

Application Development

(FinByte)



**Cluster Innovation Centre
University of Delhi**

**ASHUTOSH JHA
SHUBHAM CHAUHAN**

Nov 2021

Summer Internship Report

CERTIFICATE OF ORIGINALITY

The work embodied in this report entitled “**Application development** (*FinByte*)” has been carried out by **Ashutosh Jha and Shubham Chauhan** for “**Summer Internship Report**”. We declare that the work and language included in this project report is free from any kind of plagiarism.

The work submitted is original and has not been submitted earlier to any institute or university for the award of any degree or diploma.

Ashutosh Jha

(11811)

Shubham Chauhan

(11838)

ABSTRACT

The objective of the summer internship was to apply the theoretical as well as practical knowledge of Application development to build finbyte's application. The main purpose is to make mobile apps by first analyzing various requirements using software specifications and constructing a software specifications document and design document.

The prime objective is to build Finbyte application on android framework and hosted on Firebase which offers backend as a service.

TABLE OF CONTENTS

- 1. Introduction & Objective**
 - 1.1. Problem Statement
 - 1.2. Features & Goals
- 2. System Analysis**
 - 2.1. Non-functional Requirements
 - 2.2. Technical Requirements
 - 2.3. Input Design
 - 2.4. Output Design
- 3. Project Schedule**
 - 3.1. Project Phases
 - 3.2. Project Schedule
- 4. IMPLEMENTATION**
 - 4.1. Implementation Tools
 - 4.2. Tools Used
 - 4.3. System Implementation
 - 4.4. System Maintenance
- 5. DEVELOPMENT**
 - 5.1. Designing Layouts
 - 5.2. Implementing the classes and functions
 - 5.3. Hosting on Firebase and setting required methods
- 6. TESTING**
 - 6.1. Types of testing done
- 7. CONCLUSION**
- 8. REFERENCES**

Abbreviations and terms

Android	An open source platform designed for mobile devices
OS	Operating system
SDK	Software development kit
JRE	Java Runtime Environment
IDE	Integrated development environment
GUI	Graphical user interface
Android Emulator	Container for running Android OS along with applications
XML	Extensible Markup Language
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
RAM	Random Access Memory
IT	Information Technology
GHz	GigaHertz
API	Application Programming Interface
ICT	Information and Communication Technology
UI	User Interface
URL	Universal Resource Locator
HD	Hard Disk
CPU	Central Processing Unit

1. INTRODUCTION & OBJECTIVE

The primary objective of the project was to build the finbyte website and a script via which the clients can buy / sell stocks and a testing system with forward and backward compatibility.

1.1. Problem Statement

This project tries to tackle various problems present in the current finbyte's system of buying and selling stocks. The project tries to :

- Develop Finbyte's website
- Automate the current trading system
- Provide a testing system
- Provide forward and backward compatibility to the trading system

1.2. Features & Goals

- Authorization & Authentication (login screen): The application will have an authorization & authentication interface where users can login. All the users having an email ID can sign up directly or we can provide them login credentials individually.
- Home Screen: We are planning to implement different home screens for the different clients according to their levels.
- Trading System: A script via which clients / admins can automate their trading given they provide a stop loss, an entry point, an exit point, no. of orders etc.
- Testing System: The automated system should have a testing system which tests the trading results and generate reports for further analysis like why the trade failed, the success percentage etc,

2. SYSTEM ANALYSIS

Analysis can be defined as breaking up any whole so as to find out their nature, function etc. It defines design as to make preliminary sketches of; to sketch a pattern or outline for plan. To plan and carry out especially by artistic arrangement or in a skillful way. System analysis and design can be characterized as a set of techniques and processes, a community of interests, a culture and an intellectual orientation.

It is the most creative and challenging phase of the system life cycle. The output of this phase is a description of the recommended alternative solution. The steps involved during system analysis process are:

- Understanding application
- Planning
- Scheduling
- Forming theory background
- Developing the solution
- Performing the test analysis
- Recommending alternative solutions
- Launching the proposed solution

System analysis can include looking at end-user implementation of a software package or product and involves gathering requirements for the system. In System Analysis more emphasis is given to understanding the details of an existing system or a proposed one and then deciding whether the proposed system is desirable or not and whether the existing system needs improvements. Thus, system analysis is the process of investigating a system, identifying problems, and using the information to recommend improvements to the system. The project should address a real world interface design and be implementable. Feasibility Study is a major process in System Analysis. It helps in determining whether the project will yield a desired output with realistic and economic use of available resources.

