

**A PROJECT REPORT  
ON  
“ATTENDANCE SYSTEM USING QR CODE”  
SUBMITTED FOR PARTIAL FULFILMENT OF AWARD OF  
BACHELOR OF TECHNOLOGY  
IN  
COMPUTER SCIENCE & INFORMATION TECHNOLOGY**

**BY**  
**RIDDHI GUPTA (1902300110032)**

**UNDER THE SUPERVISION OF -**

**Prof. Parul Singh  
Assistant Professor  
(Department of CSE/IT)**

**DRONACHARYA**

**DRONACHARYA GROUP OF INSTITUTIONS, GREATER NOIDA (Affiliated  
to AKTU - Lucknow, Uttar Pradesh)**

**MAY , 2023**

## **CERTIFICATE**

This is to certify that the project entitled "**Attendance System Using QR code**" is being submitted by **Riddhi Gupta**(1902300110032) in partial fulfilment for the degree of bachelors of technology in Computer science and Information technology of Dronacharya Group of institutions, greater Noida ( (Affiliated to AKTU-Lucknow) is a record of their own work, carried out under my supervision.

Prof Parul Singh

(Assistant Professor)

Department of CSIT

Prof Bipin Pandey

(Head Of Department)

Department of CSIT

## **ABSTRACT**

Today's academic landscape requires more lecture preparation and student engagement than ever before. As such, the need has arisen for educational tools that can provide instructors with the insights they need to create the most educational curriculum possible while simultaneously incentivising students to make the most of their classroom experience.

An Attendance based system can be a valuable asset as manual computation produces errors, and wastes a lot of time. For the Stated reason an efficient cloud based attendance system is designed to track the student's activity in the class. This application takes the attendance using QR Code. The Data will be Stored in the **firebase** through which teachers, admin can Know the student who were present in the class.

Cloud Storage for Firebase is a powerful, simple, and cost-effective object storage service built for Google scale.

**File storage:** You can store all types of information in the cloud, including files and email.

**Backing up data:** You can also use the cloud to protect your files. Admin can access the data from anywhere, anytime

## **ACKNOWLEDGEMENT**

It is always a difficult task to acknowledge all those who have been of tremendous help in an academic project of this nature and magnitude nevertheless, we have made a sincere attempt to express our gratitude to all those who have contributed to the successful completion of this project through this project report.

As we represent this report on “Attendance System Using QR Code ” we are aware of humanity and grateful to all the individuals who have so kindly offered us their time, skill, knowledge, advice, and facilities or guidance.

We are extremely grateful to Prof. Parul Singh, Assistant Professor DGI-GN for giving his valuable input and opportunity to develop this project and for making all the resources available to us to succeed this project.

We would also like to thank Prof. Bipin Pandey, H.O.D. CSE/IT DGI-GN for his idea, advice, knowledge, and all his support throughout this project.

We also take this opportunity to express my deepest gratitude to our Project Guide and our faculty, Prof. Parul Singh, for his constant support, guidance, and encouragement and thus being a constant source of inspiration for us.

Last but not least we would like to thank my parents, all the employees and all my friends who have always been strong support during the entire course of my project and without their co-operation the completion of this project would not have been possible.

Riddhi Gupta

(1902300110032)

## **TABLE OF CONTENTS**

	<b>Page No.</b>
Certificate	i
Abstract	ii
Acknowledgment	iii
Chapter 1 : Introduction	1
Chapter 2: Literature Survey	2 - 7
2.1 Android -A Mobile Operating system	2
2.2 Quick Response (QR) Code	3
2.3 Firebase Overview & Usage	4
2.4 Application of Firebase	5
Chapter 3: Concept	8
Chapter 4 : Procedure	9
Chapter 5 : Data Flow Diagram	10 - 11
Chapter 6 : Modelling	12 - 14
Chapter 7 : Hardware & Software Requirements	15 - 20
7.1 Cloud Used	16
7.2 JAVA	17
7.3 Kotlin	18
7.4 API (Application Programming Interface)	19
7.5 User Interface (UI / UX)	20

	<b>Page No.</b>
Chapter 8 : Android Studio	21 - 22
Chapter 9 : Implementation	23 - 28
9.1 Android Studio code Overview	23
9.2 Firebase Overview	24
9.3 Authentication in Firebase	25
9.4 Realtime Database Rules	25
9.5 Database	26
9.6 Crashlytics	27
9.7 Analytics of firebase	28
Chapter 10 : Admin GUI	29 - 34
10.1 Application View	29
10.2 Faculty SignIn and Options	30
10.3 Faculty Generate QR code	31
10.4 Attendance Report	32
10.5 Student Login View	33
10.6 Scan QR	34
Chapter 11 : Result	35
Chapter 12 : Conclusion & Future Scope	36
Chapter 13 : Reference	37

## **LIST OF FIGURES**

S.No.	Name of Figure	Page No.
Figure 2.1	Real Time Access To The Clients	4
Figure 2.2	Main & Wall Activity Of App	6
Figure 5.1	Data Flow Diagram	11
Figure 6.1	Use Case Diagram	13
Figure 6.2	Flow Chart	14

## **LIST OF TABLES**

S.No.	Name of Table	Page No.
Table-1	Comparison With Existing System	3
Table-2	Comparison Between Firebase and SQL	5
Table-3	Comparison Between Firebase and MS SQL Server	5

# **CHAPTER 1**

## **INTRODUCTION**

An attendance system using QR code is an innovative approach to tracking the attendance of students or employees. The system works by generating a unique QR code for each individual that is linked to their identity, and scanning the code using a smartphone or tablet to record their attendance. This eliminates the need for traditional paper-based attendance registers and provides a more efficient and accurate way of tracking attendance.

The implementation of an attendance system using QR code technology offers several advantages over traditional methods. Firstly, it is more convenient for both students/employees and teachers/managers, as they can quickly scan the QR code and update attendance records in real-time. Secondly, it reduces the chances of errors in attendance records, which can occur due to illegible handwriting or manual data entry errors. Finally, it allows for greater transparency in attendance tracking, as the system records the exact time and date of each scan, making it easier to identify patterns of absences or tardiness.

Overall, an attendance system using QR code technology is a modern and effective way to manage attendance records. It saves time, reduces errors, and provides a more transparent and reliable way of tracking attendance. This technology has been adopted by many educational institutions and workplaces around the world and is becoming increasingly popular due to its numerous benefits.

## **CHAPTER 2**

### **LITERATURE SURVEY**

A QR code-based attendance system is an innovative solution for schools, universities, and organisations to track attendance quickly and accurately. It eliminates the need for manual attendance taking, which can be time-consuming and prone to errors. In this literature survey, we will explore some of the existing research on QR code-based attendance systems and their implementation in different settings.

#### **2.1 Android -A Mobile Operating system**

A unique QR code is generated for each participant or attendee. This can be done using various QR code generation tools or libraries. The system generates a unique QR code for each student, which they can scan using the mobile app to mark their attendance. The attendance data is then uploaded to a centralised database, where it can be accessed and monitored by teachers and administrators. The proposed system was tested on a sample of students, and the results showed that it was efficient, reliable, and reduced the time taken to take attendance.

The authors highlight the advantages of the proposed system, including its ability to eliminate the need for manual attendance taking, reduce errors and manipulation, and provide real-time monitoring of attendance data. They also note that the system can be easily implemented in various educational settings, including schools, colleges, and universities. The paper provides a detailed description of the system architecture, including the mobile application, server-side software, and database design. The authors also discuss the security aspects of the system, including data encryption and access control mechanisms. [No. 2.1]

## 2.2 Quick Response (QR) Code based Attendance Marking System

	Cost (to administration)		Accuracy	Scaling	Time of operation	Security(against proxy)
	Devices	Operational				
Biometric	High	Low	Very high	Normal	Very high	Very strong
RFID	Very high	Moderate	Very high	Easy	High	Poor
Face recognition	Moderate	High	Moderate to low	Difficult	Low	Poor
Proposed System	Low	High	Very High	Easy	Very high	Very Strong

**Table 1 : Comparison with existing systems**

The paper "Quick Response (QR) Code based Attendance Marking System" proposes a system for marking attendance using QR codes. The system uses a mobile application that generates a unique QR code for each student. The teacher scans the code using their mobile phone, and the attendance is automatically marked in the system. The system was tested on a sample of students, and the results showed that it was efficient, accurate, and reduced the time taken to take attendance. The proposed system is ideal for schools, universities, and organisations looking for automated and reliable attendance tracking system.

This paper also shows the mechanism of various different methods of taking attendance in an institute.

In this paper, it has been discussed about the analysis of QR codes as well as their applications. The capacity of these codes to store data is very high plus they are damage resistance which makes them overcome one of the key concerns of security. [No. 2.2]

In the past decade or so, the application of QR codes in public domains like supermarkets and in educational purposes like book scanning or stationary scanning has been increased rapidly and it will continue to thrive in more fields as the awareness will increase. The QR code technique is getting popular day by day and at the same time it is becoming increasingly secure as the technology is enhancing.

