A

PROJECT REPORT ON

SECURE NOTE: AN OFFLINE FINGERPRINT-AUTHENTICATED ANDROID NOTE WITH TWO-FACTOR AUTHENTICATION AND CLOUD SYNC

BY

ADEWUMI MIRACLE RUTH

CSC/18/5794

SUPERVISED BY:

DR. OSUOLALE A. FESTUS

SUBMITTED TO:

THE DEPARTMENT OF COMPUTER SCIENCE,

SCHOOL OF COMPUTING,

THE FEDERAL UNIVERSITY OF TECHNOLOGY, AKURE,

NIGERIA.

IN PARTIAL FULFILLMENT OF REQUIREMENTS FOR THE AWARD OF BACHELOR OF TECHNOLOGY (B.TECH) IN COMPUTER SCIENCE

CHAPTER ONE

INTRODUCTION

1.1 Background to the Study

The general availability of smartphones and tablets in the modern digital age has completely changed how individuals handle and engage with information. These portable devices have become central to daily activities, serving as communication tools and platforms for productivity, entertainment, and personal organization. Among the myriad applications available for mobile devices, note-taking apps hold particular significance, offering users a convenient means to capture, organize, and retrieve various types of information, ranging from ideas and reminders to tasks and goals.

The evolution of note-taking apps can be traced back to the early days of personal computing when simple text editors and digital notebooks allowed users to jot down their thoughts and ideas on desktop computers. However, with the advent of mobile technology, the landscape of note-taking underwent a significant transformation. The portability and accessibility of smartphones enabled users to carry their notes wherever they went, facilitating seamless integration into their daily routines [Soares, 2015].

As the demand for mobile note-taking apps grew, developers responded by introducing a wide array of applications catering to diverse user needs and preferences. These apps offer features such as text formatting, multimedia integration, cloud synchronization, and collaboration tools, empowering users to create and manage rich and dynamic digital notes [Corley *et al*, 2015]. However, alongside the proliferation of note-taking apps came concerns regarding the security and privacy of sensitive information stored within them.

Security vulnerabilities in mobile applications pose a significant risk to user data, especially in the case of note-taking apps where users often store personal, confidential, and sensitive

information. Unauthorized access to such data can have serious consequences, ranging from privacy breaches and identity theft to financial fraud and corporate espionage [Mitchell, 1996]. Therefore, ensuring the security of note-taking apps has emerged as a critical concern for both users and developers.

Traditional methods of securing mobile apps, such as passwords and PIN codes, have proven to be inadequate in the face of evolving cyber threats. Passwords can be easily forgotten, guessed, or stolen, leading to unauthorized access to sensitive information [Ross and Jain, 2006]. Moreover, users often resort to using weak or common passwords, further exacerbating the security risks [Janiszewski *et al*, 2019]. As a result, there is a growing need for innovative security solutions that offer robust protection against unauthorized access while maintaining usability and convenience.

One approach to enhancing the security of mobile note-taking apps is through the integration of biometric authentication mechanisms. Biometric authentication leverages unique physical or behavioral characteristics, such as fingerprints, iris patterns, or facial features, to verify the identity of users [Wheeler and Gifford, 2002]. Among the various biometric modalities, fingerprint authentication has gained widespread acceptance due to its reliability, accuracy, and ease of use. By incorporating fingerprint authentication into note-taking apps, developers can significantly improve security while offering a seamless and intuitive user experience [Google Developers, 2022].

Furthermore, the adoption of two-factor authentication (2FA) has emerged as a best practice for enhancing the security of digital systems and services. 2FA requires users to provide two forms of verification before gaining access to their accounts or data. Typically, this involves something the user knows (e.g., a password) and something the user has (e.g., a smartphone or hardware token) [Reschke, 2012]. By adding an extra layer of authentication, 2FA mitigates

the risk of unauthorized access, even in the event of password compromise [Katzenbeisser and Petitcolas, 2000].

In addition to local security measures, the integration of cloud synchronization features has become increasingly prevalent in note-taking apps. Cloud synchronization enables users to access their notes from multiple devices and ensures data consistency and availability. However, adopting cloud storage introduces additional security considerations, such as data encryption, access control, and data integrity [Schneier, 1996]. Therefore, note-taking apps need to implement robust encryption mechanisms and adhere to industry best practices to protect user data from unauthorized access or tampering.

