R. P. Shaha University

- an institution of Kumudini Welfare Trust of Bengal (BD) Ltd.

An Android Application

For

Algorithm Visualization And Analysis

Submitted by

MD. SAKIB HOSSAIN

ID: 18300021

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

NAIMA ISLAM NODI

Lecturer

Department of Computer Science and Engineering,

R.P. Shaha University

NARAYANGANJ, 2023

CERTIFICATION

This is to certify that the work presented in the project is an outcome of the investigation carried out by **Md. Sakib Hossain (18300021)** under the supervisor of **Naima Islam Nodi**, Lecturer, Department of Computer Science and Engineering, R. P. Shaha University, Narayanganj. It is declared that this project has been submitted only for the award of graduation.

Submitted by:

____

MD. SAKIB HOSSAIN

ID: 18300021

Department of CSE

R. P. Shaha University

Supervised by:

____

NAIMA ISLAM NODI

Lecturer

Department of CSE

R. P. Shaha University

Certificate of Acceptance of the Project

The project titled **"AN ANDROID APPLICATION FOR ALGORITHM VISUALIZATION AND ANALYSIS"** submitted by Md. Sakib Hossain (18300021) Session: 2018, has been accepted as satisfactory in partial fulfillment of the degree of Bachelor of Science in Computer Science and Engineering requirement on 10/05/23.

....

Supervisor

Naima Islam Nodi

Lecturer

Department of Computer Science and

Engineering

R. P. Shaha University, Shitalakhya,
Narayanganj-1400, Bangladesh

....

Head of the Dept.

Dr. Kingkar Prosad Ghosh

Associate Professor and Head

Department of Computer Science and
Engineering

R. P. Shaha University, Shitalakhya,
Narayanganj-1400, Bangladesh

ACKNOWLEDGEMENT

I take this occasion to thank Allah, almighty for blessing me with his grace and tracking my endeavor to a successful culmination. I sincerely thank my caring parents for guiding me throughout my life, including my project. I would like to thank the Department of computer science and Engineering, R. P. Shaha University, Shitakhya, Narayanganj-1400, Bangladesh, for supporting this project. Special Thanks to the Chairman, our head of dept. Dr. Kingkar Prosad Ghosh sir for guide and support.

I would like to express my heartiest gratitude to my advisor, Naima Islam Nodi for her constant support, and inspiration. She provided me with my Bachelor Project study. Patience, motivation, supervision, and vast knowledge were my thorough guide till the end.

ABSTRACT

The Android application titled "An Android Application for Algorithm Visualization and Analysis" falls under the category of educational applications. The main objective of this project is to eliminate the fear factor associated with learning algorithms and to foster an interest in the topic. Algorithms are often considered a challenging topic to comprehend, and many students tend to avoid it, which is unfortunate as it is a crucial subject in computer science. Not just top tech companies like Google, Amazon, and Facebook, but every company prefers candidates who possess a strong understanding of algorithms, rather than those who have simply memorized them. They need individuals who can grasp how algorithms work and where to apply them effectively. Therefore, this application helps users visualize how algorithms work, step by step, by analyzing each code and illustrating how data performs with each operation. Finding relevant and reliable information on algorithm visualization or analysis on the internet can be challenging, as many resources are paid, which students cannot afford. This application has been designed to be user-friendly, easily accessible, and comprehensible for students.

Keywords: Java, Algorithm, Data Structure, Visualization, Analysis.

TABLE OF CONTENTS

LIST OF IMAGES

LIST OF TABLES

Chapter 1

Introduction

In this chapter, we will discuss the essential components of an Android project overview. Firstly, we will provide a summary to give an overview of the project to readers. Next, we will discuss the project motivation that highlights the reasons why the project was initiated. This will help readers understand the purpose of the project and its significance.

Moreover, we will also talk about the project objectives, which define the goals that the project aims to achieve. It is crucial to identify these objectives at the outset to ensure that the project is aligned with its purpose. Additionally, we will elaborate on the project features that describe the functions and capabilities that the project will provide to its users. These features must be closely linked to the project objectives and designed to achieve them.

Furthermore, the project problem statement will be discussed, which refers to a concise and clear description of the problem faced while creating the project. This statement identifies the challenge and proposes an approach to overcome it.

Finally, we will touch on the social impact of the Android project, which refers to the positive or negative effects that it may have on society or specific groups of people. It is crucial to consider the broader implications of the project, as it will impact the wider community. By taking into account the social impact, the project can be designed to bring about positive change.

Overall, this chapter will provide readers with a comprehensive understanding of the essential components that must be considered when creating an Android project. By following these guidelines, developers can create impactful projects that meet their objectives and have a positive impact on society.

1.1. Project Overview

An algorithm is a set of instructions or steps that are followed in a specific order to solve a problem or accomplish a task. Algorithms can be expressed in any language, including natural languages, pseudocode, and programming languages. They are used in a wide range of fields, including computer science, mathematics, operations research, and engineering. In computer science, algorithms are used to perform various tasks such as sorting, searching, and pattern matching, as well as finding the shortest path between two points in a graph. Algorithms can be divided into two main categories: deterministic algorithms, which always produce the same output given the same input, and randomized algorithms, which make use of random numbers to produce a result.

Algorithms are often analyzed and compared based on their time and space complexity, which describe the number of resources (e.g. time or memory) required to run the algorithm as a function of the size of the input. This analysis helps to understand how the algorithm performs and how it scales when applied to larger inputs.

Learning algorithms are important for several reasons:

1. **Problem–solving:** Algorithms provide a step-by-step approach to solving complex problems, making it easier to understand and implement solutions.
2. **Efficiency:** Algorithms can be designed to be efficient in terms of time and space complexity, which is crucial when working with large data sets or in real-time applications.
3. **Automation:** Algorithms can automate repetitive tasks and perform them faster and more accurately than humans, which can save time and resources.
4. **Machine Learning:** Algorithms are the backbone of machine learning and artificial intelligence. Machine learning algorithms are used to train models to perform tasks such as image recognition, natural language processing, and prediction, which have a wide range of applications in industry, medicine, and other fields.
5. **Data Analysis:** Algorithms are also used for analyzing data, for example, clustering, classification, and anomaly detection are used to extract meaningful information from raw data.
6. **Innovation:** Algorithms are used to come up with new ideas and innovations. For example, genetic algorithms are used in engineering to optimize designs and in finance to optimize portfolios.

Therefore, understanding the principles and techniques behind different algorithms allows you to make better decisions in the design and development of software systems, and also plays a key role in many fields like data science, AI, computer science, and many more.

The "Algorithm Visualization and Analysis" is an Android application that has been developed using Java programming language and XML markup language. The application is designed to assist learners in understanding various algorithms by providing visualization and analysis features.

The application comprises different sections for each algorithm type, including Sorting algorithms, Searching Algorithms, Graphs, Trees, Strings, and more. Each section contains various algorithms that are associated with the algorithm type. For example, the Sorting section includes algorithms such as Selection sort, Bubble sort, Merge sort, and others.

Every algorithm in the application is accompanied by a brief description and complexity (Time & Space). Additionally, every specific algorithm has visualization and analysis features that enable users to observe and understand its working procedure. For instance, in the Sorting section, each algorithm demonstrates how to sort. Some algorithms offer faster and more efficient solutions, while others provide the worst solutions. However, just knowing this information is insufficient to fully comprehend the algorithms. To better understand them, learners need to know how algorithms work internally and what types of operations they can perform. Through analysis, users can observe how the operations are performed internally and visually see what the algorithm can do.

The project's objective is to help learners understand algorithms better by providing both visual and analytical representations of their functionality. This way, users can learn the theory and facts behind the algorithms simultaneously, making it easier to comprehend the concepts. By visualizing and analyzing the algorithms, users can get a better grasp of how the algorithms work internally and the types of operations they can perform.

In conclusion, the "Algorithm Visualization and Analysis" Android application is designed to provide an immersive learning experience for algorithms. The application's visualization and analysis features allow users to observe and comprehend the internal workings of algorithms better. By utilizing this application, learners can easily understand complex algorithms more engagingly and interactively.

1.2. Motivation

The motivation behind creating this project stemmed from my desire to work at a top tech company. Despite participating in online and offline coding competitions, I realized that I had a weakness in algorithms. With my skills in Java and Android, I began searching for ways to improve my understanding of algorithms [1] [2].

I came across an application on the Play Store called "Algorithms," which provided helpful visualizations. However, I wanted to gain a deeper understanding of the internal workings of algorithms, which I couldn't find online. I then purchased a paid course called "Logic Mojo" from India, which provided code analysis and visualizations [2]. While studying, I learned about a few Google engineers who received offers from the company because of their project that visualized a few algorithms. This inspired me to think about how I could use my skills to create a unique and helpful project that could make a significant impact in the future. As a result, I came up with the idea of creating an algorithm visualizer and analysis application that could provide both visualization and analysis features. This project would be unique and extremely helpful to those looking to improve their understanding of algorithms.

With this idea in mind, I began working on the project and developed the "Algorithm Visualization and Analysis" Android application. My goal was to create an immersive and interactive learning experience for algorithms, which would allow users to comprehend their internal workings through visual and analytical representations.

