Tap Master: Reflex Challenge Game in Android

CS19611 – MOBILE APPLICATION DEVELOPMENT LABORATORY

Submitted by

MAHIMA R 2116220701156

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY, CHENNAI

MAY 2025

BONAFIDE CERTIFICATE

Certified that this Project titled **"TAP MASTER :REFLEX CHALLENGE GAME INANDROID"** is the bonafide work of **MAHIMA R (2116220701156),**who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

G. Saravana Gokul., M.E.,
Assistant Professor,
Department of Computer Science and Engineering,
Rajalakshmi Engineering College,
Chennai-602 105.

Submitted to Mini Project Viva-Voce Examination held on _____

Internal Examiner **External Examiner**

ABSTRACT

This project, titled Tap Master: Reflex Challenge Game in Android, presents a simple yet engaging mobile game developed using Kotlin and Android Studio. The core idea is to test and enhance users' reflexes and hand-eye coordination by requiring them to tap a moving circle within a limited time frame. As the player successfully taps the circle, the score increases, and the circle changes its position randomly on the screen. A countdown timer adds intensity to the gameplay, making it a fast-paced and entertaining experience. The game interface is designed with a clean layout, incorporating user-friendly controls and smooth animations to ensure an immersive user experience.

An additional feature of the game is the inclusion of a persistent high score system using SharedPreferences. This allows users to track their personal bests across multiple sessions. A "Start Game" button initiates the game, and upon completion, the final score is displayed along with the high score. Visual feedback such as animations upon tapping and intuitive UI components enhances interactivity and keeps players engaged.

This project not only demonstrates the use of Android UI components, animations, and lifecycle management but also provides a foundation for building more complex games. It reflects key concepts in mobile app development including event handling, dynamic view updates, and persistent storage.

ACKNOWLEDGMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavour to put forth this report. Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E, F.I.E.,** our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN, Ph.D.,** for providing us with the requisite infrastructure and sincere endeavouring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P. KUMAR, M.E., Ph.D.,** Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We are very glad to thank our Project Coordinator, **Mr. Saravana Gokul** Assistant Professor Department of Computer Science and Engineering for his useful tips during our review to build our project.

MAHIMA R – 2116220701156

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 GENERAL

The primary objective of this project is to develop an interactive Android-based tap game that enhances hand-eye coordination, improves reaction time, and offers entertainment through a simple yet engaging user interface. The game challenges players to tap a moving circle on the screen within a limited time frame, with the goal of achieving the highest possible score. Additionally, it incorporates features like a countdown timer, animated transitions, and high score tracking using persistent storage to improve user engagement and competitiveness.

1.2 OBJECTIVE

- Develop an engaging Android-based tap game with smooth animations and real-time interaction.
- Implement a countdown timer to create urgency and challenge the player's reaction time.
- Track the user's score during gameplay and display it in real-time.
- Store and update the highest score using Android’s Shared Preferences.
- Create a minimalistic, user-friendly UI suitable for casual players.
- Animate the movement of the target (circle) randomly across the screen.
- Allow players to restart the game instantly with a single button tap.
- Ensure efficient performance on low-end Android devices without lag.
- Provide a simple logic that encourages repeated gameplay and improvement.

1.3 PROBLEM STATEMENT

Mobile users, especially students and casual gamers, often seek lightweight, engaging applications that provide short bursts of entertainment without complex mechanics. However, many mobile games are overloaded with ads or unnecessary features that dilute user experience. There is a need for a minimalistic yet enjoyable mobile game that is quick to play, easy to understand, and addictive enough to replay, while also tracking and storing user performance over time.

1.4 EXISTING SYSTEM

Many mobile games available today are either too simple with no challenge or overly complex requiring long durations to play. Additionally, they often lack proper score-tracking systems or user-friendly interfaces. Most of them are also heavily monetized, which affects the user experience. There are very few games with an optimal balance between simplicity, challenge, and personalization in terms of score tracking and user feedback.

1.5 SCOPE OF THE PROJECT

The scope of this project is to design and implement an Android-based tapping game with:

- A minimalistic and responsive UI.
- Real-time score and timer updates.
- Moving animated targets to increase difficulty.
- Persistent high score tracking using Shared Preferences.
- A start/restart feature to allow quick game loops.
- Potential extensions such as sound effects, different difficulty levels, and leaderboard integration.