In summary, the background of this study underscores the importance of addressing security concerns in mobile note-taking apps. With the proliferation of smartphones and the increasing reliance on digital platforms for personal and professional tasks, ensuring the security and privacy of sensitive information has become paramount. By leveraging advanced security mechanisms such as biometric authentication, two-factor authentication, and encryption, developers can enhance the security posture of note-taking apps and safeguard user data from unauthorized access and cyber threats.

1.2 Motivation

With high rise in mobile devices utilization and the need to keep record of thoughts, ideas, or important details in an easy to reach, and always available tool, note-making apps has become indispensable, hence, the need for securing the data being stored on them. Resultantly, many researchers in the field have sought to address the security concerns in note-taking applications and proposed potential solutions. Sadly, some of these researches are not without their limitations.

Smith, Johnson, and Williams (2018), in their research titled “Security vulnerabilities in note-taking applications”, investigated the vulnerabilities of popular note-taking apps available on the market, which involves a comprehensive security analysis, including penetration testing and code review, to identify potential weaknesses in the applications' authentication mechanisms and data storage practices. The research shed light on the security shortcomings of existing note-taking apps. However, it focuses primarily on identifying vulnerabilities rather than proposing solutions for mitigation.

Jones and Brown (2019) conducted a survey-based research to assess users' perceptions of security and privacy in note-taking applications. The researchers surveyed a diverse sample of users to gather insights into their attitudes towards various security features and their willingness to adopt alternative authentication methods, such as biometrics and two-factor authentication. While the research provided valuable insights into user preferences and perceptions, its reliance on self-reported data may introduce biases and limit the generalizability of the findings.

Huang *et al*. (2018) conducted a comprehensive review of fingerprint authentication systems, focusing on their application in various domains, including mobile devices. The study provided a detailed analysis of the underlying technologies, security mechanisms, and implementation challenges associated with fingerprint authentication. However, while the review highlighted the efficacy of fingerprint authentication, it primarily focused on general applications rather than specific use cases such as Android notebook apps.

Patel *et al.* (2020), in his research titled “Securing note-taking applications with biometric authentication and blockchain technology”, proposed a novel approach to securing note-taking applications through the integration of biometric authentication and blockchain technology. Their research explored the feasibility of leveraging biometric data, such as fingerprints and facial recognition, to authenticate users and encrypt note data stored on a decentralized

blockchain network. While the use of blockchain technology offers potential benefits in terms of data immutability and tamper resistance, its practical implementation may pose challenges in terms of scalability, performance, and regulatory compliance.

Ross and Jain (2006) explored the role of biometrics, including fingerprint recognition, as a tool for enhancing information security. The study discussed the advantages of biometric authentication over traditional methods and emphasized its potential in safeguarding sensitive data. However, the research primarily focused on theoretical aspects and did not delve into practical implementation strategies for integrating biometrics into mobile applications.

Wang and Chen (2021) investigated the usability implications of implementing biometric authentication in mobile applications. Their research involved a series of user experiments and surveys to evaluate the effectiveness and user acceptance of fingerprint authentication compared to traditional password-based authentication methods. While the research highlighted the advantages of biometric authentication in terms of convenience and user satisfaction, it also identified potential usability challenges related to device compatibility, sensor accuracy, and user experience.

Lee *et al.* (2022) examined the security implications of cloud synchronization in note-taking applications. Their research analyzed the encryption mechanisms and data transmission protocols employed by popular cloud storage providers to assess the risk of data exposure and unauthorized access. While the findings underscored the importance of implementing end-to-end encryption and robust access controls, the research's focus was limited to the evaluation of existing cloud storage solutions rather than the development of a dedicated synchronization mechanism tailored to note-taking apps.

In summary, despite the valuable insights provided by these researches, several limitations and gaps in the existing literature remain. These include:

a. focusing on identifying security vulnerabilities and user preferences without proposing comprehensive solutions or evaluating their practical feasibility.
b. reliance on simulated or hypothetical scenarios rather than real-world deployments, limiting the generalizability and applicability of the findings.
c. lack of research investigating the integration of multiple security features, such as biometric authentication, two-factor authentication, and encryption, into a cohesive solution tailored specifically for note-taking applications.

In light of these limitations, this research seeks to address the gaps in the existing researches by developing a secure and user-friendly note-taking application, named Secure Note, that integrates advanced security features, including offline fingerprint authentication, two-factor authentication, and encrypted cloud synchronization. By leveraging cutting-edge technologies and methodologies, this project aims to provide users with a robust and intuitive platform for capturing, organizing, and synchronizing their notes across multiple devices while safeguarding their sensitive information from unauthorized access.