Overall, my motivation for creating this project was to improve my understanding of algorithms while also helping others achieve a deeper comprehension of these complex concepts. I believe that this application can make a significant positive impact on the field of computer science and assist individuals in achieving their career goals.

1.3. Objective

The main object of this project is to make the algorithm easier to understand. A person must know Java to understand the Algorithm from this application. Everything will be offline that’s for anyone can learn the algorithm anytime. We will provide problems for each algorithm so that they will have implementation skills. How they can implement when they learn algorithms. Shortly we can say the objective of the application is,

- Visualize Algorithms
- Analyze Algorithms

1.4. Feature

Project Features provide a complete understanding of the problem or opportunity that the project aims to address. Here are some of the features the algorithm visualization analysis project has given below:

- Onboarding Screen for a brief overview of the project
- Every algorithm has a description and related task(problems)
- User can visualize the algorithm
- User can analyze algorithm

1.5. Problem Statement

The Algorithm Visualization Analysis project I undertook turned out to be a very complex and challenging task. As I began working on the project, I found that there were no good resources available to help me with it. I tried to reach out to experienced Android developers and expert programmers to get some ideas on how to proceed, but unfortunately, they didn't respond to my requests for assistance [3]. I felt quite disappointed and demotivated.

However, I did not give up hope and continued my search for resources to help me complete the project. I joined several online communities such as Facebook groups, Telegram groups, and Discord groups, but still couldn't find the support I needed.

Eventually, I found a resource that told me that what I was attempting to do was very difficult and that I would need to invest in a paid course to learn the necessary skills. The course covered physics-based animation, which would be instrumental in creating the visualization section of the project [5]. When it came to the analysis part of the project, I faced another hurdle. I was unable to find any developers who believed it was possible to create a debugging application for a specific algorithm. I was quite shocked by this revelation, and it made me wonder if I would be able to complete the project in time.

Despite the challenge, I persevered and tried to implement a brute-force approach to the analysis part of the project. It took a lot of effort, but I eventually managed to make it work effectively, even if it wasn't as efficient as I would have liked.

Overall, the journey to complete the Algorithm Visualization Analysis project was a difficult one, but I learned a lot along the way. The challenges I faced helped me develop my problem-solving skills, and the project taught me the value of persistence and determination.

1.6. Social Impact

The Algorithm Visualization Analysis android project has the potential to have a significant social impact by making the study of algorithms more accessible and engaging for a wide range of people.

First, by providing a user-friendly, interactive, and dynamic visualization of algorithms, the app can help computer science students and professionals to better understand and analyze algorithms. This can lead to improved performance in their studies and careers and can help to close the skills gap in the technology industry.

Additionally, the app can also make the study of algorithms more accessible to people who are not computer science professionals but are interested in learning about algorithms. This can include people who are interested in computer science as a hobby, as well as people who are interested in learning about algorithms for other reasons, such as data analysis, cryptography, or artificial intelligence.

Another potential impact of the Algorithm Visualization Analysis Android project is that it can make it easier for people with learning disabilities or cognitive impairments to understand and analyze algorithms. The app's user-friendly interface and interactive visualizations can help to make the study of algorithms more accessible and engaging for these individuals.

Lastly, the app's ability to visualize complex algorithms in real time could be used in various fields, such as education, research, and business, where understanding and analyzing algorithms is important. As a result, the Algorithm Visualization Analysis android project could have a positive impact on society by making the study of algorithms more accessible and engaging, and by helping to close the skills gap in the technology industry.

Conclusion: In this whole chapter, I have discussed a general understanding of the project’s reason to build the project. Its achievement and functionality. The problem I faced and the impact that can occur over the project.

Chapter 2

Background Study

The background of a study includes a review of the existing literature on the area of any research, leading up to the topic. This is also a review of current information surrounding the issue, previous studies on the issue, and relevant history on the issue. Some topics of background studies of this project have been discussed in this chapter.

2.1. Identifying the problems

The problem that the Algorithm Visualization Analysis Android project aims for is to reduce the difficulty of algorithms that many students and professionals face problem in understanding and analyzing algorithms. Despite the importance of algorithms in computer science and other fields, many people struggle to understand the underlying concepts and mechanics of these complex algorithms.

2.2. Research existing solutions

When it comes to learning algorithms, traditional resources such as textbooks and lectures can be difficult to understand due to their dense and technical language and static diagrams. While software and apps that can visualize algorithms exist, they are often limited to certain algorithms and lack user-friendliness. Furthermore, there is currently no Android application that is capable of analyzing algorithms. The idea for this project was inspired by a paid Indian website that focuses on helping individuals get hired by top tech companies, as well as the story of Clément Mihailescu, a former Google engineer who landed his job thanks to his algorithm visualizer project [6] [2].

2.3. Project Scope

The Algorithm Visualization Analysis Android project aims to improve upon existing solutions by providing a user-friendly, interactive, and dynamic visualization of algorithms. By allowing users to see the inner workings of algorithms in real-time and in a visual format, the project aims to make the study of algorithms more accessible and engaging.

2.4. Target audience

The target audience for the Algorithm Visualization Analysis android project includes computer science students and professionals, as well as anyone else who is interested in learning about algorithms. The app will be designed with a user-friendly interface that makes it easy to use and understand, regardless of prior experience or knowledge of algorithms.

2.5. Technology Research and Selection

The Algorithm Visualization Analysis Android project is a complex undertaking that requires various technologies and tools to ensure its successful development. These include the Android SDK, Java programming language, and Code View for displaying code. Additionally, Meow Bottom Navigation is a useful tool that provides a customizable bottom navigation bar with an animated tab indicator, which will enhance the user experience. Also, implementing animations for the project may present a challenge, which will require the expertise of a skilled developer. Overall, the Algorithm Visualization Analysis Android project is an exciting initiative that will require careful planning, execution, and testing to ensure its success. With the right tools and resources, it has the potential to make a significant impact in the field of algorithm education and help students learn more effectively.

Table 2.1: Platform & API

Platform	Android
Font – End	XML
Back – End	Java
API	Meow Bottom Navigation
API	Code View

2.5.1. Android

Android is a popular open-source operating system developed by Google for mobile devices like smartphones and tablets. It is based on the Linux kernel and uses Java-based programming. Developers can use the Android software development kit (SDK) to create various apps, including games and enterprise-level apps.

One of the main advantages of the Android platform is its flexibility and customization. Users can modify their devices with various themes and launchers, and the OS can run on a wide range of devices, from low-end smartphones to high-end tablets and smartwatches. Additionally, the platform includes built-in apps like a web browser, email client, and calendar, and has an active developer community that provides resources to help create new apps.

Due to its wide range of features and capabilities, Android has a large market share in the mobile operating system market and is the most used mobile OS worldwide. This makes it a crucial platform for businesses and developers to reach a vast audience [7] [8] [9].

2.5.2. XML

XML stands for Extensible Markup Language, which is a markup language similar to HTML that is used for describing data. It was derived from the Standard Generalized Markup Language (SGML), and unlike HTML, XML tags are not predefined. Instead, developers need to define the tags to describe and organize the data.

XML is an ideal choice for storing and organizing data due to its scalability and simplicity. It is also lightweight, making it easy to develop and implement. In Android development, XML is mainly used for implementing UI-related data, and it does not make layouts heavy. This is because XML only contains tags, which are invoked as needed during implementation [10].

2.5.3. Java

Java is a widely-used programming language that is well-suited for building applications that can run on a variety of platforms, including mobile devices. Android, on the other hand, is an open-source platform that provides a framework for developing mobile applications.

Android applications can be developed using the Java programming language and the Android SDK. The Android SDK provides developers with a set of tools and APIs for building applications that can interact with the Android operating system and the hardware of the device.

One of the key benefits of using Java for mobile application development is that it is a managed code platform. This means that the Java Virtual Machine (JVM) manages the execution of the code, providing features such as garbage collection and memory management. These features make it easier for developers to write applications that are more efficient and reliable.

In addition, the Java programming language has a large and active developer community, which provides access to a wealth of libraries and tools that can be used to build mobile applications. The Android platform also has a large and growing community of developers, which provides support and resources for developing applications that can run on Android devices.

Java is a popular choice for mobile application development, particularly for developing applications that can run on Android devices. Its robust set of features, including managed code execution and a large developer community, make it a powerful tool for building efficient and reliable mobile applications [11].

2.5.4. Meow Bottom Navigation

MeowBottomNavigation is an open-source Android library designed to simplify the creation of a customizable bottom navigation bar with an animated tab indicator. Developed by Shahbaz Youssefi, the library is freely available on GitHub and aims to provide developers with an intuitive and straightforward way to incorporate a feature-rich bottom navigation bar into their Android applications.

One of the key benefits of using MeowBottomNavigation is its ease of use and customizability. With support for multiple navigation items, icons, and text, developers can easily create a navigation bar that fits the needs of their app. Additionally, MeowBottomNavigation provides support for badges, allowing developers to add notifications or other visual indicators to their navigation items.