2.1. *Non functional requirements:*

Non-functional requirements are not concerned with the functions of the system. Instead, they look at the criteria to which the application is expected to conform to. Non-functional requirements can include things like response time and reliability. Some of the Non-functional requirement for the applications are:

- All the components of the application should be fully loaded within reliable time without downgrading performance.
- Should be user friendly and content should be readable by all types of users.
- Should take minimal time, effort, resources or cost to create the android application.
- Should provide the correct information about all the modules.
- Should consider the Response times
- Android applications should strictly follow Material Design guidelines
- The data should get retrieved from the server even in all types of network connectivity
- All the communications between server and app should have a secure endpoint

2.1.1. *Usability Requirements:*

The android applications can be accessed by users on their android devices, given they install the desired apks. We will be pushing this app to the playstore where the different users can download their respective apps.

2.1.2. *Efficiency Requirements:*

Mean Time to Repair (MTTR) - Even if the system fails, the system will be recovered back up within an hour or less.

2.1.3. *Accuracy:*

The system should accurately provide real time information taking into consideration various concurrency issues. The system shall provide 100% access reliability.

2.1.4. *Safety Requirements:*

- As the app is connected to the internet occasionally, an antivirus should be installed on the system for its safety purpose.
- Since, app does not require any body movement to work properly, there is no physical safety requirement.

2.1.5. *Performance Requirements:*

- The program must be able to run concurrently by multiple users.
- Transmission of data to the database shall occur spontaneously.
- Queries upon the database shall be performed in less than 5 seconds.
- Reading and writing data shall occur instantly with the required HTTP protocols

2.1.6. *Maintainability and Portability Requirements:*

Changes (new parts in addition, password changes, and database changes) must be verified once per day at least.

2.1.7. *Security Requirement:*

- Re-authentication will be required once every week.
- The security and privacy policies will be open source and disclosed at the time of installation.

2.1.8. *Software Quality Attributes:*

- Availability: The system shall be available to all the users through information systems.
- Reliability: Overall reliability of the system and roll information shall be achieved through the process of database manipulation.
- Reusability: The app shall be able to be reused for each new semester.
- Robustness: If no network connection can be established to sync the data then the user will be allowed to do that later. If a network connection is lost during syncing, the app will allow the user to continue roll syncing at a later time.
- Updatability: The system shall allow for addition or deletion of classes, subjects, assignments and notes while incorporating new semesters.

- Usability: Usability of the system shall be achieved through help page, FAQs and an introductory training guide for all students upon installation of the program.

2.1.9. Feasibility Study:

After studying and analyzing all the existing or required functionalities of the system, the next task is to do the feasibility study for the project. All projects are feasible – given unlimited resources and infinite time.

Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

2.1.9.1. Economic Feasibility:

This is a very important aspect to be considered while developing a project. We decided the technology based on the minimum possible cost factor. All hardware and software cost has to be borne by us.

2.1.9.2. Technical Feasibility:

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system and checked if everything was possible using different types of frontend and backend platforms.

2.1.9.3. Operational Feasibility:

No doubt the proposed system is fully GUI based and is very user friendly and all inputs to be taken are all self-explanatory even to a layman.

2.2. Technical Requirements

2.2.1. *Hardware Requirements:*

The minimum requirements for PC are:

- Operating System: Windows 10
- Application system: Android Studio, Android SDK
- Development environment: Java, XML, Android SDK, React
- IDE: Android Studio 4.1.1, Visual Studio Code
- Frameworks: Android framework , Firebase (Authentication Methods, Realtime Database, Cloud Storage, Cloud Functions, Cloud Messaging)

The minimum requirements for Android device are:

- Android version: Nougat (7.1)
- API level: 25
- RAM: 2 GB recommended
- Processor: Quad Core 1.2GHz
- Internet Connectivity
- Google Play Services

2.2.2. *Software Requirements:*

- Application system : Android Studio, Android SDK
- Language : Java, XML, Android Framework, React, HTML, CSS

2.3. Input Design

Input design is part of overall system design that requires special attention. For designing input data, the data entered should be easy and free from errors. The input forms and dialog boxes in applications are designed using the controls available. Validation is made for each and every data that is entered. Help information is provided for the users.

Input design is the process of converting the user originated inputs to a computer based format. The collection of input data is considered to be the most expensive part of the system design. Since the input has to be planned in such a manner so as to get relevant information, extreme care is taken to obtain pertinent information.