## 2.3 FIREBASE - OVERVIEW AND USAGE

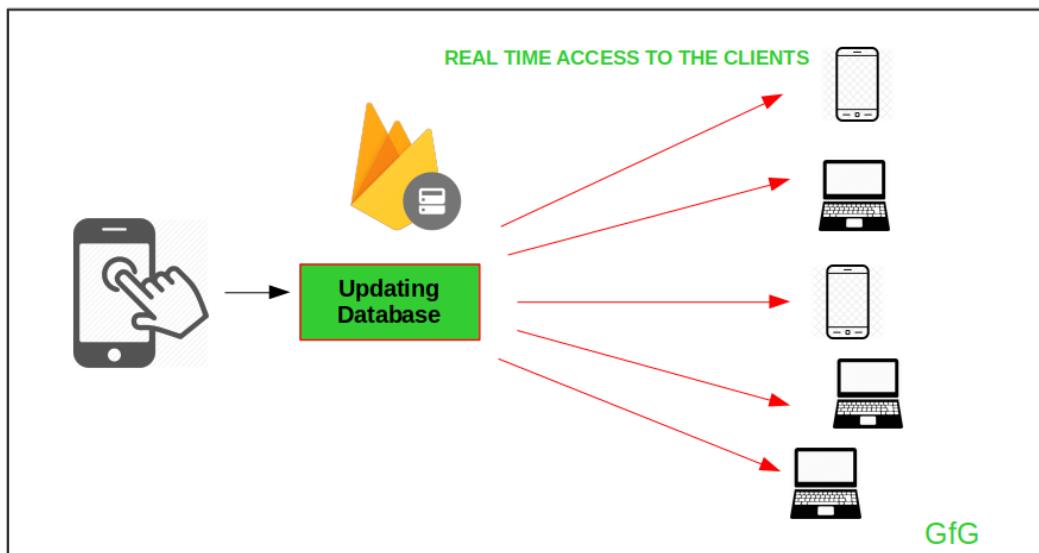


figure 2.1: Real Time Access To The Clients

The paper "Firebase - Overview and Usage" Firestore is a flexible, scalable, and fully managed NoSQL document database provided . The paper discusses the features and capabilities of Firebase, including real-time database, hosting, authentication, and storage. The authors highlight the advantages of using Firebase in application development, including its ease of use, scalability, and flexibility. They also provide a detailed description of the various services offered by Firebase, along with examples of how they can be used in real-world applications.[No. 2.3]

The paper also discusses the integration of Firebase with other technologies, such as AngularJS and ReactJS. The authors provide step-by-step instructions for setting up a Firebase project, including creating a new project, configuring authentication, and deploying the application.

Basis of comparison	Firebase	SQL(RDBMS)
Data Storage	Stored as JSON Tree	Stored in a Relational Model as Rows and Columns (Tables)
Schema flexibility	Dynamic Schema, data can be added, updated or deleted anytime	Fixed schema. Altering will result in going offline temporarily
Specialty	Data which has no definite type or Structure	Data whose type is known in advance
Technique	Synchronize data	Fire Query

**Table 2 :Comparison between Firebase and SQL**

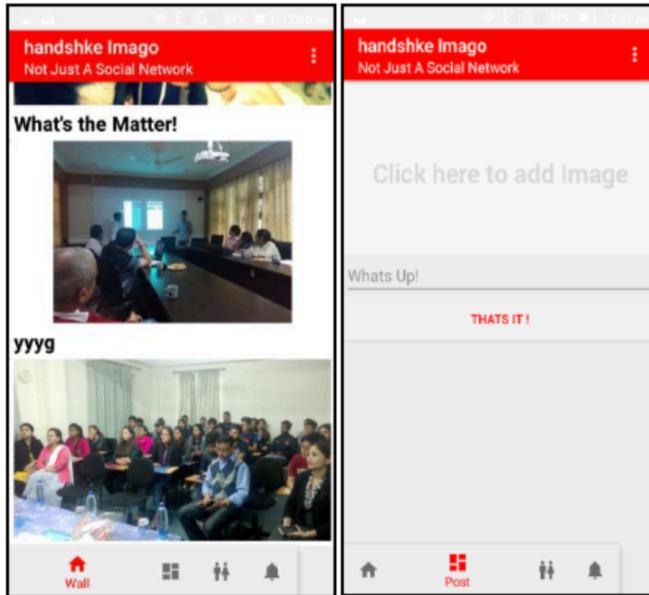
Overall, the paper provides a comprehensive overview of Firebase and its usage in mobile and web application development. It is a useful resource for developers who are new to Firebase and want to learn more about its features and capabilities. The paper also demonstrates the potential of Firebase to improve application development and delivery, and provides insights into best practices for using Firebase .

## 2.4 Application of Firebase in Android App Development

Basis of Comparison	Firebase	MS SQL Server
Data Model	Stored as JSON Tree	Table
DB Engine Ranking	Score-2.79 Rank-79 Overall, 13 Key-Value Pairs.	Score-94 Rank-3 Overall, 3 RDBMS
Website	<a href="https://firebase.google.com">Firebase.google.com</a>	<a href="https://www.microsoft.com/en-us/sql-server">Microsoft.com./en-us/SQL-server</a>
Developer	Google	Microsoft

**Table 3 : Comparison between Firebase and MS SQL Server**

1. Firebase is a platform that provides a range of services and tools for mobile and web application development.
2. It offers real-time database, authentication, cloud messaging, and other services that can be integrated into Android apps.



**figure 2.2 :**

**(a)**

**(b)**

**a) Main activity of app which allows user to upload images along with caption.**

**b) Wall activity of app where users can view post of other users.**

3. Firebase provides a cloud-based backend that can be used to store data and perform server-side operations.
4. Firebase offers a wide range of libraries and APIs that can be used to develop Android apps.
5. Firebase has a comprehensive documentation that makes it easy for developers to get started.
6. Firebase offers analytics and crash reporting features that help developers understand how their app is being used and identify and fix issues.

7. Firebase's authentication service provides secure authentication options that can be integrated into Android apps.
8. Firebase's real-time database offers a powerful solution for storing and synchronising data in real-time across devices.
9. Firebase Cloud Messaging (FCM) enables developers to send push notifications to their app users.
10. Firebase Remote Config allows developers to manage app configuration settings remotely.
11. Firebase Storage provides a scalable and secure way to store user-generated content, such as images and videos.
12. Firebase offers easy integration with Android Studio, making it easy for developers to incorporate Firebase services into their apps.
13. The paper provides a detailed explanation of how to integrate Firebase into an Android app, including step-by-step instructions and code snippets.
14. The paper also provides a case study of an Android app that utilizes Firebase's services and tools.
15. Overall, Firebase offers a powerful and easy-to-use platform for Android app development, and can help developers build better apps more efficiently. [No. 2.4]

## **CHAPTER 3**

### **CONCEPT**

The concept of an attendance system using QR codes involves generating unique QR codes for each user and then using a scanner to read the codes to mark attendance.

The system can be divided into two parts: admin and user.

1. The admin side of the system is responsible for creating and managing user accounts. The admin would generate a unique QR code for each user and assign it to their account. The admin can also view attendance reports for each user and generate reports for a specific time period.
2. On the user side, each user would have their own unique QR code that they would present to a scanner to mark their attendance. The scanner would read the QR code and mark the user as present. The user can also view their own attendance report.

### **3. Technologies to be used**

This project will be a web application developed by using Android Studio , having Java and Kotlin as frontend and Firebase as backend.

- Database Design
- Form Design
- Coding
- Testing
- Reporting Tool

This type of system is useful for tracking attendance in various settings, such as schools, businesses, and events. It eliminates the need for manual attendance tracking and provides a more efficient and accurate way of keeping track of attendance.

Overall, an attendance system using QR codes is a useful and innovative solution for attendance tracking. It offers a more efficient and accurate way of keeping track of attendance while also providing useful data for analysis and reporting.

## **CHAPTER 4**

### **PROCEDURE**

Here is a step-by-step procedure for building an attendance system using QR code:

1. Define the requirements: Determine the specific requirements of the attendance system, such as the number of students, the frequency of attendance, the types of data to collect, and the reporting requirements.
2. Create a database: Create a database to store the data. The database should include tables for students, courses, and attendance records.
3. Generate QR codes: Create a unique QR code for each student that includes their ID number and other relevant information.
4. Develop a mobile application: Develop a mobile application that can read the QR codes and connect to the database to record attendance. The application should be compatible with Android and iOS devices.
5. Connect the application to the database: Use an API to connect the mobile application to the database. This will allow the application to access the information stored in the database and update it in real-time.
6. Test the system: Test the system to ensure that it works correctly. Verify that the mobile application can read the QR codes and update the database accurately.
7. Deploy the system: Deploy the system to the relevant classrooms and provide instructions to the students and teachers on how to use it.
8. Monitor the system: Monitor the system to ensure that it continues to work correctly. Make any necessary adjustments or updates as needed.
9. Generate reports: Use the data collected by the system to generate reports on attendance, course progress, and other relevant information. These reports can be used to track student performance, identify areas for improvement, and provide feedback to teachers and students.

## **CHAPTER 5**

### **DATA FLOW DIAGRAM (DFD)**

A data flow diagram (DFD) is a graphical representation of a system that shows the flow of data and processes involved. Here's an example of a DFD for an attendance system using QR code:

**Level 0 DFD:**

At the highest level of the DFD, we have three main components: the students, the mobile application, and the database.