MeowBottomNavigation supports different animation styles to provide users with a smooth and visually appealing experience. The library is built on top of the AndroidX library and is compatible with Material Design guidelines, ensuring that the navigation bar looks and feels great on all devices. Furthermore, MeowBottomNavigation supports a minimum SDK version of 16, making it accessible to a wide range of Android devices.

MeowBottomNavigation is an excellent option for developers who want to implement a customizable, easy-to-use bottom navigation bar in their Android applications. Its range of features and support for customization makes it a great tool for developers looking to enhance the user experience of their apps [12].

2.5.5. Code View

The "Code View" refers to a graphical interface that presents source code in a clear and organized way, facilitating the readability and comprehension of the code by developers.

However, the lack of specific information about this "Code View API" makes it challenging to provide a more detailed explanation. Depending on the context of its usage, a "Code View API" may provide features such as syntax highlighting, code folding, line numbering, search and replace, and other editing functionalities. Additionally, it could also allow developers to customize the visual appearance of the code view, including font size and color schemes.

Overall, the "Code View API" can be a powerful tool for developers, enabling them to create user-friendly interfaces for viewing and editing code efficiently [13].

Chapter 3

Methodology

Algorithm Visualization Analysis Android project involves the development of an Android application that allows users to visualize and analyze various algorithms. To ensure the successful completion of this project, a well-defined methodology is necessary. Which will be discussed below.

3.1. Project Definition

The Algorithm Visualization Analysis Android project is an innovative software application that aims to help computer science students, professionals, and enthusiasts understand and analyze algorithms more effectively. The project addresses the challenges that people face in comprehending and analyzing algorithms by providing a user-friendly, interactive, and dynamic visualization of algorithms. The Android SDK and Java programming languages will be utilized for app development, while Code View will be used to enable users to view the app's code. The app will also have a database to store user information. Extensive research and data analysis will be conducted to ensure that the app is intuitive and effective in helping users understand algorithms.

The Algorithm Visualization Analysis Android project aims to make the study of algorithms more accessible and engaging, while also serving as a valuable tool for students and professionals to understand the underlying concepts and mechanics of complex systems. The app has been designed to be user-friendly and responsive, utilizing XML for the front end and Java for the back end. Developing the UI/UX aspect, as well as coding, problem-solving, designing, and animation skills, are critical components of the project. The app will be offline, without any need for a database.

Key features of the Algorithm Visualization Analysis Android app include its full responsiveness, intuitive and user-friendly design, algorithm analysis and visualization, and provision of problems in each algorithm section. The app is free of annoying ads, ensuring a seamless user experience.

By utilizing advanced technology and design principles, the Algorithm Visualization Analysis Android project will serve as an essential tool for individuals seeking to improve their understanding and analysis of algorithms, while also enhancing their skills and knowledge in computer science.

3.2. Design and Development

Designing and developing an algorithm visualization and analysis Android project involves several steps, including: Develop a user interface design that is easy to use and visually appealing. Create wireframes or mockups of the screens to show how the interface will work. Different types of model need to create how the application interact when button clicks. What steps the application follows? How the classes are designed to make this project is done. In Development we write the code for the application using Java. Use Android SDK and Android Studio to create the layout and logic of the application. It is also important to test. Test the application in this level of test called unit test. Code view library such as Syntax View or CodeView can be used to display the code in the application. Algorithm visualization we will use physics based animation no library used in this section.

It's important to note that this is just a basic overview of the steps involved in designing and developing an algorithm visualization and analysis android project, and the specifics will depend on the project's requirements and constraints.

3.2.1. Use Case Diagram

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

- Scenarios in which your system or application interacts with people, organizations, or external systems
- Goals that your system or application helps those entities (known as actors) achieve
- The scope of your system

When to apply use case diagrams

A use case diagram doesn't go into a lot of detail—for example, don't expect it to model the order in which steps are performed. Instead, a proper use case diagram depicts a high-level overview of the relationship between use cases, actors, and systems. Experts recommend that use case diagrams be used to supplement a more descriptive textual use case.

UML is the modeling toolkit that you can use to build your diagrams. Use cases are represented with a labeled oval shape. Stick figures represent actors in the process, and the actor's participation in the system is modeled with a line between the actor and use case. To depict the system boundary, draw a box around the use case itself.

UML use case diagrams are ideal for:

- Representing the goals of system-user interactions
- Defining and organizing functional requirements in a system
- Specifying the context and requirements of a system
- Modeling the basic flow of events in a use case

Use case diagram components

To answer the question, "What is a use case diagram?" you need to first understand its building blocks. Common components include:

Actors: The users that interact with a system. An actor can be a person, an organization, or an outside system that interacts with your application or system. They must be external objects that produce or consume data.

System: A specific sequence of actions and interactions between actors and the system. A system may also be referred to as a scenario.

Goals: The end result of most use cases. A successful diagram should describe the activities and variants used to reach the goal.

Use case diagram symbols and notation

The notation for a use case diagram is pretty straightforward and doesn't involve as many types of symbols as other UML diagrams.

- **Use cases:** Horizontally shaped ovals that represent the different uses that a user might have.
- **Actors:** Stick figures that represent the people actually employing the use cases.
- **Associations:** A line between actors and use cases. In complex diagrams, it is important to know which actors are associated with which use cases.
- **System boundary boxes:** A box that sets a system scope to use cases. All use cases outside the box would be considered outside the scope of that system. For example, Psycho Killer is outside the scope of occupations in the chainsaw example found below.
- **Packages:** A UML shape that allows you to put different elements into groups. Just as with component diagrams, these groupings are represented as file folders.

A use case diagram is a visual representation of the interactions between the users and the system in a software project. It can be a useful tool for understanding the functionality of an algorithm visualization and analysis android project. Here is the use cases for the project given below:

- **View Algorithm:** The user can view a list of algorithms available in the application and select one to view.
- **Visualize Algorithm:** The user can visualize the selected algorithm and see how it works.
- **Analyze Algorithm:** The user can analyze the selected algorithm and see its performance metrics.

Each use case is represented as an oval shape, with the name of the use case written inside. The actors, or the users interacting with the system, are represented as stick figures and are connected to the use cases with lines. It is important to keep in mind that the use case diagram is a high-level representation of the interactions between the users and the system, and it can be refined and expanded as needed.

Result: Use-Case Diagram

The use-case diagram for the Algorithm Visualization Analysis Android project consists of three primary use cases: Visualization, Analysis, and Algorithm. Users can either visualize or analyze an algorithm, but before they can access these functionalities, they must first navigate from the root to select the type of algorithm they wish to work with.

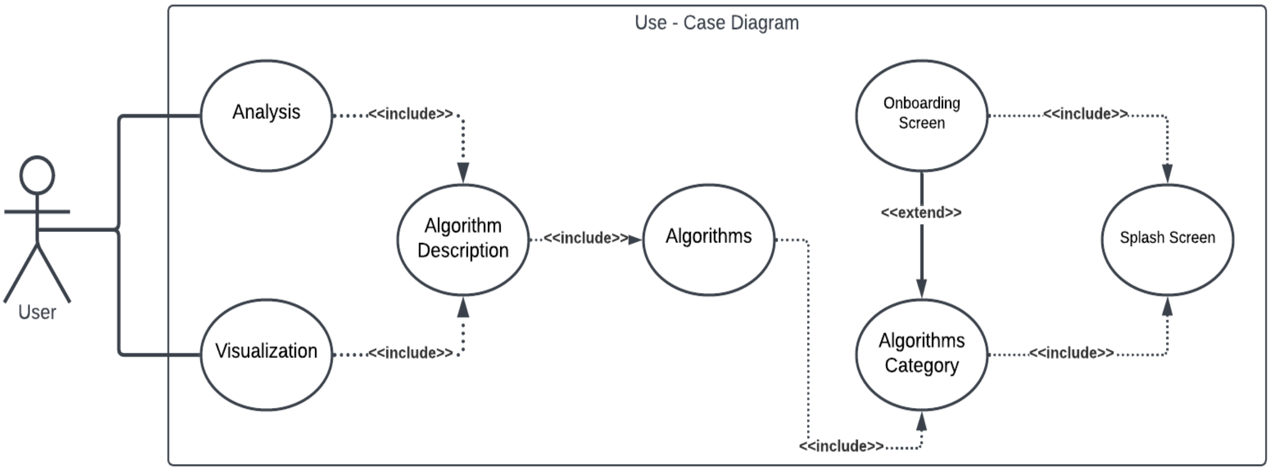


Figure 3.1: Use-Case Diagram

3.2.2. Class Diagram

A class diagram is a visual representation of the classes, interfaces, and objects in a software project, and their relationships. Class diagrams are commonly used in software engineering to illustrate the structure of a system or software application.

Member access modifiers

All classes have different access levels depending on the access modifier (visibility). Here are the access levels with their corresponding symbols:

- Public (+)
- Private (-)
- Protected (#)
- Package / Default (~)
- Derived (/)
- Static (underlined)

Additional class diagram components

Depending on the context, classes in a class diagram can represent the main objects, interactions in the application, or classes to be programmed. To answer the question "What is a class diagram in UML?" you should first understand its basic makeup.