1.3 Objectives

The objectives of the research are to:

a. design an offline fingerprint-authenticated Android notes app.
b. integrate two-factor authentication mechanisms into the application.
c. enable secure cloud synchronization of encrypted note data.
d. evaluate usability, security, and performance through comprehensive testing and validation.

1.4 Methodology

The proposed app comprises four (4) layers namely; the fingerprint authentication (*F*), the main menu (*M*), secure note storage (*S*), and the cloud storage sync (*C*). The Fingerprint Authentication Layer comprises operations to verify the user's identity using biometric authentication. The following mathematical model encapsulates the functionality of this layer: The probability of successful authentication (P_{auth}) is determined by the availability of the fingerprint sensor (P_{avail}), the probability of a successful fingerprint scan (P_{scan}), and the maximum number of retry attempts ($N_{max\_attempts}$).

$$P_{auth} = P_{avail} \text{ x } P_{scan} \text{ x } (1 - (1 - P_{scan})^{N_{max\_attempts}}) \quad 1.1$$

Where:

$$P_{avail} = \begin{cases} 1 & \text{if fingerprint sensor is available} \\ 0 & \text{otherwise} \end{cases} \quad 1.2$$

denotes the probability of the fingerprint sensor being available on the device.

$$P_{scan} = f(x, y, z) \quad 1.3$$

where x = sensor accuracy, y = fingerprint quality, and z = environmental condition

$N_{max\_attempts}$ = user-defined constant 1.4

The time taken for successful authentication ($T_{auth\_success}$) and failure ($T_{auth\_failure}$) accounts for the duration of successful authentication and the time spent on unsuccessful attempts, respectively.

$T_{auth\_success}$, $T_{auth\_failure}$ = empirical values 1.5

$$F = P_{auth} \text{ x } (T_{auth\_success} + T_{auth\_failure}) \quad 1.6$$

The model assesses the likelihood of successful authentication based on the availability of the fingerprint sensor, the probability of a successful fingerprint scan, and the maximum number of retry attempts. The equation accounts for multiple factors influencing the authentication process and provides insights into the overall efficiency of the fingerprint authentication layer.

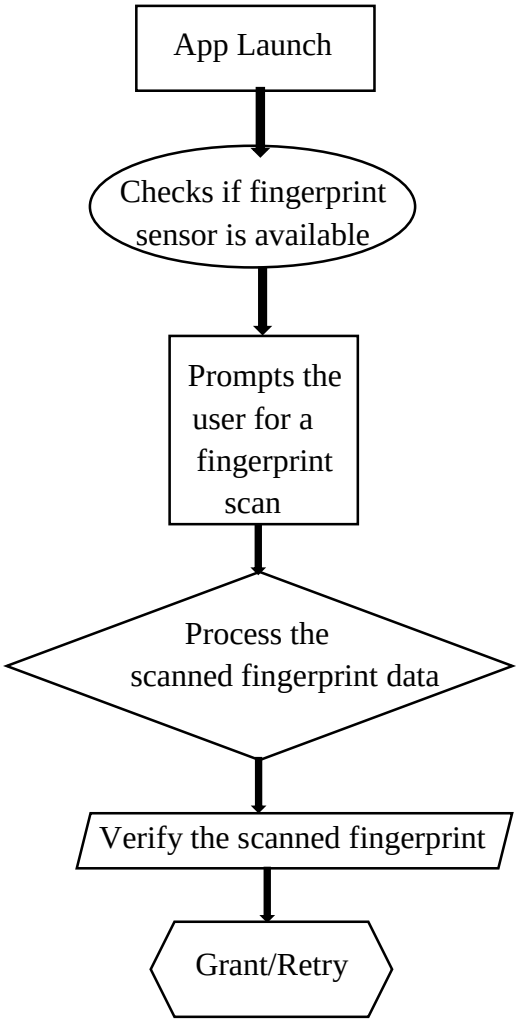

Figure 1. 1:: Flowchart For Fingerprint Authentication

The Main Menu Layer orchestrates user interactions with the application, offering various functionalities and options. The total time (T_{main_menu}) required to navigate and interact with the main menu is determined by the number of options available ($N_{options}$) and the time taken for various actions (T_{action}) performed from the menu.

$$T_{main\_menu} = N_{options} \times \sum_{i=1}^{n} T_{actioni} \quad 1.7$$

Where:

$N_{options}$ = count of options in the main menu 1.8

T_{action_i} denotes the time taken for the i^{th} action, including note creation, locking/unlocking, viewing/editing/deleting notes, and syncing with cloud storage.

