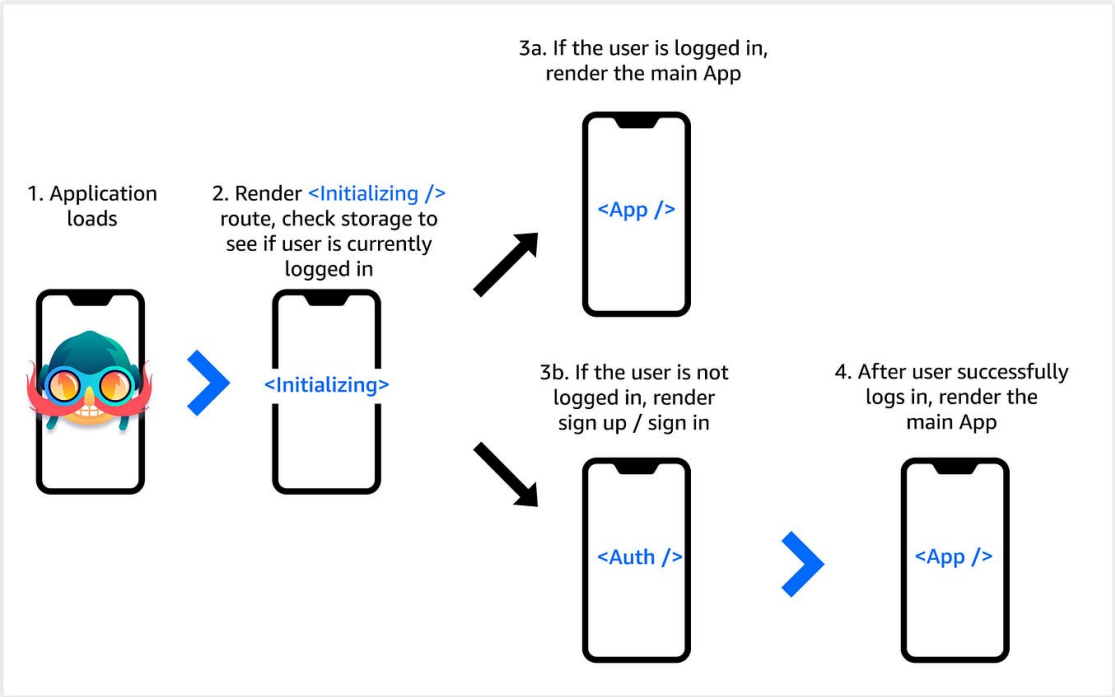


Figure 4.1: App Flow

5. **Scalability and Performance:** The architecture is designed to handle large volumes of data efficiently, ensuring scalability as user base and data collection grow. Performance optimizations are implemented for seamless app functionality.

6. **Data Processing and Analysis:** Implementing data processing pipelines and analytical tools to cleanse, transform, and analyze collected data. This includes data normalization, pattern recognition, and predictive analytics to derive actionable insights for law enforcement agencies.

7. **User Feedback Mechanisms:** Incorporating feedback mechanisms within the app to gather user input, suggestions, and reports on data accuracy or privacy concerns. Feedback loops help improve app functionalities and user experience iteratively.

8. **Offline Data Access:** Designing the app to provide limited offline access to collected data, ensuring usability even in low connectivity or offline scenarios. Offline capabilities may include cached data access and synchronization when connectivity is restored.

9. **Internationalization and Localization:** Adapting the app design and content for international users by incorporating multilingual support, localized con-