1. Students: This component represents the students who will use the system. They will have unique QR codes that can be scanned to record their attendance.
2. Mobile application: The mobile application is the tool used to scan the QR codes and record attendance. The application will communicate with the database to update the attendance records.
3. Database: The database stores all the data related to attendance, including student information, course information, and attendance records.

**Level 1 DFD:**

Next, we can break down each of these components into more detail, as follows:

1. Students:
  - a. Provide QR code: The student provides their unique QR code to the mobile application.
2. Mobile application:
  - a. Scan QR code: The mobile application scans the QR code provided by the student.
  - b. Retrieve student data: The mobile application retrieves the student data from the database using the QR code.

c. Record attendance: The mobile application updates the attendance records in the database.

### 3. Database:

- a. Store student data: The database stores all student data, including their name, ID number, and course information.
- b. Store attendance records: The database stores attendance records for each student, including the course code, date, and time.
- c. Generate reports: The database generates reports on attendance, course progress, and other relevant information using the data collected.

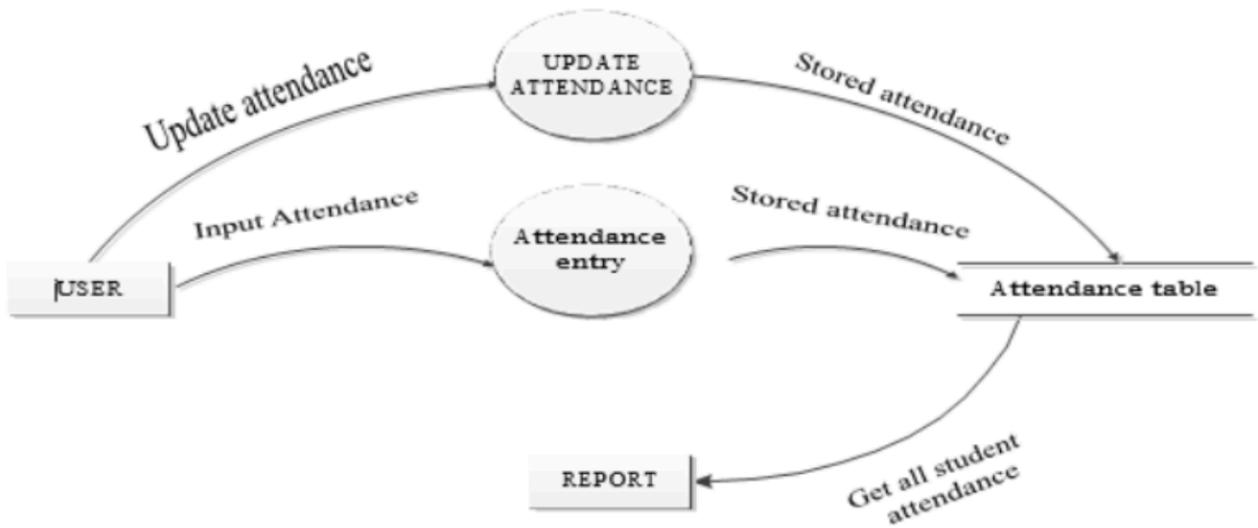


Figure 5.1 : Data Flow Diagram

## **CHAPTER 6**

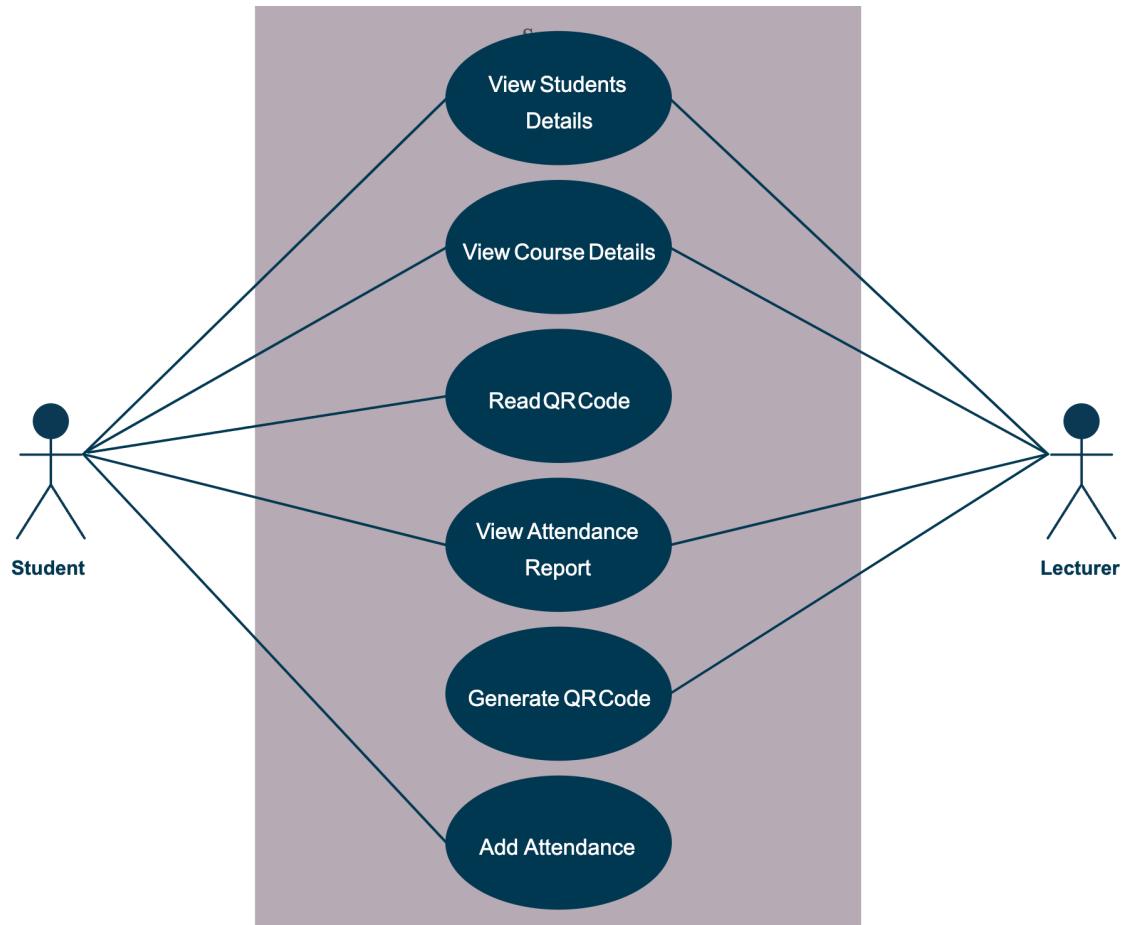
### **MODELLING**

High-level modelling of an attendance system using QR code:

1. Student data: The system should store a database of student data that includes unique identification numbers, student names, and other relevant information.
2. QR code generation: The system should generate a unique QR code for each student that includes their identification number and other relevant information.
3. Mobile application: The system should have a mobile application that can read QR codes and connect to the database to record attendance. The mobile application should be compatible with both Android and iOS devices.
4. Attendance record: The system should store an attendance record for each student in the database. Each attendance record should include the student's identification number, the date and time of attendance, and other relevant information.
5. API integration: The mobile application should integrate with the system's database using an API. This integration will allow the application to access the information stored in the database and update it in real-time.
6. QR code scanning: Students should scan their QR code using the mobile application at the beginning of each class. The mobile application should record the time and date of the attendance and update the database.
7. Reporting: The system should generate reports on attendance, course progress, and other relevant information. These reports can be used to track student performance, identify areas for improvement, and provide feedback to teachers and students.

Overall, the modelling of an attendance system using QR code should include the generation of unique QR codes for each student, the development of a mobile application that can read QR codes and connect to the database, the recording of attendance in the database, and the generation of reports. By implementing this system, schools and educational institutions can streamline their attendance-taking process, reduce the risk of errors, and improve accuracy in attendance records.

The various factors of the system along with their functionality are described in this use case diagram :



**Figure 6.1 : Use Case Diagram**

The flow chart of the problem is shown in the figure below :