- **Classes**: A template for creating objects and implementing behavior in a system. In UML, a class represents an object or a set of objects that share a common structure and behavior. They're represented by a rectangle that includes rows of the class name, its attributes, and its operations. When you draw a class in a class diagram, you're only required to fill out the top row—the others are optional if you'd like to provide more detail.
  I. **Name**: The first row in a class shape.
  II. **Attributes**: The second row in a class shape. Each attribute of the class is displayed on a separate line.
  III. **Methods**: The third row in a class shape. Also known as operations, methods are displayed in list format with each operation on its own line.
- **Signals**: Symbols that represent one-way, asynchronous communications between active objects.
- **Data types**: Classifiers that define data values. Data types can model both primitive types and enumerations.
- **Packages**: Shapes designed to organize related classifiers in a diagram. They are symbolized with a large tabbed rectangle shape.
- **Interfaces**: A collection of operation signatures and/or attribute definitions that define a cohesive set of behaviors. Interfaces are similar to classes, except that a class can have an instance of its type, and an interface must have at least one class to implement it.
- **Enumerations**: Representations of user-defined data types. An enumeration includes groups of identifiers that represent values of the enumeration.
- **Objects**: Instances of a class or classes. Objects can be added to a class diagram to represent either concrete or prototypical instances.

- **Artifacts**: Model elements that represent the concrete entities in a software system, such as documents, databases, executable files, software components, etc.

It can be a useful tool for understanding the structure and organization of an algorithm visualization and analysis android project. Here is an example of a class diagram for such a project:

1. **Algorithm**: This class represents the algorithms used in the application. It contains properties such as name, type, and data.
2. **Visualization**: This class represents the visualization of the algorithms. It contains methods for displaying the algorithms in different forms such as graph, chart, etc.
3. **Analysis**: This class represents the analysis of the algorithms. It contains methods for calculating performance metrics such as time complexity and space complexity.
4. **User**: This class represents the users of the application. It contains properties such as name, email, and password.

Each class is represented as a rectangle, with the name of the class written at the top. The properties and methods of the class are listed inside the rectangle. The relationships between the classes are represented by lines connecting the classes. The arrowheads on the lines indicate the direction of the relationship. Inheritance is represented by an arrow pointing from the subclass to the superclass.

Result: Class Diagram

The class diagram for the Algorithm Visualization Analysis Android project includes MainActivity, which serves as the primary Java activity, and all other activities perform their respective tasks from here. The class diagram can be divided into three main parts, as discussed previously. The majority of the variables are represented with the '~' special symbol, indicating that they are default variables. It's worth noting that in Java, variables and methods are package-private by default, meaning they can only be accessed within the package in which they are defined.

MainActivity is connected to Sorting through a composition sign, which signifies the whole-part relationship between these two classes. This relationship suggests that the part (i.e., Sorting) cannot exist separately from the whole (i.e., MainActivity). The Helper class, MainAdapter, is linked to MainActivity since it assists with various functionalities.

MainAdapter implements ClickListener, and as a result, is linked to the ClickListener interface sign. The same is true for the Sorting section, which also has a helper class and interface. Sorting is associated with sorting description and is also linked to Selection Sort because both belong to the Sorting category and are therefore associated with each other.

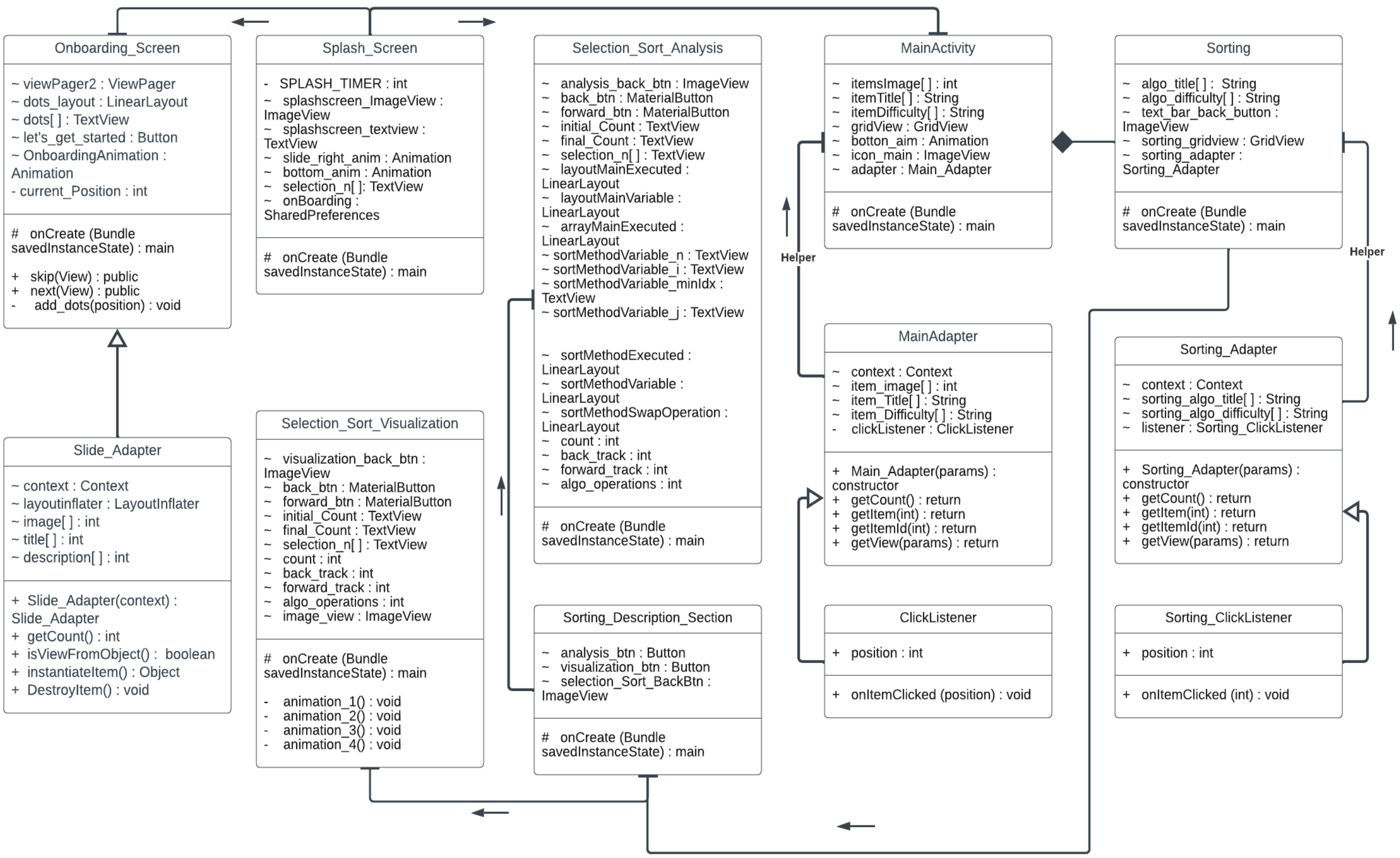

Onboarding_Screen
~ viewPager2 : ViewPager
~ dots_layout : LinearLayout
~ dots[ ] : TextView
~ let's_get_started : Button
~ OnboardingAnimation : Animation
- current_Position : int
# onCreate (Bundle savedInstanceState) : main
+ skip(View) : public
+ next(View) : public
- add_dots(position) : void
Splash_Screen
- SPLASH_TIMER : int
~ splashscreen_ImageView : ImageView
~ splashscreen_textview : TextView
~ slide_right_anim : Animation
~ bottom_anim : Animation
~ selection_n[ ]: TextView
~ onBoarding : SharedPreferences
# onCreate (Bundle savedInstanceState) : main
Selection_Sort_Analysis
~ analysis_back_btn : ImageView
~ back_btn : MaterialButton
~ forward_btn : MaterialButton
~ initial_Count : TextView
~ final_Count : TextView
~ selection_n[ ] : TextView
~ layoutMainExecuted : LinearLayout
~ layoutMainVariable : LinearLayout
~ arrayMainExecuted : LinearLayout
~ sortMethodVariable_n : TextView
~ sortMethodVariable_i : TextView
~ sortMethodVariable_minIdx : TextView
~ sortMethodVariable_j : TextView
~ sortMethodExecuted : LinearLayout
~ sortMethodVariable : LinearLayout
~ sortMethodSwapOperation : LinearLayout
~ count : int
~ back_track : int
~ forward_track : int
~ algo_operations : int
# onCreate (Bundle savedInstanceState) : main
MainActivity
~ itemsImage[ ] : int
~ itemTitle[ ] : String
~ itemDifficulty[ ] : String
~ gridView : GridView
~ botton_aim : Animation
~ icon_main : ImageView
~ adapter : Main_Adapter
# onCreate (Bundle savedInstanceState) : main
Sorting
~ algo_title[ ] : String
~ algo_difficulty[ ] : String
~ text_bar_back_button : ImageView
~ sorting_gridview : GridView
~ sorting_adapter : Sorting_Adapter
# onCreate (Bundle savedInstanceState) : main
Helper
Helper
MainAdapter
~ context : Context
~ item_image[ ] : int
~ item_Title[ ] : String
~ item_Difficulty[ ] : String
- clickListener : ClickListener
+ Main_Adapter(params) : constructor
+ getCount() : return
+ getItem(int) : return
+ getItemId(int) : return
+ getView(params) : return
Sorting_Adapter
~ context : Context
~ sorting_algo_title[ ] : String
~ sorting_algo_difficulty[ ] : String
~ listener : Sorting_ClickListener
+ Sorting_Adapter(params) : constructor
+ getCount() : return
+ getItem(int) : return
+ getItemId(int) : return
+ getView(params) : return
Slide_Adapter
~ context : Context
~ layoutinflater : LayoutInflater
~ image[ ] : int
~ title[ ] : int
~ description[ ] : int
+ Slide_Adapter(context) : Slide_Adapter
+ getCount() : int
+ isViewFromObject() : boolean
+ instantiateItem() : Object
+ DestroyItem() : void
Selection_Sort_Visualization
~ visualization_back_btn : ImageView
~ back_btn : MaterialButton
~ forward_btn : MaterialButton
~ initial_Count : TextView
~ final_Count : TextView
~ selection_n[ ] : TextView
~ count : int
~ back_track : int
~ forward_track : int
~ algo_operations : int
~ image_view : ImageView
# onCreate (Bundle savedInstanceState) : main
- animation_1() : void
- animation_2() : void
- animation_3() : void
- animation_4() : void
Sorting_Description_Section
~ analysis_btn : Button
~ visualization_btn : Button
~ selection_Sort_BackBtn : ImageView
# onCreate (Bundle savedInstanceState) : main
ClickListener
+ position : int
+ onItemClicked (position) : void
Sorting_ClickListener
+ position : int
+ onItemClicked (int) : void