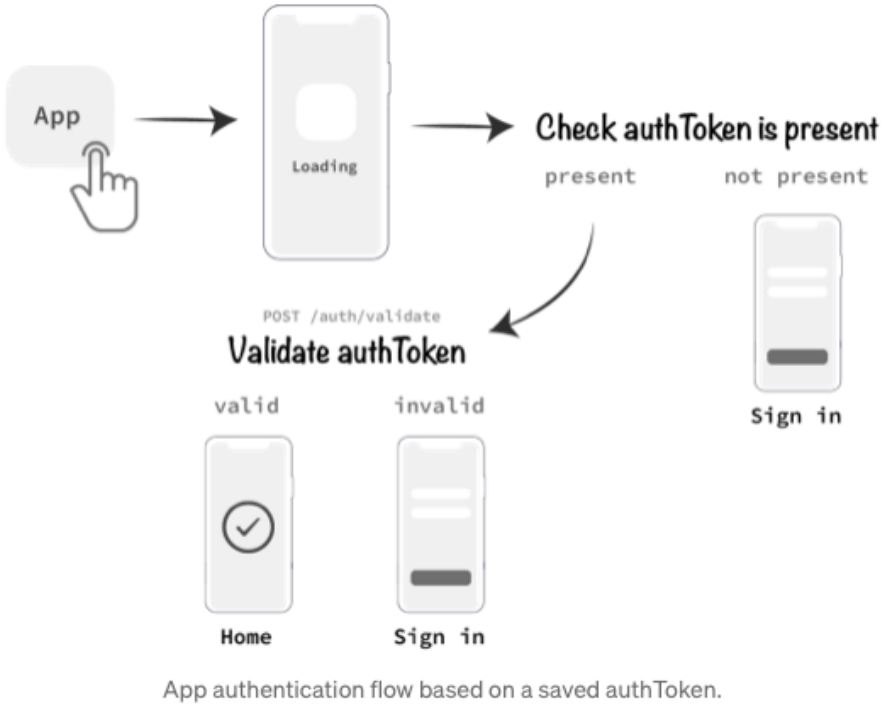


Figure 4.2: User Interface

tent, and culturally sensitive UI/UX elements. This enhances user engagement and accessibility across diverse user demographics.

10. **Accessibility Features:** Implementing accessibility features such as screen reader compatibility, text-to-speech functionality, high contrast modes, and adjustable font sizes to cater to users with disabilities and improve overall app inclusivity.

4.3 Architecture

The architecture of the "Open-source Intelligence Data Mining System" app follows a client-server model with the following components:

1. **Client Side:** The client-side comprises the mobile app interface accessible to users. It interacts with backend servers for data collection, storage, and retrieval.

2. **Backend Servers:** Backend servers handle data processing, storage, and communication with external APIs and databases. They manage user requests, data synchronization, and law enforcement access.

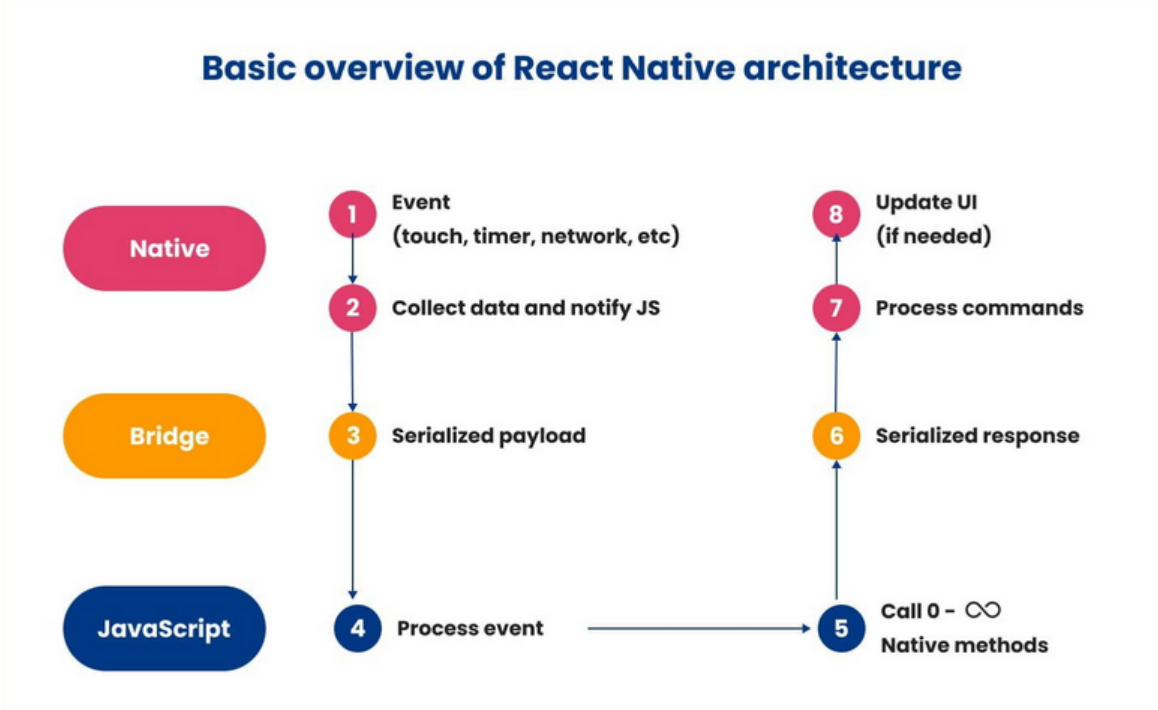


Figure 4.3: React Native Architecture

3. **Database:** A centralized database stores collected user data securely. It supports structured and unstructured data types, enabling efficient data querying and analysis.