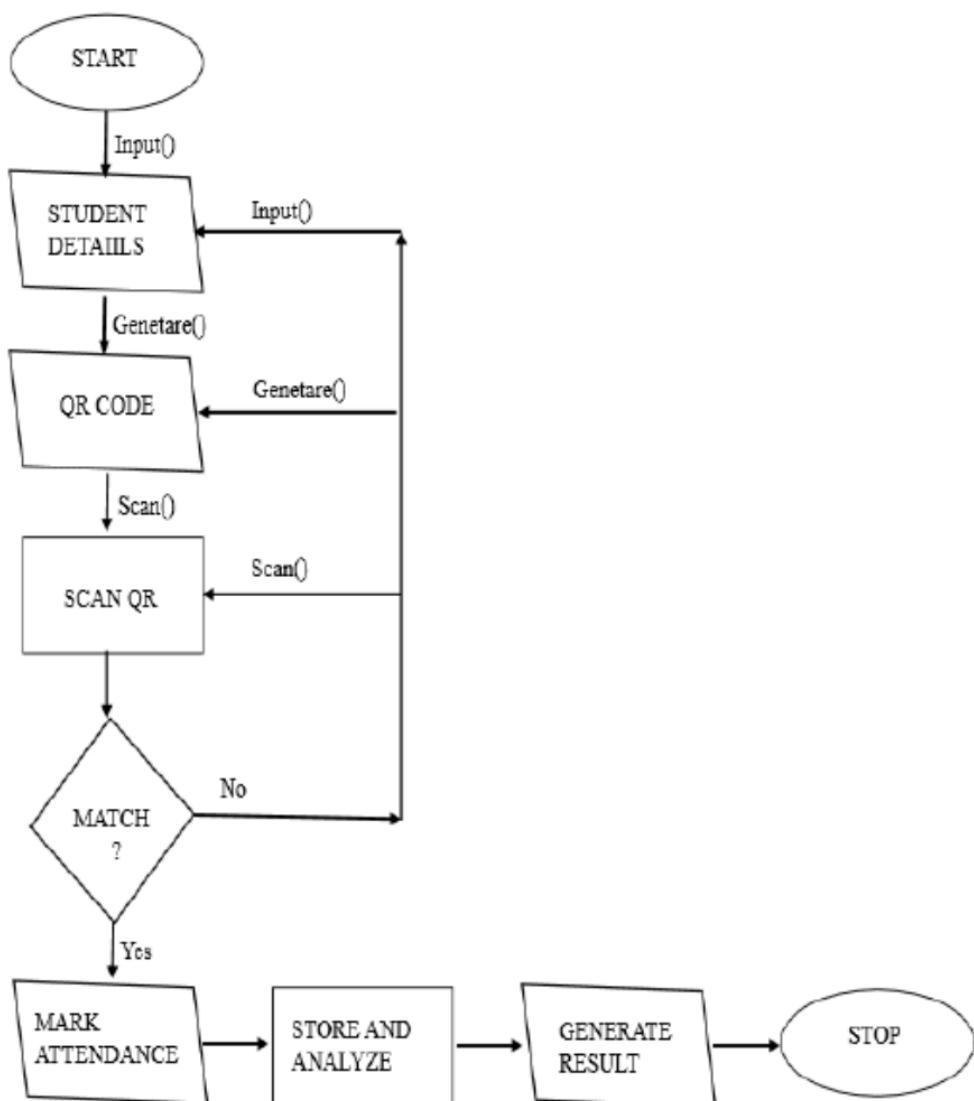


Figure 6.2 : Flow Chart of the application system

## CHAPTER 7

### **HARDWARE AND SOFTWARE REQUIREMENTS**

The software requirement specification can produce at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by established a complete information description, a detailed functional description, a representation of system behaviour, and indication of performance and design constraint, appropriate validate criteria, and other information pertinent to requirements.

Software Requirements:

- Operating system : Mac OS X 10.8.5.
- Coding Language : Kotlin , Java
- Front-End : Android Studio
- Data Base : Firebase

Hardware Requirements:

- System : MacBook Air
- HardDisk : 256GB
- Ram : 8GB

## 7.1 Cloud Used



Google Firebase is a Google-backed application development software that enables developers to develop iOS, Android and Web apps. Firebase provides tools for tracking analytics, reporting and fixing app crashes, creating marketing and product experiment.

Firebase offers a number of services, including:

- Analytics – Google Analytics for Firebase offers free, unlimited reporting on as many as 500 separate events. Analytics presents data about user behavior in iOS and Android apps, enabling better decision-making about improving performance and app marketing.
- Authentication – Firebase Authentication makes it easy for developers to build secure authentication systems and enhances the sign-in and onboarding experience for users. This feature offers a complete identity solution, supporting email and password accounts, phone auth, as well as Google, Facebook, GitHub, Twitter login and more.
- Cloud messaging – Firebase Cloud Messaging (FCM) is a cross-platform messaging tool that lets companies reliably receive and deliver messages on iOS, Android and the web at no cost.
- Realtime database – the Firebase Realtime Database is a cloud-hosted NoSQL database that enables data to be stored and synced between users in real time. The data is synced across all clients in real time and is still available when an app goes offline.

## 7.2 Java



Java is a popular programming language that is widely used in Android development. It provides a platform-independent environment for building robust and scalable mobile applications. Java offers a rich set of features such as object-oriented programming, automatic garbage collection, and support for multi-threading, making it an ideal choice for developing high-performance Android applications. Additionally, Java provides a large and active community, extensive documentation, and a vast collection of libraries and frameworks that make Android app development more accessible and efficient. Overall, Java's versatility and popularity have made it a widely adopted language for Android development, making it a critical component of the Android platform.

### 7.3 Kotlin



Kotlin is a modern, open-source, statically-typed programming language developed by JetBrains. It is designed to be more concise, expressive, and safe than Java, the traditional language for Android app development. Kotlin is fully interoperable with Java, which means developers can easily mix Kotlin and Java in their Android projects. Kotlin has become increasingly popular for Android development due to its many benefits, including:

1. Concise syntax: Kotlin has a concise, expressive syntax that makes code easier to read and write, reducing the amount of boilerplate code required.
2. Null safety: Kotlin has a built-in null safety feature that helps prevent null pointer exceptions, a common issue in Java.
3. Interoperability: Kotlin is fully interoperable with Java, which means developers can easily mix Kotlin and Java in their Android projects.
4. Coroutines: Kotlin supports coroutines, a powerful tool for asynchronous programming that makes it easier to write responsive, scalable Android apps.
5. Android Studio support: Kotlin is fully supported by Android Studio, the official IDE for Android app development.

## 7.4 API (Application Programming Interface)



An API, or Application Programming Interface, is a set of protocols, routines, and tools for building software applications. It allows different software applications to communicate and interact with each other. APIs define the way that different software components should interact with each other, allowing developers to build software applications that are modular, scalable, and flexible.

APIs can be used to access data from external sources, integrate with other software applications, or automate tasks. They provide a standard way for developers to access and manipulate data, regardless of the underlying software or hardware.

APIs can be created for internal or external use, and can be public or private. Public APIs are typically available to developers outside of the organisation, while private APIs are used within an organisation to connect different software components. APIs can be accessed using a variety of programming languages and protocols, including REST, SOAP, and GraphQL.

## 7.5 User Interface ( UI / UX)



UI/UX (User Interface/User Experience) design plays a crucial role in the success of any Android application. A well-designed UI/UX can enhance the user experience, improve usability, and increase engagement. Here are some key considerations for UI/UX design in Android development:

1. Material Design: Google's Material Design guidelines provide a set of principles, patterns, and components for designing intuitive, responsive, and visually appealing Android applications.
2. Consistency: Consistency in design across the entire application can enhance usability and make the app more intuitive. Consistent use of colors, typography, and icons can create a visual language that users can easily recognize and understand.
3. Navigation: Navigation within the app should be simple and intuitive. A clear and consistent navigation menu, breadcrumbs, and in-app search can help users find what they are looking for quickly.
4. Responsive Design: Android devices come in various screen sizes, resolutions, and aspect ratios. Responsive design can ensure that the application looks and functions well across all devices.
5. Gestures: Gestures can provide a more natural and intuitive way of interacting with the app. Common gestures include swiping, pinching, tapping, and dragging.

## CHAPTER 8

### Android Studio



Android Studio provides a unified environment where you can build apps for Android phones, tablets, Android Wear, Android TV, and Android Auto. Structured code modules allow you to divide your project into units of functionality that you can independently build, test, and debug.

### **Advantages of QR Code Based Smart Attendance**

#### **System:**

- Provide better security.
- Maintenance of the system is easy and cost effective.
- Generate the result quickly.
- Provide accurate and efficient data.
- User friendly.

#### **Problem statement :**

- Development of a SMART QR CODE BASED ATTENDANCE SYSTEM.
- Integrating Android device with QR code and Google Firebase to store attendance results.

- Analysing the attendance on weekly basis.

### **Characteristic of proposed system :**

- User Friendly
- Reports are easily generated
- Very less paper work
- One spot solution for attendance calculation

