

A
Project Report
on
DiG-Pass: Enhance and Secure Solution for
Gate-Pass using QR Code

Submitted for the Course of BE in Computer Engineering by

Mr. Omkar Arun Jadhav [B191204289]

Mr. Parag Rajendra Shirsat [B191204327]

Mr. Abhishek Ninad Soundankar [B191204332]

Mr. Om Sugandh Suryawanshi [B191204334]

Under the guidance of

Prof. M. P. Gangawane



Department of Computer Engineering

**Guru Gobind Singh College of Engineering and
Research Centre**

Nashik-422009

2023-24

**GURU GOBIND SINGH COLLEGE OF
ENGINEERING AND RESEARCH CENTRE**

Nashik-422009

2023-2024

Department of Computer Engineering



CERTIFICATE

This is to certify that the PROJECT REPORT entitled

**DiG-Pass: Enhance and Secure Solution for
Gate-Pass Using QR Code**

is submitted as fulfilment of the

Project Examination BE in Computer Engineering

BY

Mr. Omkar Arun Jadhav [B191204289]

Mr. Parag Rajendra Shirsat [B191204327]

Mr. Abhishek Ninad Soundankar [B191204332]

Mr. Om Sugandh Suryawanshi [B191204334]

Prof. M. P. Gangawane

Project Guide

Prof. P. K. Bachhav & P. C. Patil

Project Coordinator

Prof. S. G. Shukla
Head of the Department

Dr. N. G. Nikam
Principal

SAVITRIBAI PHULE PUNE UNIVERSITY



CERTIFICATE

This is to certify that,

Mr. Omkar Arun Jadhav [B191204289]

Mr. Parag Rajendra Shirsat [B191204327]

Mr. Abhishek Ninad Soundankar [B191204332]

Mr. Om Sugandh Suryawanshi [B191204334]

of BE in Computer Engineering was examined in the

Project Examination entitled

**DiG-Pass: Enhance and Secure Solution for
Gate-Pass Using QR Code**

on 30/05/2024

At

Department of Computer Engineering

**GURU GOBIND SINGH COLLEGE OF ENGINEERING
AND RESEARCH CENTRE**

Nashik-422009

2023-2024

Internal Examiner

External Examiner

Acknowledgement

It is a great pleasure to acknowledge those who extended their support and contributed time to this project work.

While the project is still in progress, I would like to thank my project guide **Mr. Manish. P. Gangawane**, for his valuable and skillful guidance, assessment, and suggestions from time to time improved the quality of work in all respects. I would like to take this opportunity to express my deep sense of gratitude towards him, for his invaluable contribution to the completion of this project.

I am also thankful to **Mr. Sandeep. G. Shukla**, Head of the Computer Engineering Department for his timely guidance, inspiration, and administrative support without which my work would not have been completed.

I am also thankful to all staff members of the Computer Engineering Department and the Librarian, Guru Gobind Singh College of Engineering and Research Centre, Nashik.

Also I would like to thank my colleagues and friends who helped me directly and indirectly to complete this project.

Mr. Omkar Arun Jadhav

Mr. Parag Rajendra Shirsat

Mr. Abhishek Ninad Soundankar

Mr. Om Sugandh Suryawanshi

Abstract

In an increasingly digital and security-conscious world, the "DiG-Pass: Enhance and Secure Solution for Gate-Pass using QR Code" emerges as a versatile solution for access control, visitor management, and event ticketing across various industries. This innovative mobile application leverages the ubiquity of Android devices and the efficiency of QR codes to streamline access procedures while enhancing security and convenience. This project harnesses the power of QR codes to generate digital gate passes that serve as secure credentials for individuals seeking entry into diverse environments. Whether managing visitors in corporate offices, regulating access to educational institutions, or facilitating seamless entry at events and tourist attractions, this application offers a flexible and scalable solution. Key features of this project include pass generation and real-time access validation. The application gives administrators the tools to create and distribute QR code passes effortlessly, ensuring that only authorized individuals gain exit. It also records entry and exit times, aiding in tracking, auditing, and compliance reporting.