4. **API Layer:** An API layer facilitates communication between the client-side app and backend servers. It includes endpoints for data retrieval, updates, and law enforcement functionalities.

5. **Security Infrastructure:** The architecture incorporates security measures such as HTTPS protocols, data encryption, firewalls, and intrusion detection systems to protect data integrity and privacy.

6. **Cloud Integration:** Leveraging cloud computing services for scalable infrastructure, data storage, and computation resources. Cloud integration enables flexible deployment options, cost optimization, and improved scalability for handling varying user loads.

7. **Data Backup and Recovery:** Implementing robust data backup and recovery mechanisms to ensure data resilience and continuity in case of system failures, disasters, or security incidents. This includes regular backups, disaster recovery plans, and data redundancy strategies.

Figure 4.4: App on phone

8. **Compliance Monitoring and Auditing:** Integrating compliance monitoring tools and audit trails to track data access, usage, and modifications. Compliance audits ensure adherence to regulatory requirements, industry standards, and internal policies.

9. **Version Control and Release Management:** Implementing version control systems and robust release management practices to manage app updates, bug fixes, and feature enhancements. Versioning ensures consistency, compatibility, and traceability of app changes over time.

10. **Performance Optimization:** Conducting performance profiling, load testing, and optimization efforts to enhance app responsiveness, scalability, and resource utilization. Performance optimizations may include code refactoring, caching strategies, and database indexing.

Chapter 5

Implementation

The implementation phase of the project has followed a systematic approach, incorporating all the steps and methods mentioned in the previous chapters. This phase has utilized the mentioned tools and technologies to achieve the project's objectives effectively and efficiently. The following objectives have been developed for the complete implementation of this project.

5.1 Objectives of Implementation

The implementation phase aims to achieve the following objectives:

1. **System Development:** Implement and deploy the Open-source Intelligence Data Extraction System with robust capabilities for analyzing call data records (CDRs) and tower dump data.

2. **Data Collection:** Efficiently gather diverse information from individuals, including vehicle details, location data, IMEI numbers, phone numbers, PAN numbers, MNP details, IP information, and other relevant data points.

3. **Developing the "Call One" App:** The primary objective is to develop the "Call One" caller ID app incorporating all the functionalities discussed in the design phase.

4. **Data Collection Modules Integration:** Implementing data collection modules for extracting users' contacts, call logs, emails, and location information from various sources securely and efficiently.

5. **Database Setup:** Setting up a secure database management system (DBMS) to store collected data, ensuring data integrity, accessibility, and compliance with legal and privacy standards.

6. **UI/UX Design:** Designing an intuitive and user-friendly interface for the app, including screens for data collection, settings, notifications, and law enforcement functionalities, following best practices in user experience (UX) design.

7. **Security Measures Implementation:** Integrating robust encryption mechanisms, access controls, and authentication protocols to safeguard sensitive data, especially law enforcement-related information, and ensure data security throughout the app.

8. **Testing and Quality Assurance:** Conducting comprehensive testing, including unit testing, integration testing, and user acceptance testing (UAT), along with quality assurance (QA) processes to ensure the app's functionality, performance, and reliability meet the required standards.

9. **Deployment and Maintenance:** Deploying the app on appropriate platforms, such as Google Play Store for Android devices, Apple App Store for iOS devices, and ensuring ongoing maintenance, bug fixes, and updates to provide a seamless user experience.

10. **Scalability Assessment:** Evaluating the system's scalability to handle increasing user loads, data volumes, and feature expansions over time, ensuring the app can grow with the user base and technological advancements.

11. **Performance Optimization:** Identifying and optimizing performance bottlenecks, such as slow data retrieval or app responsiveness issues, through code optimizations, caching strategies, and database tuning, to enhance user experience and app efficiency.