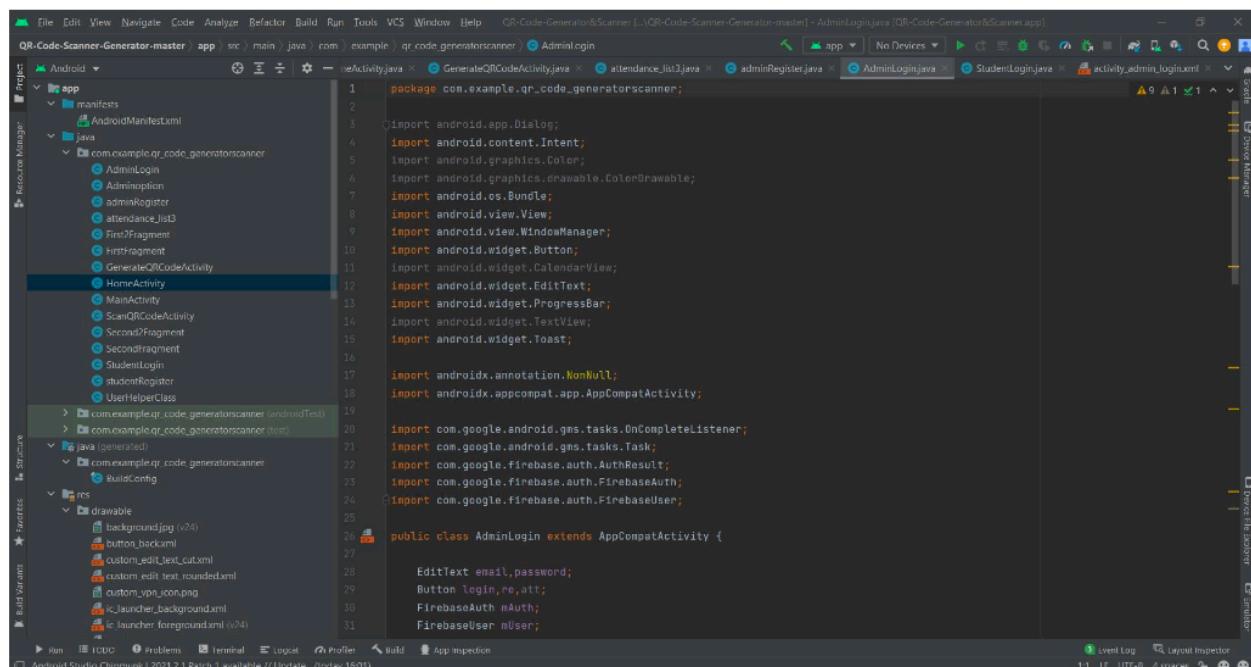
### **Methodology :**

- This is an android application which contain two interface for admin and other one for student.
- Here the student and admin has to register by providing initial credential.
- Here the admin can generate the QR code where each QR code contain data regarding date and subject code.
- And in the other interface the student can scan the QR Code and the data will be stored into the firebase cloud which will form a dedicated connection between application and cloud server via internet.
- The admin is authorised to view the records of all the Students. This system allows to keep up to date record of the Student date wise.

# Chapter 9

## Step by Step Implementation of Firebase

### 9.1 Android Studio code Overview



The screenshot shows the Android Studio interface with the project 'QR-Code-Scanner-Generator-master' open. The 'app' module is selected. The code editor displays the 'AdminLogin.java' file, which contains the implementation for an activity. The code imports various Android and Firebase libraries and defines a class 'AdminLogin' extending 'AppCompatActivity'. The interface includes an email input field, a password input field, and a login button. The code also handles Firebase authentication.

```
1 package com.example.qr_code_generatorscanner;
2
3 import android.app.Dialog;
4 import android.content.Intent;
5 import android.graphics.Color;
6 import android.graphics.drawable.ColorDrawable;
7 import android.os.Bundle;
8 import android.view.View;
9 import android.view.WindowManager;
10 import android.widget.Button;
11 import android.widget.CalendarView;
12 import android.widget.EditText;
13 import android.widget.ProgressBar;
14 import android.widget.TextView;
15 import android.widget.Toast;
16
17 import androidx.annotation.NonNull;
18 import androidx.appcompat.app.AppCompatActivity;
19
20 import com.google.android.gms.tasks.OnCompleteListener;
21 import com.google.android.gms.tasks.Task;
22 import com.google.firebase.auth.AuthResult;
23 import com.google.firebase.auth.FirebaseAuth;
24 import com.google.firebase.auth.FirebaseUser;
25
26 public class AdminLogin extends AppCompatActivity {
27
28     EditText email,password;
29     Button login,re,att;
30     FirebaseAuth mAuth;
31     FirebaseUser mUser;
```

## 9.2 Firebase Overview

The screenshot shows the 'Project Overview' section of the Firebase console. On the left sidebar, under 'Build', 'Authentication' is selected. The main area displays project details:

- Project name:** QR-Code-Generator
- Project ID:** qr-code-generator-e5661
- Project number:** 575869581255
- Default GCP resource location:** nam5 (us-central)
- Parent org/folder in GCP:** vit.ac.in
- Web API Key:** AlzaSyDmgIMlwccolyI-7x83vhUhs1HSTvL3V2o

**Environment:** Unspecified

**Public settings:** Public-facing name: project-575869581255; Support email: Not configured

The screenshot shows the 'Project settings' page. Under 'Users and permissions', it lists members:

Member	Roles
Hitesh Lalwani 21MCA0113 hitesh.lalwani2021@vitstudent.ac.in	Owner
tushar.gupta2021@vitstudent.ac.in	Owner

6 service accounts also have access to this project.

[Advanced permission settings](#)

## 9.3 Realtime Database Rules

The screenshot shows the Firebase Realtime Database Rules editor. On the left, there's a sidebar with project settings like Authentication, App Check, Firestore Database, and Realtime Database. The Realtime Database section is selected. The main area has tabs for Data, Rules, Backups, and Usage. The Rules tab is active, showing a code editor with the following rules:

```
1 + {
2 +   "rules": {
3 +     ".read": true,
4 +     ".write": true
5 +   }
}
```

Below the code editor are buttons for 'Edit rules' and 'Monitor rules'. A 'Rules playground' button is also visible.

## 9.4 Database

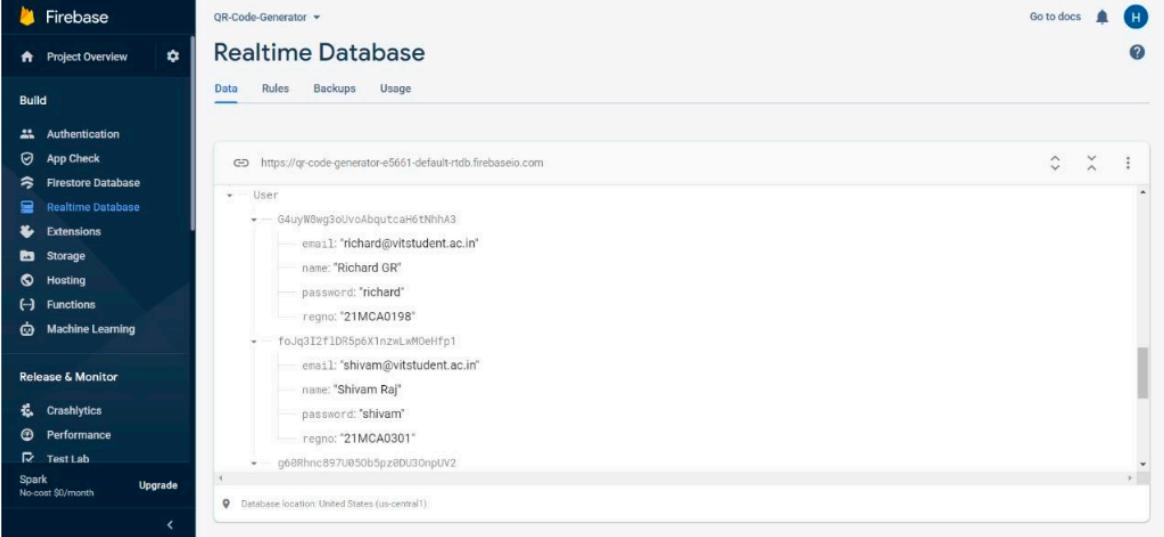
### Admin

The screenshot shows the Firebase Realtime Database Data tab. The sidebar on the left is identical to the previous screenshot. The main area displays a hierarchical view of database data under the 'Admin' node. Two users are listed: '4uHloKFkF6hQdE7Jtv2p8Dxy3Z2' and '095DA6AYVCVJBHFFBlyHufsnkcZ1'. Each user node contains the following fields:

- email: "tushar@vitfaculty.ac.in"
- name: "Tushar Gupta"
- password: "tg192001"
- regno: "ITA6009"

A message at the bottom indicates the database location is United States (us-central1).