Keywords:- *DiG-Pass, Secure solution, Gate-Pass, QR Code, mobile application, streamline, security, pass generation, exit times.*

Abbreviation

Sr No.	Abbriviation	Full Form
1	MVC	Model View Controller
2	CSV	Comma-Separated Values
3	QR	Quick Response code
4	UML	Unified Modeling Language
5	DFD	Data Flow Diagram
6	API	Application Programming Interface
7	SSL	Secure Socket Layer
8	TLS	Transport Layer Security
9	HTTPS	Hypertext Transfer Protocol Secure

List of Figures

3.1	DFD Level 0	17
3.2	DFD Level 1	18
3.3	DFD Level 2	19
3.4	Class Diagram	20
3.5	State Machine Diagram	21
4.1	System Architecture	23
4.2	Proces flow of DiG-Pass App	25
4.3	ER Diagram	27
4.4	Activity Diagram	28
4.5	Use Case Diagram	29
4.6	Communication Diagram	30
4.7	Sequence Diagram	31
4.8	Component Diagram	32
4.9	Deployment Diagram	33
6.1	Timeline Diagram	40
8.1	DiG-Pass Dashboard	67
8.2	Student Login Page	68
8.3	Student Login Page with Validation	69
8.4	Student Sign-Up Page	70
8.5	Student Sign-Up Page with Error Message	71
8.6	Student Sign-Up Page with Error Message	72
8.7	Student Sign-Up Page with Error Message	73
8.8	Student Sign-Up Page with Error Message	74
8.9	Verify User Email	75
8.10	Student Dashboard	76
8.11	Student Gate-Pass Request	77
8.12	Edit Profile Page	78

8.13	Gate-Pass History	79
8.14	Staff Login Page	80
8.15	Staff Sign-Up Page	81
8.16	Staff Dashboard Page	82
8.17	Staff Gate-Pass Request	83
8.18	Notification to Admin	84
8.19	Request Details	85
8.20	QR Code Generation	86
8.21	Admin Dashboard	87
8.22	Create Admin Page	88
8.23	Guard Dashboard	89
8.24	Scan QR Code	90
8.25	QR Code Details	91
9.1	Speed Visualization	93
10.1	Plagiarism Report	97

List of Tables

3.1	Implementation Plan	22
6.1	Modes of development	36
6.2	Coefficients related to development modes for intermediate model	36
6.3	Lists Of Tasks	39
6.4	Lists of Developers	40
8.1	Test Case for successful Login	45
8.2	Test case for Student Login with invalid Email	46
8.3	Test case for invalid Password	46
8.4	Test case for Student Login with empty fields	46
8.5	Test cases for Student Login with invalid credentials	47
8.6	Test cases for Student Login with empty Email	47
8.7	Test cases for Student Login with empty Password	47
8.8	Test cases for Student Sign-Up with valid data	48
8.9	Test cases for Invalid Email	48
8.10	Test cases for Required fields	48
8.11	Test cases for Password	49
8.12	Test cases for Duplicate email	49
8.13	Test cases for Updating Student Profile	49
8.14	Test cases for Empty fields	50
8.15	Test cases for Updating Profile Image	50
8.16	Test cases for Retrieving student data from Firebase	50
8.17	Test cases for Handling failed image upload to Firebase	51
8.18	Test cases for Retrieving student data from Firebase	51
8.19	Test cases for Submit the form without File	52
8.20	Test cases for Submit the form with file	52
8.21	Test cases for Submit the form without Leaving Time	52
8.22	Test cases for Vehicle Option	53
8.23	Test cases for Reason	53

8.24	Test cases Invalid file format	53
8.25	Test cases for Sending Notification	54
8.26	Test cases for Staff Login	54
8.27	Test cases for Empty email field	54
8.28	Test cases for Empty Password field	55
8.29	Test cases for Invalid Email	55
8.30	Test cases for Incorrect Credentials	55
8.31	Test cases for Staff Sign-Up with valid data	56
8.32	Test cases for Invalid Email	56
8.33	Test cases for Required fields	56
8.34	Test cases for Password	57
8.35	Test cases for Duplicate email	57
8.36	Test cases for Updating Staff Profile	57
8.37	Test cases for Empty fields	58
8.38	Test cases for Updating Profile Image	58
8.39	Test cases for Retrieving Staff data from Firebase	58
8.40	Test cases for Handling failed image upload to Firebase	59
8.41	Test cases for Retrieving Staff data from Firebase	59
8.42	Test cases for Submit the form without File	59
8.43	Test cases for Submit the form with file	60
8.44	Test cases for Submit the form without Leaving Time	60
8.45	Test cases for Vehicle Option	60
8.46	Test cases for Reason	61
8.47	Test cases Invalid file format	61
8.48	Test cases for Sending Notification	61
8.49	Test cases for Guard Login	62
8.50	Test cases for Empty email field	62
8.51	Test cases for Empty Password field	62
8.52	Test cases for Invalid Email	63
8.53	Test cases for Incorrect Credentials	63
8.54	Test cases for Incorrect Credentials	63
8.55	Test cases for Admin Login	64
8.56	Test cases for Empty email field	64
8.57	Test cases for Empty Password field	64
8.58	Test cases for Invalid Email	65
8.59	Test cases for Incorrect Credentials	65

