**ANDROID TRACKER INVESTMENT APPLICATION DOCUMENTATION**

**PRINCE B. KALUWA**

**REG NO: 20311351013**

**GUIDE**

**MR. HOPE SOKO**

**MINI-PROJECT**

*Submitted*

*In partial fulfillment of the requirements for the Degree of*

**BACHELOR OF SCIENCE IN COMPUTER SCEINCE**

**July, 2023**



**DMI-ST. JOHN THE BAPTIST UNIVERSITY MANGOCHI,**

**MALAWI.**

# PROFORMA FOR APPROVAL OF PROJECT PROPOSAL

Proposed Project Team:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Reg. No.** | **Name of the students** | **Semester** | **Branch** |
| 1 | 20311351013 | Prince B. kaluwa | VI | BSc |

|  |  |  |
| --- | --- | --- |
| Title of the Project: | Android Tracker Investment Application (ATIA) | |
| Subject Area: | Mobile application and investment-finance detailing | |
| Name of the Guide: | Mr. Hope Soko | |
| Designation : | Lecture – II | |
| Address with Phone No.: DMI – St. John The Baptist University, P. O. Box 406, Mangochi, Malawi (+265 881 422 991) | | |
| Office: | DMI – St. John The Baptist University, P. O. Box 406,  Mangochi, Malawi | |
| Residence: | DMI-St. John the Baptist University  Mangochi, Malawi | |
| No. of projects & students currently working under the Guide: | | 19 |

**Signature of the Student Signature of the Guide with seal**

**Date.........................**

N.B.: Please do not forget to enclose the synopsis of the project and the Bio-data of the Guide. In case the complete and signed Bio-Data of the Guide is not enclosed, the proposal will not be entertained**.**

**For Office Use only:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SYNOPSIS** | APPROVED |  | NOT APPROVED |  |
| **GUIDE** | APPROVED |  | NOT APPROVED |  |

**Comments / Suggestions for reformulation of the Project.**

**Date...................... Signature of the HOD**

# 

# BIO-DATA OF THE PROPOSED GUIDE FOR PROJECT WORK

**1. PERSONAL INFORMATION**

NAME (in block letters) : Mr. HOPE SOKO

Date of Birth & Age : 05.04.96, 27

Sex : Male

Academic Qualification : Master of Science in computer science

Official Address : DMI – St. John the Baptist University, Mangochi, Malawi

Phone No. and Fax. : (+265) 88 129 9607

Residential Address : DMI – St. John the Baptist University, Mangochi, Malawi

Phone No., and e-mail id : (+265) 881 28 96 07, [hopesoko8@gmail.com](mailto:hopesoko8@gmail.com)

**2. DETAILS OF EMPLOYMENT**

Designation : Lecturer II

Field of Specialization : Data science

Teaching Experience (in years) : 3

Industrial Experience (in years) : NIL

Particulars of contribution / experience in the field of specialization: Published an article

No. of Projects guided: 19

I **MR. HOPE SOKO** do hereby accept to guide **MR. PRINCE B. KALUWA** the student of **BACHELOR OF SCIENCE IN COMPUTER SCIENCE** program of the **DMI – ST. JOHN THE BAPTIST UNIVERSITY, MANGOCHI, MALAWI.**

**Signature of the Student Signature of the Guide with Seal**

# CERTIFICATE OF THE GUIDE

This is to certify that the project work entitled, **ANDROID TRACKER INVESTMENT APPLICATION (ATIA)** is a bonafide work of **MR.** **PRINCE B. KALUWA**, Registration No**. 20311351013** in partial fulfillment for the award of the Degreeof **BACHELOR OF SCIENCE IN COMPUTER SCIENCE** OF **DMI ST. JOHN THE BAPTIST UNIVERSITY, MANGOCHI, MALAWI** under my guidance. This thesis work is original one and not submitted earlier for the award of any degree elsewhere.

**Student’s Signature:**

**Signature of the Guide:**

# DECLARATION BY THE CANDIDATE

I, **MR. PRINCE B. KALUWA** hereby declare that this project REPORT **ANDROID TRACKER INVESTMENT APPLICATION (ATIA)** submitted to **DMI-ST. JOHN THE BAPTIST UNIVERSITY, MANGOCHI, MALAWI** in the partial fulfillment of requirements for the award of the degree of **BACHELOR OF SCIENCE IN COMPUTER SCIENCE** is a record of the original work done by me under the supervision of **MR. HOPE SOKO.**

**Register No. : 20311351013**

**Date :**

**Signature :**

|  |
| --- |
| Logo_Malawi.jpg  **DMI-ST. JOHN THE BAPTIST UNIVERSITY**  **MANGOCHI - MALAWI** BONAFIDE CERTIFICATE Register No: **20311351013**  Certified that this is bonafide record of work done in **ANDROID TRACKER INVESTMENT APPLICATION (ATIA)** by **MR. PRINCE B. KALUWA** of the department of **COMPUTER SCIENCE AND INFORMATION TECHNOLOGY** at **DMI – ST. JOHN THE BAPTIST UNIVERSITY, MANGOCHI, MALAWI** during the academic year: 2020 - 2023.  **INTERNAL EXAMINER EXTERNAL EXAMINER** |

# ACKNOWLEDGEMENT

First and foremost, I am grateful to the **ALMIGHTY GOD** for the strength, ability and above all, grace that He has showered upon me throughout this project.

I stand indebted in gratitude to our beloved Founder/Chancellor of DMI & MMI **Rev. Fr. Dr. J.E. ARUL RAJ** for all the facilities provided at our institution.

I would like to thank **Dr. T.X.A Ananth** the President University Council,DMI group of Institutions, Malawi, **Dr. Ignatius A. Herman,** the Director of EducationDMI group of Institution, Malawi, **Rev. Fr Sundar** the Secretary to University Council, for providing me an opportunity to do this successfully.

I further proudly express my esteemed gratitude to **Dr JERIN LENO** the Vice Chancellor, **DR BENNEDICT MALUNGA** the Principal**, REV.FR.SAHAYA RUBIN** Vice Principal (Admin), for their encouragement by assistance and guidance towards completion of this project.

I would like to thank the **DMI - ST. JOHN THE BAPTIST UNIVERSITY, MANGOCHI, MALAWI**, for providing an opportunity to do the project work as part of our curriculum.

I sincerely thank **Mr. STAINSLAUS PHILIMON** (the HOD, School of Computer Science and Information Technology) for his valuable support; reviews critique among other things. I extend my gratitude to **Mr. HOPE SOKO** (my guide for the project for the support, critique and all contributions to the project). I thank all my friends and family for creating a good working environment and of course, all my lecturers for broadening my academic knowledge and understanding.

Lastly, I would like to thank my parents, **MR & MRS KALUWA** for their relentless loving, moral and financial support they rendered all the way until project completion. Without them, my project was void.

# TABLE OF CONTENTS

[PROFORMA FOR APPROVAL OF PROJECT PROPOSAL i](#_Toc139013635)

[BIO-DATA OF THE PROPOSED GUIDE FOR PROJECT WORK ii](#_Toc139013636)

[CERTIFICATE OF THE GUIDE iii](#_Toc139013637)

[DECLARATION BY THE CANDIDATE iv](#_Toc139013638)

[BONAFIDE CERTIFICATE v](#_Toc139013639)

[ACKNOWLEDGEMENT vi](#_Toc139013640)

[TABLE OF CONTENTS vii](#_Toc139013641)

[LIST OF FIGURES x](#_Toc139013642)

[LIST OF TABLES xi](#_Toc139013643)

[LIST OF ACRONYMS xii](#_Toc139013644)

[ABSTRACT xiii](#_Toc139013645)

[CHAPTER I 1](#_Toc139013646)

[INTRODUCTION 1](#_Toc139013647)

[1.1 ABOUT THE PROJECT 1](#_Toc139013648)

[CHAPTER II 2](#_Toc139013649)

[SYSTEM STUDY 2](#_Toc139013650)

[2.1 INTRODUCTION 2](#_Toc139013651)

[2.2 PROBLEM STATEMENT 2](#_Toc139013652)

[2.3 EXISTING SYSTEM 3](#_Toc139013653)

[2.4 PROPOSED SYSTEM 4](#_Toc139013654)

[2.5 SYSTEM OBJECTIVES 5](#_Toc139013655)

[2.6 SYSTEM SPECIFICATION 5](#_Toc139013656)

[2.6.1 SOFTWARE SPECIFICATION 5](#_Toc139013657)

[2.6.2 HARDWARE SPECIFICATION 6](#_Toc139013658)

[2.7 LITERATURE REVIEW 6](#_Toc139013659)

[2.8 METHODOLOGY 7](#_Toc139013660)

[CHAPTER III 9](#_Toc139013661)

[SYSTEM DESIGN 9](#_Toc139013662)

[3.0 INTRODUCTION 9](#_Toc139013663)

[3.1.1 USECASE DIAGRAM 9](#_Toc139013664)

[3.1.2 ACTIVITY DIAGRAM 10](#_Toc139013665)

[3.3 SYSTEM ARCHITECTURE 13](#_Toc139013666)

[CHAPTER IV 14](#_Toc139013667)

[SYSTEM DEVELOPMENT 14](#_Toc139013668)

[4.1 INTRODUCTION 14](#_Toc139013669)

[4.2 ALGORITHM AND TECHNIQUES 14](#_Toc139013670)

[4.3 MODULE DESCRIPTION 15](#_Toc139013671)

[4.4 USER AUTHENTICATION 16](#_Toc139013672)

[4.5 USER INTERFACE 16](#_Toc139013673)

[4.5 PROFILE MODULE 16](#_Toc139013674)

[4.6 PAYMENT MODULE 16](#_Toc139013675)

[4.7 PORTFOLIO MANAGEMENT 16](#_Toc139013676)

[4.8 NOTIFICATION AND ALERTS 16](#_Toc139013677)

[CHAPTER V 18](#_Toc139013678)

[SYSTEM TESTING 18](#_Toc139013679)

[5.1 INTRODUCTION 18](#_Toc139013680)

[5.2 SYSTEM TESTING 18](#_Toc139013681)

[5.2.1 UNIT TESTING 18](#_Toc139013682)

[5.2.2 INTEGRATION TESTING 19](#_Toc139013683)

[5.2.3 VALIDATION TESTING 20](#_Toc139013684)

[5.2.4 USER ACCEPTANCE TESTING 20](#_Toc139013685)

[CHAPTER VI 21](#_Toc139013686)

[SYSTEM IMPLEMENTATION 21](#_Toc139013687)

[6.1 USER TRAINING AND DOCUMENTATION 21](#_Toc139013688)

[6.2 SAMPLE CODE 28](#_Toc139013689)

[CHAPTER VII 31](#_Toc139013690)

[PROBLEMS FACED AND SOLUTIONS 31](#_Toc139013691)

[7.1 PROBLEMS FACED 31](#_Toc139013692)

[ CODE EXECUTION 31](#_Toc139013693)

[ ERROR DEBUGGING 31](#_Toc139013694)

[7.2 SOLUTIONS 31](#_Toc139013695)

[CHAPTER VIII 33](#_Toc139013696)

[SUGGESTIONS FOR PROJECT ENHANCEMENTS 33](#_Toc139013697)

[CHAPTER IX 34](#_Toc139013698)

[CONCLUSION 34](#_Toc139013699)

[REFERENCES 35](#_Toc139013700)

[PLAGIARISM 37](#_Toc139013701)

[COPYRIGHT LETTER 38](#_Toc139013702)

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **NAME OF THE FIGURE** | **PAGE NO.** |
| 2.1 | Waterfall model | 8 |
| 3.1 | Use-case diagram | 9 |
| 3.2 | Communication diagram | 10 |
| 3.3 | Activity diagram | 10 |
| 3.4 | Class diagram | 11 |
| 3.5 | Component diagram | 12 |
| 3.6 | Interaction diagram | 12 |
| 3.7 | System architecture | 13 |
| 4.1 | Screenshot of all the dependencies used. | 17 |
| 6.1 | Screenshot of the splash screen | 21 |
| 6.2 | Screenshots of the login page and create account page | 22 |
| 6.3 | Screenshots of the Dashboard and what happens when the menu icon is clicked. | 22 |
| 6.4 | Screenshots of the add asset page and portfolio page. | 23 |
| 6.5 | Screenshots of the payment module and its functions in action. | 23 |
| 6.6 | Screenshots of the Profile module and has the show profile page and create profile page. | 24 |
| 6.7 | Screenshot of the online work module. | 24 |
| 6.8 | Screenshots of all the menu functions in action. | 25 |
| 6.9 | Screenshot firebase authentication in the console | 26 |
| 6.9.1 | Screenshot firebase firestore Database used by admin. | 27 |
| 6.9.2 | Screenshot firebase storage used by admin. | 27 |

# LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **TABLE NO.** | **NAME OF THE TABLE** | **PAGE NO.** |
| 2.1 | literature review | 6 |
| 5.1 | Unit Testing Results | 18 &19 |
| 5.2 | Validation Testing (Valid Input) | 20 |
| 5.3 | Validation Testing (Invalid Input) | 20 |
| 6.1 | Modules and functions that use firebase (database) | 25 & 26 |

# LIST OF ACRONYMS

|  |  |
| --- | --- |
| ATIA……………..  IDE………………. | Android Tracker Investment Application  Integrated development environment |
| GB………………. | Gigabits |
| APK……………... | Android package Kit |
| XML...…………... | Extensible Markup Language |
| JDK…………….... | Java Development Kit |
| RAM ……………. | Random Access Memory |
| ROM ……………. | Read Only Memory |
| SDK……………... | Software Developer Kit |
| API…………….... | Application Programming Interface |
| NoSQL…………... | Non-relational database. |
| Git……………….. | Distributed version control system |
| ADMIN………….. | The person in-charge of the managing the firebase console. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |  |

# ABSTRACT

Android Tracker Investment Application is the full expansion of the term ‘ATIA’. The Android Tracker Investment Application is a mobile app designed to provide investors with comprehensive analytics on their portfolio. The app has several features including the ability to track investment portfolio performance, provide insights into market trends, and give investors personalized advice on their investments. The portfolio tracking feature allows users to monitor the performance of their investments in real-time by providing detailed information and charts reflecting changes in their portfolio. Investors can choose to group investments by various criteria such as sector, asset class or geography. The app also has an alert feature which notifies the user when changes occur in their portfolio, or when important market news is reported. In addition to this, the application provides investors with insights into market trends providing them with information about economic events on a global scale and their potential impact on investments. Users also have the ability to personalize the app to their preferences and receive personalized advice from financial experts who share their investment philosophy. And the Android Tracker Investment Application is a comprehensive and user-friendly tool designed for investors to successfully track and manage their portfolios. Its real-time monitoring, alert, and insightful features provide unparalleled analytics and information to investors making it a must-have investment tool for any serious investor.

**Keywords:** Android Tracker Investment Application, Portfolio, investments and assets.

# CHAPTER I

# INTRODUCTION

### 1.1 ABOUT THE PROJECT

Android Tracker investment Application is the full expansion of the term ‘ATIA’. The term ‘Android’ is used to give insight of mobile device. The ‘Tracker’ part emphasizes that it targets a specific thing and monitors. The ‘Investment’ implies the financial matter of assets. ATIA entails a comprehensive approach that starts with understanding the user's needs. The application's design must be user-friendly, intuitive and easily navigable to ensure that the end-user is provided with an optimal experience.

User authentication is a vital component of the development process to ensure that sensitive financial information is adequately protected. The application must include secure login mechanisms that authenticate users and prevent unauthorized access to their financial data.

The user interface is another critical component of an Android tracker investment application. It should be aesthetically pleasing, intuitive and offer seamless navigation between different functionalities. A well-designed user interface not only enhances the user experience but also enables users to track their investments easily.

Alerts and notifications are a fundamental part of the application that helps users stay up to date with their investments. Notifications can be triggered by various events such as changes in stock prices, significant portfolio fluctuations, or the achievement of pre-set targets. Such alerts should be timely, precise, and actionable to enable users to make informed investment decisions.

Portfolio management is the core functionality of the Android tracker investment application. It entails tracking and analyzing investments, providing portfolio summaries, monitoring stock prices, and facilitating trade execution. The application should provide a detailed portfolio view that enables users to track their investments' performance, diversification, and asset allocation.

In conclusion, an Android tracker investment application requires a development approach that considers user authentication, user interface design, alerts & notifications, and portfolio management. The successful implementation of these modules will lead to a robust application that meets users' needs and helps them make informed investment decisions.

# CHAPTER II

# SYSTEM STUDY

## INTRODUCTION

In today’s world, investing in any form has become a trend. The increasing accessibility and availability of investment opportunities have given people the chance to grow their wealth and achieve their financial goals. However, with the growing number of investment options, it has become difficult for individuals to keep track of their portfolio effectively and make informed decisions. In this context, an Android Tracker Investment application is an innovative solution that can solve these challenges. This system study will analyze the application in detail, discussing its purpose, features, technical specifications, and usability.

## PROBLEM STATEMENT

The Android Tracker Investment application project aims to address the problem of monitoring and managing investment portfolios on-the-go for individual investors. The current investment applications available in the market lack the necessary features to provide a comprehensive, user-friendly, and efficient interface for investors to track and analyze their investments regularly.

Additionally, investors face challenges in keeping track of market trends and news and linking them to their specific investments to make informed decisions. They need a helpful and reliable investment application that can deliver real-time news, quotes and analysis, and customizable alerts.

Therefore, there is a need to develop an Android investment application that can ease the burden of investors by providing comprehensive management, complete tracking, and real-time updates on their investment portfolios. The application will utilize modern technology to deliver reliable data, custom alerts, and analysis, enabling investors to make informed decisions and improve their investment performance.

Overall, the Android Tracker Investment application project will provide an innovative and user-friendly solution to help individual investors efficiently manage and monitor their investment portfolios and achieve financial success.

## EXISTING SYSTEM

Investing has become an intrinsic part of modern-day life. With technological advancements and digitalization, investing in stocks and shares has become very accessible and easy to manage. Not too long ago, the process of investing and managing stocks and shares was conducted through a manual system, which could be an arduous and time-consuming task. This was due to the fact that users had to write down all of their investments on a piece of paper. However, with the advent of technology, this process has become much simplified.

One of the most significant problems with the manual system was that errors often arose. There were many factors that could lead to these errors, such as forgetting to write down a day's record, or mis-matching of records. Additionally, instances of incorrect calculations were also common, leading to incorrect investment values. All of these factors made managing a portfolio through a manual system a hassle.

Today, there are various stock management applications on the market that make this process much easier and accessible. One of the popular apps is the Blackberry Stock Manager. This app is designed to help users keep track of their investments and assist in making informed decisions. The app gives real-time updates and alerts users in case of any changes, enabling them to monitor their portfolio actively.

Another app, the Symbian My Portfolio app, is ideal for those who like to have all their financial information in one place. The app allows users to keep track of all their accounts, including bank accounts, credit cards, and investments. The app is particularly suited to those who want an overview of all their financial data in one easily accessible location.

The Nasdaq Portfolio Manager app for iPhone is another great option for investors. This app enables users to track stocks and mutual fund investments in real-time, providing up-to-the-minute updates. The app also provides customized alerts and news alerts tailored to the user's investment interests, making it easier to keep track of stocks that may be of particular interest.

Robinhood is an increasingly popular app that allows users to invest in different stocks, including cryptocurrencies and exchange-traded funds (ETFs). This app is designed to be user-friendly, with a fast and straightforward interface that makes investing in stocks easy to understand. The app also does not charge a commission for trades, making it particularly appealing to those who wish to avoid the transaction costs associated with most traditional online brokerage accounts.

Another popular option is Public, IWM mobile app. This app allows users to invest in certain stocks, called "themes." This feature enables users to invest in industries and sectors that interest them and align with their values. Additionally, the app also allows for fractional shares purchases, enabling investors to purchase a smaller portion of a stock that may be too expensive to buy in its entirety.

Overall, investing through a manual system can be challenging and time-consuming, with a high likelihood of errors creeping in. However, with the various stock management apps available, this process has become much easier and accessible. These apps offer real-time updates on investments, customized alerts, and an easy-to-use interface, making them a popular choice for anyone seeking to invest easily and efficiently. Investing has never been more accessible, and with these new technologies, anyone can become a successful investor.

## PROPOSED SYSTEM

Investment tracking is one of the most crucial activities for investors. It plays a vital role in managing risk and optimizing returns. Nowadays, with the advancement of technology, investors are relying more on digital tools to help them track their investments effectively. Android Tracker Investment application is a tool that offers ease of maintenance, tracks investments, and is user-friendly with authentication security.

Maintenance is a crucial factor in any app that involves financial data. An application that crashes or has regular bugs can lead to loss of data and loss of confidence amongst its users. Android Tracker Investment application offers an easy-to-maintain interface that is regularly updated with the latest advancements in technology. This ensures that the app is running on the latest frameworks and is bug-free, thus providing its users with a reliable investment tracking tool.

The dashboard is the most critical aspect of the Android Tracker Investment application, as it displays all the relevant information needed by the investors. The dashboard provides investors with an overview of their investments, including assets, portfolios, returns, and market movements. With the dashboard, investors can keep track of all their investments in one place, saving time and enabling them to make informed decisions regarding their investments.

Adding assets is a crucial function in any investment tracking app. Android Tracker Investment application offers a user-friendly interface that enables investors to add and remove assets with ease. With the ability to add, remove, or update assets, investors can keep their investment records up-to-date and adjust their investment strategies as needed.

The user interface of Android Tracker Investment application is designed to be user-friendly, providing investors with an intuitive and visually appealing layout. The UI has been designed to ensure that even novice investors can use it without any difficulty. This feature ensures that investors can use the app without the need for any special training or support.

Security is a significant concern for any investment tracking app. Android Tracker Investment application offers authentication security features that ensure that only verified users can access their investment data. The security features provided by the app, such as two-factor authentication, ensure that investors' data is secure and private.

In conclusion, Android Tracker Investment application is an ideal investment tracking tool that offers investors a reliable and user-friendly interface. With its easy-to-maintain dashboard, investors can track their investments and keep up-to-date with their investment strategies. The app's user-friendly interface and authentication security features ensure that investors can use the app with ease and confidence. By choosing Android Tracker Investment application, investors can manage their investments effectively, optimize returns, and minimize risk.

## SYSTEM OBJECTIVES

The first key objective of this project is to design and develop an application program that meets the required industry standards for asset management and oversight. Achieving this objective will require a thorough understanding of the requirements and specifications for the program, as well as an in-depth knowledge of the latest software development tools and methodologies.

Another key objective of the project is to develop functional program modules that can perform the specific tasks required for efficient asset management and oversight. These modules will need to be designed, developed, and tested in accordance with best practices and industry standards to ensure that they are reliable, scalable, and easy to maintain.

Finally, the project also aims to create an efficiently compatible program that can function effectively across a range of different hardware and software platforms. Achieving this objective will require careful consideration of the various system requirements and compatibility issues that may arise, as well as the use of efficient programming techniques for optimizing performance and reducing errors.

Overall, the successful completion of this project will require a strong focus on software engineering best practices, as well as a deep understanding of the specific requirements and challenges associated with asset management and oversight in today's complex technological landscape.

## SYSTEM SPECIFICATION

The specifications required to run this software application via mobile devices are:

* Android version 10 or higher.
* 3 GB and above storage free storage space
* 4GB Ram

### SOFTWARE SPECIFICATION

To allow ATIA to run correctly, user must have:

* java
* Updated versions of android version or the latest versions

### HARDWARE SPECIFICATION

ATIA can run on machines such as:

* Mobile Android device
* Touch screen

## LITERATURE REVIEW

The purpose of this literature review is to explore the existing knowledge and advancements in the field of Android tracker investment applications. These applications aim to provide users with real-time tracking and analysis of investment portfolios, enabling informed decision making and maximizing investment returns. Through this review, we will analyze the current state of research, identify gaps, and highlight potential opportunities for innovation and improvement.

And the survey includes the list below.