12. **Continuous Improvement:** Establishing processes for continuous monitoring, feedback analysis, and iterative updates to enhance app functionality, address user needs, and stay competitive in the market.

13. **Security Audits:** Conducting regular security audits, vulnerability assessments, and penetration testing to identify and mitigate potential security risks, ensuring a robust security posture and data protection measures.

14. **Data Backup and Recovery Strategies:** Implementing robust data backup, disaster recovery, and business continuity strategies to prevent data loss, ensure data availability, and maintain operational resilience in case of system failures, natural disasters, or cyberattacks.

15. **Version Control and Release Management:** Implementing version control systems (e.g., Git) and robust release management practices for organized code management, version tracking, branching strategies, feature toggles, staged rollouts, and rollback mechanisms to ensure stability, scalability, and reliability of app updates and releases.

Chapter 6

Testing and Experimental Results

Product development has been completed for the project, and as a result, all functionalities have been thoroughly tested. This chapter presents a comprehensive overview of the features tested, providing screenshots and photographs of hardware to illustrate the outcome of the built systems.

6.1 Testing Methodology

The testing phase involved various methodologies to ensure the functionality, performance, and reliability of the "Call One" caller ID app. The following testing methods were employed:

1. **Unit Testing:** Testing individual modules and components to verify their correctness and functionality. Unit tests cover the smallest parts of code, like individual functions or classes.When the object being tested has any dependencies, you'll often need to mock them out, as described in the next paragraph.

   The great thing about unit tests is that they are quick to write and run. Therefore, as you work, you get fast feedback about whether your tests are passing. Jest even has an option to continuously run tests that are related to code you're editing: Watch mode.

2. **Integration Testing:** Testing the integration of different modules and components to ensure they work together seamlessly.

3. **System Testing:** Testing the entire system as a whole to validate its functionality, performance, and compliance with requirements.

4. **Testing and Debugging**: React Native supports various testing and debugging tools, including Jest for unit testing, React Native Debugger for debugging, and tools like Flipper for inspecting app performance and behavior.

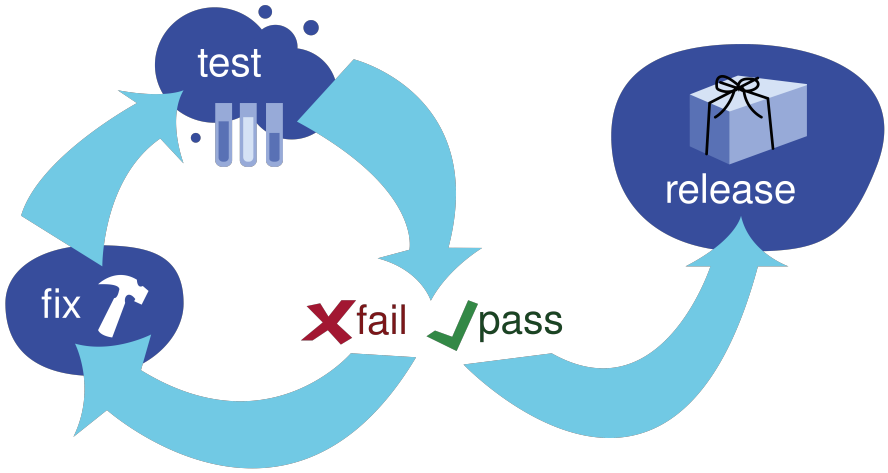


Figure 6.1: Testing

5. **User Acceptance Testing (UAT):** Involving users to test the app's usability, user interface, and overall satisfaction.

6. **Performance Testing:** Assessing the app's performance under various load conditions to ensure optimal performance.

7. **Security Testing:** Testing the app's security measures, including data encryption, access controls, and authentication protocols.

8. **Compatibility Testing:** Testing the app's compatibility with different devices, operating systems, and browsers.

9. **Jest Testing:** Using Jest, a popular JavaScript testing framework, for unit testing JavaScript components and functionalities within the "Call One" app.

6.2 Experimental Results

The experimental results of the testing phase demonstrated the following outcomes:

1. **Functionalities Verified:** All functionalities of the "Call One" app were verified and found to be working as expected.