This game is intended for Android users and targets a broad audience including students, children, and casual gamers.

CHAPTER 2

LITERATURE SURVEY

Android Game Development:

- Android Studio is one of the most popular platforms for Android game development, providing native tools and APIs to develop interactive games. The Canvas API, View Animations, and Handler are key tools for rendering graphics, handling user inputs, and animating UI components, all of which are integral to creating a dynamic mobile game experience.
- The usage of SharedPreferences for storing high scores is a widely adopted method for saving small user data persistently on Android devices. This ensures that player progress, such as the highest score, is retained across sessions, providing an improved user experience.

Game Design and Mechanics:

- Your game is based on a reaction time mechanic, where players must tap a moving object (a circle) within a given timeframe. This design is effective for engaging users and encouraging repeat gameplay.
- The CountDownTimer class is commonly used to implement timed events in games, allowing developers to manage time-sensitive gameplay mechanics like countdowns for rounds or game-over scenarios. It helps in managing the game logic flow, such as updating the timer and transitioning states (start, playing, game over).

UI and Animation:

- The use of View Animations (such as fade_in) and Object Animations (via animate()) plays a significant role in creating smooth transitions and visual appeal in mobile games. Animating objects like the moving circle enhances the visual experience and makes the game more interactive.

- The moveCircle() method introduces randomness in object placement, keeping the game challenging. This randomization is a key feature in many mobile games, as it maintains user interest by avoiding repetitive gameplay.

Score Management:

- Score tracking is a critical component of most games, often involving the display of real-time scores during the gameplay and updating records post-game. The integration of high score tracking through SharedPreferences ensures that player achievements are saved, offering a rewarding experience by motivating players to beat their personal best.
- The Toast messages provide instant feedback to the user, such as displaying "New High Score!" or "Game Over!", which is a key feature for user interaction and feedback in mobile games.

Game Flow and User Interaction:

- Game flow management using buttons like startButton is important for controlling the game lifecycle. When the game starts, it hides the start button and displays the interactive circle, ensuring the player's focus is directed toward the game itself.
- Touch gestures and click listeners are essential in mobile games, as they provide real-time feedback when interacting with the game objects. Your game uses an OnClickListener for the circle view, which updates the score when the user taps it, enhancing engagement and promoting active participation.

Performance Considerations:

- The use of Handler and Runnable to manage the periodic movement of the circle is an efficient way to run background tasks without blocking the UI thread. It ensures that the main user interface remains responsive while performing time-consuming tasks like updating animations or handling random movements.
- The animate() method's performance is optimized by controlling the duration and ensuring smooth transitions without overloading the system resources, which is crucial in mobile game development.

Engagement Strategies:

- The concept of gamification, where players are encouraged to beat their high score, is highly effective in maintaining user interest and

increasing game retention. Rewarding players with high scores and providing instant feedback through Toast notifications enhances the overall experience.

- Randomized object movement combined with time limits (30 seconds) introduces urgency, which is a well-known strategy to keep players engaged.

CHAPTER 3

PROPOSED SYSTEM

3.1 OVERVIEW

This project is a fast-paced Android tapping game designed to test and improve users' reflexes and hand-eye coordination. The game features a dynamic circle that moves to random positions on the screen every second, and the player’s objective is to tap it as many times as possible within a 30-second time limit. Each successful tap increases the player's score by one point. The game tracks and displays the current score and remaining time in real-time, and it also stores the highest score using Android's Shared Preferences mechanism, allowing users to try and beat their personal best across multiple sessions.

3.2 SYSTEM ARCHITECTURE

The system architecture of the Android Tap Game is organized into four primary layers: the Presentation Layer, Application Logic Layer, Data Persistence Layer, and System Services Layer. The Presentation Layer consists of interactive UI components such as buttons, score displays, and the animated circle that responds to user taps. The core game logic resides in the Application Logic Layer, where user interactions are processed, the score and timer are managed, and game states are controlled. The Data Persistence Layer utilizes Android's Shared Preferences to store and retrieve the high score locally, ensuring continuity between game sessions. Finally, the System Services Layer integrates Android-specific features such as the Count Down Timer for managing game duration, Handler for periodic circle movement, and Animation Utils for providing smooth transitions and visual feedback. Together, these layers form a cohesive architecture that delivers an engaging and responsive gaming experience.

