

Android App Development Using Android Studio

Dr.N. Baggyalakshmi^{1*}, A.K. Gayathri², Dr.R. Revathi³

^{1*}Assistant Professor, Department of Computer Science, PSGR Krishnammal College for Women, Coimbatore.

E-mail: baggyanethra@gmail.com

²Student, Bsc Computer Science, PSGR Krishnammal College for Women, Coimbatore. E-mail: 23bcs020@psgrkcw.ac.in

³Assistant Professor, Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore.

E-mail: revathilakshay@gmail.com

Abstract--- This internship program is fully focused on Android app development using the industry-standard software, Android Studio. The internship aims to provide aspiring developers with hands-on experience and practical knowledge in creating Android applications using the powerful features and tools offered by Android Studio Software. The development environment setup section outlines the steps to install and configure Android Studio on a developer's machine, ensuring compatibility with the Android Software Development Kit (SDK) and other required dependencies. It emphasizes the importance of setting up virtual devices or using physical devices for testing and previewing app behavior. The user interface design section discusses Android Studio's Visual editor, known as the Layout Editor, which simplifies the creation of app interfaces through a drag- and -drop approach. In summary, this abstract provides an overview of Android app development using Android Studio, showcasing its comprehensive set of features, tools, and resources that empower developers to create robust and visually appealing Android applications.

Keywords--- Android Development, Android Studio, Android Software Development Kit (SDK).

I. INTRODUCTION

ANDROID Studio is the official integrated development environment (IDE) for Android app development [1]. It is a comprehensive software package designed to assist developers in creating, testing, and debugging Android applications. Android Studio provides a range of tools and features that streamline the app development process and enhance productivity [2]. Android app development using Android Studio software involves the process of creating, designing, coding, testing, and deploying Android applications. Android Studio is the primary tool used by developers to build Android apps, and it provides a robust and efficient environment for the development process.

An Android Developer is responsible for developing Android Applications for devices that are powered by Android Operating System. An Android Developer not only develops Android Application's but also remove bugs, update software, software maintenance, etc. [3]. The future of Android application development includes new technology from Google — Flutter. This framework allows developing apps with a native interface for platforms like Android and iOS. A lot of developers choose Flutter as a tool for mobile app development service due to its flexibility and combinability.

Application designing is a multi-step process for drawing clear visual directions and offering an abstract of the final product [4]. Development – During this stage, a

working prototype is developed to validate the functionality, assumptions, and understanding of the project scope.

The IDE we have worked with for developing this app is Android Studio. Android Studio is the new official IDE for Android. Version 1 was developed in Eclipse [5]. However, the Application Programming Interface (APIs) released in future will not be available for Eclipse. To get the latest updates, one must switch to Android Studio. The existing system for Android Studio refers to the current version and its associated features, tools, and capabilities. As of my knowledge cutoff in September 2021, the existing system for Android Studio was Android Studio 4.2. However, please note that there might have been newer versions released since then, and it's always recommended to check for the latest version on the official Android Studio website.

Components in the Existing System for Android Studio

Android SDK: The Android Software Development Kit (SDK) is an essential part of Android Studio. It includes libraries, APIs, and tools necessary for developing Android applications.

Gradle Build System: Android Studio uses the Gradle build system to automate the build, testing, and deployment of Android apps. It provides flexibility and customizability in managing dependencies and building multi-module projects.

Profiling and Performance Analysis: Android Studio offers tools for profiling and analyzing the performance of Android apps. It allows developers to monitor CPU usage,

memory allocation, network traffic, and other performance metrics.

Version Control Integration: Android Studio seamlessly integrates with popular version control systems like Git. It provides features for managing source code repositories, branching, merging, and collaboration with other developers.

Support for Libraries and APIs: Android Studio offers extensive support for integrating third-party libraries, frameworks, and APIs into your app. It provides tools for managing dependencies, importing libraries, and configuring project settings.

Android Debug Bridge (ADB): ADB is a command-line tool that comes bundled with Android Studio. It enables developers to interact with physical and virtual Android devices for tasks such as installing apps, debugging, and accessing device logs.

II. LITERATURE REVIEW

Sarkar et al [6] designed his open-source operating system, which is built on the Linux kernel and is very flexible thanks to its modification capabilities, is the leading mobile operating system. Java is the programming language used to create Android apps. With the help of the Google Android SDK, developers may easily create Android applications on a platform that offers a unique software stack. Additionally, developers have freedom by utilizing the pre-existing Java IDEs. The process of developing third-party applications primarily uses Java libraries. Cross-platform techniques ensure that programmers are spared from creating platform-specific programs. These methods make it possible to distribute an application across several platforms without requiring coding changes. Unfortunately, most consumers are unaware of Android's increased susceptibility to security flaws.