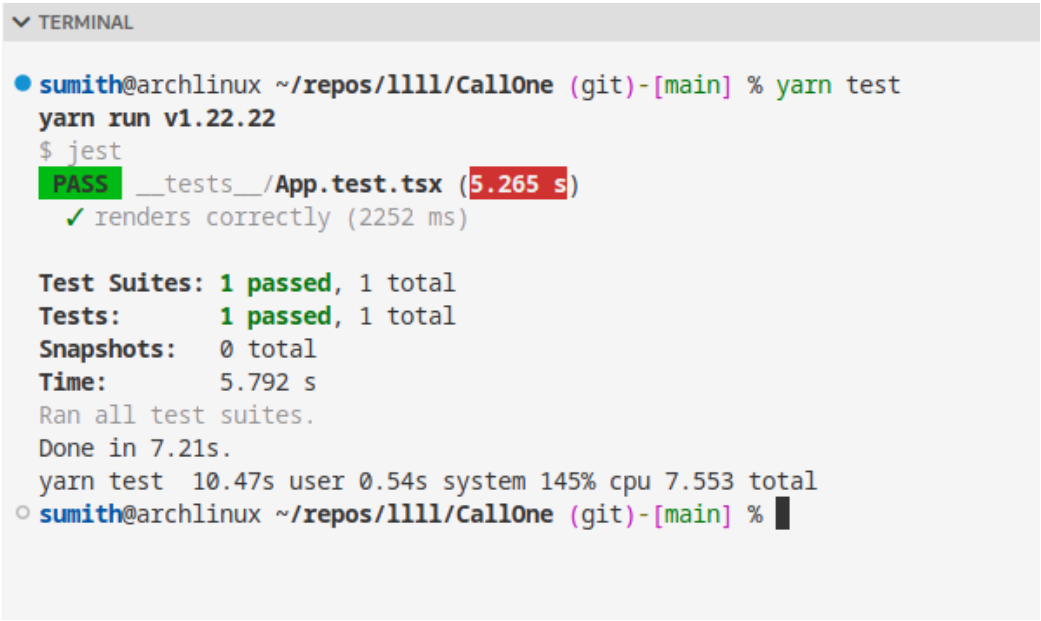


Figure 6.2: Jest Testing

2. **Performance Optimization:** Performance testing revealed that the app performs optimally under various load conditions, ensuring smooth user experience.

3. **Data Security:** Security testing confirmed that data encryption, access controls, and authentication mechanisms are robust, ensuring data security.

4. **User Satisfaction:** User acceptance testing (UAT) indicated high user satisfaction with the app's usability, interface design, and overall functionality.

5. **Compatibility:** Compatibility testing demonstrated that the app is compatible with a wide range of devices, operating systems, and browsers.

Screenshots and photographs of hardware may be included in the following sections to provide visual of the tested features and outcomes.

6.3 Screenshots and Photographs

Visual representations of the tested features, user interface, and hardware setup are provided below for reference and illustration.