Fig 3.2: Class Diagram

3.2.3. Activity Diagram

An activity diagram is a type of behavioral diagram in UML (Unified Modeling Language) used to model the dynamic aspects of a system. It describes the flow of activities and actions that take place within a system and the relationships and interactions between those activities.

An activity diagram is composed of various elements, including:

- **Initial Node**: represents the starting point of the flow of activities.
- **Final Node**: represents the end of the flow of activities.
- **Activity Nodes**: represent the specific actions or tasks that are performed in the flow of activities.
- **Transitions**: represent the flow of control from one activity node to another. They are shown as arrows connecting activity nodes.
- **Decision Nodes**: represent decisions that need to be made in the flow of activities. They are shown as diamond-shaped nodes.

Activity diagrams are useful for modeling the dynamic behavior of a system, as well as for defining the flow of activities and interactions between activities. They are used in a variety of applications, including business process modeling, software development, and systems engineering.

Result: Activity Diagram

In this activity, each section is defined by class and the activities each class will perform are given on the activity diagram. The given activity diagram (Fig3.3) shows that the flow of activities and actions that take place within the application. When the user pressed on the application the Splash_Screen class come with in the Splash_Screen multiple activities can occur either Onboarding_Screen or Main_Activity. Onboarding_Screen occur once when the application first executed else go to Main_Activity. On Main_Activity. Main_Activity contains different types of algorithms. Since this class contains multiple activities we should pick one of them. Sorting also contains different types of sorting algorithms. So, let pick one algorithm Selection_Sort. Selection_Sort contains visualization and analysis activity. Each algorithm will contain the same activities as Selection_Sort have. After visualization or analysis no other action will perform. So, we end the activity.

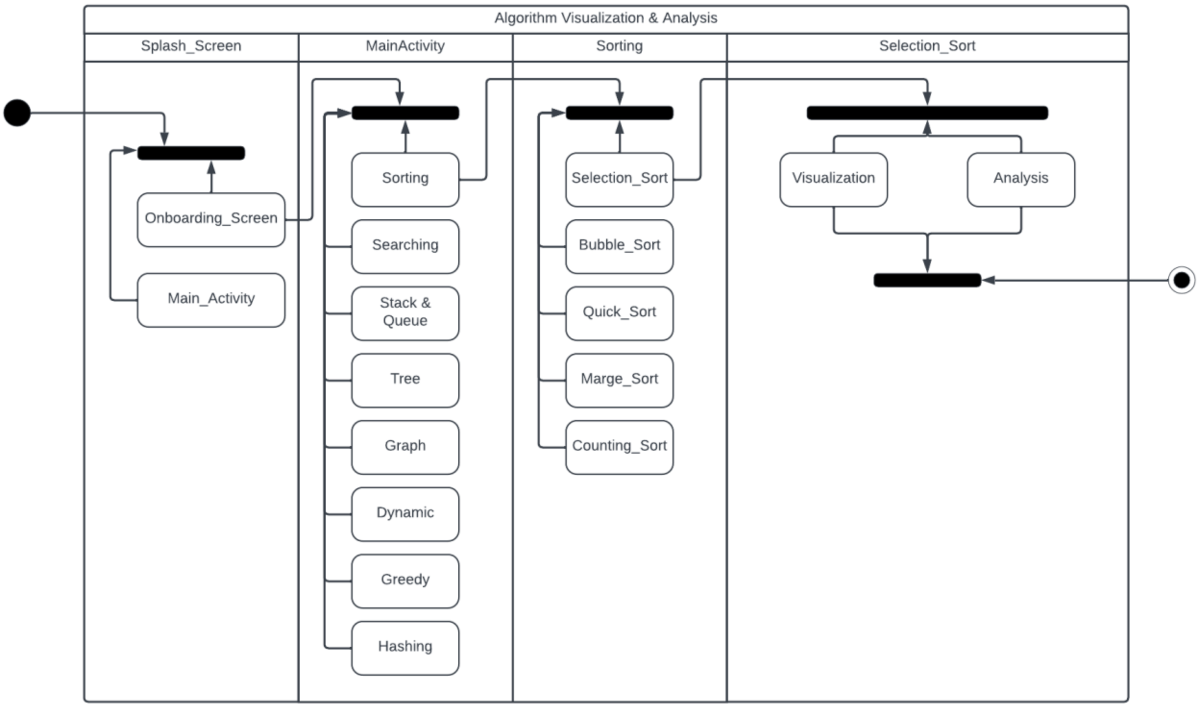


Fig 3.3: Activity Diagram

Conclusion: This could be stated that the visualization accurately displayed the steps taken by algorithms. The project was easy to use but it's to use more functionality so that users can learn more in a short time. Need more research on the topic to make it better and more useful

Chapter 4

Testing

The problem that the Algorithm Visualization Analysis android project aims for is to reduce the difficulty of algorithms that many students and professionals face problem in understanding and analyzing algorithms. Despite the importance of algorithms in computer science and other fields, many people struggle to understand the underlying concepts and mechanics of these complex algorithms.

Testing is one of the most important phases of the software development life cycle (SDLC). Testing can expose most of the defects existing in the present system. And therefore, provides a way of reducing defects in a system. Testing is done at various levels to ensure the quality of the software. Here we going to talk about UI testing called Integration Testing and Unit Testing.

UI testing and Unit testing are two common types of testing in software development. UI testing (User Interface testing) refers to testing the graphical user interface of a software application to ensure that it meets the specified requirements and behaves correctly. It involves testing the look, feel, and behavior of the application from the user's perspective, including testing the buttons, forms, dialog boxes, and other elements that the user interacts with. UI testing helps to catch any issues or bugs in the user interface, such as incorrect layout, missing text, or broken links.

Unit testing, on the other hand, is a software testing method in which individual units or components of a software application are tested in isolation from the rest of the application. The goal of unit testing is to validate that each unit of the software application is working as intended, and to identify any issues or bugs before they impact the larger system. Unit tests are usually automated, and they are run frequently during development to catch any problems early.

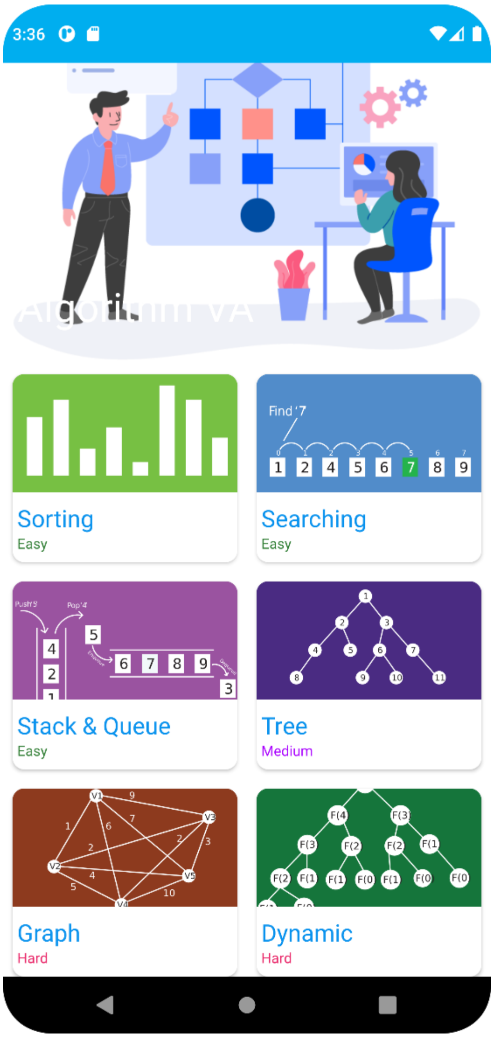
4.1 Integration Testing

Integration testing is a software testing method where individual units or components of a system are combined and tested as a group to determine if they function correctly together. It is important because it helps to detect and resolve issues that may arise when different components interact with each other, which are often not seen during unit testing. Integration testing helps to ensure that the system as a whole function as expected and meets the requirement specifications, which helps to increase the overall quality and reliability of the system. Integration test is also called UI testing. So, Tested UI’s are given below:

Splash Screen: This screen will appear when the application launched. Used theme to perform splash screen operation. Implemented most efficient approach. Splash screen represents first impression of a application.