8.60	Test cases for Displaying the gate-pass requests	65
8.61	Test cases for handling failure of fetching details	66
8.62	Test cases for Accepting the gate-pass request	66
8.63	Test cases for Rejecting the gate-pass request	66
9.1	Performance Matrix	93

Contents

Acknowledgement	i
Abstract	ii
Abbreviation	iii
List of Figures	v
List of Tables	viii
1 Introduction	1
1.1 Overview	2
1.2 Aim	2
1.3 Objectives	3
1.4 Organization of Report	4
2 Literature Survey	5
2.1 Conclusion From Literature Survey	7
3 Software Requirement Specification	8
3.1 Introduction	8
3.1.1 Purpose	8
3.1.2 Intended audience and reading suggestion	9
3.1.3 Project Scope	10
3.1.4 Design and Implementation Constrain	10
3.1.5 Assumption and Dependencies	10
3.2 System Features	12
3.2.1 User Roles and Authentication	12
3.2.2 Gate Pass Request	12
3.2.3 Gate Pass Approval Workflow	12
3.2.4 QR Code Generation	12

3.2.5	QR Code Scanning	12
3.2.6	User Notifications	12
3.2.7	Access Control and Permissions	12
3.3	External Interface Requirement	13
3.3.1	User Interface	13
3.3.2	Software Interface	13
3.3.3	Communication Interface	14
3.4	Non Functional Requirements	15
3.5	Other Requirements	16
3.5.1	Database Requirements:	16
3.6	Analysis Model	17
3.6.1	Data Flow Diagram	17
3.6.2	Class Diagram	20
3.6.3	State Machine Diagram	21
3.7	System Implementation Plan	22
3.7.1	Implementation Plan	22
4	System Design	23
4.1	System Architecture	23
4.1.1	Working:	25
4.2	UML Diagrams	27
4.2.1	Entity Relationship Diagram	27
4.2.2	Activity Diagram	28
4.2.3	Use Case Diagram	29
4.2.4	Communication Diagram	30
4.2.5	Sequence Diagram	31
4.2.6	Component Diagram	32
4.2.7	Deployment Diagram	33
5	Technical Specifications	34
5.1	Technology details used in the project	34
6	Project Estimation Schedule and Team Structure	35
6.1	Project Estimate	35
6.1.1	Equations:	36
6.1.2	Organic project:	36
6.1.3	Calculation	37
6.2	Project Schedule and Team Structure	39

7	Software Implementation	41
7.1	Introduction	41
7.2	Databases	42
7.3	Important module and algorithms	42
7.3.1	Modules	42
7.3.2	Algorithm	44
7.4	Business logic	44
8	Software Testing	45
8.1	Introduction	45
8.1.1	Test cases for Student Login Page	45
8.1.2	Test cases for Student Sign-Up Page	48
8.1.3	Test Cases For Student Edit Profile Page	49
8.2	Test Cases For Student Gate Pass Request Page	52
8.2.1	Test cases for Staff Login Page	54
8.2.2	Test cases for Staff Sign-Up Page	56
8.2.3	Test Cases For Staff Edit Profile Page	57
8.2.4	Test Cases For Staff Gate-Pass Request Page	59
8.2.5	Test cases for Guard Login Page	62
8.2.6	Test cases for Admin Login Page	64
8.2.7	Test cases for Manage Requests Page	65
8.3	Snapshots of Test Cases	67
9	Result	92
9.1	Result	92
10	Deployment and Maintenance	94
10.1	Deployment and Maintenance	94
10.1.1	Installation and un-installation	94
10.1.2	Maintenance	94
	Conclusion and Future Scope	95
	References	96
	Plagiarism Report	97
	Paper Publication and Certificate Details	97

Chapter 1

Introduction

The security of individuals has become a top priority for businesses and organizations across all industries in recent years. The welfare of the students is the responsibility of the learning institution. Thus, authorization is needed to enter or exit a campus to maintain security. Over the past few decades, not much has changed in the permissions process. A letter of request is written by the concerned party and submitted for processing. After manual verification, a written permit that can be used at entry and exit points is given out. The concerned faculty can view any specific user's records of any gate pass that has been issued to them upon request, according to the up-to-date cloud database for that purpose.