Shaheen et al [7] proposed the features of Android apps and the Android platform provided a comprehensive overview of the foundation for Android applications from the perspective of developers. Android devices that boot up have the "hub" as their primary navigation and information display. Comparable to desktop PCs, these are found on Android devices. To show the fundamental operations of Android application components, let's have a look at a basic music player. It uses the media player service component from the class of library layers to play music. This paper aims to offer help on comprehending the workings of Android applications and offer some context for creating applications for the Android platform.

Okediran et al [8] the creation and advancement of smartphones, which enable users to connect to the application stores and other online marketplaces of device manufacturers to obtain new apps and services. Numerous characteristics of a full-fledged computer have been incorporated to these smartphones, including fast processors, lots of storage, multitasking, sharp screens and cameras, versatile connection devices, and more. However, a variety

of technology platforms, such as various operating systems (OS) and application development platforms, dominate the market for these devices, leading to a wide range of competing solutions offered by various players. This essay included a thorough examination and comparison of various technological platforms' aspects.

Azis [9] Humans benefit from the smartphone industry's rapid progress. These benefits include convenience, quickness, and ease of access when doing daily tasks. Numerous smartphone applications—such as library applications—have been developed to make daily tasks easier for people. Users may quickly decide the books they want to seek for or borrow by using the library app. Students frequently don't understand where the book is located. In response to these issues, a mobile-friendly library application was developed so that students could locate books more easily without physically visiting the library. The waterfall technique along with the Software Development Life Cycle (SDLC) is the research method employed. This library application uses a MySQLite database and the Java programming language, and it is built on Android. Black Box Testing is the method used for application testing. All of the features in this Android-based library application have successfully performed as intended, according to the findings of the Black Box Testing that was conducted on it.

Maryam and Purwono [10] explain the creation of an Android mobile application that serves as a service to convey information to stakeholders, or the intended users, about space weather conditions in Indonesia. The service uses the pushed-notification approach to do this. The most efficient approach to engage users is with a notification that appears instantly on their device. To provide information contents for the notification and application that are appropriate for use cases that arise and satisfy user preferences, a cross-platform message data builder based on GSON and Retrofit interface is implemented. Moreover, the program manages the engagement process provided by Firebase, a Google cloud messaging system, specifically through token identification on mobile devices.

III. PROPOSED METHODOLOGY

3.1. About Google Search Console



With the free Google Search Console, customers may monitor site traffic, examine keyword performance, resolve difficulties, and receive messages regarding their website

from Google. Insights on a site's organic search performance and suggestions for improving its position in the Google index are both provided by this tool. (However, Search Console differs from Google Analytics in that it merely gives data on search-related traffic—not direct traffic, ad traffic, or referral traffic—and does not break down this data further). Google introduced Search Console, formerly known as "Webmaster Tools," almost fifteen years ago. Even though it has undergone numerous name changes and functional overhauls since then, its original intent remains unchanged. Google Search Console isn't a place to make direct edits to your site, but it can submit pages to Google's index, check for issues across your domain property, and make sure your site's URLs are healthy. Having said that, Android Studio and Google Search Console are completely unrelated. When it comes to creating apps for Android, the official IDE is Android Studio. Its main purpose is to aid developers in creating, testing, and debugging Android applications.

3.2. Tips for Android Studio in Google Search Console

- **App Indexing:** You can use Google Search Console to associate your Android app with your website. This allows Google to index and display deep links to your app's content in search results when relevant.
- **Mobile Usability:** Google Search Console provides a Mobile Usability report that highlights any mobile usability issues on your website. Ensuring that your website is mobile-friendly is crucial for a positive user experience, as well as for search rankings on mobile devices.
- **Performance Analysis:** While primarily designed for websites, you can still use Google Search Console to monitor the performance of web pages associated with your Android app. This can help you identify any issues that may affect your app's visibility and user experience.
- **Rich Results:** Google Search Console offers insights into the rich results generated by your website. Rich results include features like app deep links, app install buttons, and more. Monitoring and optimizing these rich results can improve your app's visibility in search results.