**Table 2.1 Literature Review**

|  |  |  |  |
| --- | --- | --- | --- |
| **AUTHORS** | **YEAR** | **PROBLEM** | **JOURNAL** |
| A N Adaoanawar,  Vikrant kadam, Arjun Raval,  Ashish ujjawal | 2012 | Tracking and managing investment | Investment portfolio  Manager-mobile application |
| Parvathy s. Nair, Atul Shiva,  Nikhil Yadav, Priyanka Tandon | 2022 | Digitalizing retail functions | Determinants of mobile apps adoption by retail investors for online trading in emerging financial markets. |
| Lee-lee chong | 2020 | Mobile stock trading among young investors | Acceptability of mobile stock trading application. |
| Shana clor-proell,Ryan guggenmos,Kristina m. rennekamp | 2019 | Information dissemination via mobile device application. | Mobile devices and investment apps |
| Suzanee malhotra | 2020 | Increase in stock trade | Study of features of mobile trading app. |
| Sayan chaudhry | 2021 | Democratize investing | Design patterns of investing apps and their effects om investing behaviour. |
| Lara stocchi | 2022 | Integrative review of existing marketing research | Marketing on mobile apps |

## METHODOLOGY

The Waterfall Model is a sequential design process that is utilized in software development. It is a linear approach that consists of stages that occur one after another and it has a sequential flow; meaning that each stage must be completed before the next stage can begin. This model has been the most popular model in software development for many years, but it has been slowly phased out in recent years for more agile methods.

The Waterfall Model consists of five main stages; requirements gathering, design, implementation, testing, and maintenance. In requirements gathering, the business requirements for the software are identified, defined and documented. During the design phase, the software is designed in detail, including the software architecture, programming languages, modules, and interfaces.

On the coding stage, the use of modularity is essential. Modularity is an approach used in software engineering where software is divided up into separate, distinct, and self-contained modules, which makes it easier to understand, reuse, and maintain. By separating the software into these smaller modules, there can be more agile development, better-defined interfaces, and improved code maintainability.

After the software is designed, the implementation stage begins. In this stage, the software design is translated into a programming language, software code is written, and the software is built. The software is then tested in the testing phase where both functional and non-functional aspects of the software are reviewed and checked. Finally, in the maintenance stage, the software is deployed, bugs are fixed, and updates are made to ensure the software is always functioning at its best.

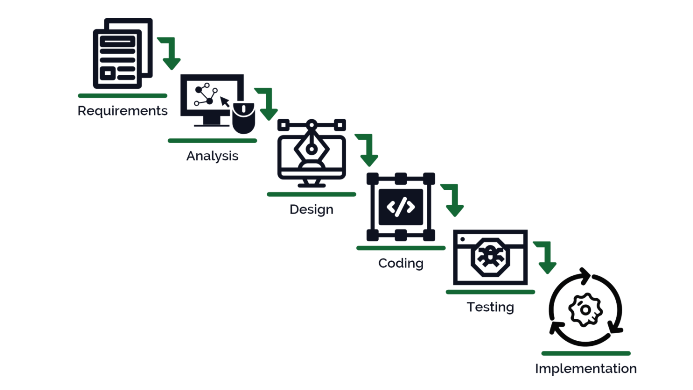
Models include user authentication, user interface, portfolio management, notification and alerts model. User authentication refers to the process of verifying a user's identity, usually through the use of login credentials such as a username and password. This is a crucial aspect of software development to ensure that user data is secure.

The user interface is a critical part of any software system, and it refers to the visual design of the software. A good user interface is easy to use, intuitive, and aesthetically pleasing. The user interface is critical to the overall experience of the software, and it can make or break the success of the software. Proper attention must be paid to designing an excellent user interface to ensure users are satisfied with the software.

Portfolio management is another important aspect of software development. Software developers create software for many different purposes ranging from financial trading to educational purposes, and so on. Good portfolio management allows developers to ensure that the software they are creating is addressing the needs of their customers.

Notification and alerts models are also crucial in software development. It refers to the system in place that notifies users of new information or changes that occur within the software. These notifications could include new updates and software change alerts; they are crucial in keeping the user up-to-date and engaged with the software.

In conclusion, the Waterfall Model is an effective and logical approach to software development that remains popular today. The use of modularity is essential in the coding stage, and models like user authentication, user interface, portfolio management, and notification and alerts systems are all important components of any software system. By following these best practices in software development, developers can create software that is efficient, effective and caters to the needs of the users.

**

*Figure 2.1: waterfall model*

# CHAPTER III

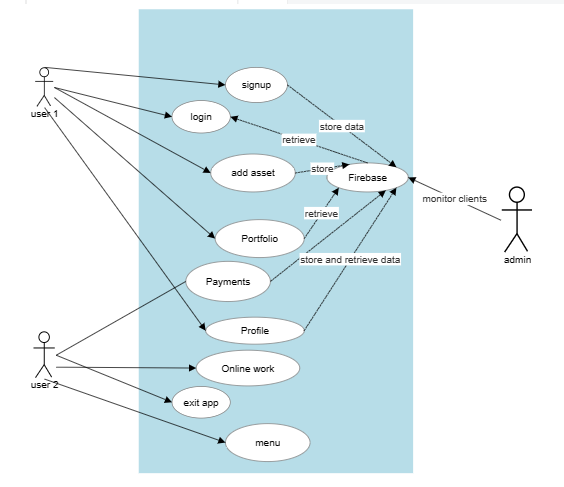
# SYSTEM DESIGN

## INTRODUCTION

The basic layout of the entire system can be summarized into picture form using diagrams. This displays the system designs of front-end and back-end process to reach completion and successfully running the project. In other words, it is a series of diagrams shown as quick scenes to displaying how the project looks internally and externally.

**3.1 UML DIAGRAMS**

## USECASE DIAGRAM

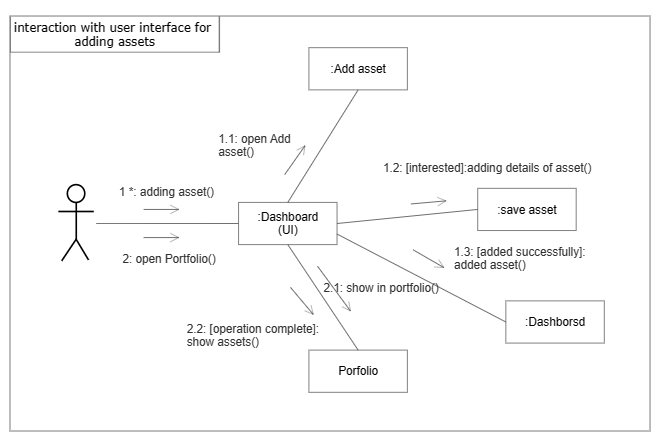


This diagram illustrates the all the operations present in the application and shows with operations are connected to firebase and how they use it.

And it shows the abstraction of the users to firebase while the admin has full access to firebase.

*Figure 3. 1: Use-case Diagram*

**3.1.2 COMMUNICATION DIAGRAM**

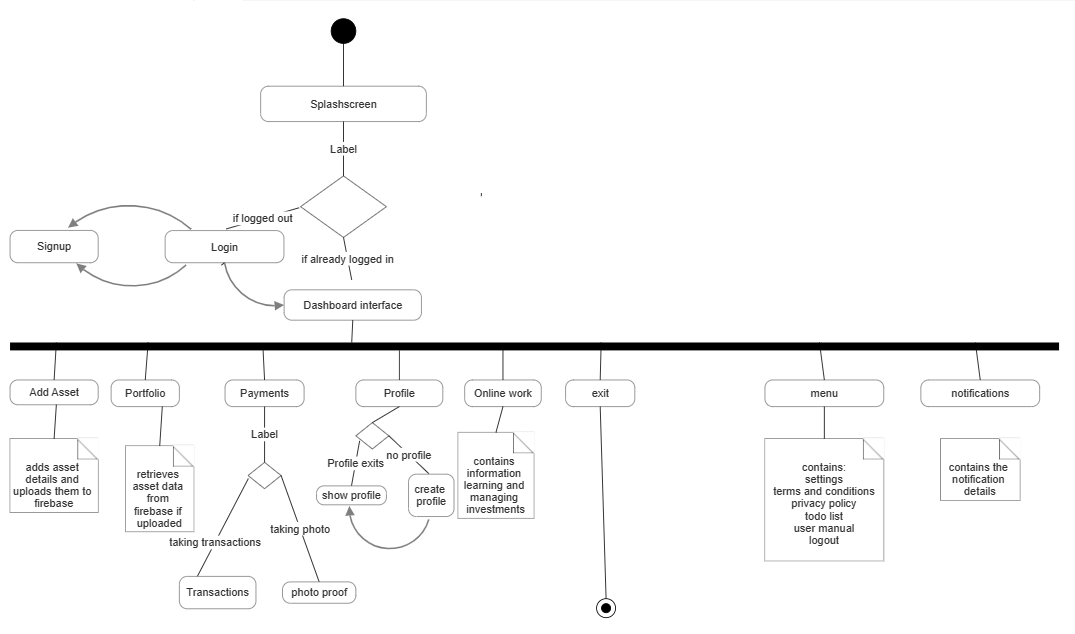


*This illustrates how the user interface interacts with the other modules when given specific commands.*

In this scenario we have adding asset and open Portfolio task done by clink on the add asset button and Portfolio button.

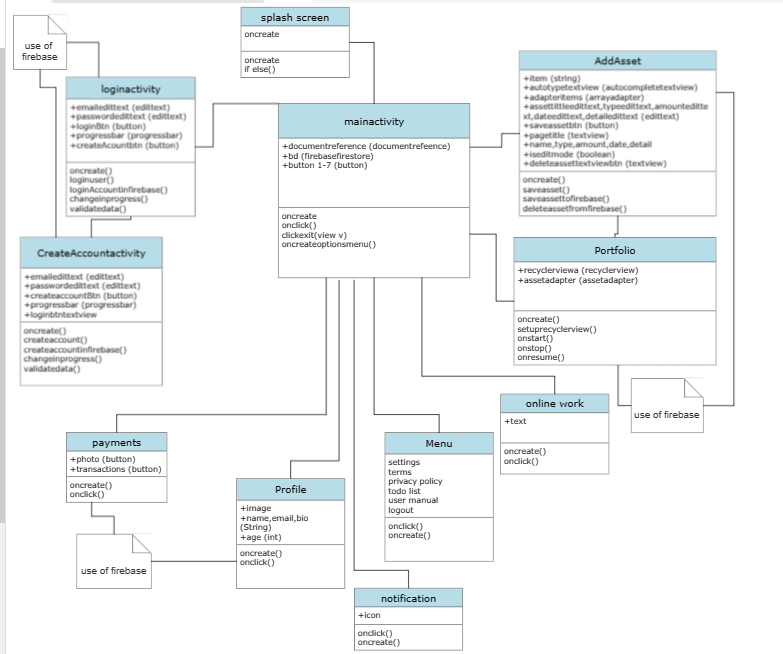
*Figure 3. 2: Communication Diagram*

## ACTIVITY DIAGRAM



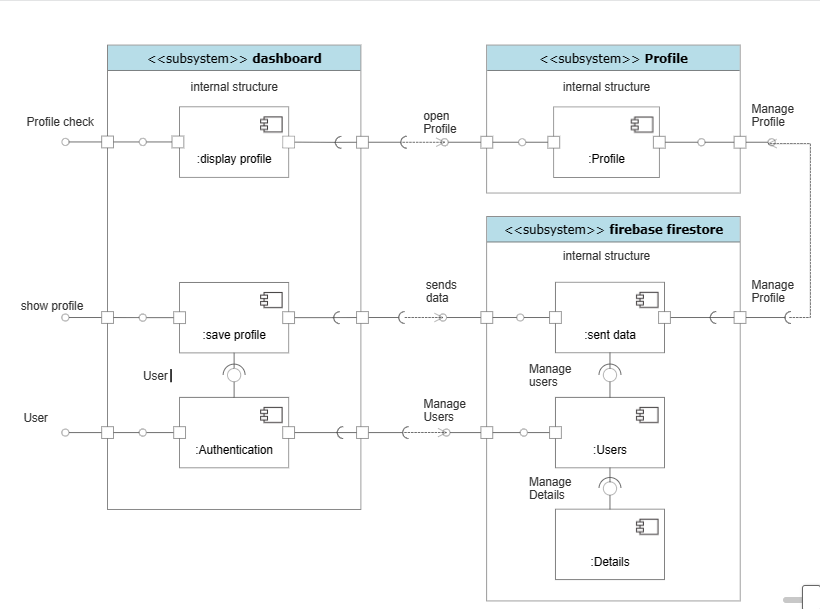
*Figure 3. 3: Activity Diagram*

The diagram above illustrates all the class found in the application and how they are integrated with each other with the activities they all present.