The process may involve several stages for human verification. This laborious process involves a great deal of physical labor. The project's objective is to automate and launch this system. By doing this, both parties will save a range of resources. The project also maintains transparency and reduces conflict between the student, the warden, and the student coordinator. Making gate passes is an essential step in ensuring the security of any educational facility. Traditional gate pass generation methods involve laborious, error-prone manual processes. This project utilizes a QR code to overcome these obstacles. Nothing can use a unique QR Code.

1.1 Overview

The DiG-Pass project aims to introduce a comprehensive solution for enhancing and securing gate-pass management using QR code technology. By leveraging the Android smartphone, the project seeks to streamline the process of issuing, verifying, and managing gate passes for various organizations. DiG-Pass allows users to conveniently generate digital gate passes encoded with encrypted information such as identification details, access permissions, and validity periods. QR codes are then easily scanned by security guards using the app to authenticate users and grant access. The project emphasizes both enhanced security and user experience, ensuring that sensitive data is protected while providing a seamless and efficient solution for gate-pass management. DiG-Pass also includes administrative functionalities for managing user accounts, access permissions, and generating comprehensive reports to facilitate smooth operation and accountability. By offering a modern, user-friendly, and secure approach to gate-pass management, DiG-Pass aims to revolutionize access control systems across various industries and sectors.

1.2 Aim

The project aims to develop a modernized system for managing gate pass system that leverages QR code technology. The project seeks to enhance the efficiency and security of traditional gate-pass systems by implementing a digital solution. By utilizing QR codes, the project aims to streamline the process of issuing, validating, and managing gate passes, ultimately improving access control and reducing administrative burden. Additionally, the project aims to enhance security measures by incorporating encryption and authentication protocols to prevent unauthorized access and ensure data integrity.

1.3 Objectives

1. Enhance college campus security:

Improve college campus security by implementing an efficient gate-pass management system that regulates access control, ensuring only authorized individuals enter designated areas, thereby enhancing overall safety and security on college campus.

2. Issue digital gate passes:

Introduce a digital gate-pass issuance process for students and staff, replacing traditional paper-based methods with a streamlined digital solution. This allows for quicker issuance and validation of passes, reducing the risk of pass duplication or loss.

3. Reduce paperwork and processing time:

Minimize the administrative burden associated with traditional gate-pass issuing processes by transitioning to a digital system. This reduces paperwork and eliminates manual work for both administrators and users.

1.4 Organization of Report

The rest of this report is organized in the following manner. In all chapters, related contents are described in detail.

- **Introduction (Chapter 1):** In this chapter, the overview of existing systems and their problem is discussed. This chapter describes the aim, motivation, and objectives of the software system.
- **Literature Survey (Chapter 2):** In this chapter, Related work done in the Previous papers has advantages and disadvantages. Related information is available in standard Books, Journals, Transactions, Internet Websites, etc. is discussed.
- **Software Requirement Specification (Chapter 3):** In this chapter, the detailed description of requirements is specified.
- **System Design (Chapter 4):** This chapter discusses the proposed system with the help of system architecture, system design, and UML diagrams
- **Technical Specifications (Chapter 5):** This chapter, discusses the technical details used in the project
- **Project Estimation Schedule and Team Structure (Chapter 6):** This chapter discusses project estimate, brief of COCOMO model, and related calculation and team structure
- **Software Implementation (Chapter 7):** This chapter discusses important module and algorithm also business logic and archite
- **Software Testing (Chapter 8):** This chapter gives a briefing about testing for various modules
- **Software Testing (Chapter 9):** This chapter discusses installation and uninstallation of project as well as maintenance
- **Conclusion and Future Scope (Chapter 10):** This chapter summarizes and concludes the project report and give the future scope.
- **Plagiarism Report(Chapter 11):** This chapter shows the plagiarism report.

Chapter 2

Literature Survey

DiGintry-Securing gated premises using QR-code

Authors: [Ms. Ashwini Jarali, Ms. Snehal Kodilkar, Mr. Siddharth Patel, Mr. Shubham Tondare, Mr. Ganesh Kudale] (June 2020)

DiGintry is an Intelligent Security Management for Gated Premises that digitizes manual undertakings at the primary gates.[1] The system begins with client registration under an organization's administrator, generating time-limited QR codes. Clients enter by scanning their QR codes, which validate their entry in real time. Administrators scan codes upon exit, storing entry and exit details in temporary and permanent databases. This dual-database approach facilitates client tracking, and unrevealed exits trigger communication with the administrator.