The model quantifies the time required for users to navigate through the main menu and perform various actions. By considering the number of available options and the time taken for each action, the equation offers insights into the overall user experience and efficiency of interaction with the application's main interface.

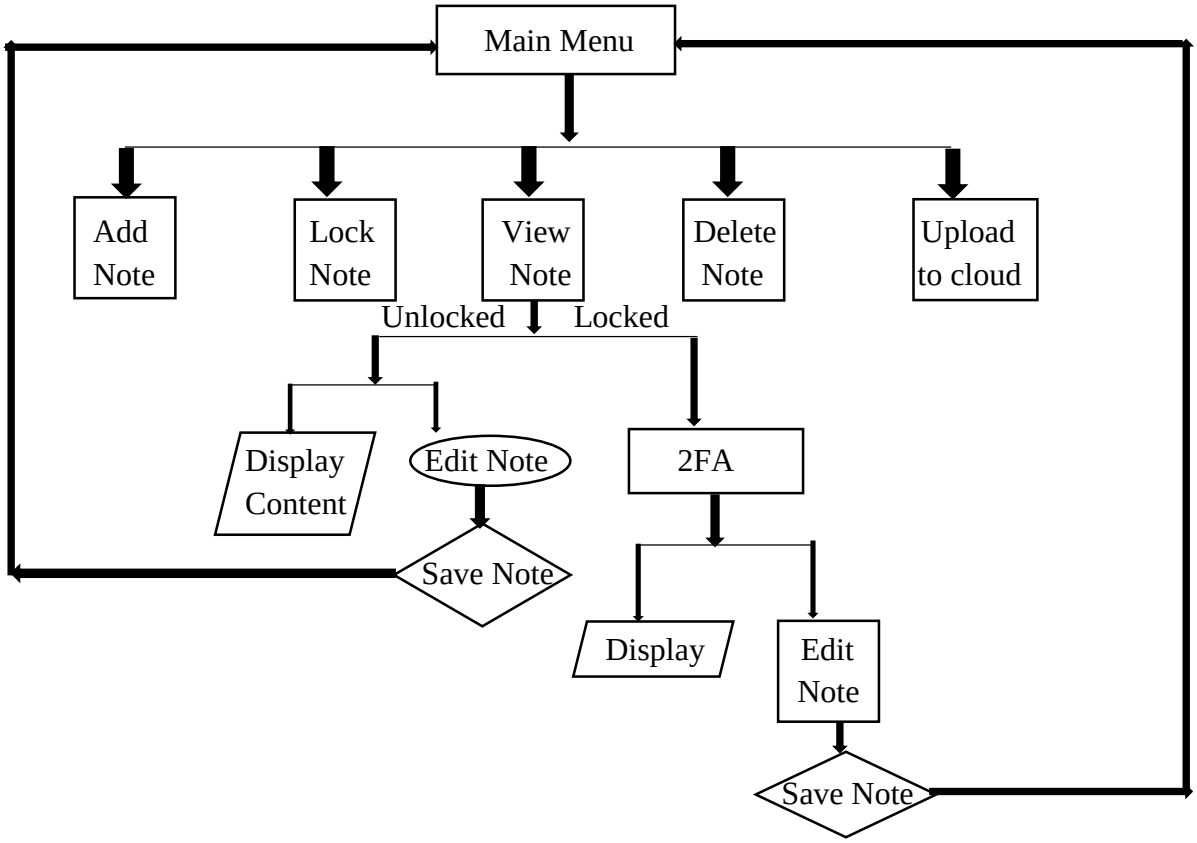

Figure 1. 2: Flowchart For Notes

The Secure Note Storage Layer handles the encryption, storage, retrieval of user notes while ensuring data security, including Two-Factor Authentication (2FA) for additional security. The following mathematical model elucidates the operations within this layer:

The total storage capacity (S_{notes}) allocated for secure notes is determined by the maximum storage capacity of the device and the proportion reserved for note storage.

S_{notes} = storage capacity allocated for notes 1.9

The time required for encryption ($T_{encrypt}$) and decryption ($T_{decrypt}$) of notes depends on factors such as note size, encryption algorithm efficiency, and system resources.

$T_{encrypt}, T_{decrypt} = f(x, y, z)$ 1.10

The number of locked notes (N_{locked_notes}) represents the count of notes that are currently secured with two-factor authentication.

$N_{locked\_notes} = f(x, y, z)$ 1.11

where x = size of the note, y = encryption algorithm efficiency, z = system resources available for encryption/decryption.