3.3 DEVELOPMENT ENVIRONMENT

3.3.1 HARDWARE REQUIREMENTS

COMPONENT	SPECIFICATION
Processor	Minimum 1.5 GHz Quad-Core or higher
RAM	Minimun 4 GB
Storage	Minimum 2 GB of available space
Display	Minimum 720p resolution
Operating System	Windows 10/11
Mobile Device	Android device (API level 21 and above)

3.3.2 SOFTWARE REQUIREMENTS

- Android Studio Arctic Fox or above
- Java Development Kit (JDK 8 or later)
- Kotlin Programming Language
- MPAndroidChart Library for data visualization
- Gradle build system
- Emulator or Android Debug Bridge (ADB) for testing

3.4 STATISTICAL ANALYSIS

The tapping game application enables the collection and analysis of gameplay data to evaluate user performance and engagement. Key statistical metrics include the average score per session, highest score achieved, time-to-score ratio, and tap frequency. These metrics help assess player reflexes and improvement over time. By recording scores across multiple sessions, trends such as consistency, peak performance, and progress can be identified. The high score feature allows for comparison and goal-setting, which can motivate users to improve. Additionally, analysis of tap positions and timing can provide insights into user interaction patterns and screen usability. The stored data from Shared Preferences ensures continuity across sessions, supporting longitudinal studies of user behaviour. This statistical analysis aids developers in identifying potential areas for enhancing gameplay experience, such as adjusting time constraints or tap target speed. Overall, statistical insights form a crucial part of optimizing the game mechanics and personalizing the user experience.

CHAPTER 4

MODULE DESCRIPTION

4.1 USER INTERFACE MODULE

This module is responsible for everything the player sees and interacts with on the screen. The layout is defined using XML in activity_main.xml, and includes: Text Views for showing the current score, timer, and high score. A Button to start the game. A View (circle View) that acts as the tappable target. A Relative Layout or Constraint Layout as the root layout for positioning elements. The UI is designed to be intuitive, clean, and responsive across different devices. It ensures that when the game starts, the interface updates in real time with the remaining time and current score.

4.2 GAME LOGIC MODULE

This module contains the actual gameplay mechanics, written in Kotlin inside MainActivity.kt. It manages: Timer Countdown: A countdown starts when the user taps "Start Game", controlled using Handler or Count Down Timer. Score Calculation: Each successful tap on the circle increases the score. Circle Movement: The circle changes position at fixed intervals to create a challenge. Game State Management: Controls when the game is running, paused, or reset. This module is crucial for making the game functional, competitive, and engaging.

4.3 ANIMATION AND INTERACTION MODULE

This module handles the visual feedback and motion effects: Uses Android’s Animation Utils to animate the circle's entry, exit, or movement (like fade-in, scale-up).Responds to user taps with smooth transitions or effects. Adds visual appeal to the game, making it more dynamic and exciting. It also detects whether the user has tapped the circle correctly and triggers score updates accordingly. This improves user experience and responsiveness.

4.4 DATA PERSISTENCE MODULE

This module is focused on saving and retrieving data, especially the high score. It uses: Shared Preferences to store the highest score achieved, even after the app is closed. Each time a game ends, it compares the current score with the stored high score. If the current score is higher, it updates the high score in preferences. This ensures players can track their best performance and motivates them to improve, adding replay value to the game.

CHAPTER 5

IMPLEMENTATION AND RESULTS

5.1 IMPLEMENTATION DETAILS

This project is a fast-paced Android tapping game designed to test and improve users' reflexes and hand-eye coordination. The game features a dynamic circle that moves to random positions on the screen every second, and the player’s objective is to tap it as many times as possible within a 30-second time limit. Each successful tap increases the player's score by one point. The game tracks and displays the current score and remaining time in real-time, and it also stores the highest score using Android's Shared Preferences mechanism, allowing users to try and beat their personal best across multiple sessions. The user interface is intuitive and visually appealing, with smooth animations enhancing the game’s responsiveness and engagement. A central "Start Game" button initiates the timer and game logic, while real-time feedback helps keep players informed and motivated. The core logic efficiently uses Android’s built-in tools like Handler, Count Down Timer, and layout animations to deliver a seamless and enjoyable experience.