Authenticated Gate-Pass Generating Application Using QR-Code

Authors: [Akshay ET, Afsal M, Abhinav R, Rahul C, Prof. Mohammed Mailk CK, Ass. Prof. Haseena M] (April 2023)

Generating gate passes for students is an essential aspect of ensuring the safety and security of any educational institution.[2] The System Generates the QR code. Using a QR code generating tool, generate a unique QR code for each student, print it on paper, and deliver it to all the students. Then, the admin can monitor and track each gate pass holder.

Implementation of Smart and Secure Gate Pass System using QR Code

Authors: [Deepanshu Jaiswal, Devansh Singh, Ms.Aarushi Thusu] (Feb 2023)

The scope of the proposed work that is automated gate pass system is to record the arrival and departure of the students in institution.[3] The process of identifying an individual on the basis of Mail ID and Password. Students have to fill out the online form mentioning details like Name, Roll Number, Room No, Mail, Branch, and Reason. After the submission of the form by a student the warden has rights either to accept or to reject the request. On rejection, a mail would be sent to the student with a reason for rejection. The student needs to get the code scanned by the scanner installed at the exit gate.

Face Recognition Based Gate Pass System

Authors: [Dr. Sunil Bhutada, Dr. Sreenivas Mekala, Mayukhi Gandham, Rishika Bhat, Ruchitha Upadhyayula] (June 2022)

It is a software-based web application based on face recognition. Students can generate the gate pass by entering their Roll Number, Name, Address, Class, etc. After the generation of gate passes, when a student arrives to leave the campus, his/her face will be compared with the stored images in the database. If it is valid, then it allows students to leave the campus. The Face Recognition based Gate pass system assists both the organization and the guest in managing their Gate passes.[4]

Gatepass Generation and Management System Using QR Code

Authors: [Abhijit Alane, Shrinivas Chalikwar, Ganesh Pekam, Padmavati Sarode, Pranav Pekam] (May 2022)

In this paper, students have to register with a valid email ID and details like Name, Class, and Roll Number. Then they can log in to the app and generate the QR code for Gate Pass. The guard will scan the QR code generated by visitors, then that guard will scan it to mark users as in or out, if the QR code is valid, then our entry and exit are automatically saved in the database. One of the main advantage of using this system is maintain all the entry and exit record of a person is saved and we can easily access it.[5]

Gate Pass System

Authors: [V. Sellam, Medha Shree, Shreya Chopdar, Shambhavi] (Dec 2019)

The objective of this work is to make the hectic process of getting a gate pass easier and less stressful.[6] An Application is designed for hostel students to get gate passes for going outside the hostel. The user will get a unique Username and Password for accessing the application. The request is first sent to the class in charge. If they approve, then the request is sent to the coordinator. After the approval of the coordinator, the warden will receive a notification for permission, and the warden issues a gate pass to students.

E-Gatepass System

Authors: [Chaitanya Lengure, Laxmikant Kakde, Mamta Bargat, Saachi Jambhulkar, Prof. Ashish Palandurkar, Prof. Hemant Wade] (March 2018)

“E-Gatepass System” is a Client-Server application software. It uses the concept of MVC (Model View Controller) to implement the application.[7] In this paper, the authorized clerk may provide the gate pass to the legal student. Students will fill in details like Name, Branch, Mobile Number, and Reason. The administrator can enable or disable unauthorized users from the system. The guard will get a notification about gate passes. They will allow students to leave the campus if they have a legal gate pass.

2.1 Conclusion From Literature Survey

After conducting the literature survey on our project topic we found that there is a need to replace the traditional gate-pass system with a new digital gate-pass system. In this digital era, the conventional gate pass system demands a lot of administrative and paperwork. By designing the digital gate-pass system, we reduce administrative burden, paperwork, time, and security personnel. By using a QR code, we eliminate duplicate gate passes. By conducting a literature survey, we got a deep understanding of the gate-pass system and how the traditional way is inefficient in the current era.