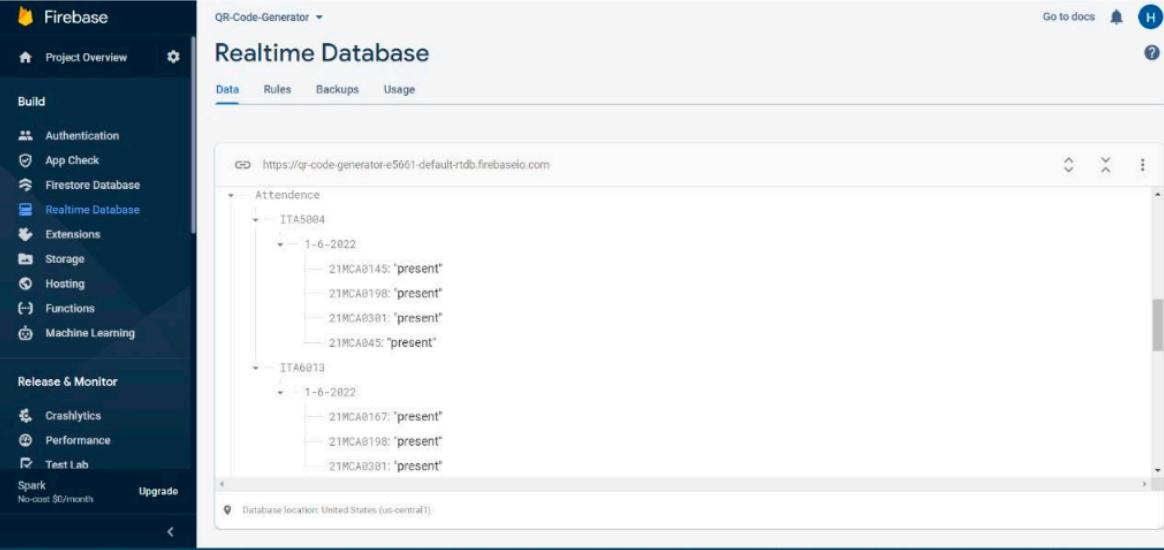
## User



The screenshot shows the Firebase Realtime Database interface. On the left, the navigation sidebar includes sections for Authentication, App Check, Firestore Database, Realtime Database (selected), Extensions, Storage, Hosting, Functions, and Machine Learning. Under Release & Monitor, it lists Crashlytics, Performance, and Test Lab. The main area displays the Realtime Database with a URL: https://qr-code-generator-e5661-default-rtdb.firebaseio.com. The database structure under 'User' contains three child nodes: G4uyN0lg3oJvoAbqutcaH6TNhhA3, foJq3I2f1DR5p6X1nzLw#0eHfp1, and g69Rhc897U050b5pz8DU30npUV2. Each node has fields for email, name, password, and regno.

```
graph TD; User --> G4uyN0lg3oJvoAbqutcaH6TNhhA3; User --> foJq3I2f1DR5p6X1nzLw#0eHfp1; User --> g69Rhc897U050b5pz8DU30npUV2; G4uyN0lg3oJvoAbqutcaH6TNhhA3 --> email1["email: 'richard@vitstudent.ac.in'"]; G4uyN0lg3oJvoAbqutcaH6TNhhA3 --> name1["name: 'Richard GR'"]; G4uyN0lg3oJvoAbqutcaH6TNhhA3 --> password1["password: 'richard'"]; G4uyN0lg3oJvoAbqutcaH6TNhhA3 --> regno1["regno: '21MCA0198'"]; foJq3I2f1DR5p6X1nzLw#0eHfp1 --> email2["email: 'shivam@vitstudent.ac.in'"]; foJq3I2f1DR5p6X1nzLw#0eHfp1 --> name2["name: 'Shivam Raj'"]; foJq3I2f1DR5p6X1nzLw#0eHfp1 --> password2["password: 'shivam'"]; foJq3I2f1DR5p6X1nzLw#0eHfp1 --> regno2["regno: '21MCA0301'"]; g69Rhc897U050b5pz8DU30npUV2 --> email3["email: 'shivam@vitstudent.ac.in'"];
```

## Attendance



The screenshot shows the Firebase Realtime Database interface. The navigation sidebar is identical to the previous screenshot. The main area displays the Realtime Database with a URL: https://qr-code-generator-e5661-default-rtdb.firebaseio.com. The database structure under 'Attendance' contains two child nodes: ITA5004 and ITA6013. Each node has a child node for the date 1-6-2022, which in turn contains four student IDs (21MCA0145, 21MCA0198, 21MCA0301, 21MCA045) each associated with the value "present".

```
graph TD; Attendance --> ITA5004; Attendance --> ITA6013; ITA5004 --> 1_6_2022["1-6-2022"]; ITA6013 --> 1_6_2022; 1_6_2022 --> 21MCA0145["21MCA0145: 'present'"]; 1_6_2022 --> 21MCA0198["21MCA0198: 'present'"]; 1_6_2022 --> 21MCA0301["21MCA0301: 'present'"]; 1_6_2022 --> 21MCA045["21MCA045: 'present'"];
```

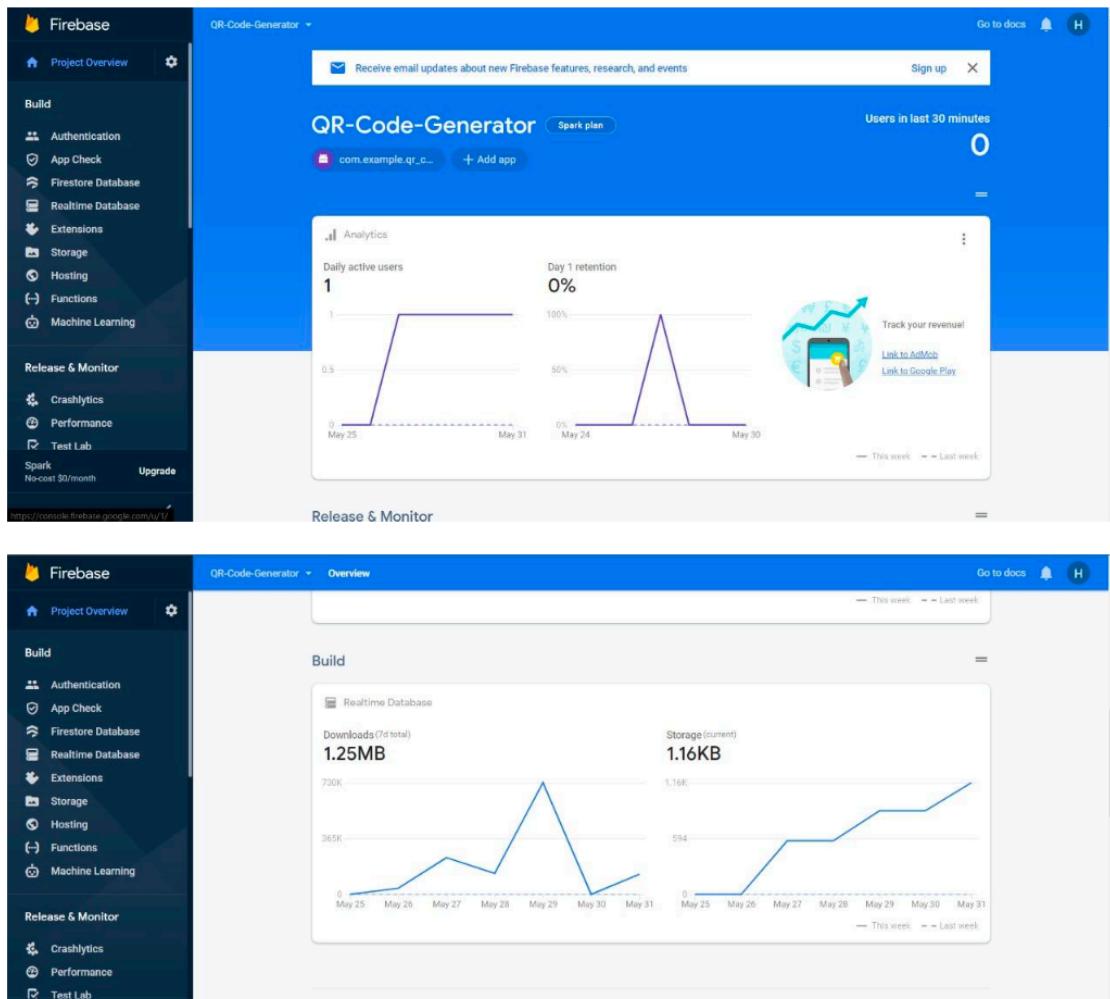
## 9.5 Crashlytics

The screenshot shows the Crashlytics dashboard for the project 'QR-Code-Generator'. The left sidebar includes sections for Build (Authentication, App Check, Firestore Database, Realtime Database, Extensions, Storage, Hosting, Functions, Machine Learning), Release & Monitor (Crashlytics, Performance, Test Lab), and general Firebase services (Project Overview, Upgrade). The main area displays 'Crash-free statistics' with a line chart showing a drop from 100% to 0% on May 29. It also shows 'Trends' for Crashes, Users, Non-fatales, and ANRs over the last 7 days. A section for BigQuery integration is present, and the bottom shows a list of open issues.

This screenshot shows a detailed view of the 'Issues' section in the Crashlytics dashboard. It lists seven entries, each with a checkbox, issue title, event type, versions, events, and users affected. Most issues are labeled as 'Fresh issue'. The titles include 'GenerateQRCodeActivity.java ~ line 75', 'GenerateQRCodeActivity.java ~ line 73', 'attendance\_List3.java ~ line 56', 'GenerateQRCodeActivity.java ~ line 79', 'GenerateQRCodeActivity.java ~ line 74', 'ScanQRCodeActivity.java ~ line 121', and 'attendance\_List3.java ~ line 58'.