3.3. Advantages of Android Studio

- **Widely Used Development Environment:** Android Studio is the official Integrated Development Environment (IDE) for Android app development, backed by Google. It is widely adopted by developers and has a large community of users, making it easier to find resources, tutorials, and support.
- **Official Support and Updates:** As the official IDE, Android Studio receives regular updates and improvements directly from Google. This ensures that you have access to the latest features, security patches, and performance enhancements.

- **Rich Set of Tools and Features:** Android Studio provides a comprehensive set of tools and features specifically designed for Android app development. It includes a powerful code editor, debugging tools, layout editor, performance profilers, and built-in emulators to test your app on different virtual devices.
- **Efficient Code Editing:** Android Studio offers smart code completion, code refactoring, and easy navigation features that significantly enhance the development workflow. It helps you write code faster, reduces errors, and improves overall productivity.
- **Advanced Debugging Capabilities:** Debugging is an essential part of the development process, and Android Studio provides robust debugging tools. You can set breakpoints, inspect variables, step through code, and track the flow of your application to identify and fix issues effectively.
- **Seamless Integration with Android SDK:** Android Studio seamlessly integrates with the Android Software Development Kit (SDK) and supports all the latest Android platform versions and APIs. It allows you to leverage the full potential of the Android platform and access a wide range of APIs and libraries.
- **Emulator and Device Testing:** Android Studio offers a built-in emulator that allows you to test your app on virtual Android devices with different configurations. Additionally, you can connect physical Android devices to the IDE for real-time testing and debugging, ensuring compatibility across various devices.
- **Performance Profiling:** Android Studio includes powerful profiling tools that help you optimize your app's performance. You can analyze CPU usage, memory allocation, network activity, and other metrics to identify bottlenecks and improve the efficiency of your app.
- **Support for Multiple Programming Languages:** Android Studio supports multiple programming languages, including Java and Kotlin. This flexibility allows you to choose the language that best suits your preferences and project requirements.
- **Collaboration and Version Control:** Android Studio integrates seamlessly with version control systems like Git, making it easier to collaborate with other developers and manage code repositories effectively.

3.4. Disadvantages of Android Studio

- **Steeper Learning Curve:** Android Studio is a professional IDE with a rich set of features, which can make it complex and overwhelming for beginners. The learning curve may be steeper compared to simpler development tools, requiring

additional time and effort to become proficient.

- **Resource Intensive:** Android Studio is a resource-intensive software that demands a relatively powerful computer to run smoothly. It can consume a significant amount of memory, CPU, and storage, which may pose challenges if you are using a low-end or older computer.
- **Initial Setup and Configuration:** Setting up Android Studio and configuring the development environment can be a bit involved, especially for individuals who are new to software development. It requires installing the correct SDKs, tools, and dependencies, which can be time-consuming and error-prone.
- **Emulator Performance:** While Android Studio provides built-in emulators for testing apps, they can be sluggish and slow, especially when running on lower end machines. This can impact the testing and debugging experience, making it less responsive and efficient.
- **Limited Platform Compatibility:** Android Studio is primarily designed for developing Android applications and does not support other platforms like iOS or Windows Phone. If you want to develop cross-platform applications, you may need to explore additional tools or frameworks.
- **IDE Complexity:** With its extensive feature set, Android Studio can be overwhelming for beginners. Navigating through its various menus, settings, and options may take some time to get used to. It's important to invest time in understanding the IDE to maximize its capabilities.
- **Continuous Updates:** While regular updates are beneficial for staying up-to date with new features and improvements, they can also introduce compatibility issues. Updating Android Studio or its associated components may sometimes lead to conflicts with existing project configurations or libraries.
- **Lack of Visual Design Tools:** Android Studio provides a layout editor for designing user interfaces, but it may not be as intuitive or feature-rich as dedicated visual design tools. In some cases, you may need to supplement it with other tools or hand-code certain UI elements.

IV. SYSTEM SPECIFICATION

4.1. Hardware Requirement

Hardware is the physical parts or components of a computer such as the monitor, keyboard, computer, data storage, graphics card, sound card, and mother board.

Table 1: Hardware Requirement

| | |
|------------------|---|
| Processor | Intel i5 7th gen |
| Hard Disk | 8 GB of available disk space minimum (IDE + Android SDK + Android Emulator) |
| RAM | 8 GB RAM |

4.2. Software Requirement

Computer software or simply software is a part of a computer system that consist of data or computer system that helps in the development of this system.