Chapter 3

Software Requirement Specification

3.1 Introduction

3.1.1 Purpose

The Software Requirements Specification (SRS) document for this project outlines the purpose, features, and constraints of the system. The purpose of SRS is to provide an unambiguous description of the system's requirements, ensuring a shared understanding among stakeholders, guiding the development process, and serving as a basis for quality assurance and change management. The scope of the SRS document encompasses system functions, non-functional requirements, user interfaces, database specifications, external interfaces, and change control procedures, ultimately defining the project's boundaries and ensuring the successful development of a user-friendly and efficient gate pass system.

Why the project is Chosen?

The "DiG-Pass: Enhanced and Secure Solution for Gate-Pass using QR Code" was chosen to address critical shortcomings in traditional gate pass procedures within college campuses. These outdated methods often involve manual processes, leading to inefficiencies, security vulnerabilities, and inconvenience for students and staff alike. By leveraging QR code technology, this project aims to revolutionize gate pass management by providing a modern, streamlined, and secure solution. A convenient and dependable way of authentication and access control is provided by QR codes, which improve campus security and make the gate pass procedure easier for users. Moreover, the digital nature of the system allows for real-time synchronization of data, scalability, and integration with existing campus systems. Ultimately, this project was chosen to enhance the overall campus experience, promote efficiency, and ensure the safety and security of college

students and staff. Overall, the choice to implement DiG-Pass is driven by the desire to enhance efficiency, security, convenience, and adaptability to modern technology in the college environment.

3.1.2 Intended audience and reading suggestion

Intended audience

- **Security Guard:** Individuals tasked with verifying QR code and granting permission to exit from college campus.
- **Administrators:** Those responsible for managing user accounts, permissions, and access control policies within the DiG-Pass system.
- **Student Developers:** Undergraduate or graduate students who are tasked with developing the Android app "DiG-Pass" as part of a college course or project.

Reading Suggestions

- **Understanding Project Objectives:** Students should carefully review the project objectives outlined by the faculty supervisor to ensure alignment with the intended goals of the DiG-Pass.
- **Clarification of Requirements:** If there are any aspects of the project requirements that are unclear, students should seek clarification from the faculty supervisor or project stakeholders.
- **Mobile Application Development Guide:** Explore resources on Android app development to gain insights into best practices, user interface design, and security considerations for mobile applications.
- **Seeking Feedback:** Regularly seek feedback from the faculty supervisor and project stakeholders to ensure that the development process is on track and meeting expectations.

3.1.3 Project Scope

The scope of the project includes the development of an Android mobile application that allows students and faculty to generate QR codes for access control within the college campus. This system will streamline exit procedures and replace traditional paper gate passes with digital QR codes. Key features include an easy-to-use Android app, secure QR code generation, and validation through the in-app security guard login module, integration with university management systems, and privacy compliance to improve security, efficiency, and user experience at the same time to protect the environmental effects of paper-based systems.

3.1.4 Design and Implementation Constrain

1. Ensure compatibility with a wide range of Android devices and screen sizes.
2. Optimize database structure and queries for efficient storage and retrieval of gate pass data.
3. Implement robust encryption techniques to secure sensitive user information within the app.
4. Integrate Firebase Authentication for secure user authentication and authorization.
5. QR code generation and scanning must be seamless and user-friendly.
6. Employ efficient algorithms for real-time QR code scanning and validation.

3.1.5 Assumption and Dependencies

Assumptions

1. **User Availability:** Users are assumed to have access to Android smartphones with internet connectivity.
2. **Data Connectivity:** It is assumed users have access to a stable internet connection for functionalities such as submitting gate-pass requests, receiving notifications, and syncing data with the server.
3. **User Registration:** To access the features of the app and create an account, users are presumed to have to go through a registration process. It is expected that users will register using true and accurate information.

4. **Administrative Access:** The app assumes that there will be designated administrators responsible for managing gate-pass requests, user accounts, and access permissions.

Dependencies

1. **Backend Services:** The app depends on backend services such as Firebase Authentication, Realtime Database for User Authentication, Data Storage, and Push Notifications.
2. **Third-Party Libraries:** Dependencies on third-party libraries like ZXing (Zebra Crossing) for QR code scanning and generation may exist to facilitate specific functionalities within the app.
3. **Device Features:** The app may depend on device features such as camera access for QR code scanning, and network connectivity for data synchronization registration.
4. **Permissions:** The app depends on the user granting necessary permissions, such as camera access and network access, for proper functioning.

3.2 System Features

3.2.1 User Roles and Authentication

Identify the various user roles (e.g., students, staff, admin, security guard) and define the authentication methods.