Fig - 4.1: Splash Screen

Algorithm Category (Dashboard)**:** Where All algorithms categories will be shown in the dashboard section. Which types algorithms the user wants to learn will show in the dashboard.



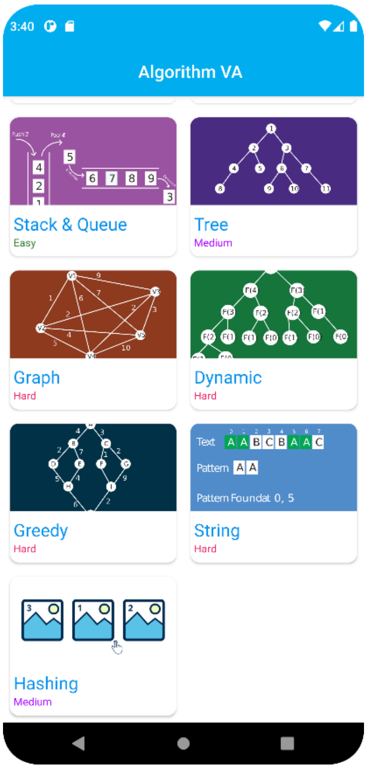
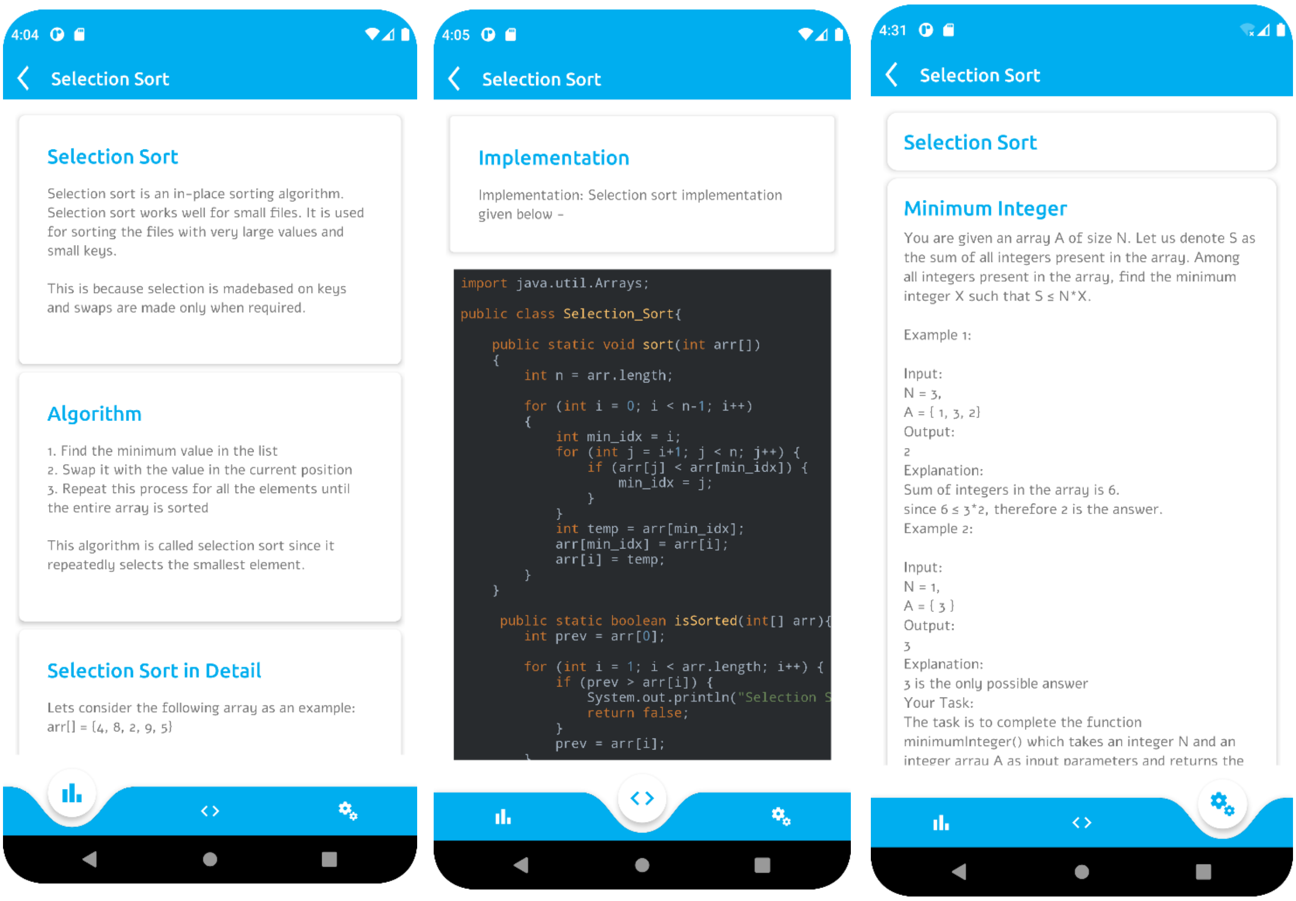


Fig - 4.2: Algorithm Category

Algorithm Description (Specific):

Algorithm description which describes specific algorithm description. This section is divided into 3 sections. 1st section is about the description of the algorithm. The next section is about whether the learner wants to learn algorithms visually or analytically. Lastly, the problem section for practice.



Selection Sort

Selection Sort

Selection sort is an in-place sorting algorithm. Selection sort works well for small files. It is used for sorting the files with very large values and small keys.

This is because selection is madebased on keys and swaps are made only when required.

Algorithm

1. Find the minimum value in the list
2. Swap it with the value in the current position
3. Repeat this process for all the elements until the entire array is sorted

This algorithm is called selection sort since it repeatedly selects the smallest element.

Selection Sort in Detail

Lets consider the following array as an example:
arr[] = {4, 8, 2, 9, 5}

Selection Sort

Implementation

Implementation: Selection sort implementation given below -

```
import java.util.Arrays;

public class Selection_Sort{

    public static void sort(int arr[])
    {
        int n = arr.length;

        for (int i = 0; i < n-1; i++)
        {
            int min_idx = i;
            for (int j = i+1; j < n; j++) {
                if (arr[j] < arr[min_idx]) {
                    min_idx = j;
                }
            }
            int temp = arr[min_idx];
            arr[min_idx] = arr[i];
            arr[i] = temp;
        }
    }

     public static boolean isSorted(int[] arr){
        int prev = arr[0];

        for (int i = 1; i < arr.length; i++) {
            if (prev > arr[i]) {
                System.out.println("Selection S
                return false;
            }
            prev = arr[i];
```

Selection Sort

Selection Sort

Minimum Integer

You are given an array A of size N. Let us denote S as the sum of all integers present in the array. Among all integers present in the array, find the minimum integer X such that S ≤ N*X.

Example 1:

Input:
N = 3,
A = { 1, 3, 2}
Output:
2
Explanation:
Sum of integers in the array is 6.
since 6 ≤ 3*2, therefore 2 is the answer.
Example 2:

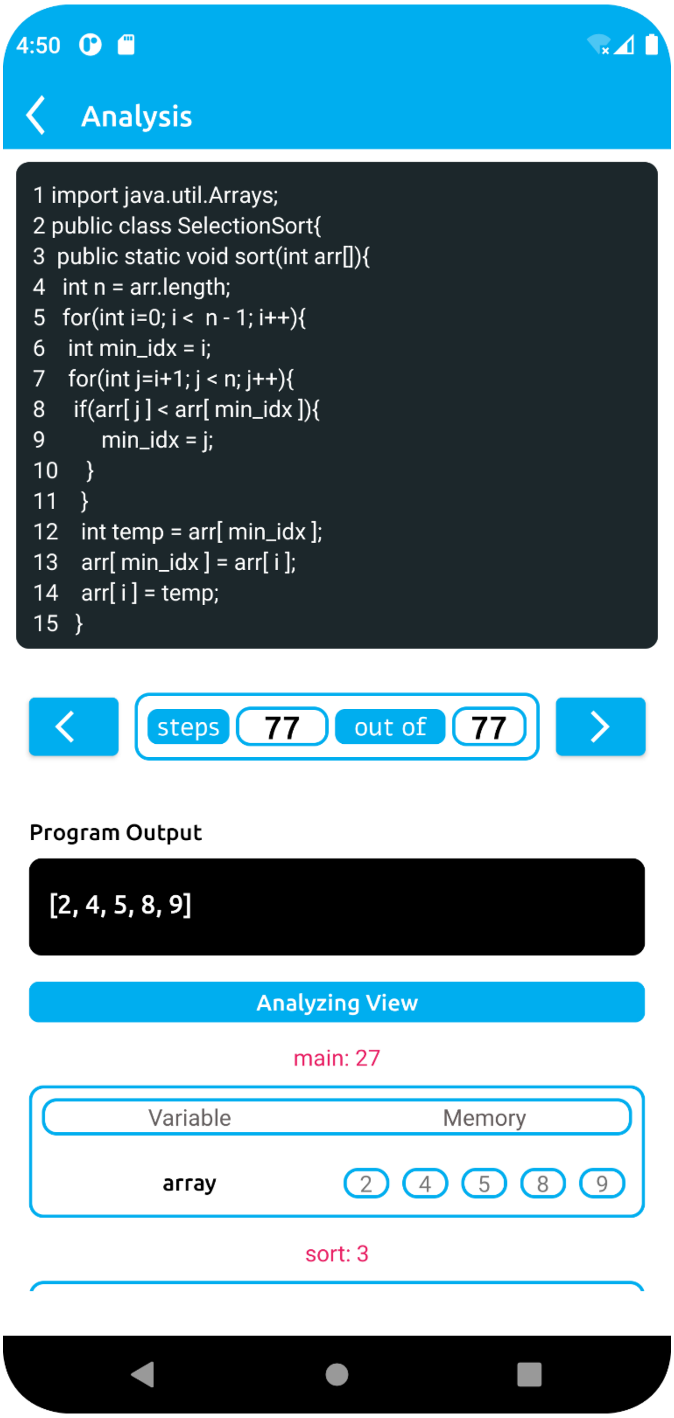
Input:
N = 1,
A = { 3 }
Output:
3
Explanation:
3 is the only possible answer
Your Task:
The task is to complete the function minimumInteger() which takes an integer N and an integer array A as input parameters and returns the

Fig - 4.4: Algorithm Description

The description section contains the Specific algorithm in detail, Algorithm Implementation, and lastly problem section for the specific algorithm.

Algorithm Analysis (Specific)**:**

Selection sort analysis are showing here. This section also divided into 3 part such as: Code, Analysis View, and handler. Code section will show code for an algorithm. Analysis view will show how data are performing operations based on algorithm. Lastly, Handler will handle each of the code and analysis view.



```
import java.util.Arrays;
public class SelectionSort{
 public static void sort(int arr[]){
  int n = arr.length;
  for(int i=0; i <  n - 1; i++){
   int min_idx = i;
   for(int j=i+1; j < n; j++){
    if(arr[ j ] < arr[ min_idx ]){
        min_idx = j;
    }
   }
   int temp = arr[ min_idx ];
   arr[ min_idx ] = arr[ i ];
   arr[ i ] = temp;
 }
```

Program Output

```
[2, 4, 5, 8, 9]
```

main: 27

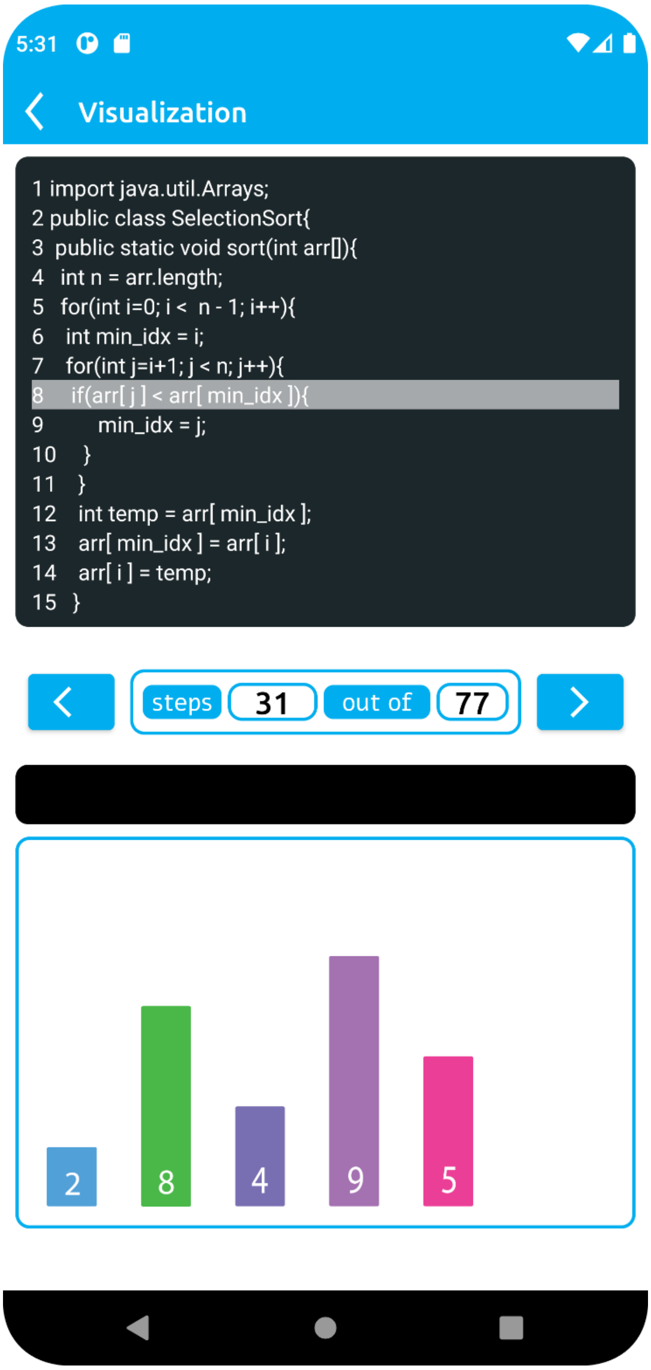
Variable	Memory
array	2 4 5 8 9

sort: 3

Fig - 4.5: Algorithm Analysis

Algorithm Visualization (Specific)**:**

Selection sort visualization are showing here. This section divided into 3 part such as: Code, Visualization View, and handler. Code section will show code for an algorithm. Visualization will show how data are performing operations based on algorithm visually. Lastly, Handler will handle each of the code and visualization view.



```
import java.util.Arrays;
public class SelectionSort{
 public static void sort(int arr[]){
  int n = arr.length;
  for(int i=0; i <  n - 1; i++){
   int min_idx = i;
   for(int j=i+1; j < n; j++){
    if(arr[ j ] < arr[ min_idx ]){
       min_idx = j;
    }
   }
   int temp = arr[ min_idx ];
   arr[ min_idx ] = arr[ i ];
   arr[ i ] = temp;
  }
```

Fig - 4.6: Algorithm Visualization

4.2 Unit Testing

Unit testing is a software testing method where individual units or components of a software application are tested in isolation from the rest of the system.

Table 6: Unit testing result

SL No.	Test Cases	Data Input	Expected Output	Actual Output	Pass / Fail
1	Visualization	Click the forward and back buttons for visualize	Successful	Successful	Pass
2	Analysis	Click the forward and back buttons for analyzing	Successful	Successful	Pass
3	Topic section	Click the button	Successful	Successful	Pass

Conclusion:

In conclusion, testing an Android project is crucial for ensuring the quality and functionality of the application. It helps to identify bugs, improve performance, and enhance user experience. There are various types of testing that can be performed, including unit testing, functional testing, integration testing, and more. A combination of manual and automated testing methods can be used to validate the application and its components. It's important to have a comprehensive testing strategy in place to ensure that the application meets the needs of users and meets the expected standards of quality.

Chapter 5

Result Analysis

The outcome results of the whole project and briefly discussed what we need and what we have to make it a good application. Every question answer will be here. The performance section will consider the response of the application after some operation. How much useable is the application? How effective and functional? What about security and scalability? Everything will be discussed part by part.

5.1. Performance

There are several ways to describe the performance of an Android application, including:

1. **Speed**: This refers to how quickly the application responds to user inputs and performs tasks. This can be measured in terms of the time it takes for the application to launch, load pages, or perform certain actions.
2. **Stability**: This refers to the application's ability to function smoothly and without crashing or freezing.
3. **Memory usage**: This refers to the amount of memory the application uses while running. High memory usage can lead to slow performance and stability issues.
4. **Battery usage**: This refers to the amount of battery the application uses while running. High battery usage can lead to shorter battery life for the device.
5. **Network usage**: This refers to the amount of data the application uses while running. High network usage can lead to higher data usage charges and slow performance if the device is on a limited data plan.
6. **User experience**: This refers to the overall satisfaction of the users when using the application. User experience can be measured by conducting user surveys and interviews or by analyzing user reviews and ratings.

It is important to note that performance can vary depending on the device and its specifications. It is also important to test the application on different devices and versions of Android to ensure compatibility.

The speed of the project is satisfactory, considering it is completely built from raw code without any library involvement for the main tasks such as visualization and analysis. Although crashing or freezing can occur, I have thoroughly tested the application and have not encountered any issues so far. Additionally, the application is designed to take up minimal memory since it stores images in .xml format. Therefore, the application is expected to take up less space and consume less battery.