Table 2: Software Requirement

| | |
|-------------------------|--------------------------------|
| Operating System | 64-bit OS – Microsoft Windows. |
|-------------------------|--------------------------------|

4.3. Software Features

- Project Structure
- Code Editor
- Layout Editor
- Gradle Build System
- Emulator and Device Testing
- Performance Profiling
- Debugging and Logging
- Version Control Integration
- Code Templates and Snippets
- Extensibility and Plugins
- **Project Structure:** Android Studio allows you to create and manage projects easily. It provides project templates, allowing you to start with a basic structure for different types of apps.
- **Code Editor:** Android Studio offers a powerful code editor with features like syntax highlighting, code completion, code refactoring, and error checking. It supports multiple programming languages, including Java and Kotlin.
- **Layout Editor:** The Layout Editor in Android Studio provides a visual interface for designing app layouts. It allows you to drag-and-drop UI components and preview how they will appear on different screen sizes and orientations.
- **Gradle Build System:** Android Studio utilizes the Gradle build system, which automates the process of building, testing, and packaging Android apps. It manages dependencies, compiles code, and generates the APK (Android Package) file.
- **Emulator and Device Testing:** Android Studio includes an emulator that simulates different Android devices and versions, allowing you to test your app without needing physical devices. It also supports USB debugging for testing on real devices.
- **Performance Profiling:** The Profiler tool in Android Studio helps you analyze and optimize your app's performance. It provides insights into CPU, memory, and network usage, allowing you to identify and fix performance bottlenecks.
- **Debugging and Logging:** Android Studio offers robust debugging capabilities, allowing you to set breakpoints, inspect variables, and step through code execution. It also provides logging tools for capturing and analyzing runtime logs.
- **Version Control Integration:** Android Studio has built-in support for version control systems like Git. It allows you to manage your project's source code, commit changes, and collaborate with other developers using popular version control

workflows.

- **Code Templates and Snippets:** Android Studio provides a range of code templates and snippets to speed up development. These templates offer pre written code for common tasks, such as creating activities, fragments, or adapters.
- **Layout Inspector:** The Layout Inspector tool helps you inspect the UI hierarchy and properties of your running app. You can analyze the layout structure, view properties, and debug UI issues using this tool.
- **Extensibility and Plugins:** Android Studio is highly extensible through plugins. You can install and use various plugins to enhance functionality, integrate with third-party libraries, and customize the IDE to suit your development needs.

Websites for Android Studio

A website, often spelt "web site," is an online publication consisting of a network of interconnected web pages and other content published on a shared server and accessible via an established domain name. Some well-known examples include Amazon.com, Google.com, and Wikipedia.org.

In its entirety, the World Wide Web consists of all websites that are accessible to the public. Some websites, like an internal company website, are private and can only be visited by authorised users on a private network.

Whether it's for news, education, business, entertainment, or social networking, most websites have a specific focus. The navigation of a website, which typically begins with a home page, is guided by hyperlinks between pages.

Desktop computers, notebooks, tablets, and smartphones are just some of the many devices that users can use to view websites. The software that runs on these gadgets is known as a web browser.

Static and interactive websites are the two main types. As a subset of the Web 2.0 network, interactive websites facilitate two-way communication between site administrators and their users. Sites that are static only collect data and do not enable users to interact with the content in any way. Some websites are made by hobbyists, provide information, or are just for fun.

4.4. Many Websites do Aim to Make Money, using one or more Business Models, Including

- Posting interesting content and selling contextual advertising either through direct sales or through an advertising network.
- E-commerce: products or services are purchased directly through the website · Advertising products or services available at a brick-and-mortar business · Freemium: basic content is available for free but premium content requires a payment (e.g., WordPress website, it is an open-source platform to build a blog or website.)

4.5. Tools Used in Android Studio

Android Studio is the official integrated development environment (IDE) for Android app development. It is a comprehensive software package designed to assist developers in creating, testing, and debugging Android applications. Android Studio provides a range of tools and features that streamline the app development process and enhance productivity. Android app development using Android Studio software involves the process of creating, designing, coding, testing, and deploying Android applications. Android Studio is the primary tool used by developers to build Android apps, and it provides a robust and efficient environment for the development process. Here are some of the key tools used in Android Studio:

- **Android Virtual Device (AVD) Manager:** The AVD Manager allows you to create and manage virtual Android devices for testing your app. You can create emulators with different configurations, such as screen size, Android version, and hardware characteristics.
- **Android Debug Bridge (ADB):** ADB is a command-line tool that allows you to communicate with a connected Android device or emulator. It provides various commands to install and uninstall apps, transfer files, capture screenshots, and more.
- **Profiler:** The Profiler tool helps you monitor and analyze the performance of your app. It provides real-time data about CPU, memory, network usage, and other system metrics. You can identify and optimize performance bottlenecks in your app using this tool.
- **Android Asset Packaging Tool (AAPT):** AAPT is a command-line tool used by Android Studio to compile resources (such as XML layouts, images, and strings) into the binary format that can be used by the Android operating system.
- **Android Device Monitor:** This tool allows you to monitor and debug Android devices connected to your computer. It provides features like logcat viewer, file explorer, process viewer, and more.
- **Layout Inspector:** The Layout Inspector tool helps you inspect the UI hierarchy and properties of your running app. You can analyze the layout structure, view properties, and debug UI issues using this tool.

V. RESULTS AND DISCUSSION

5.1. Input Design

STEP 1: Install Android Studio

STEP 2: Go to <http://developer.android.com/sdk/index.html> to download Android Studio.

STEP 3: Use the installer to install Android Studio following its instructions.



Figure 1 - Android Studio IDE

STEP 3: In Welcome to Android Studio Setup Window click Next to continue.



Figure 2 - Android Studio Setup

STEP 4: Open Android Studio.

STEP 5: Under the "Quick Start" menu, select "Start a new Android Studio project."

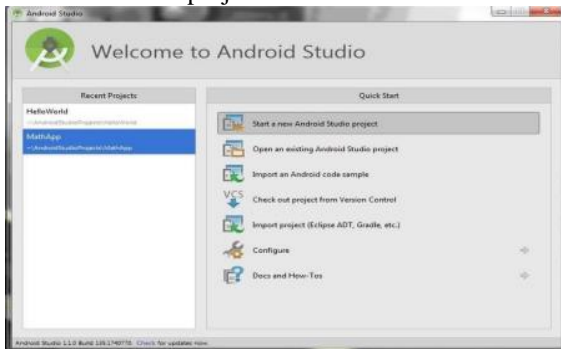


Figure 3 - Android Studio Front Page

STEP 6: On the "Create New Project" window that opens, name your project "HelloWorld".

STEP 7: If you choose to, set the company name as desired*.

STEP 8: Note where the project file location is and change it if desired.

STEP 9: Click "Next."

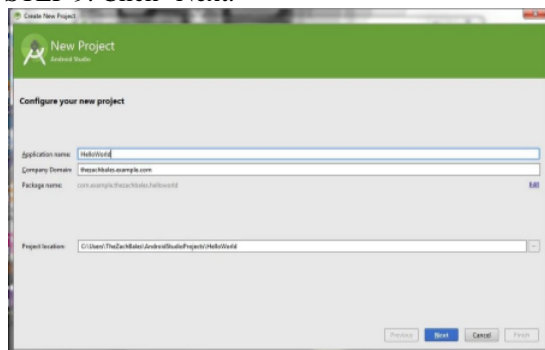


Figure 4 - Create New Project

STEP 10: Make sure on that "Phone and Tablet" is the only box that is checked.

STEP 11: If you are planning to test the app on your phone, make sure the minimum SDK is below your phone's operating system level.

STEP 12: Click "Next."

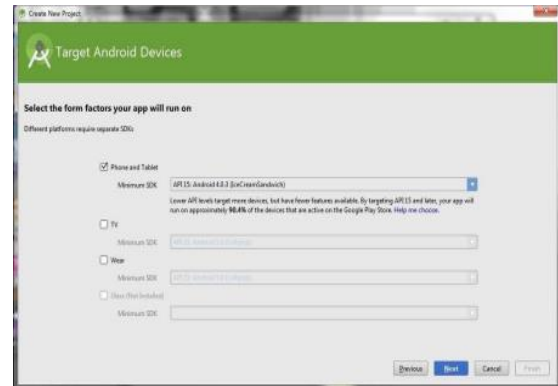


Figure 5 - Target Android Device

5.2. Output Design

STEP 13: Select "Blank Activity."

STEP 14: Click "Next."

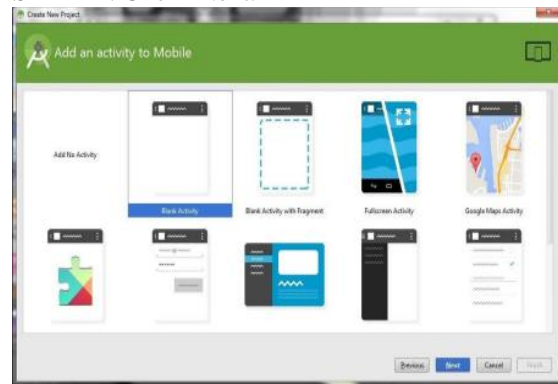


Figure 6 - Add a Activity to Mobile

STEP 15: Leave all of the Activity name fields as they are.