3.2.2 Gate Pass Request

Enable students to send a request for a gate pass by providing the purpose, date, time, and duration.

3.2.3 Gate Pass Approval Workflow

Define a workflow for gate pass approval, involving relevant stakeholders (e.g., security personnel, administrators). Implement approval/rejection mechanisms, notifications, and alerts.

3.2.4 QR Code Generation

The App will automatically generate a unique QR code for each gate pass after approving the request from the admin. QR code contains essential information such as the user's name, purpose, and validity.

3.2.5 QR Code Scanning

Equip security guards with the ability to scan QR codes using mobile devices or dedicated scanners. Verify the authenticity and validity of gate passes then permit to exit from campus.

3.2.6 User Notifications

Send notifications to users via email, SMS, or mobile app alerts about the status of their gate pass requests (approved, rejected, pending).

3.2.7 Access Control and Permissions

Define access control rules to restrict users' access based on their roles and permissions. Allow administrators to configure access permissions for different campus areas or facilities.

3.3 External Interface Requirement

3.3.1 User Interface

1. **Android Application:** The main user interface for generating and scanning QR codes is the Android mobile application, which has to be simple to use and intuitive. It should give users feedback and clear instructions.
2. **Administrative Dashboard:** To control and keep an eye on the system, an administrative interface is needed. It should have features for user management, configuring access control rules, and system reporting, and it should only be accessible by authorized personnel.
3. **Alert Notifications:** The Android application should have the capability to display alert notifications to users regarding important updates, such as the status of the gate pass.
4. **Gate-Pass History:** Within the Android application, users should be able to access their history, providing details of their previous access requests and approvals. This feature enhances transparency and allows users to track their activity.

3.3.2 Software Interface

1. **Backend Services:** The Android app should interact with backend services that manage data storage, access control, user authentication, and QR code validation. For this interaction, web APIs or other communication protocols are needed.
2. **Database:** To store and retrieve user data, access logs, and other information, the system needs to communicate with a database system. It should support the required query language and database management system.
3. **Authentication Services:** Integration with authentication services may be necessary to ensure secure user authentication within the system. This allows users to log in securely using their existing credentials, enhancing usability and security.
4. **Third-Party Integration:** The system may need to integrate with third-party services or APIs for additional functionalities such as identity verification services for enhanced security measures.

3.3.3 Communication Interface

1. **Network Protocols:** The Android app and backend services should communicate securely over network protocols (e.g., HTTPS) to protect data in transit. Secure Socket Layer (SSL) or Transport Layer Security (TLS) may be employed for encryption.
2. **Push Notifications:** To notify users via push notifications about the status of gate-pass requests, communication interfaces might be required.
3. **Real-Time Updates:** The communication interface should support real-time updates between the Android app and backend services. This ensures that any changes made to user permissions, access rules, or gate-pass status are immediately reflected in the app. Real-time updates enhance user experience by providing timely information and reducing the risk of access control errors.
4. **Multi-Channel Communication:** The communication interface should support multiple channels for interacting with users, including in-app messaging, and email notifications. This multi-channel approach ensures that important information reaches users through their preferred communication channels, increasing the likelihood of timely responses and actions.

3.4 Non Functional Requirements

1. **Performance Requirements:** To guarantee prompt access, performance requirements outline how quickly the system must generate and validate QR codes. They discuss scalability to keep the system responsive as the college expands as well as the system's ability to manage large volumes of requests during peak times.
2. **Safety Requirements:** Safety requirements focus on data integrity, ensuring that users' information and access logs are protected from unauthorized access or tampering. They also include redundancy measures to guarantee system operation during failures and provisions for emergency access during crises.
3. **Security Requirements:** User authentication mechanisms are emphasized by security requirements to confirm users' identities. They also cover access control procedures to stop unauthorized access to restricted areas and data encryption techniques for safe data transmission and storage.
4. **Availability:** Requirements for availability specify the acceptable amount of system downtime as well as interruption-minimization techniques. They consist of redundant systems to guarantee high availability, dependability, and maintenance scheduling during low activity periods.
5. **Functionality:** The system's essential features, such as the creation of QR codes, scanning, user management, and system integration with college systems, are outlined in the functionality requirements.