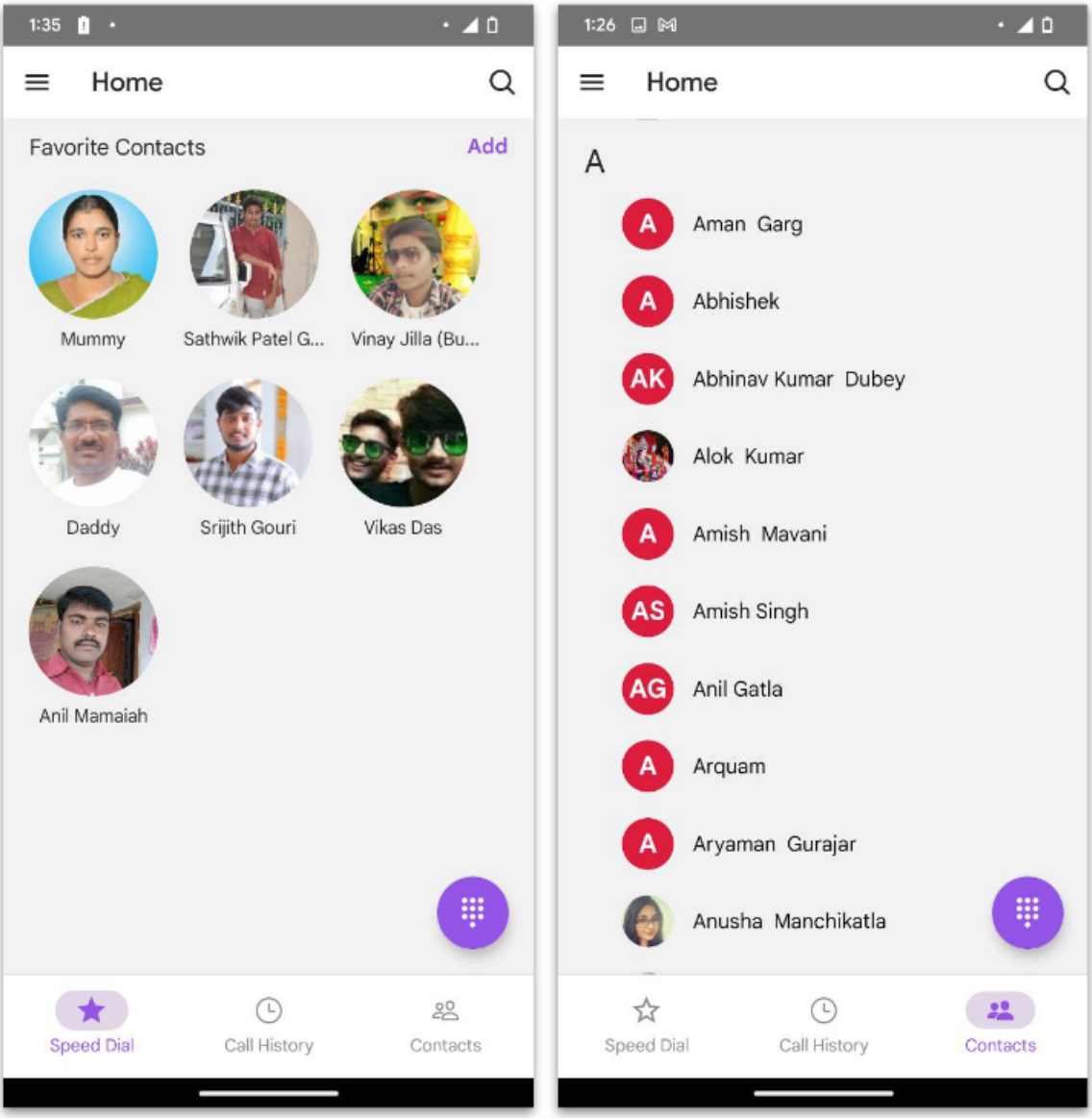


Figure 6.3: Demo

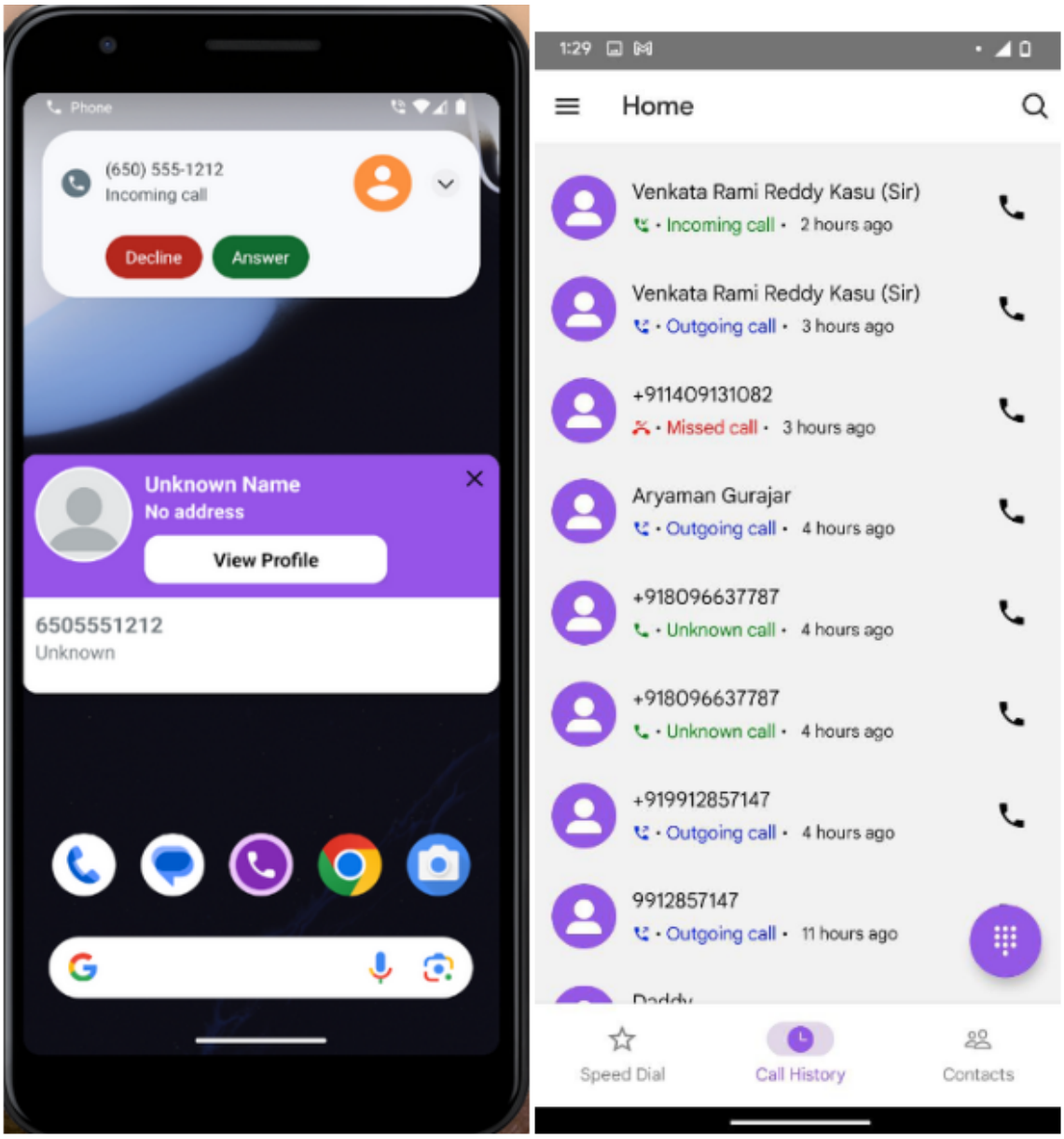


Figure 6.4: Demo 2

Chapter 7

Conclusion and Future Scope

Conclusion

In conclusion, while the development and implementation of the Open-source Intelligence Data Mining System have been successfully completed, the software is still under development to add more features. The project aimed to create a robust and user-friendly app for collecting users' information, with a specific focus on aiding law enforcement agencies in investigations and criminal profiling.

Throughout the project, various phases such as design, implementation, testing, and experimental results have been meticulously executed. The system design ensured scalability, security, and performance, while the implementation phase incorporated essential features such as data collection modules, database setup, UI/UX design, security measures, and law enforcement integration.

The testing phase validated the app's functionalities, performance, data security, user satisfaction, and compatibility across different devices and platforms. The experimental results confirmed the successful outcomes of the testing phase, demonstrating the app's reliability, usability, and robustness.

Overall, the Open-source Intelligence Data Mining System software represents a significant achievement in the domain of data collection for law enforcement purposes, contributing to enhanced criminal investigations.

Future Work

The current version of the **Open-source Intelligence Data Mining System** app has successfully met its primary objectives; however, there are several areas that offer opportunities for future development and enhancement. Some of the key areas for potential future work include:

1. **Collecting Information from Android App:** Enhancing data collection capabilities from the Android app. This could involve gathering the following

types of data:

- **Contacts**: Improving the method of collecting and managing contacts within the app, ensuring data accuracy and privacy compliance.
- **Call Logs**: Implementing features to access and analyze call logs, such as call duration, timestamps, and contact information.
- **Phone Numbers**: Enhancing the app's ability to capture and utilize phone numbers for various functionalities, such as call management and contact synchronization.

2. **Watch Location Integration:** Integrating functionalities to gather location data from smartwatches. This could involve developing apps for various smartwatch platforms, improving synchronization between the app and smartwatch, and ensuring accurate location data retrieval.

3. **Caller ID Features:** Integrating advanced caller identification features to provide users with more information about incoming calls. This could include displaying caller location, identifying spam or fraudulent calls, and offering customizable caller profiles.

4. **Backup and Restore Functionality:** Implementing robust backup and restore mechanisms to allow users to securely backup their app data, settings, and preferences. This could involve cloud storage integration, encryption protocols for data security, and seamless restoration options.