	Event type	Versions	Events	Users
GenerateQRCodeActivity.java ~ line 75 com.example.qr_code_generatorscanner.GenerateQRCodeActivity\$2.onClick ↳ Fresh issue	Crash	1.0 ~ 1.0	11	1
GenerateQRCodeActivity.java ~ line 73 com.example.qr_code_generatorscanner.GenerateQRCodeActivity\$2.onClick ↳ Fresh issue	Crash	1.0 ~ 1.0	9	1
attendance_List3.java ~ line 56 com.example.qr_code_generatorscanner.attendance_list3\$1.onDataChange ↳ Fresh issue	Crash	1.0 ~ 1.0	8	1
GenerateQRCodeActivity.java ~ line 79 com.example.qr_code_generatorscanner.GenerateQRCodeActivity\$2.onClick ↳ Fresh issue	Crash	1.0 ~ 1.0	7	1
GenerateQRCodeActivity.java ~ line 74 com.example.qr_code_generatorscanner.GenerateQRCodeActivity\$2.onClick ↳ Fresh issue	Crash	1.0 ~ 1.0	5	1
ScanQRCodeActivity.java ~ line 121 com.example.qr_code_generatorscanner.ScanQRCodeActivity\$1.onCodeScanned ↳ Fresh issue	Crash	1.0 ~ 1.0	5	1
attendance_List3.java ~ line 58 com.example.qr_code_generatorscanner.attendance_list3\$1.onDataChange	Crash	1.0 ~ 1.0	5	1

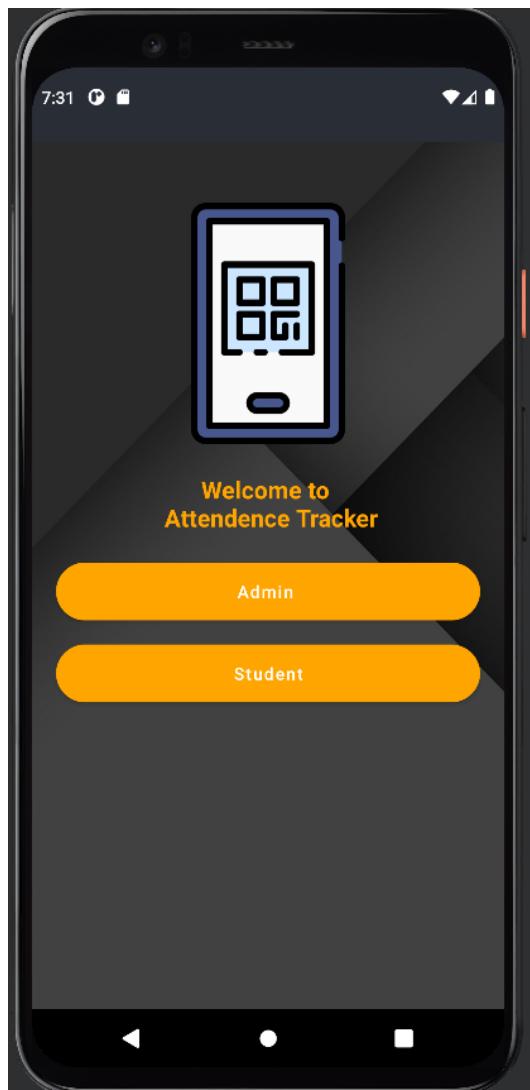
## 9.6 Analytics of firebase



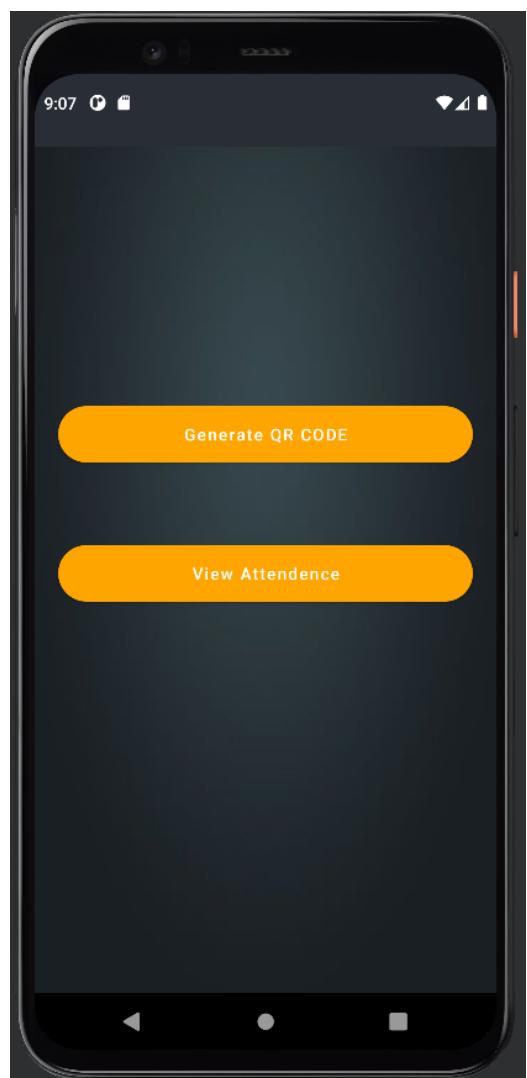
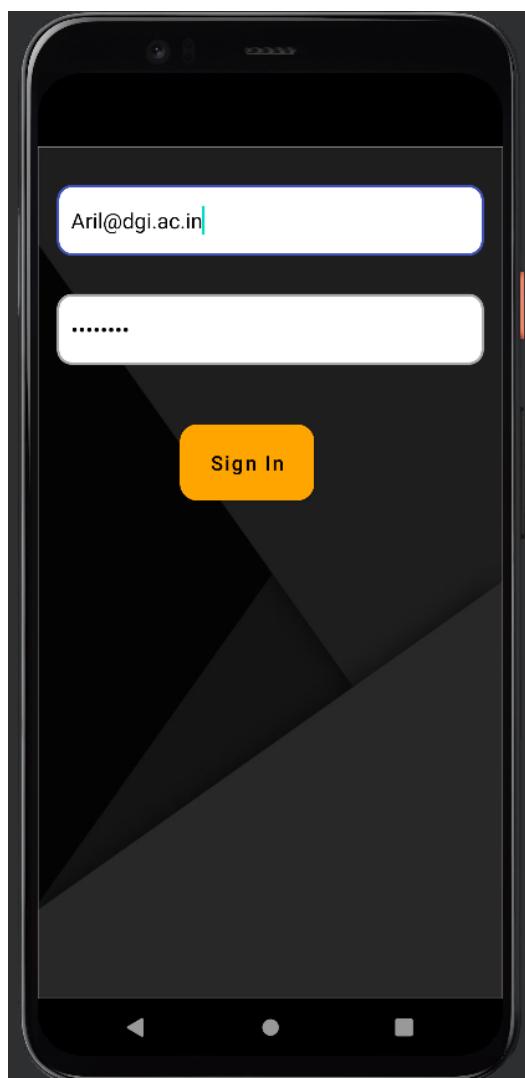
# CHAPTER 10