The application gives an interface for registering and validating users and the data entered by users are converted to suitable models and uploaded to firebase for further use

2.4. Output Design

Output design of the applications generally refers to the results and information that are generated by the system for many end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application. The output is designed in such a way that it is attractive, convenient and informative. As the outputs are the most important sources of information to the users, better design should improve the system's relationships with us and also will help in decision making. The application presents the information in an interactive, user friendly manner. Moreover, the user can further print Invoices and can interact with the app.

3. PROJECT SCHEDULE

3.1. Project Phases

It is very important to follow up phases as the pre-requirement for every next phase is the completion of the previous one. Each project was divided broadly into 4 phases

- Designing Layouts
- Implementing the classes and functions
- Hosting on Firebase and setting required methods
- Integrating APIs

3.2. Project Schedule

Tasks	Time Allocation (person days)	Planned Start	Planned End	Optimistic Time Estimates (person days)	Most Likely Time Estimates (person days)	Pessimistic Time Estimates (person days)	Estimated Time $T_e = (T_0 + 4T_m + T_p) / 6$
Requirement Analysis and Project Planning	12	06/06/2021	18/06/2021	5	10	13	10
Setting up the Environment	10	20/6/2021	2/7/2021	3	5	14	6
Software Construction	28	3/7/2021	30/7/2021	9	19	30	19

Unit Testing	12	23/7/2021	7/8/2021	5	10	14	10
System Testing	5	8/8/2021	14/8/2021	2	4	8	4

4. IMPLEMENTATION

4.1. Implementation Tools

The various implementation tools are listed below:

- Operating System - Windows 10
- Development environment - Java, XML, Android SDK
- IDE - Android Studio 4.1.1

4.2. Tools Used

The front is an abstraction, simplifying the underlying component by providing a user-friendly interface. There are several tools available that can be used to develop the front end of android application

4.2.1. *Android framework:*

Android is one of the Open source platforms. It is created by Google and owned by Open Handset Alliance. It is designed with the goal “accelerate innovation in mobile”. As such Android has taken over a field of mobile innovation. It is definitely a free and open platform that differs hardware from software that runs on it. It results in many more devices running the same application. Also it gives the possibility of a friendlier environment for developers and consumers. Android is a complete software package for a mobile device. Since the beginning the android team offers the developing kit (tools and frameworks) for creating mobile applications as quickly and easily as possible. In some cases you do not especially need an android phone but you are very welcome to have one. It can work right out of the box, but of course users can customize it for their particular needs. For manufacturers it is a ready and free solution for their devices. Except specific drivers, the android community provides everything else to create their devices.

4.2.2. *Firebase:*

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014. As of November 2020, the Firebase platform has 18 products, which are used by 1.5 million apps. We used it for hosting purposes. The features we used in our app are

- Authentication Methods
- Realtime Database
- Cloud Storage
- Cloud Functions
- Cloud Messaging

4.3. System Implementation

System implementation is the important stage of a project when the theoretical design is tuned into practical system. The main stages in the system implementation are Planning, Training, System testing and Changeover planning

4.4. System Maintenance

Software maintenance is far more than finding mistakes. Provision must be made for environment changes, which may affect either the computer, or other parts of the computer based systems. Such activity is normally called maintenance. It includes both the improvement of the system functions and the corrections of faults, which arise during the operation of a new system. It may involve the continuing involvement of a large proportion of computer department resources. The main task may be to adapt existing systems in a changing environment. Backup for the entire database files are taken and stored in cloud systems so that it is possible to restore the system at the earliest. If there is a breakdown or collapse, then the system gives provision to restore database files. Storing data in a separate secondary device leads to an effective and efficient maintenance of the system.

5. DEVELOPMENT

5.1. Designing Layouts

Layout defines the structure for a user interface in your app, such as in an activity. All elements in the layout are built using a hierarchy of View and ViewGroup objects. A View usually draws something the user can see and interact with. Whereas a ViewGroup is an invisible container that defines the layout structure for View and ViewGroup objects.

The View objects are usually called "widgets" and can be one of many subclasses, such as Button or TextView. The ViewGroup objects are usually called "layouts" and can be one of many types that provide a different layout structure, such as LinearLayout or ConstraintLayout.

You can declare a layout in two ways:

- Declare UI elements in XML: Android provides a straightforward XML vocabulary that corresponds to the View classes and subclasses, such as those for widgets and layouts.
- Instantiate layout elements at runtime : App can create View and ViewGroup objects (and manipulate their properties) programmatically.