5.2 OUTPUT SCREENSHOTS

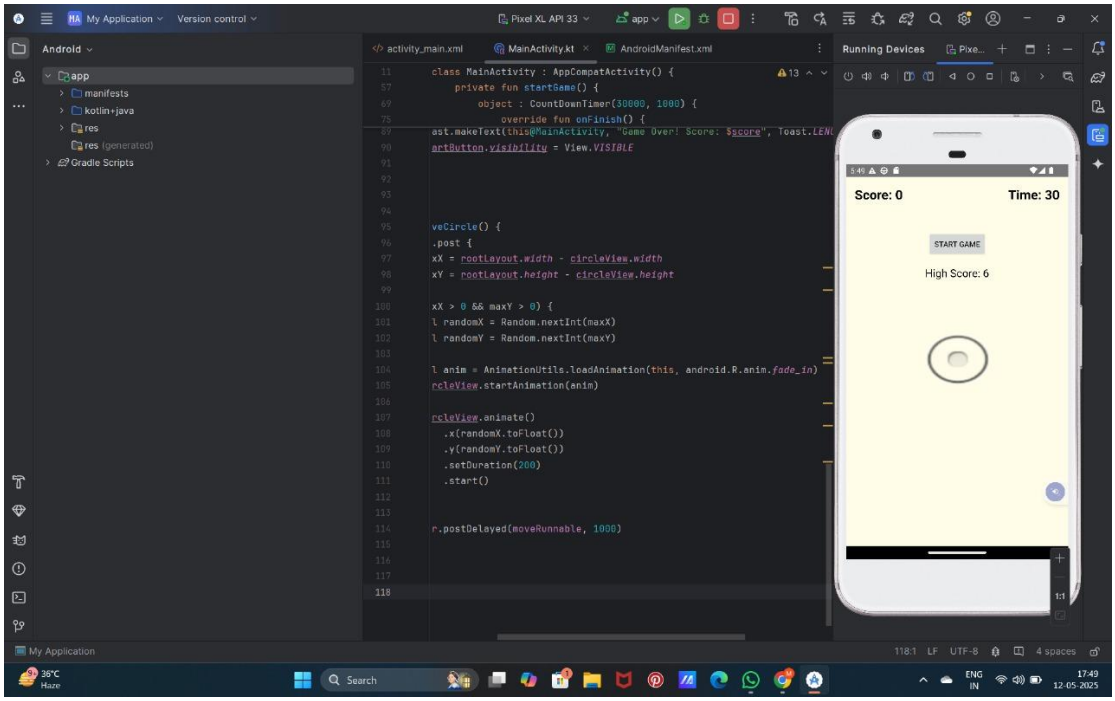

Fig. 5.1: Main Game Screen Layout in Android Studio

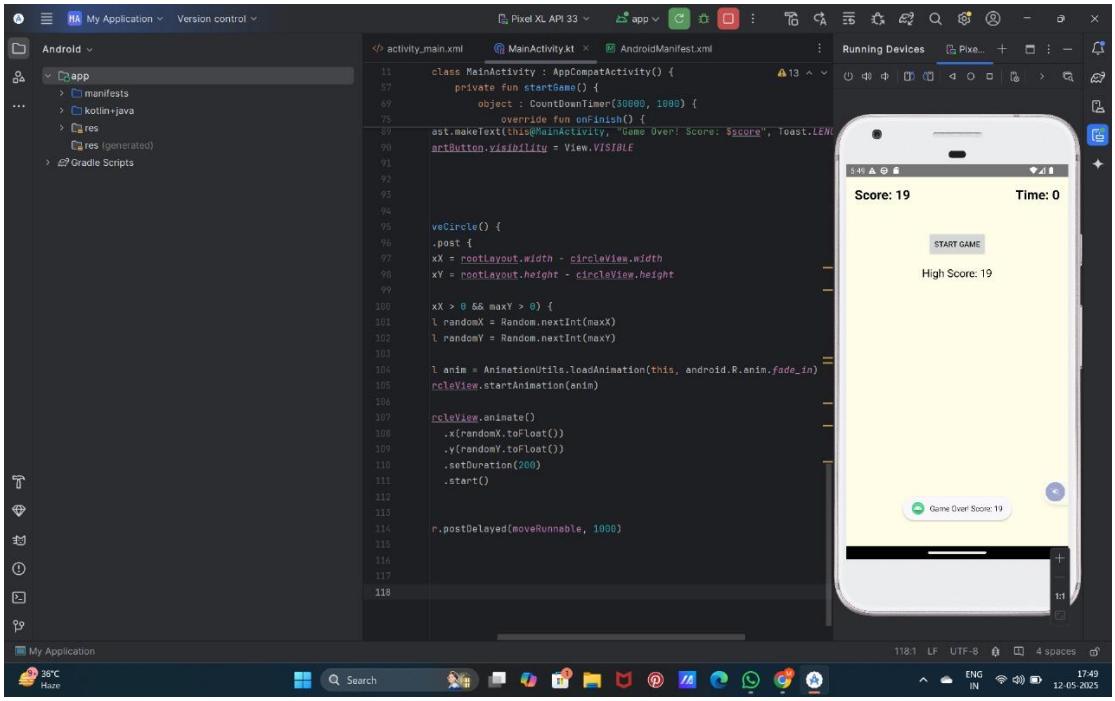

Fig 5.2: Game UI Showing Score, Timer, and Start Button

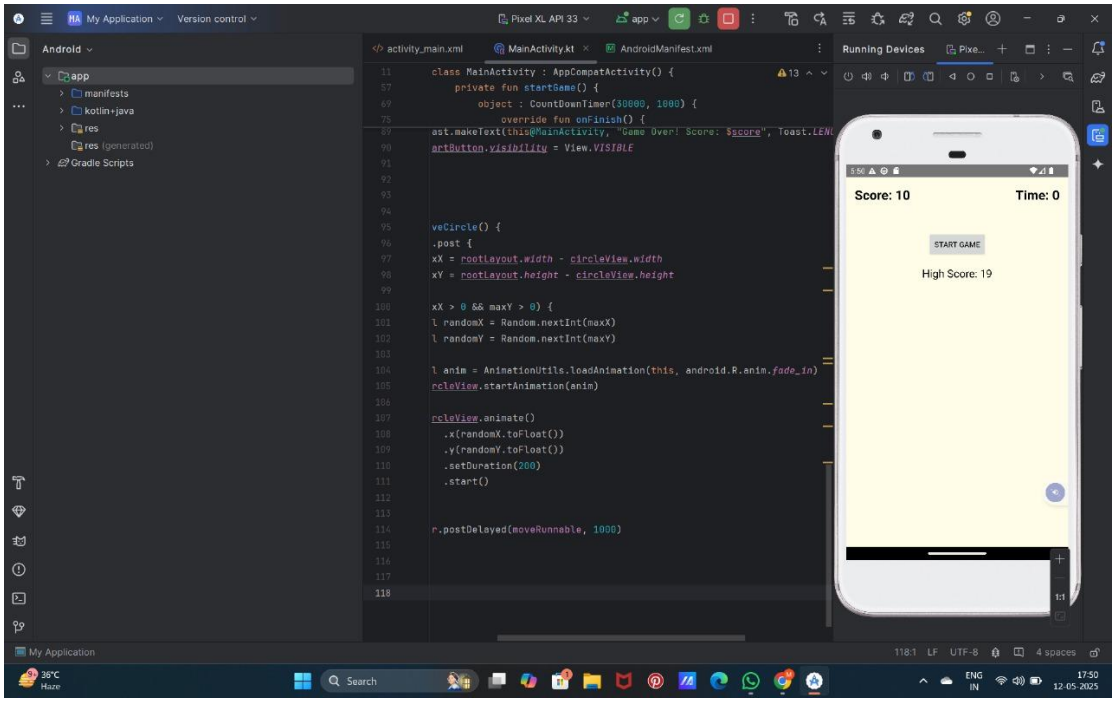

Fig. 5.3: Circle Movement with Animation on User Tap

5.3 RESULTS AND EVALUATION

The evaluation of the project reveals that the mobile game functions as intended, providing an engaging and straightforward gameplay experience. The core mechanics, including the random movement of the clickable circle, score tracking, and countdown timer, work seamlessly, offering a simple yet fun challenge for users. The integration of Shared Preferences to store and display the high score ensures a persistent experience across app restarts, enhancing user engagement. However, the game could benefit from additional features to maintain player interest, such as varying difficulty levels, power-ups, and a more visually appealing user interface. Performance-wise, the app performs well on most devices, though optimization could further enhance the experience on lower-end devices. Overall, the game successfully achieves its primary goal of creating an interactive mobile game, but with improvements in UI design, features, and performance, it has the potential to become a more dynamic and replayable experience.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