## ADMIN GUI

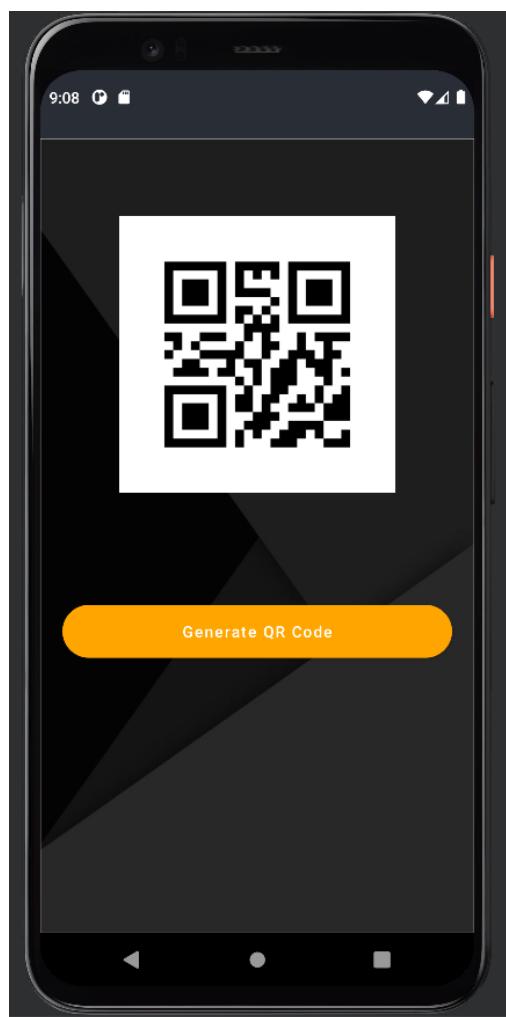
### 10.1 Application View



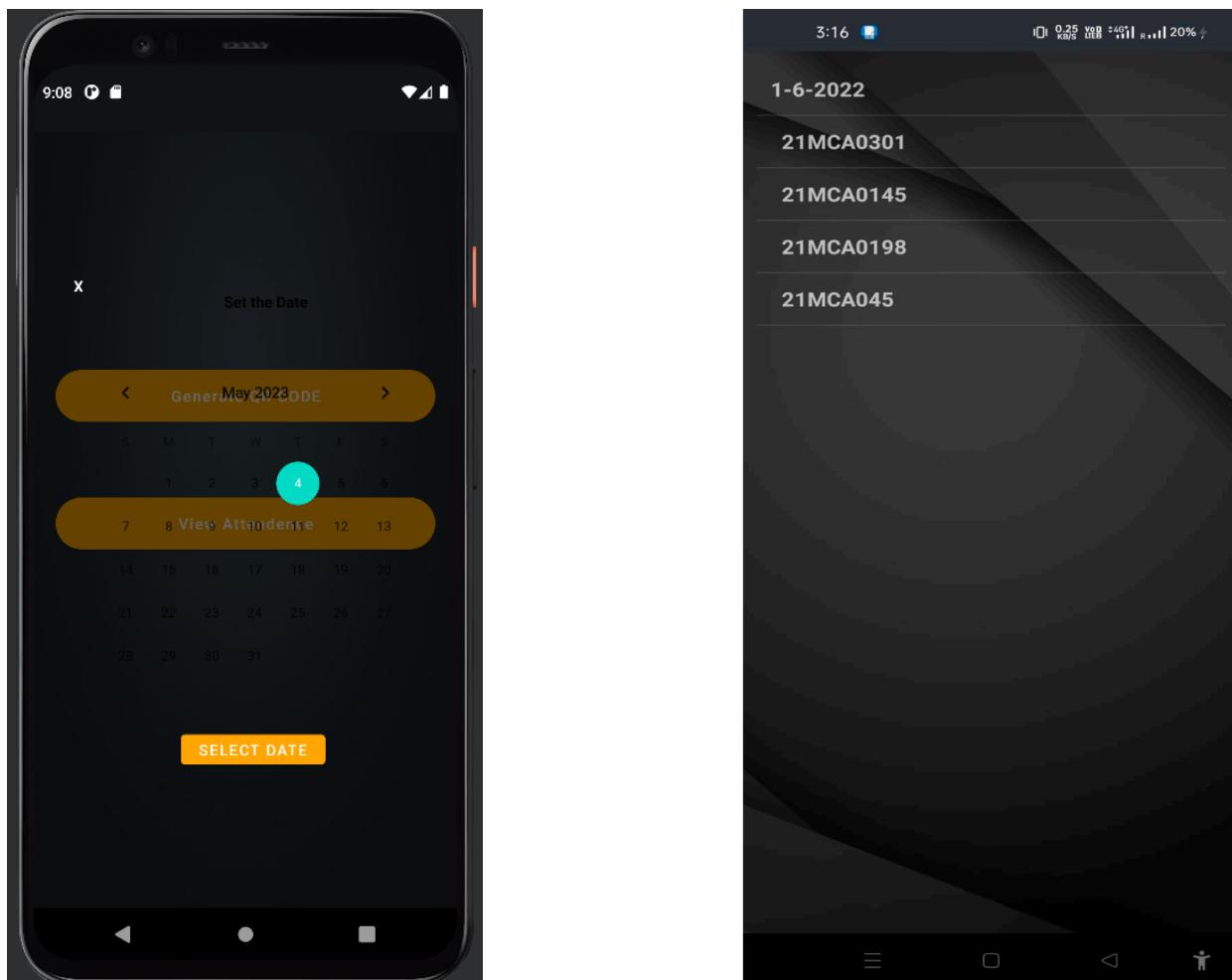
## 10.2 Faculty Sign in and Options



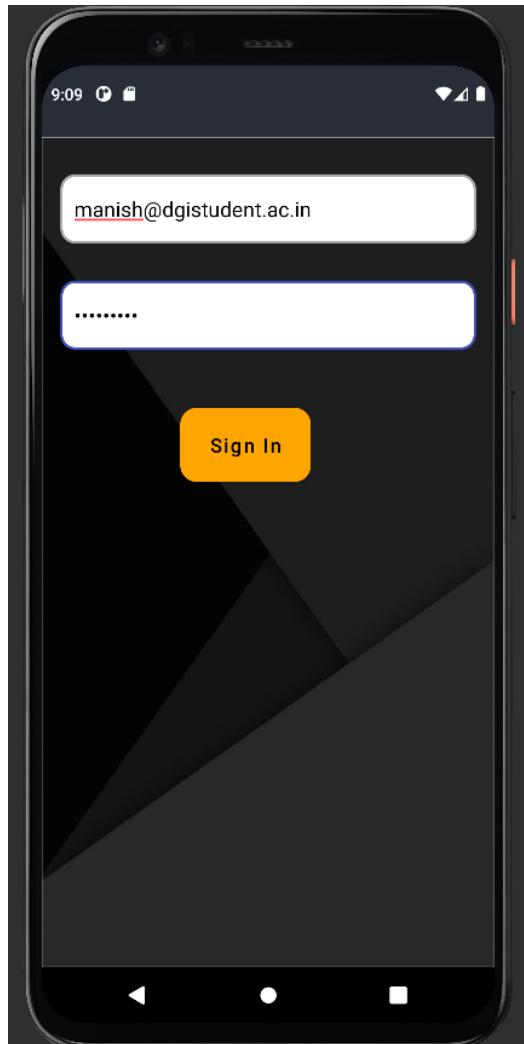
### 10.3 Faculty Generate QR Code



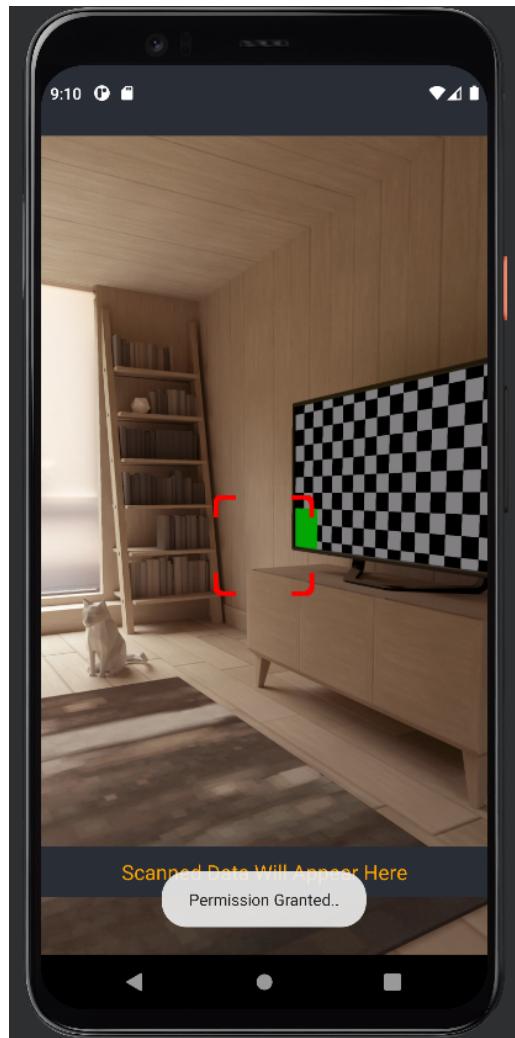
## 10.4 Attendance Report



## 10.5 Student Login View



## 10.6 Scanner QR



## **CHAPTER 11**

### **RESULT**

An attendance system using QR code is an innovative way to automate the process of taking attendance. The system uses QR codes, which are scanned by a mobile device with a camera and an internet connection. The system is designed to reduce the workload of teachers and staff, and to make the process of taking attendance more efficient.

Some of the benefits of implementing an attendance system using QR codes:

1. Time-saving: The system reduces the amount of time required to take attendance, as the process is automated.
2. Accuracy: The system eliminates the possibility of errors, as the QR codes are scanned by the mobile device and the data is automatically recorded.
3. Transparency: The system provides transparency in attendance tracking, as students can view their attendance record in real-time.
4. Cost-effective: The system is cost-effective as it does not require any additional hardware, apart from mobile devices with a camera and an internet connection.
5. Real-time tracking: The system provides real-time tracking of attendance, which allows teachers to take corrective measures if necessary.

Overall, an attendance system using QR codes is an efficient and cost-effective solution for schools and colleges to automate the attendance tracking process. It reduces the workload of teachers and staff, provides real-time tracking, and improves transparency and accuracy.

## **CHAPTER 12**

### **CONCLUSION AND FUTURE SCOPE**

QR code-based attendance systems are becoming increasingly popular in various industries, including education, healthcare, and corporate settings. With this technology, users can simply scan a QR code using their mobile devices to register their attendance, eliminating the need for manual processes and reducing errors.

The future scope of QR code-based attendance systems is quite promising. As more and more organisations embrace digital transformation, they are likely to adopt this technology to streamline their attendance tracking processes. This is especially true in the post-COVID era, where contactless solutions are becoming the norm.

QR code-based attendance systems also offer several benefits over traditional methods, such as:

1. Improved accuracy: QR code-based systems eliminate the risk of manual errors and make attendance tracking more accurate.
2. Time-saving: With QR code-based systems, attendance can be recorded in a matter of seconds, saving time and effort for both the user and the organisation.
3. Cost-effective: These systems are often more cost-effective than traditional methods, as they eliminate the need for paper-based processes and reduce administrative overhead.
4. Real-time reporting: QR code-based systems can provide real-time attendance reports, allowing organisations to monitor attendance and identify issues quickly.

In conclusion, QR code-based attendance systems have a bright future ahead. With their many benefits and increasing adoption, they are likely to become the go-to solution for attendance tracking in various industries.

## **CHAPTER 13**

### **REFERENCES**

[2.1] <https://www.irjet.net/archives/V8/i1/IRJET-V8I101.pdf>

[2.2] [https://www.ijcaonline.org/archives/volume177/number3\\_3/shah-2020-ijca-919816.pdf](https://www.ijcaonline.org/archives/volume177/number3_3/shah-2020-ijca-919816.pdf)

[2.3] <https://www.ijaiem.org/Volume8Issue7/IJAIEM-2019-07-02-3.pdf>

[2.4] [https://www.irjmets.com/uploadedfiles/paper/volume\\_3/issue\\_12\\_december\\_2021/17917/final/fin\\_irjmets1640499489.pdf](https://www.irjmets.com/uploadedfiles/paper/volume_3/issue_12_december_2021/17917/final/fin_irjmets1640499489.pdf)

[2.5] <https://www.ijcaonline.org/archives/volume179/number46/khawas-2018-ijca-917200.pdf>