**3.1.3 CLASS DIAGRAM**

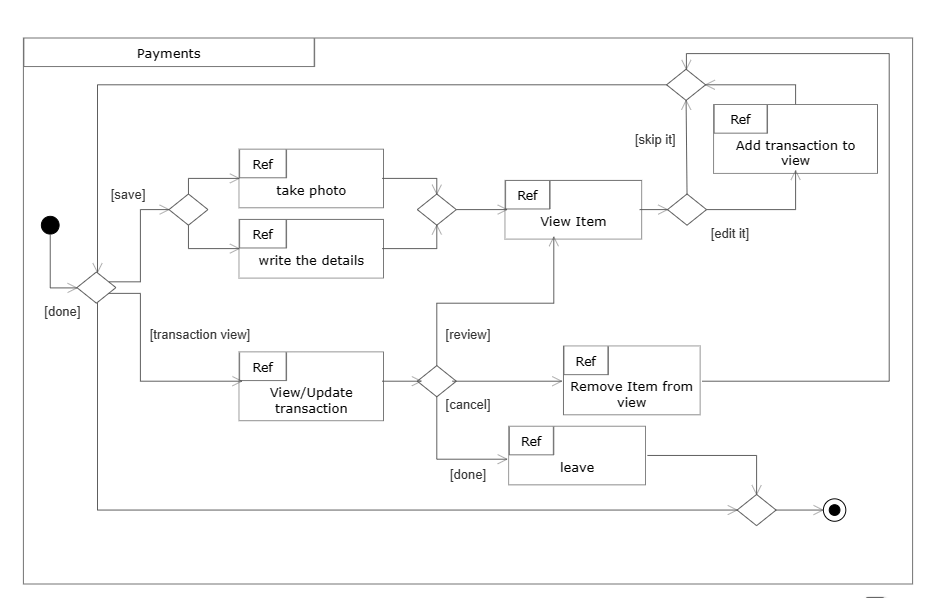
*This class diagram represents all the classes found in the application and all the necessary details for their functionality.*

*Figure 3. 4: Class Diagram*

**3.7 COMPONENT DIAGRAM**

*The emphasis is that the components such the authentication and Profile management and how its handled within the application.*

*Figure 3. 5: component Diagram*

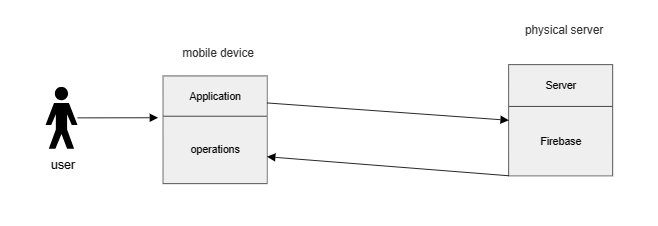
**3.8 INTERACTION DIAGRAM**

*This interaction diagram shows how the Payment module is used and how its logical processes are carried out.*

*Figure 3. 6: Interaction Diagram*

## SYSTEM ARCHITECTURE

This diagram shows how modelling moved from stage to stage to attain the best visuals ATIA can put out.



This diagram represents the functional relationship between the user, application and firebase which illustrates that the primary operations are dependent on firebase.

*Figure 3.7: System Architecture*

# CHAPTER IV

# SYSTEM DEVELOPMENT

## INTRODUCTION

System development involves a process of solution production with help of system related software. Apart from gathering smart, knowledgeable and field experienced colleagues, for development completion, the software development process is also need’s systematical and practical technique to successfully complete a project.

A software’s life cycle is the series of identifiable stages the software product undergoes during its usage. ATIA will continue to run dependent on user domain website license.

## ALGORITHM AND TECHNIQUES

User Authentication:

Implement Firebase Authentication to handle user registration and login.

Use Firebase Authentication APIs to validate user credentials and manage user sessions securely.

User Interface:

Utilize Android XML layouts to design and create various screens for the application.

Implement navigation between screens using activities or fragments.

Use appropriate UI components like TextView, EditText, Button, RecyclerView, etc., to display information and interact with users.

Profile Management:

Implement functionality for users to create and update their profiles.

Allow users to input details such as name, contact information, and preferences.

Store profile information in Firebase Firestore or Realtime Database.

Payment Handling:

Integrate a secure payment recording system with firebase for users to add, remove, or update payment details.

Ensure the security of user payment information by utilizing encryption techniques.

Portfolio Management:

Develop data structures to store user investment portfolios.

Implement CRUD (Create, Read, Update, Delete) operations for managing portfolios.

Calculate and display portfolio performance metrics, such as total investment value, returns, and investment distribution.

Firebase Integration:

Utilize Firebase Realtime Database or Firestore for storing and retrieving user data.

Implement appropriate Firebase APIs to handle data synchronization, querying, and updates.

Ensure data security and access control through Firebase Authentication and Firestore/Firebase Realtime Database rules.

Notification System:

Utilize Firebase Cloud Messaging to send push notifications to users regarding portfolio updates, payment confirmations, or important announcements.

Implement algorithms to determine which notifications users should receive based on their preferences.

## MODULE DESCRIPTION

This is an era of information technology where automation of each and every activity is gaining importance tech-wise. ATIA introduces the site to lead in the automation of self and efficient investment management. Which it is better than some of the old ways of tracking the investments.

It has modules like user authentication, user interface, profile module, payments module, portfolio management and notification and alerts module.

* user authentication: it contains both create and login pages.
* user interface: contains the dashboard and menu plus its where all the other modules are accessed.
* profile module: contains show profile and create profile pages.
* Payment module: contains camera function and record keeping function.
* Portfolio management module: contains the add asset page, Portfolio page and online work page.
* Notification and alerts module: inform the user on necessary information.

## USER AUTHENTICATION

This module provides functionality for user authentication and authorization. It enables users to register, log in, and securely manage their account credentials. It also includes two java files which login and create account with their independent xml files. It also uses the firebase authentication service.

## USER INTERFACE

This module focuses on creating an intuitive and user-friendly interface for the Android tracker investment application. It includes designing and implementing the layout, navigation menus, buttons, and other interactive elements. Its primary goal is to provide a seamless user experience.

And it gives access to the other operations which is very vital

## PROFILE MODULE

The profile module allows users to create and manage their profile information within the Android tracker investment application. It provides options for users to edit their personal details, contact information, profile pictures, and other relevant information. It also uses the firebase firestore cloud service and storage service.

## 4.6 PAYMENT MODULE

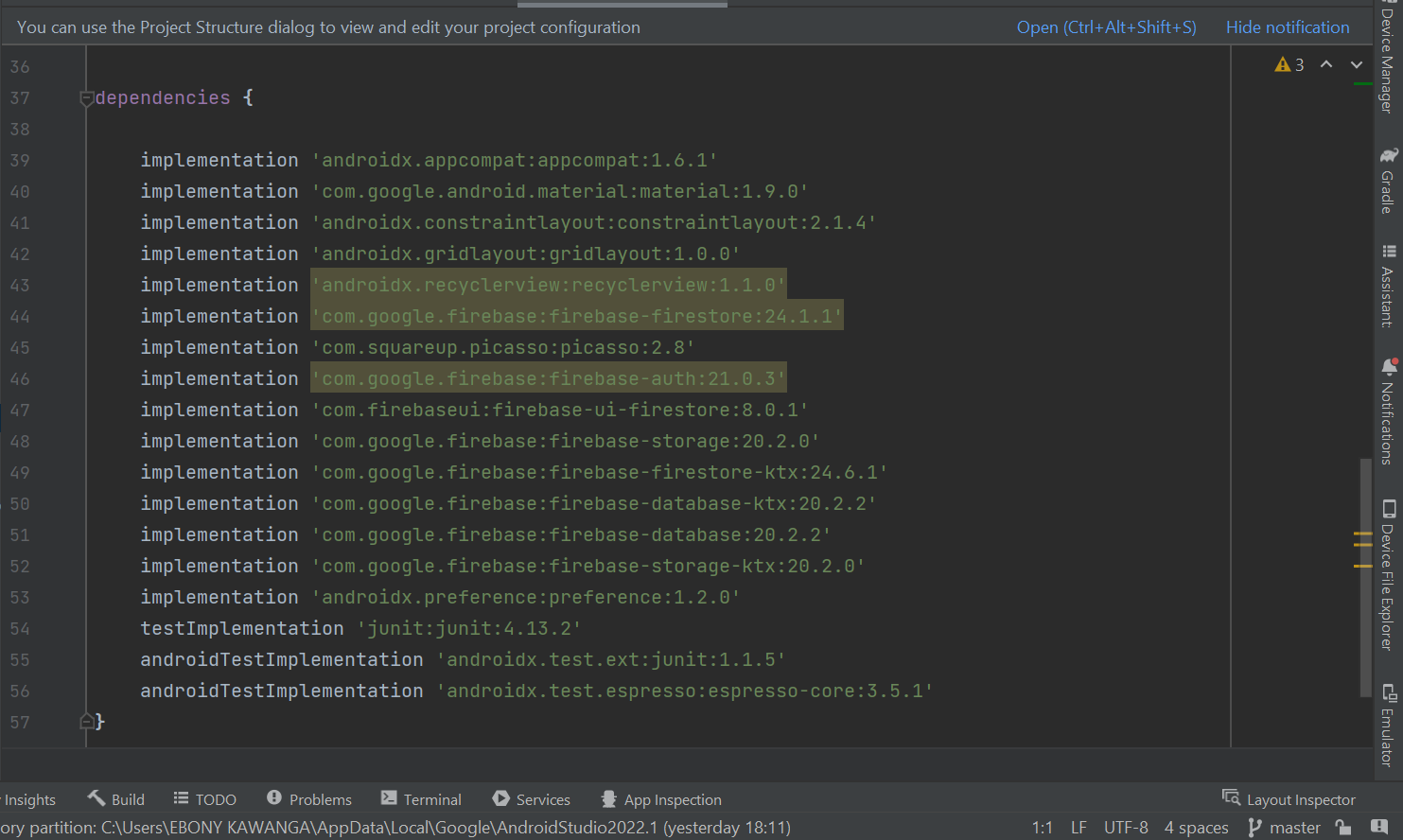
The payment module mainly allows users to record all their transactions involving their investments. It has operations like taking photographic proof and text input records. It also uses the firebase firestore cloud service.

## PORTFOLIO MANAGEMENT

This module provides functionality for managing investment portfolios. It allows users to add, view, and track investment assets. Users can monitor the performance of their investments, view historical data, and make informed investment decisions. It also uses the firebase firestore cloud service.

## NOTIFICATION AND ALERTS

This module handles sending real-time notifications and alerts to users. It may include features like push notifications for portfolio updates, market trends, important announcements, payment reminders, and other relevant events. Users can customize their notification preferences based on their interests and requirements.



*Figure 4.1: screenshot of all the dependencies used.*

<https://github.com/B-kaluwa/ATIA-CHECK-.git>

above is the repository where all the code for the project can be found.

# CHAPTER V

# SYSTEM TESTING

## INTRODUCTION

This chapter describes the levels of system testing soon after its development during the course of implementing for use. Every system must be tested before presenting, advertising or marketing. Testing gives room for error checking and identification, professionalism and implementation detection for project perfection. Self-testing, client, user, customer and student testing is therefore important.

## SYSTEM TESTING

System testing is the initial phase in the software development life cycle where the application I tested overall. The application is tested completely to confirm that it meets all of the functional and technical specifications required. The application must therefore be tested by the client, student, and customer to see if it best performs as described.

It also allows to not only test, but approve the organization’s necessities with its applications architecture. ATIA was cumulatively tested. It required any user really that logged in as admin user or student. No errors were made present even after continuous updates.