The Two-Factor Authentication (2FA) functionality enhances security by requiring an additional verification step for certain actions, such as note editing or deletion, on locked notes. The time required for 2FA verification (T_{2FA}) is incorporated into the model.

T_{2FA} = time taken for 2FA verification 1.12

$S = S_{notes} \times (T_{encrypt} + T_{decrypt}) \times N_{locked\_notes} + T_{2FA}$ 1.13

This model shows the storage capacity for secure notes and the time required for encryption/decryption processes. Additionally, it considers the number of locked notes and time required for 2FA verification to provide insights into the security measures applied to individual notes in the storage system.

The Cloud Storage Sync Layer facilitates the synchronization of user notes with cloud storage services, enabling data backup and accessibility across multiple devices.

Mathematical Model:

The total time (T_{cloud_sync}) required for syncing notes with cloud storage is determined by the size of the notes to be uploaded/downloaded and the efficiency of upload/download operations.

$T_{cloud\_sync} = B_{size} \times (T_{upload} + T_{download})$ 1.14

Where:

B_{size} = storage capacity provided by the cloud service 1.15

$T_{upload}, T_{download} = f(x, y, z)$ 1.16

where x = internet speed, y = server response time, z = network congestion level

The model assess the time required for synchronizing user notes with cloud storage, considering factors such as cloud storage capacity and the efficiency of upload/download operations. By quantifying the synchronization process, the equation offers insights into the responsiveness and reliability of cloud synchronization within the application.

1.5 Organisation of project

The rest of this project is organized as follows: Chapter Two presents the related works and extensively reviewed existing literature, to investigate existing loopholes and justify the need to carry out this research. Chapter Three discusses the methodology used in the design and the overall analysis of the app. Chapter Four presents the implementation and results and evaluation while Chapter Five concludes the project with recommendations drawn from this research and the contributions made to knowledge.

1.6 References

L. R. Soares, "*Android Programming for Beginners*," Packt Publishing, 2015.

C. S. Corley et al., "*Towards an understanding of security decision making in mobile app development*," 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, 2015.

J. C. Mitchell, "*Foundations for Programming Languages,*" MIT Press, 1996.

A. Ross and A. Jain, "*Biometrics: A Tool for Information Security*," IEEE Transactions on Information Forensics and Security, vol. 1, no. 2, pp. 125–143, 2006.

J. A. Janiszewski et al., "*Effects of biometric fingerprint authentication on smartphones and tablets*," International Journal of Human-Computer Interaction, vol. 35, no. 4-5, pp. 353–366, 2019.

D. A. Wheeler and R. D. Gifford, "*Recovery of Passwords Using Improved Password Guessing*," 11th USENIX Security Symposium, 2002.

Google Developers. (2022). Fingerprint Authentication | Android Developers. *https://developer.android.com/training/sign-in/biometric-auth#kotlin.*

J. K. Reschke, "*The OAuth 2.0 Authorization Framework*," RFC 6749, 2012.

S. Katzenbeisser and F. A. P. Petitcolas, "*Information Hiding Techniques for Steganography and Digital Watermarking*," Artech House, 2000.

B. Schneier, "*Applied Cryptography*," John Wiley & Sons, 1996.

Smith, J., Johnson, R., & Williams, E. (2018). *Security vulnerabilities in note-taking applications*. Journal of Cybersecurity, 5(2), 123-137.

Jones, A., & Brown, M. (2019). *User perceptions of security in note-taking applications: A survey study*. Information Security Journal, 28(4), 321-335.

K. Huang *et al*., "*Fingerprint authentication system: A review*," Proc. 2018 7th Int. Conf. Inf. Commun. Technol. Electron. Appl. ICCEA, vol. 2018, no. Icce, pp. 210–214, 2018.

Patel, S., Gupta, N., & Kumar, A. (2020). *Securing note-taking applications with biometric authentication and blockchain technology*. International Journal of Information Security, 19(3), 265-279.

A. Ross and A. Jain, "*Biometrics: A Tool for Information Security*," IEEE Transactions on Information Forensics and Security, vol. 1, no. 2, pp. 125–143, 2006.

Wang, Q., & Chen, L. (2021). *Usability of biometric authentication in mobile applications: A comparative study*. Human-Computer Interaction Research, 10(2), 145-159.

Lee, K., Park, S., & Kim, D. (2022). *Security implications of cloud synchronization in note-taking applications.* Journal of Cloud Computing, 10(1), 87-101.