STEP 16: Click "Finish."

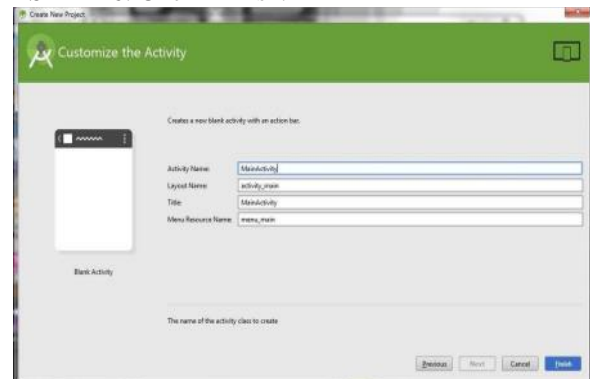


Figure 7 - Customize the Activity

VI. CONCLUSION

Android Studio is an essential tool for Android development and will be a valuable asset during my internship. It offers a wide range of features and tools that

simplify the development process and enhance productivity. Android Studio provides a comprehensive integrated development environment (IDE) for Android app development. It allows you to create, manage, and navigate through your app project structure efficiently. The Layout Editor enables you to design and modify the user interface of your app visually. The code editor offers powerful editing capabilities and supports multiple programming languages, such as Java and Kotlin. Android Studio utilizes the Gradle build system to automate the build process and handle dependencies. The Android Emulator and device testing tools facilitate app testing on virtual and physical devices. During my internship, take advantage of the resources available for Android Studio, such as online documentation, tutorials, and the Android developer community. This will help you familiarize yourself with the various features and best practices for Android app development. Remember to seek guidance from your mentor or supervisor whenever you encounter challenges or have questions. Android Studio, combined with your enthusiasm and dedication, will assist you in successfully completing my internship and gaining valuable hands-on experience in Android development.

In Journal of Physics: Conference Series, 2214(1), 1-7. IOP Publishing.

REFERENCE

- [1] Verma, N., Kansal, S., & Malvi, H. (2018). Development of native mobile application using android studio for cabs and some glimpse of cross platform apps. *International Journal of Applied Engineering Research*, 13(16), 12527-12530.
- [2] Ejyiyi, C.J., Deng, J., Ejyiyi, T.U., Salako, A.A., Ejyiyi, M.B., & Anomihe, C.G. (2021). Design and Development of Android Application for Educational Institutes. *In Journal of Physics: Conference Series*, 1769(1), 1-8. IOP Publishing.
- [3] Asritha, R., & Arpitha, R. (2020). A Survey Paper on Introduction to Android and Development Process. *International Research Journal of Engineering and Technology [en línea]*, 7(6), 2395-0056.
- [4] Asritha, R., & Arpitha, R. (2020). A Survey Paper on Introduction to Android and Development Process. *International Research Journal of Engineering and Technology [en línea]*, 7(6), 2395-0056.
- [5] Tewari, A., & Singh, P. (2021). Android App Development: A Review. *Journal of Management and Service Science (JMSS)*, 1(2), 1-6.
- [6] Sarkar, A., Goyal, A., Hicks, D., Sarkar, D., & Hazra, S. (2019). Android application development: A brief overview of android platforms and evolution of security systems. *In IEEE Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, 73-79.
- [7] Shaheen, J.A., Asghar, M.A., & Hussain, A. (2017). Android OS with its architecture and android application with Dalvik virtual machine review. *International Journal of Multimedia and Ubiquitous Engineering*, 12(7), 19-30.
- [8] Okedirán, O.O., Arulogun, O.T., Ganiyu, R.A., & Oyeleye, C.A. (2014). Mobile operating systems and application development platforms: A survey. *International journal of advanced networking and applications*, 6(1), 2195-2201.
- [9] Azis, N.U.R. (2022). Android based Library Application Development Using SQLite Crud. *International Journal of Multi Science*, 3(01), 71-79.
- [10] Maryam, S., & Purwono, A. (2022). Android application development for push notification feature for Indonesian space weather service based on Google Cloud Messaging.