### UNIT TESTING

**Table 5.1 Unit Testing Results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST CASE** | **PURPOSE** | **PROCEDURE** | **EXPECTED RESULTS**  **(Pass/Fail)** | **ACTUAL RESULTS (Pass/Fail)** |
| User authentication | To test if user can login without error | Entering user email and password | Pass | Pass |
| Login out | To see if the user can log out successfully | Pressing the logout button | Pass | Pass |
| Exit button | To test if application can be exited | Single click to the button. | Pass | Pass |
| Add asset button | To test if by clicking the button the screen will go to add asset activity. | By clicking on the button labelled add asset. | Pass | Pass |
| Portfolio button | To test if labeled button Portfolio goes to the portfolio page. | Pressing on the Portfolio button | Pass | Pass |
| Payment button | To test if user can access the payment page with the button. | Pressing on the payments button. | Pass | Pass |
| Profile button | To test if user can access the Profile page with the button. | Pressing on the profile button. | Pass | Pass |
| Online work button | To test if the user can access the online work page | By pressing on the online work button | Pass | Pass |
| Menu icon | To test if the user can access the menu options like settings, terms and conditions, privacy policy, to-do list and logout. | By pressing on the menu icon represented by three vertical dots. | Pass | Pass |
| Notification icon | To see if user can see their notifications. | By pressing on the bell icon | Pass | Pass |

### INTEGRATION TESTING

Integration testing is any kind of software testing that looks to check the interface between parts against a software design. Every part of the software must be integrated in an iterative manner altogether.

It works to uncover absconds in the interface and interaction between coordinated modules. ATIA has a user friendly interface that is simple and straight forward. In regards to random testing methods where it was necessary to test, students and other users founds no difficulty is exploration. The table testing process was filled with Wongani lungu, Ebony Kawanga.

### VALIDATION TESTING

A validation testing compares the speed of the project, the adaptability, as well as dependability qualities of the project under testing phase against the thought set or assumed by the users to be useful and effective.

**Table 5.2 Validation Testing (Valid Input)**

|  |  |
| --- | --- |
| **FIELD** | **REQUIREMENT** |
| Email (user) | Required: Enter a string |
| Password (user) | Required: Enter integer or string not less than 6 |

**Table 5.3 Validation Testing (Invalid Input)**

**Null Data Test Case**

|  |  |  |
| --- | --- | --- |
| **Field** | **Test Data** | **Error Message** |
| Email (user) | Null | Enter email account |
| Password (user) | Null | Enter password |

### USER ACCEPTANCE TESTING

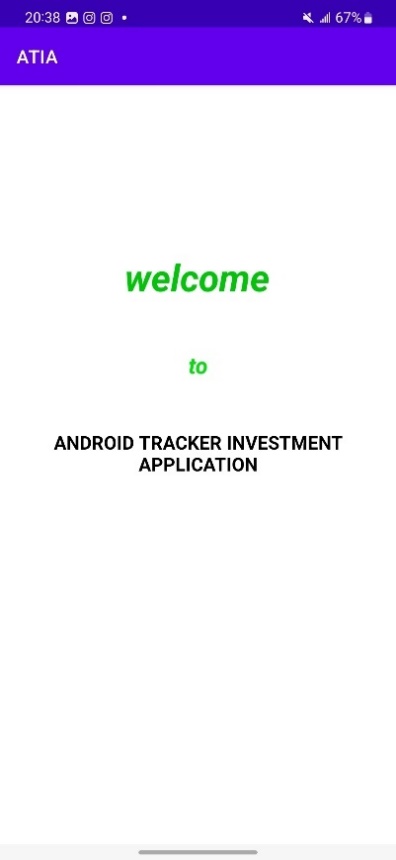
This is the most significant sort of testing as it is led by a quality assurance team who check whether the application meets the planned particulars and if it fulfills the users who will necessities. A question and answer group usually has an arraignment of pre-composed scenes and test cases that will be utilized to test the project.

# CHAPTER VI

# SYSTEM IMPLEMENTATION

## USER TRAINING AND DOCUMENTATION

The system developed is mobile application that allows the user to monitor their investments and take necessary records. Firebase is used in the authentication, Profile, Payment and Portfolio management module which are online operations. For mobile devices with android version 10 and above the application is very compatible with ATIA and can run correctly and efficiently.

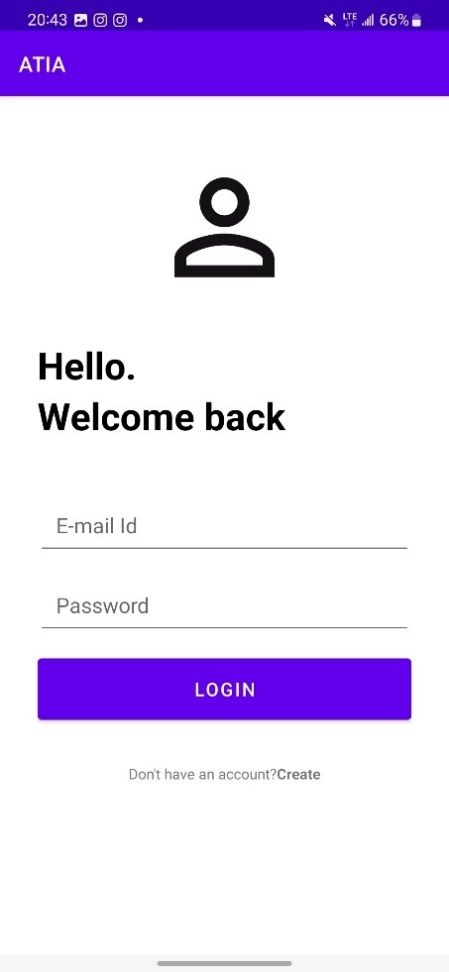
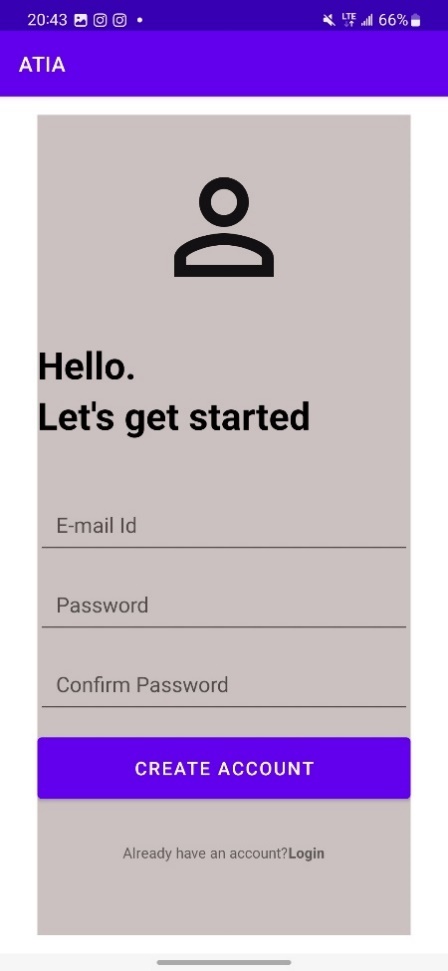
**

*This is the welcoming page of the application which stays for*

*About 5 seconds then proceeds login page when user is not logged in*

*But it will proceeds to the Dashboard when user is already logged in*

*Figure 6.1: screenshot of the splash screen*

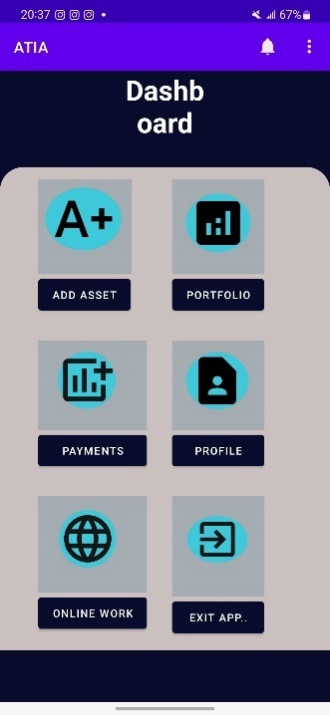
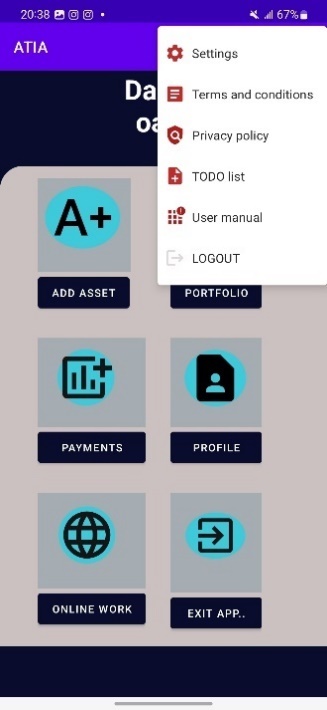
First will have the create account page where the first-time user will create their account and after creating the account, they have to verify it at the emails with the given link.

User can go to login page from this page by pressing or tapping on ‘Login’

Second, we have the login page where the user will enter the details like the email account registered and the password afterwards press on the button ‘LOGIN’.

User can go to the create account page from this page by pressing on ‘Create’

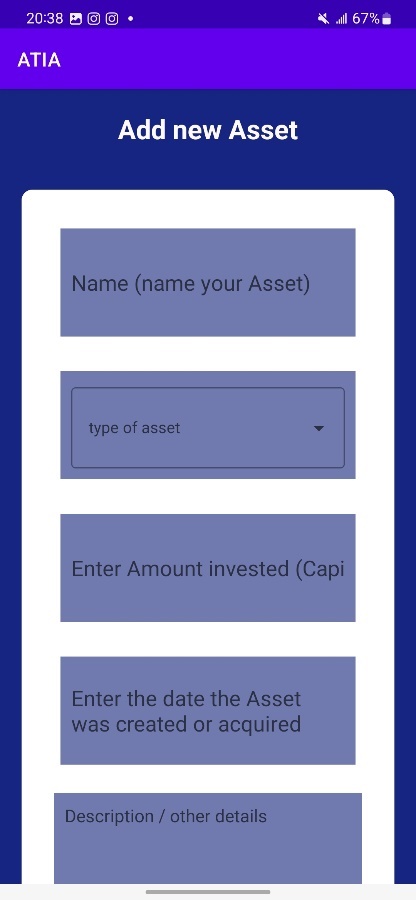
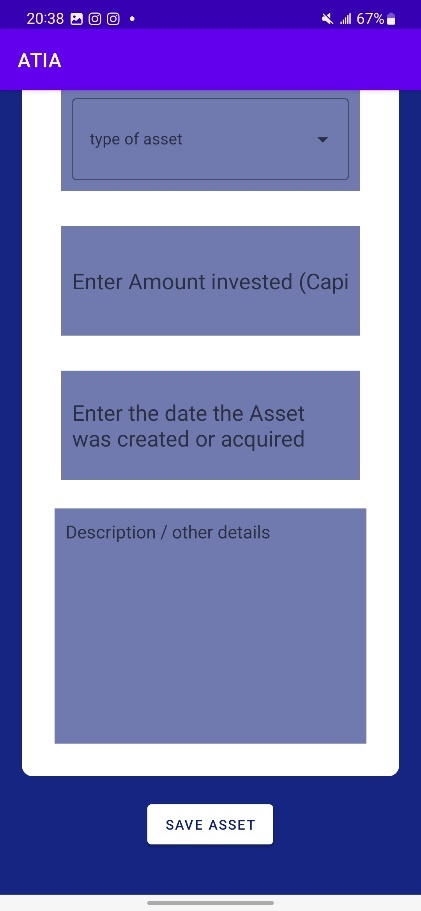
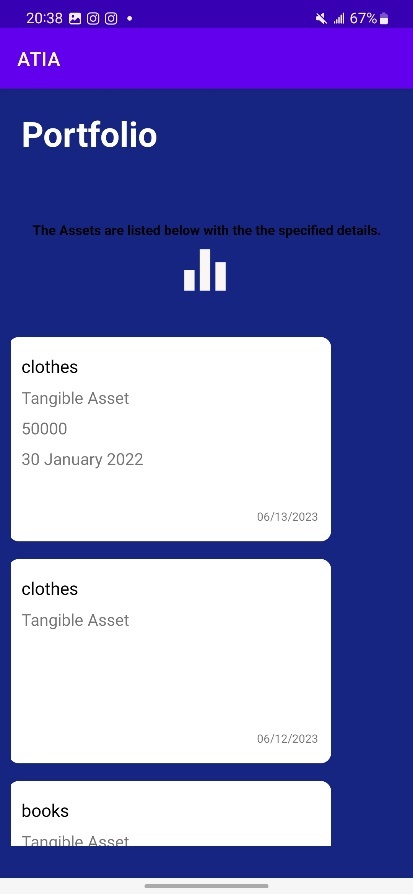
*Figure 6.2: screenshots of the login page and create account page.*



We have the Dash board where the user can access most of the features like add asset, Portfolio, payment, Profile, Online work, exit app and the menu options by pressing on the buttons labelled for each purpose.