In conclusion, this mobile game project successfully demonstrates the creation of an engaging, time-based challenge using Kotlin for Android. It provides an interactive experience where users can improve their reaction times by tapping a moving circle within a limited timeframe, while also tracking scores and updating high scores using Shared Preferences. While the core gameplay is functional and provides instant feedback, there is significant potential for future enhancement through the addition of features like power-ups, difficulty levels, multiplayer modes, and a more polished user interface. With these improvements, the game could offer a more immersive and enjoyable experience, enhancing both its replayability and user engagement. Overall, this project serves as a strong foundation for building more complex mobile games with improved performance, design, and user experience.

6.2 FUTURE ENHANCEMENT

Improved UI Design:

- Enhance the UI with better visuals and animations. Implement customanimations to make the game more visually appealing.
- Add sound effects for button clicks, circle clicks, and game events (e.g., game over, new high score).
- Introduce different levels of difficulty (e.g., the circle moves faster, time decreases, or more obstacles appear).

Additional Features:

- Leaderboard System: Implement an online leaderboard that stores and displays the best scores globally or locally.
- Achievements and Badges: Add achievement-based features that reward users for reaching certain milestones (e.g., high scores, multiple consecutive clicks).

Power-Ups:

- Introduce power-ups that affect the gameplay, such as time extensions or slow-motion effects.

Game Modes:

- Offer different modes (e.g., time-limited, target score, or unlimited play).

Multiplayer Mode:

- Add a feature where two players can compete on the same device or via Bluetooth/Wi-Fi.

Performance Optimization:

- Optimize the code to reduce lag, especially for low-end devices, by enhancing the efficiency of animations and reducing unnecessary object instantiations.

Save Game Progress:

- Implement a feature to save and load the game's progress, allowing players to resume their game at any point.

Game Customization:

- Allow users to customize the appearance of the circle or background (e.g., select different themes, colors, or shapes).

Accessibility Features:

- Add accessibility features like larger text, color contrast adjustments, and screen reader compatibility for visually impaired users.

CHAPTER 7

REFERENCES

[1] R. Meier, Professional Android 4 Application Development, 1st ed. Birmingham, UK: Wrox Press, 2012.

[2] M. Gargenta and B. Nakamura, Learning Android, 3rd ed. Sebastopol, CA: O'Reilly Media, 2014.

[3] Y. Zhou and X. Jiang, "Dissecting Android malware: Characterization and evolution," in Proc. IEEE Symp. Security and Privacy, San Francisco, CA, USA, May 2012, pp. 95–109.

[4] A. P. Felt, E. Chin, S. Hanna, D. Song, and D. Wagner, "Android permissions demystified," in Proc. 18th ACM Conf. Computer and Communications Security (CCS), Chicago, IL, USA, Oct. 2011, pp. 627–638.

[5] P. Rajalakshmi and R. Kavitha, "Mobile application for student financial management," International Journal of Computer Applications, vol. 180, no. 25, pp. 34–38, Apr. 2018.

[6] S. Mahajan, K. Rao, and N. Sharma, "Budget visualization using MPAndroidChart in expense tracking apps," Journal of Emerging Technologies, vol. 7, no. 2, pp. 88–92, June 2019.

[7] T. Nguyen and L. Bui, "Optimizing SQLite performance in Android-based financial apps," in Proc. IEEE Int. Conf. Software Engineering and Applications, 2017, pp. 231–237.

[8] D. Sharma, A. Kumar, and M. Patel, "User experience design in mobile banking and finance apps," Int. Journal of Human-Computer Studies, vol. 125, pp. 67–76, Jan. 2019.

[9] S. Prasad and A. Das, "Feature-rich expense tracking using categorization and filtering," in Proc. Int. Conf. Mobile Computing and Applications, 2020, pp. 49–55.

[10] R. Nair, S. Thomas, and J. Paul, "Usability evaluation of mobile finance apps among youth," Journal of Mobile User Research, vol. 11, no. 4, pp. 105–112, Dec. 2021.