5. **App Settings Features:** Enhancing the app's settings menu to provide users with more control and customization options. This could include theme selection, notification preferences, language settings, accessibility options, and user profile management.

6. **Scraping Information from Websites and Apps:** Developing capabilities to scrape relevant information from websites and other apps to enrich the app's content and functionality. This could include news aggregation, product price comparisons, weather forecasts, and more.

7. **Integrating Government APIs:** Leveraging government APIs to integrate valuable services and data into the app. This could include real-time traffic updates, public transportation schedules, weather alerts, emergency services information, and government announcements.

By focusing on these areas for future work, the **Open-source Intelligence Data Mining System** app can continue to evolve, offering enhanced features, improved user experience, and increased functionality to meet the changing needs and expectations of its users.

References

[1] Company Site, [Online]. Available: https://cdrsoftwares.com/. Accessed on: March 15, 2023.

[2] Call One App Site, [Online]. Available: https://callones.com/. Accessed on: March 15, 2023.

[3] VS Code, [Online]. Available: https://code.visualstudio.com/. Accessed on: March 15, 2023.

[4] React Native, [Online]. Available: https://reactnative.dev/. Accessed on: March 15, 2023.

[5] React Native Firebase, [Online]. Available: https://rnfirebase.io/. Accessed on: March 15, 2023.

[6] RN Developer Tools, [Online]. Available: https://reactnative.dev/docs/react-devtools. Accessed on: March 15, 2023.

[7] Kotlin, [Online]. Available: https://kotlinlang.org/. Accessed on: March 15, 2023.

[8] Android Broadcasts overview, [Online]. Available: https://developer.android.com/develop/background-work/background-tasks/broadcasts. Accessed on: March 15, 2023.

[9] postgreSQL, [Online]. Available: https://www.postgresql.org/. Accessed on: March 15, 2023.

[10] Flipper, [Online]. Available: https://fbflipper.com/. Accessed on: March 15, 2023.

[11] JSON Web Token, [Online]. Available: https://jwt.io/. Accessed on: March 15, 2023.

[12] NPM Documentation, [Online]. Available: https://docs.npmjs.com/. Accessed on: March 15, 2023.

[13] Yarn Documentation, [Online]. Available: https://yarnpkg.com/. Accessed on: March 15, 2023.

[14] Node Modules Documentation, [Online]. Available: https://nodejs.org/api/modules.html. Accessed on: March 15, 2023.

[15] HTML, [Online]. Available: https://www.w3schools.com/html/. Accessed on: March 15, 2023.

[16] CSS, [Online]. Available: https://www.w3schools.com/css/default.asp. Accessed on: March 15, 2023.

[17] JavaScript, [Online]. Available: https://www.w3schools.com/js/default.asp. Accessed on: March 15, 2023.

[18] ExpressJS, [Online]. Available: https://expressjs.com/. Accessed on: March 15, 2023.

[19] NodeJS Documentation, [Online]. Available: https://nodejs.org/en/docs. Accessed on: March 15, 2023.

[20] ReactJS Documentation, [Online]. Available: https://react.dev/learn. Accessed on: March 15, 2023.

[21] Node Fetch, [Online]. Available: https://www.npmjs.com/package/node-fetch. Accessed on: March 15, 2023.

[22] Nginx, [Online]. Available: https://www.nginx.com/. Accessed on: March 15, 2023.

[23] TypeScript, [Online]. Available: https://www.typescriptlang.org/. Accessed on: March 15, 2023.

[24] Next.JS, [Online]. Available: https://nextjs.org/. Accessed on: March 15, 2023.

[25] Httpie, [Online]. Available: https://httpie.io/. Accessed on: March 15, 2023.

[26] Puppeteer, [Online]. Available: https://pptr.dev/. Accessed on: March 15, 2023.

[27] Baileys, [Online]. Available: https://whiskeysockets.github.io/. Accessed on: March 15, 2023.

[28] Leak OSINT, [Online]. Available: https://leakosint.com/en. Accessed on: March 15, 2023.

[29] PDF Kit, [Online]. Available: https://pdfkit.org/. Accessed on: March 15, 2023.

[30] GitHub, [Online]. Available: https://docs.github.com/en. Accessed on: March 15, 2023.