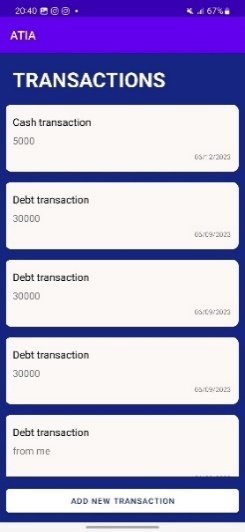
On the first screenshot the menu options are shown.

*Figure 6. 3: screenshots of the Dashboard and what happens when the menu icon is clicked.*

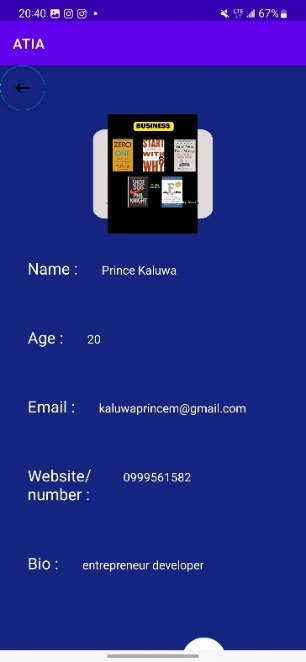
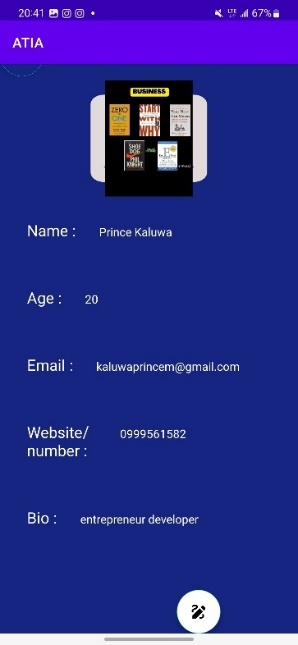
*Figure 6.4: screenshots of the add asset page and portfolio page.*

Add asset to help user add their asset details like name, type of asset, capital, date and other details and portfolio is where all those assets are displayed and can be edited by the user.

*Figure 6.5: screenshots of the payment module and its functions in action.*

Payments where the user can take records of all the transactions details either by photo proof or text details by pressing on the button with the desired option.

*Figure 6.6: screenshots of the Profile module and has the show profile page and create profile page.*

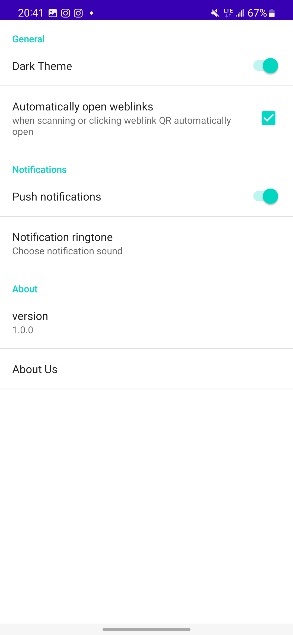
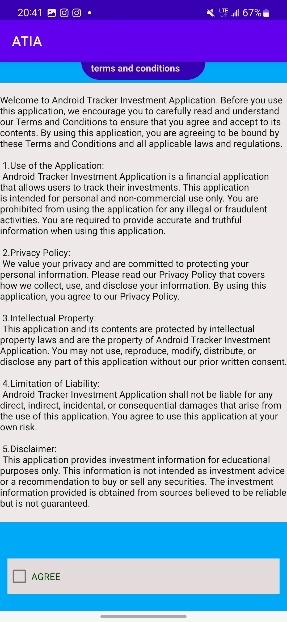
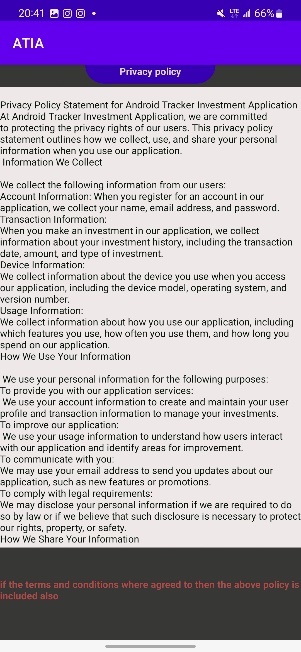
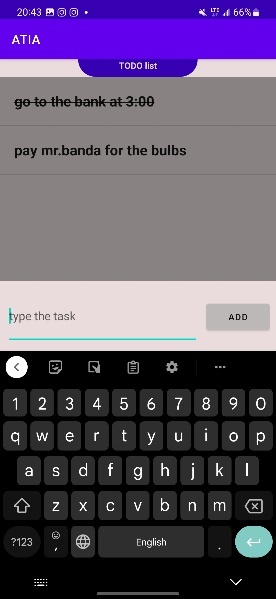
This is where the use can view and edit their profile and it holds vital personal details. There is the show profile page where the user can only see the details and they can go to the edit profile page by clicking the icon with the pencil and then place the details and save them by they must include the profile picture.



*This is where the user can learn and manage their investments using various websites depending on the task at hand.*

This is done by clicking on the desired text which will redirect the user to the appropriate web page.

*Figure 6.7: screenshot of the online work module*

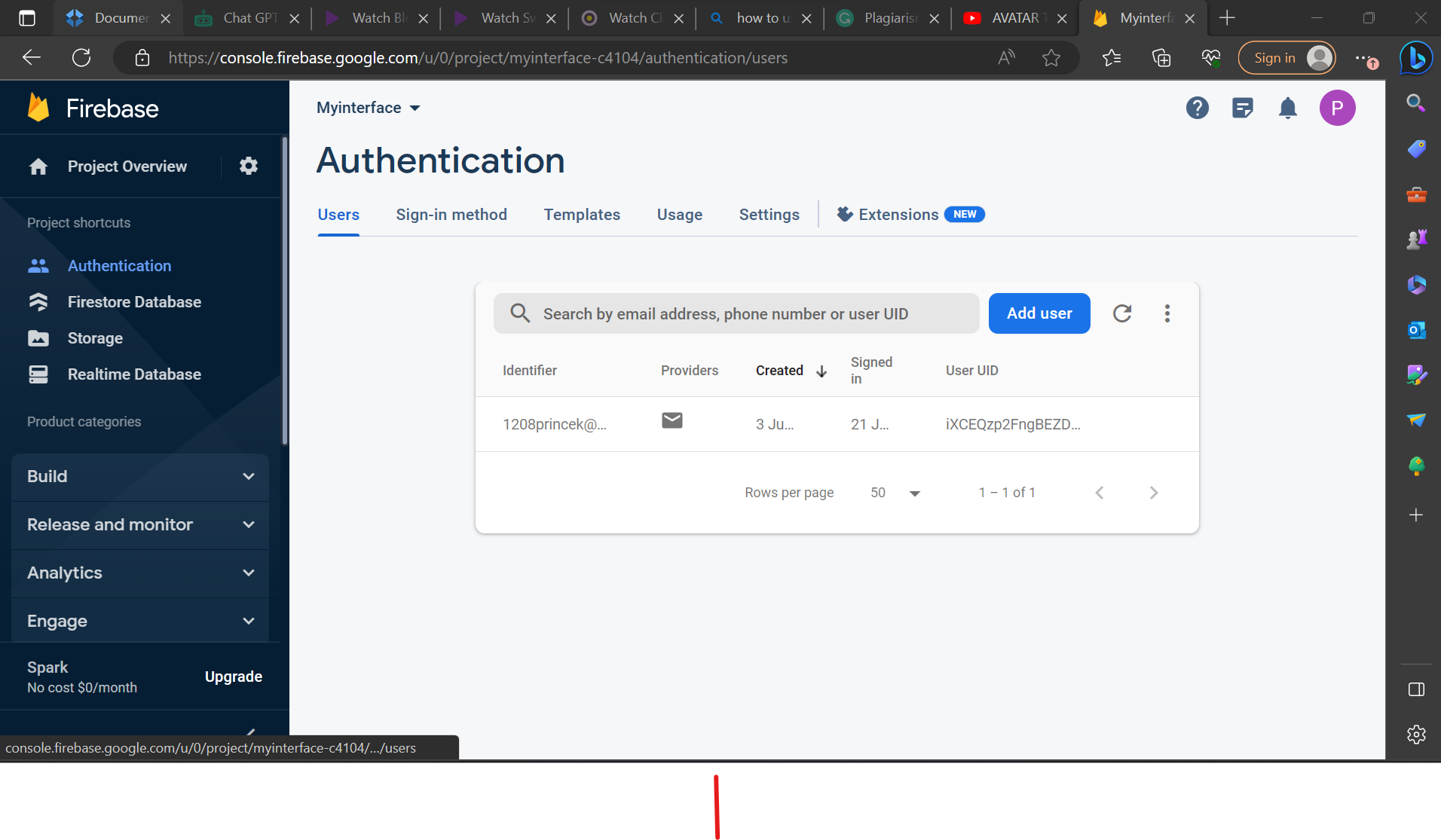
*Figure 6.8: screenshots of all the menu functions in action.*

Above all the menu options are shown what they do the first is the settings option to set the preferences that the user likes, terms and conditions agreement which outlines the terms of the application, privacy policy which outlines mostly the policies on the user data, simple to-do list and a user manual which will help users in some simple functionality.

**Table 6.1 modules and functions that use firebase (database)**

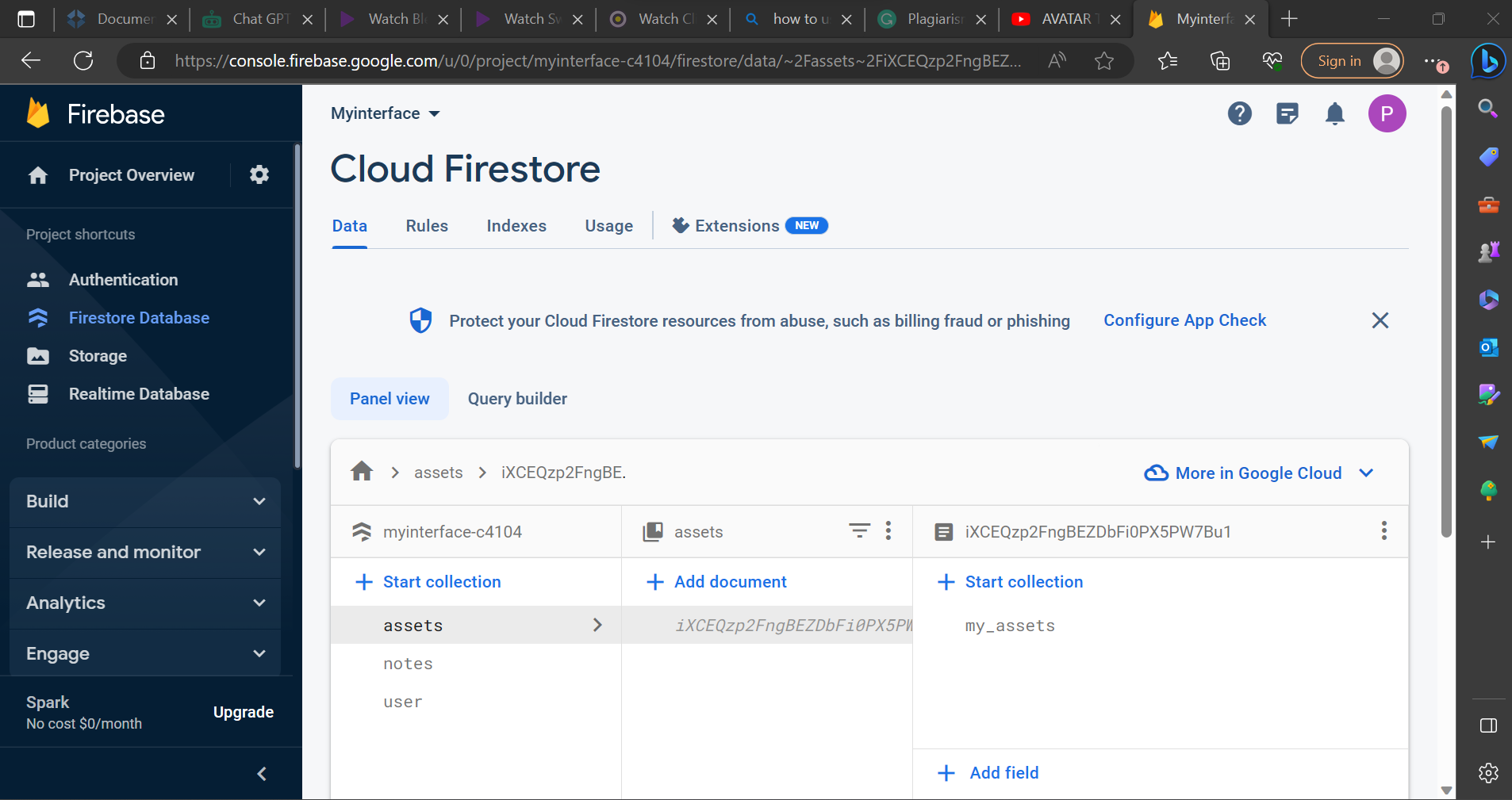
**Brief explanation of the uses of firebase**