This application does not require high network connectivity as it is an offline application. However, since we are using Firebase Real-time Database, data can be stored and synced even when the device is offline. However, write operations such as push, update, or delete will not be available when the device is offline.

I have also conducted several surveys on the project, and the feedback has been positive, which further validates the application's reliability and effectiveness.

Overall, we can say that this application performance will be good enough.

5.2. Usability

In general, the usability of an algorithm visualization android project would depend on factors such as the design of the user interface, the ease of navigation, and the effectiveness of the visualizations in helping users understand the algorithms being presented. A well-designed algorithm visualization android project should be easy to use and understand, even for users with little or no prior experience with the specific algorithms being presented. [1]

Additionally, an Algorithm Visualization Analysis android project should be able to provide the user with a clear understanding of the algorithm being visualized, the input and output of the algorithm, and its performance characteristics.

5.3. Functionality

An algorithm learning educational android application should have several functionalities that can help users effectively learn and understand algorithms. These functionalities can include:

Interactive visualization: Where we can visualize our algorithms.

Code execution: Don’t forget we have analysis which is doing code execution work analytically.

Data Structures and Algorithm Explanation: In the description section, we tell what how a specific algorithm can do and how it works.

Interactive quizzes and exercises: Yes, every section will have related problem to practice algorithms.

Offline access: 100% this application is completely offline application.

5.4. Security

An algorithm-learning educational Android application should have several types of security to protect user data and ensure the integrity of the application. These include:

Authentication: To ensure that only authorized users can access the application and its features. Not implemented yet because we didn’t provide any tracking functionality on our application. No need for authentication for this application right now.

Authorization: To ensure that users can only access the resources and features they are permitted to use. Similarly, we can say that until we didn’t provide any tracking progress functionality we can ignore this section also.

Data encryption: To protect user data from unauthorized access and tampering while it is stored on the device or transmitted over the network. For, now we are not concerned with data encryption because we’re not taking any user data except their credentials.

Input validation: To prevent malicious data from being entered into the application and potentially causing harm. At this stage, input validation is also not necessary to be implemented. There is no way it can occur.

Secure network communication: To ensure that data transmitted between the application and any external servers is protected from eavesdropping and tampering. No need because this feature has not been implemented yet. To communicate with others.

Secure storage: To ensure that the data is stored in an encrypted format or in a secure location on the device. No data will be taken or given so, no need to worry but secure storage.

Regular updates and monitoring: To ensure that any security vulnerabilities are identified and patched as soon as possible. This will be needed when scoring and tracking feature will be implemented on this application. Until now it’s not important in this stage.

Device management: To restrict access to the application on lost or stolen devices. If the device stolen the person can learn from the application nothing else can do with it.

Overall, Discussion we can say the users don’t need to worry about security. We’re not taking their data except credentials.

5.5. Scalability

An algorithm-learning educational android application should have several types of scalability to ensure that it can handle increasing numbers of users and data without performance degradation. These include:

Horizontal scalability: The ability to add more servers to the system to handle increased traffic and load. This project lite weight project algorithm is madden in such a way so that no crash could occur.

Vertical scalability: The ability to increase the resources (such as CPU, memory, and storage) of existing servers to handle increased traffic and load. As I said the project is very lite weight so no need for vertical scalability.

Auto-scaling: The ability to automatically add or remove servers based on the current load, ensuring that resources are always used efficiently. This application doesn't need any server right now. It can handle simple task as it is designed.

Load balancing: The ability to distribute traffic and load evenly across multiple servers, reducing the risk of a single point of failure. Server not important in this stage.

Caching: The ability to temporarily store frequently accessed data in memory, reducing the number of database queries and improving performance. No database queries are implemented which can reduce application performance. The database queries which are implemented those are ignorable.

Data partitioning: The ability to split large datasets into smaller, manageable chunks, reducing the time and resources required to query and analyze the data. Since data are too small it is not necessary to implement data partitioning on this application.

Asynchronous processing: The ability to perform tasks in the background, rather than blocking the user interface, ensuring a smooth user experience. This application doesn't need to perform task in the background. User need to do operation to learn so, asynchronous processing not important right now.

Database scalability: The ability to handle large amounts of data, such as distributed databases, NoSQL databases and column-family databases. As, I said no data will be taken from user for example: How much point they got while learning or what is the current progress of each user or what is the overall rank or individual user. This feature will be implemented in future work. So, until now no need for database scalability. Overall, Discussion we got that the application is lite weight and scalable.

Chapter 6

Conclusion

Algorithms have been found very hard. Algorithms are generally considered one of the hardest topics to learn in Computer Science. Every computer science student must have a good grasp of algorithms. It's going to take a lot of time and effort to master these topics. Some algorithms are genuinely hard, and some even seem unapproachable, but if you learn some basic patterns the algorithms will start making sense. This application is created to reduce scariness and complexity of learning algorithms. This application going to help you not only understanding basic patterns but also help how you too implement this knowledge step by step. This application going to be an asset for you and others because this is not only an application this is your home tutor. This application going to help you to reach your goal.

In conclusion, the Algorithm Visualization Analysis android project aims to address the difficulty that many students and professionals face in understanding and analyzing algorithms. By providing a user-friendly, interactive, and dynamic visualization visualization and analysis of algorithms, the app aims to make the study of algorithms more accessible and engaging. The app is intended for computer science students and professionals, as well as anyone else who is interested in learning about algorithms. The app will be developed using Android, Java, and Code View. The app will also be designed with user-friendly interface that makes it easy to use and understand. The Algorithm Visualization Analysis android project aims to fill the gap in the market and provide an innovative solution to the problem of understanding algorithms, making it a valuable tool for students and professionals alike.

6.1. Benefits of Project

This is a very simple design and easy to manipulate. It has got following features:

- It's an android-based project.
- Completely Offline. Learn anytime without an internet connection.
- Analysis or Visualization using Java. Most preferred language by the software company.
- Easy to understand every step of this application. User Friendly.
- Visualize algorithms.
- Analyze algorithms.
- Related Problems in every section of an algorithm to increase implementation skills.
- Introduction of each algorithm section. Space and time complexity also be provided.
- Free of cost.

6.2. Limitations

This is application has some limitations such as:

- API level less than 30 users can’t able to get full functionality of this application.
- UI design poor quality
- Description section needs more improvements
- Visualization animation needs more improvements

6.3. Future Work

In a nutshell, it can be summarized that the future scope of the project circles around maintaining information regarding:

Ranking: Ranking of the best performer who did best on algorithms. This will make it challenging and interesting for the user.

Certification: Those who pass the test will be able to get the certificate as an achievement. This will make them motivated because people love something in return after hard work.

Tracking Process: The tracking process will help them to complete the specific task effectively.

Earn Coin or Payment: The learner can unlock the algorithm based on the coin or using the payment method. The more coin the learner gets the more rank the person will get.

Interactive Dashboard: Interactive dashboard will make them feel better to use this application.

Competition: Algorithm Competition The more competition the learner wins the more coin the person can get to unlock new algorithms.

Quiz: Quiz to test their skills and make them confident in specific algorithms.

Push Notification: They can forget or they may need the remainder so that they can learn in time. This way they can able to gain active learning skills.

References

[1] D. S. Halim, "VisuAlgo," 2 December 2011. [Online]. Available: https://visualgo.net/en. [Accessed 2022].

[2] N. Karumanchi, "Data Structures and Algorithms Made Easy in Java: Data Structures and Algorithmic Puzzles," 2011.

[3] R. Singh, "Logic Mojo," 2018. [Online]. Available: https://logicmojo.com/. [Accessed 5 April 2021].

[4] P. Chandrasekar, "Stack Over Flow," 2008. [Online]. Available: https://stackoverflow.com/users/17184390/sakib-x-hossain. [Accessed 5 2019].

[5] G. Coccari, "Udemy," [Online]. Available: https://www.udemy.com/course/the-complete-android-oreo-developer-course. [Accessed March 2022].

[6] C. Mihailescu, "YouTube," [Online]. Available: https://www.youtube.com/ watch?v=msttfIHHkak. [Accessed December 2022].

[7] G. Coccari, "Udemy," 5 Feb 2019. [Online]. Available: https://www.udemy.com/course/the-complete-android-oreo-developer-course/.. [Accessed March 2022].

[8] A. E. Rubin, "Android Developer," [Online]. Available: https://developer.android.com/courses/advanced-training/toc.

[9] J. Hossain, "BongoAcademy," [Online]. Available: https://bongoacademy.com/.. [Accessed February 2022].

[10] S. Jain, "Geeks For Geeks," [Online]. Available: https://www.geeksforgeeks.org/a-complete-guide-to-learn-xml-for-android-app-development/.

[11] H. Schildt, "Java the Complete Reference, Tenth Edition," in *Java the Complete Reference*, 2017.

[12] oneHamidreza, "Github," [Online]. Available: https://github.com/oneHamidreza/MeowBottomNavigation.

[13] AmrDeveloper, "Github," [Online]. Available: https://github.com/AmrDeveloper/CodeView.