Declaring your UI in XML allows you to separate the presentation of your app from the code that controls its behavior. Using XML files also makes it easy to provide different layouts for different screen sizes and orientations. So, the main elements used in our project were:

- Parent Views: Linear View, Relative View and Constraint view for holding widgets and child elements
- Child Elements:
 - Buttons: Whenever user clicks it, then it performs a specific action
 - EditText: To take user Input
 - CardView: Show information inside cards that have a consistent look across the platform

- GridLayout: a Layout manager that lays out a container's components in a rectangular grid. The container is divided into equal-sized rectangles, and one component is placed in each rectangle. Used along with a card view.
- TextView
- ImageView etc.
- Sub-child elements: Combination of the above child elements

5.2. Implementing the classes and functions

We used various classes and methods in our project which is beyond the scope of this report. The main methods and classes used were:-

- `setOnClickListener`: It helps us to link a listener with certain attributes. `setOnClickListener` is a method in Android basically used with buttons, image buttons etc.
- `addOnSuccessListener`
- `addOnFailureListener`
- `Toast`: For displaying user a message
- `Intent`: For taking user to other activities
- `DatabaseReference`: for referring to the firebase database according to the document ID etc.

5.3. Hosting on Firebase and setting required methods

Firebase gives us functionality like analytics, databases, messaging and crash reporting so we can move quickly and focus on the users. Firebase is built on Google infrastructure and scales automatically, for even the largest apps. The steps involved were:

- We need to set up a Firebase Account and create a new project.
- Then provide the necessary details and provide the app's SHA-1 fingerprint.
- Add the firebase configuration files and plugins.
- Add firebase SDK to the app
- In the console under sign-in method, enable Email / Password authentication and implement the required methods according to the Firebase Docs.
- Then, in the database section, select the Realtime database and the region we wish to store our data. Then make a collection and accordingly document. Each document has

a specific ID. In our case, it is the user's UID (Unique Identification Number). We use DatabaseReference to refer to documents and their subchilds.

6. TESTING

Testing is a method of assessing the functionality of a program. Testing is a set of processes aimed at investigating, evaluating and ascertaining the completeness and quality of a project. Testing refers to the process of implementing all or part of the system with the intent of finding errors. It is performed in order to find the bugs or errors in the system and minimize it. In general, testing is finding out how well something works. Testing is more than just debugging.

6.1 Types of testing done

- **White Box Testing**

It is a way of testing the software in which the tester has knowledge about the internal structure or the code or the program of the software. Knowledge of implementation is required. This type of testing of software is started after detail design document. It is generally applicable to the lower levels of software testing. Data domains along with inner or internal boundaries can be better tested. We have done path, loop and conditional testing under this category.

- **Black Box Testing**

It is a way of software testing in which the internal structure or the program or the code is hidden and nothing is known about it. No knowledge of implementation is needed. This testing can be initiated on the basis of requirement specifications document. Can be done by trial and error ways and methods. We have done functional, non-functional and regression testing under this category.

- **Unit Testing**

Each division class of every page or layout is tested in the android device. Inspecting XML, and modifying style and layout in real-time.

- **Integration testing**

This type of white box testing involves combining individual units or components of the application's source code and testing them as a group. The purpose is to expose errors in the interactions of the different interfaces with one another. It takes place after unit testing.

- **System Testing**

After completing the overall application design and development it is tested for error. We have also validated output errors with detailed debugging and have taken necessary actions.

- **Performance Testing**

Performance testing is designed to test the run-time performance of software within the context of an integrated system. Performance testing occurs throughout all steps in the testing process. Even at the unit level, the performance of an individual module may be assessed as white-box tests are conducted.

7. CONCLUSION

In this Project, prototypes of the application was developed. The application was able to connect to the server and retrieve information from the server. It can be installed and functions on several devices at the same time.

The prototype is working on a desktop as well as android platform and made on the base of Java and React framework. It uses Firebase to store and to receive information. The information is stored on Firebase and can be accessed any time.

References:

[1] How to write a good SRS for your Project

From <https://www.geeksforgeeks.org/how-to-write-a-good-srs-for-your-project/>

[2] How To Write Software Design Documents

From <https://blog.tara.ai/software-design-documents/>

[3] Important Features You Should Include During Android App Development

From <https://www.mindinventory.com/blog/features-include-during-android-app-development/>

[4] Software Development Process: How to Pick The Process That's Right For You

From <https://plan.io/blog/software-development-process/>

[5] Software Development Process : From Idea to Product

From <https://www.udacity.com/course/software-development-process--ud805>

[6] How to turn an app idea into a product: Creating a feature set

From <https://www.peerbits.com/blog/turn-an-app-idea-into-a-product.html>

[7] Android Developers Documentation

From <https://developer.android.com>