|  |  |  |
| --- | --- | --- |
| **Module or function** | **Firebase service** | **Definition and summary use** |
| Sign-up operation | Authentication | Easy sign-in with any platform. Firebase Authentication aims to make building secure authentication systems easy, while improving the sign-in and onboarding experience for end users. |
| Login operation | Authentication | Easy sign-in with any platform. Firebase Authentication aims to make building secure authentication systems easy, while improving the sign-in and onboarding experience for end users. |
| Profile module | Firestore database and storage | Cloud Firestore is a NoSQL document database that lets you easily store, sync, and query data for your mobile and web apps - at global scale. |
| Add asset operation,  Payment module,  Portfolio | Firestore database | Cloud Firestore is a NoSQL document database that lets you easily store, sync, and query data for your mobile and web apps - at global scale. |



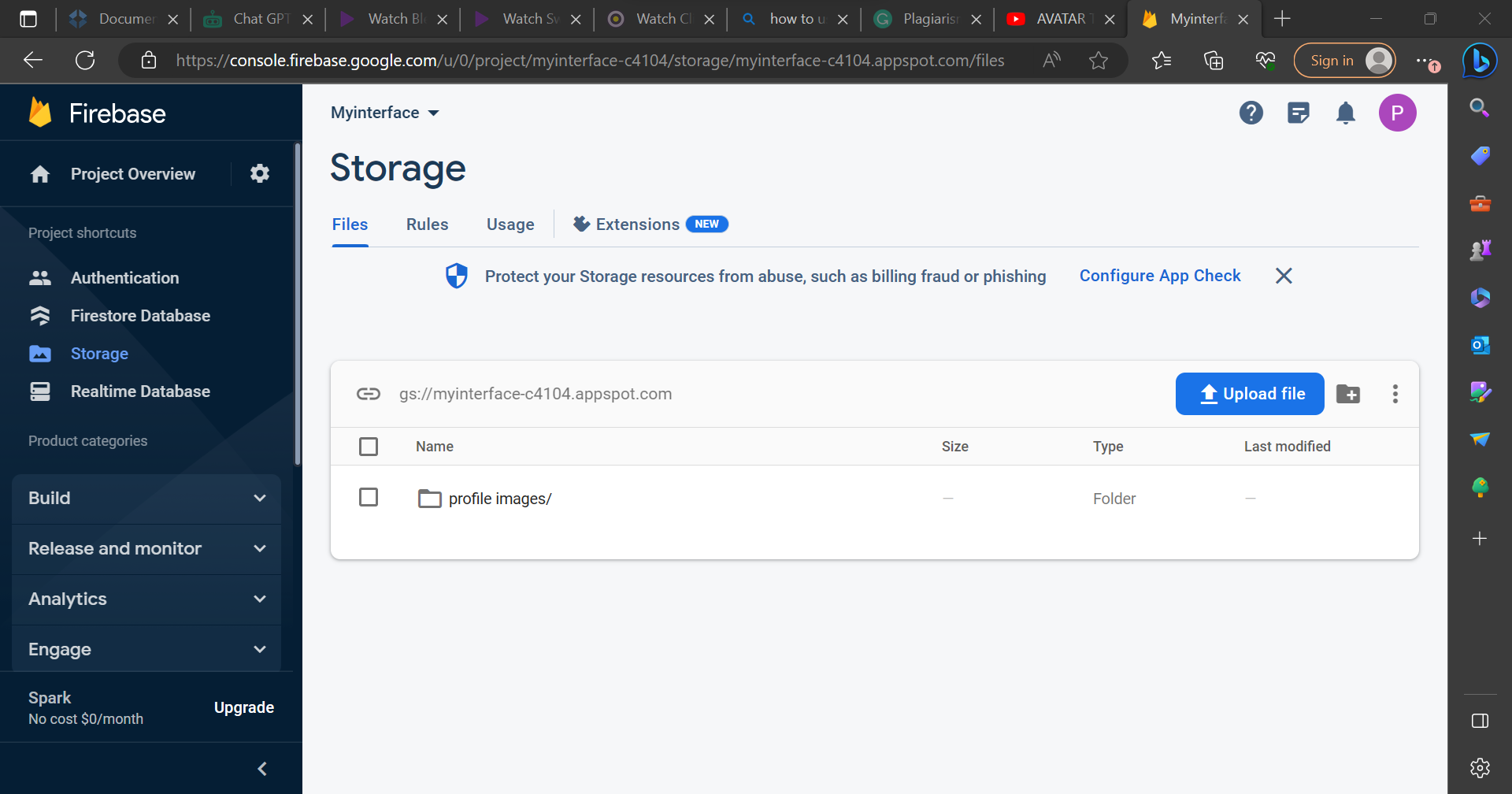
*Figure 6.9: screenshot firebase authentication in the console.*

The figure above shows the firebase console where the admin can monitors the accounts of all the users of the application



*Figure 6.9.1: screenshot firebase firestore Database used by admin.*

The diagram above shows the firebase firestore where the admin will monitor the user information for Portfolio, Payments and Profile details.



*Figure 6.9.2: screenshot firebase storage used by admin.*

The diagram above illustrates the firebase storage service where the admin can view the profile data of pictures for the users.

At this point, the project is running as anticipated and giving the desired output. Errors concerning operations have been dealt with, debugging, compiling and error correction. The system has thus been designed and implemented successfully.

## SAMPLE CODE

**4.7.1 add asset code: java code**

package com.example.myinterface;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.AutoCompleteTextView;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.Timestamp;  
import com.google.firebase.firestore.DocumentReference;  
  
public class AddAsset extends AppCompatActivity {  
  
 String[] item = {"Financial Asset", "Tangible Asset" ,"Intangile Asset"};  
 AutoCompleteTextView autotypeTextView;  
 ArrayAdapter<String> adapterItems;  
  
 EditText assettitleEditText,typeEditText,amountEditText,dateEditText,detailEditText;  
 Button saveAssetBtn;  
 TextView pageTitleTextView;  
 String name,type,amount,date,detail,docId;  
 boolean isEditMode = false;  
 TextView deleteAssetTextViewBtn;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_add\_asset*);  
  
 autotypeTextView = findViewById(R.id.*auto\_type\_text*);  
 adapterItems = new ArrayAdapter<String>(this,R.layout.*list\_item*, item);  
  
 autotypeTextView.setAdapter(adapterItems);  
  
 autotypeTextView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {  
 String item = adapterView.getItemAtPosition(i).toString();  
 Toast.*makeText*(AddAsset.this,"Item: " + item, Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
  
 assettitleEditText = findViewById(R.id.*asset\_title\_text*);  
 typeEditText = findViewById(R.id.*auto\_type\_text*);  
 amountEditText = findViewById(R.id.*asset\_amount\_text*);  
 dateEditText = findViewById(R.id.*asset\_date\_text*);  
 detailEditText = findViewById(R.id.*asset\_detail\_text*);  
 saveAssetBtn = findViewById(R.id.*save\_asset\_btn*);  
 pageTitleTextView = findViewById(R.id.*page\_title1*);  
 deleteAssetTextViewBtn = findViewById(R.id.*delete\_asset\_text\_view\_btn*);  
  
 name = getIntent().getStringExtra("name");  
 type = getIntent().getStringExtra("type");  
 amount = getIntent().getStringExtra("amount");  
 date = getIntent().getStringExtra("date");  
 detail = getIntent().getStringExtra("detail");  
 docId = getIntent().getStringExtra("docId");  
  
 if(docId!=null && docId.isEmpty()){  
 isEditMode = true;  
 }  
  
 assettitleEditText.setText(name);  
 typeEditText.setText(type);  
 amountEditText.setText(amount);  
 dateEditText.setText(date);  
 detailEditText.setText(detail);  
 if(isEditMode){  
 pageTitleTextView.setText("Edit your Asset");  
 deleteAssetTextViewBtn.setVisibility(View.*VISIBLE*);  
 }  
  
 saveAssetBtn.setOnClickListener( (v)-> saveAsset());  
  
 deleteAssetTextViewBtn.setOnClickListener((v)-> deleteAssetFromFirebase());  
 }  
  
 void saveAsset(){  
 String assetName = assettitleEditText.getText().toString();  
 String assetType = typeEditText.getText().toString();  
 String assetAmount = amountEditText.getText().toString();  
 String assetDate = dateEditText.getText().toString();  
 String assetDetail = detailEditText.getText().toString();  
 if(assetName==null || assetName.isEmpty() ){  
 assettitleEditText.setError("name of asset is required");  
 return;  
 }  
 Asset asset = new Asset();  
 asset.setName(assetName);  
 asset.setType(assetType);  
 asset.setAmount(assetAmount);  
 asset.setDate(assetDate);  
 asset.setDetail(assetDetail);  
 asset.setTimestamp(Timestamp.now());  
  
 saveAssetToFirebase(asset);  
  
 }  
  
 void saveAssetToFirebase(Asset asset){  
 DocumentReference documentReference;  
 if(isEditMode){  
 //update the note  
 documentReference = Utility.getCollectionReferenceForAssets().document(docId);  
 }else {  
 //create new note  
 documentReference = Utility.getCollectionReferenceForAssets().document();  
 }  
  
 documentReference.set(asset).addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task){  
 if(task.isSuccessful()){  
 Utility.showToast(AddAsset.this,"Asset added successfully");  
 finish();  
 }else{  
 Utility.*showToast*(AddAsset.this,"Failed while adding Asset");  
 }  
 }  
 });  
  
 }  
  
 void deleteAssetFromFirebase(){  
 DocumentReference documentReference;  
 documentReference = Utility.*getCollectionReferenceForAssets*().document(docId);  
 documentReference.delete().addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task){  
 if(task.isSuccessful()){  
 //note is deleted  
 Utility.*showToast*(AddAsset.this,"Asset deleted successfully");  
 finish();  
 }else{  
 Utility.*showToast*(AddAsset.this,"Failed while deleting Asset");  
 }  
 }  
 });  
  
 }  
}

# CHAPTER VII

# PROBLEMS FACED AND SOLUTIONS

## PROBLEMS FACED

The implementation of this system was dependent on software that required powerful computers with more than 4GBs of ram and a frequency speed of 2.30GHz at least. This is why there was a lagging in performance when extra detail was added. The school’s library had slow internet and l had to borrow a lap from Ebony since mine had technical difficulties which are resolved.

Other software modules had no download available content. Other dependencies needed to be downloaded by the IDE. This slowed down the implementation project work as well since there was need to work with what was made available.

## CODE EXECUTION

I had to get my training from YouTube and so some areas of the code logic had changed and so did the IDE due to system updates.

The ideas presented had not thought of ever becoming successful as output. This made implementation harder while verbal presentation was easier said than done. A single line could cause a syntax or logical error. Therefore, coding had to be precise.

## ERROR DEBUGGING

There were difficulties in finding errors during the debugging process stage some of which were not even defined by the script or the debugger. It is difficult to find errors in the back-end tool to provide solutions.

## SOLUTIONS

Some of the practically used solutions that helped throughout to project completion are:

* I had to go over the training videos for coding over and over again so as to find the proper.
* To eliminate some errors some dependencies had to be added in order for proper code execution.
* Manually go over the code to see where the IDE indicated errors and fix them appropriately.
* Refreshing the laptop was time to time helped with the performance.
* Downloading some dependencies that the project needed.
* Downloading JDK.
* Downloading SDK.
* Use of firebase which uses NoSQL.
* Downloading firebase dependencies.
* Importing the requires java libraries helped with code execution a lot.
* Use of suitable API made the program run correctly.

# CHAPTER VIII

# SUGGESTIONS FOR PROJECT ENHANCEMENTS

ATIA has been implemented in such a way that it accepts modifications and manual updating due to software changes. The software is user friendly and any further changes can be done easily. System restructuring can be carried out. System restructuring modifies source code in an effort to make it amenable to future changes even as the codes and programming advances to simplify work.

Integration with Additional Exchanges: The application could be extended to integrate with additional stock exchanges from around the world. This would allow users to track their investments in a wider range of markets, increasing the value of the application to a global audience.

Customized Investment Alerts: Another extension could add customized investment alerts, such as price alerts or news alerts, enabling users to stay up-to-date with their investments and make quick decisions.

Data Analysis and Visualization: The application could provide users with data analysis and visualization tools to help them make better investment decisions. These could include graphical representations of their portfolio's performance or historical data on specific stocks, allowing investors to see trends and make educated decisions.

Social Integration: The application could be extended to include social integration with other users, allowing investors to share their portfolios, compare returns, and exchange tips and advice.

Machine Learning: The application could incorporate machine learning algorithms to provide personalized investment advice to users based on their investment history and risk profile. This would be a powerful tool to help investors make informed decisions and maximize their returns.

Overall, these project extensions would enhance the Android Tracker Investment application's functionality, increasing its value to investors and providing a more comprehensive tool for investment tracking and analysis plus better features on the notification & alerts module and the settings preferences.

# CHAPTER IX

# CONCLUSION

The Android Tracker Investment application is an innovative solution that will help users manage their portfolio effectively and make informed decisions on their investments. The application will provide real-time updates and notifications, personalized investment news, customized portfolio management, and risk analysis. The application will help investors to achieve their financial goals through a transparent and efficient platform. The technical specifications and usability of the application make it a reliable application that can be used by beginners and experienced investors equally. The Android Tracker Investment application has the potential to fulfill investors' needs and add value to their investment journey.

The user interface (UI) design of the Android Tracker Investment Application was carefully crafted to provide a seamless and intuitive user experience. The interface was developed with a focus on simplicity and usability, ensuring that users can easily navigate through the application and access the desired features. The UI design incorporates visually appealing charts and graphs to present investment data in a visually engaging manner.

Throughout the project, various software development methodologies were utilized to ensure smooth coordination among team members and facilitate the timely completion of tasks. Agile methodologies, such as Firebase, were implemented to effectively manage the project and adapt to changing requirements. This approach allowed us to deliver incremental updates to the application, taking into account user feedback and continuously improving its performance.

The Android Tracker Investment Application demonstrates the practical application of computer science principles, such as data mining, machine learning, and software engineering. It serves as an example of how these concepts can be utilized to develop robust and feature-rich applications in the financial domain.

In conclusion, the Android Tracker Investment Application has successfully achieved its goal of providing users with a comprehensive platform to track and manage their investments. The project highlights the potential of computer science in developing innovative solutions for the financial sector. Moving forward, the application can be further enhanced by incorporating additional features, such as investment simulation tools and real-time news updates, to provide users with even greater control over their investment strategies. Overall, the Android Tracker Investment Application serves as a testament to the capabilities of computer science in revolutionizing the way we interact with and manage our investment portfolios.

# REFERENCES

Akdag, S., n.d. *Budget Analysis and Optimization.* s.l.:s.n.

Anon., 2017. *Android Studio 3.0 Development Essentilals.* Android 8 ed. s.l.:s.n.

Anon., 2019-2023. *coding videos.* [Online]   
Available at: www.toutube.com

Anon., 2019. *Android Studio tutorial.* s.l.:s.n.

Anon., 2023. *Android Developer Guide: The.* [Online]   
Available at: https://developer.android.com/guide/

Anon., n.d. *android investment tracker application.* [Online]   
Available at: https://nevonprojects.com/android-investment-Tracker-app/

Anon., n.d. *apk.* [Online]   
Available at: www.bing.com

Anon., n.d. *buid basic android apps with java | codecademy.* [Online]   
Available at: https://www.codecademy.com/learn/paths/introduction-to-android-with-java

Anon., n.d. *Build your first android app in java | android developers.* [Online]   
Available at: https://developer.android.com/codelabs/build-your-first-android-app

Anon., n.d. *commission-free stock trading & investing app | robinhood.* [Online]   
Available at: https://robinhood.com

Anon., n.d. *java for android.* [Online]   
Available at: https://www.coursera.org/learn/java-for-android

Anon., n.d. *java for android course.* [Online]   
Available at: https://www.coursera.org/learn/java-for-android

Anon., n.d. *java fundamentals for android.* [Online]   
Available at: https://androidatc.com/upload/editor\_upload/file/Java-Fundamentals-for-AndroidT-Application-Development-AND-404.pdf

Anon., n.d. *Learn java for android app development-A complete guide.* [Online]   
Available at: https://www.geeksforgeeks.org/learn-java-for-android-app-development-a-complete-guide/

Anon., n.d. *types of assets.* [Online]   
Available at: www.youtube.com

Anon., n.d. *Yahoo finance.* [Online]   
Available at: www.yahoofinance.com

CodingZest, n.d. *Modern Dashboard UI Design Android Studio Tutorial | Cardview Android Studio | Grid layout Android.* [Online]   
Available at: https://www.youtube.com/watch?v=ixRXEoGAEZM

Craig, A. G. a. C., 2015. *Learn Android Studio Build Apps quickly and effectively.* s.l.:Apress.

DCodeMania, n.d. *Create Option Menu In Empty Activity On Android Studio | Action Bar Menu.* [Online]   
Available at: https://www.youtube.com/watch?v=h47CqbmhdAs

Development, K., n.d. *Terms & Conditions Material Design Android Studio 2020.* [Online]   
Available at: https://www.youtube.com/watch?v=DP6pGwtUID4

EduHindiMe, n.d. *How to add privacy policy in android studio Step by step 2020.* [Online]   
Available at: https://www.youtube.com/watch?v=5gPXM7Zmf2E

flow, C. i., n.d. *Options Menu with Sub Items - Android Studio Tutorial.* [Online]   
Available at: https://www.youtube.com/watch?v=oh4YOj9VkVE

flow, C. i., n.d. *Popup Menu - Android Studio Tutorial.* [Online]   
Available at: https://www.youtube.com/watch?v=s1fW7CpiB9c

gap, f., n.d. *Dashboard UI Design Using Grid Layout in Android Studio.* [Online]   
Available at: https://www.youtube.com/watch?v=amnSAwVytHg

Gap, F., n.d. *How To Make Dropdown Menu in Android Studio.* [Online]   
Available at: https://www.youtube.com/watch?v=jXSNobmB7u4

India, E. C., n.d. *How to work Option Menu Item Click and Open Another Activity In Android Studio by Nilkanth Pawan.* [Online]   
Available at: https://www.youtube.com/watch?v=BfrIFVsvUzI&t=5s

Krantz, M., n.d. *Investing Online for Dummies.* 9 ed. s.l.:Wiley.

Liao, S., 2014. *Migrating to Android for IOS Developers.* 1 ed. s.l.:Apress.

Madden, B., n.d. *Bloomberg.* [Online]   
Available at: www.bloomberg.com

Mnr, S. P., n.d. *How to Use ScrollView in Android App - Android Studio 2.2.2 Tutorial.* [Online]   
Available at: https://www.youtube.com/watch?v=o2VfWU9EUtY

PaniBus, D., n.d. *Build a Note App with Android Studio, Java and Permanent Storage.* [Online]   
Available at: https://www.youtube.com/watch?v=48EB4HeP1kI

Peojects, T., n.d. *How to move from one activity to another in android studio on button click | Tech Projects.* [Online]   
Available at: https://www.youtube.com/watch?v=JOdWT50bWw4

Risky, A., n.d. *Modern Dashboard UI Design Android Studio Tutorial.* [Online]   
Available at: https://www.youtube.com/watch?v=LOcD1evBcSA

Smyth, N., 2020. *Android studio 4.1 Development Essentials-java edition.* s.l.:s.n.

TechiZVIbe, n.d. *How To Use ScrollView in Android | Scroll View And Constraint Layout | Android Studio Tutorial.* [Online]   
Available at: https://www.youtube.com/watch?v=h6g4NpiC0i4

Tuto, E., n.d. *Notes App With Firebase | Android | 2023.* [Online]   
Available at: https://www.youtube.com/watch?v=jzVmjU2PFbg

## PLAGIARISM



Plagiarism Checker X Originality Report

Similarity Found: 8%

Date: Monday, June 26, 2023

Statistics: 613 words Plagiarized / 7274 Total words

Remarks: Low Plagiarism Detected - Your Document needs Optional Improvement.

PROJECT DOCUMENTATION PRINCE B. KALUWA REG NO: 20311351013 MINI-PROJECT

Submitted In partial fulfillment of the requirements for the Degree of BACHELOR OF SCIENCE IN COMPUTER SCC-INCE / DMI-STJOHN THE BAPTIST UNIVERSITY MANGOCHI, MALAWI. PROFORMA FOR APPROVAL OF PROJECT PROPOSAL Proposed Project Team: S.No. \_Reg. No. \_Name of the students \_Semester \_Branch \_ \_1 \_20311351013 \_Prince B. kaluwa \_VI \_ BSc \_ \_ Title of the Project: \_Android Tracker Investment Application (ATIA) \_Subject Area: \_Mobile application and investment-finance detailing \_ \_Name of the Guide: Mr.

Hope soko \_ \_Designation : \_Lecture - Il \_ \_Address with Phone No.: DMI — St. John



The Baptist University (+265 881 422 991) \_ \_Office: \_DMI — St. John The Baptist

University Mangochi, Malawi \_ \_Residence: \_DMI-St. John the Baptist University

Mangochi, Malawi \_ \_No. of projects & students currently working under the Guide: \_2

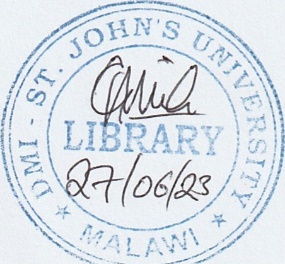
Projects, 2 Students \_ Signature of the Student Signature of the Guide

Datewith seal N.B.: Please do not forget to enclose the synopsis of the project and the Bio-data of the Guide. In case the complete and signed Bio-Data of the Guide is not enclosed, the proposal will not be entertained.For Office Use only: SYNOPSIS \_APPROVED NOT APPROVED GUIDE APPROVED

NOT APPROVED \_ Comments / Suggestions for reformulation of the Project.

Date Signature of the HOD BIO-DATA OF THE PROPOSED GUIDE FOR

PROJECT WORK 1. PERSONAL INFORMATION NAME (in block letters) : Mr. HOPE





## COPYRIGHT LETTER

I **Mr. PRINCE B. KALUWA** hereby declare that this project report **ANDROID TRACKER INVESTEMENT APPLICATION (ATIA)** submitted to **DMI-ST. JOHN THE BAPTIST UNIVERSITY**, in the partial fulfillment of requirements for the award of the degree of Science in Computer science is a record of the original work done by me under the supervision of **Mr. HOPE SOKO**.

I certify that this project thesis was indeed done and written by me and contains no material that has been accepted for the award of any other degree in any other university, to the best of my knowledge and belief and contains no material previously published or written by another person, except where due reference has been made in the text. Therefore, no part of this documentation may be reproduced in any form without prior written permission by me, except in the case of brief quotations and certain noncommercial uses. I give consent to this copy of thesis to be deposited in the university library being made available for viewing and referencing.

**Yours faithfully Signature of the Examination office**

**Mr. Prince B. kaluwa**

© Copyright by Prince B. kaluwa 2023

All Rights Reserved