3.5 Other Requirements

3.5.1 Database Requirements:

1. **Data Schema Design:** Define a clear and well-structured database schema that includes tables for users, gate passes, access logs, and any other relevant data entities. This schema should be designed to accommodate the specific data needs of the gate pass system.
2. **User Profiles and Access Permissions:** Maintain a database for user profiles and access permissions which contains information on students, staff, faculty, and visitors. This database should include fields for user profiles, access permissions, and unique identifiers. Implement the necessary user roles and access levels.
3. **QR Code Data Storage:** Store QR code data, including the information embedded in the codes such as user details, access privileges, and timestamps.
4. **QR Code Validation History:** Maintain a record of QR code validations, tracking when and where each QR code was scanned. This information can be useful for security and auditing purposes.
5. **Data Redundancy and Backup:** Implement data redundancy and regular backups to prevent data loss due to hardware failures or other unforeseen events. Backup strategies should be robust and routine.

3.6 Analysis Model

3.6.1 Data Flow Diagram

DFD Level 0

A DFD level 0, also called a context diagram, is the simplest DFD showing the entire system as a single process with its external entities (inputs and outputs). The process starts with user registration. A user can then request a gate pass. The login information is then sent to an admin who can accept or reject the request. If accepted, the user receives a notification and QR code. The user can then scan the QR code, which will show the gate pass details to a guard.

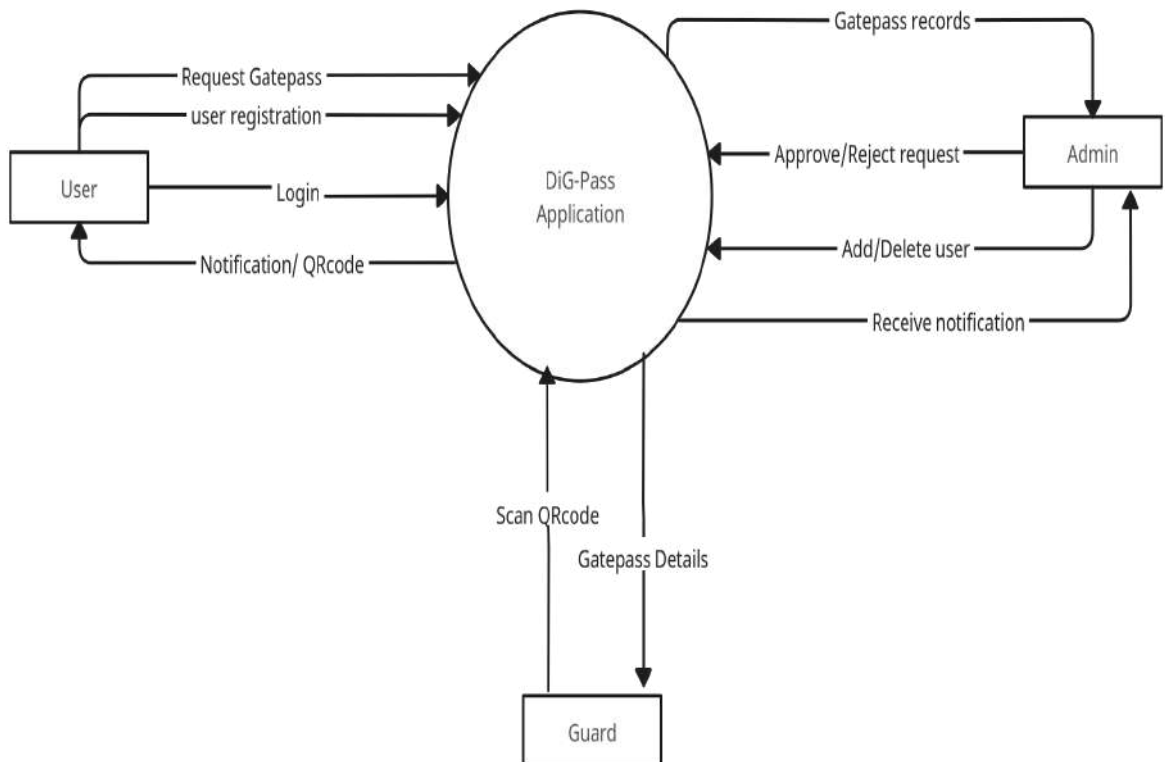


Figure 3.1: DFD Level 0

DFD Level 1

A DFD level 1 expands on the context diagram, breaking down the main system process into smaller sub-processes like user interaction, data storage, admin management, and guard access. The user interacts with the system through Login/Register, Request Management, and View Records. Gate pass data is stored and retrieved. Admin manages users (add/delete) and approves/rejects gate pass requests. Guard can view gate pass details after a user scans a QR code.

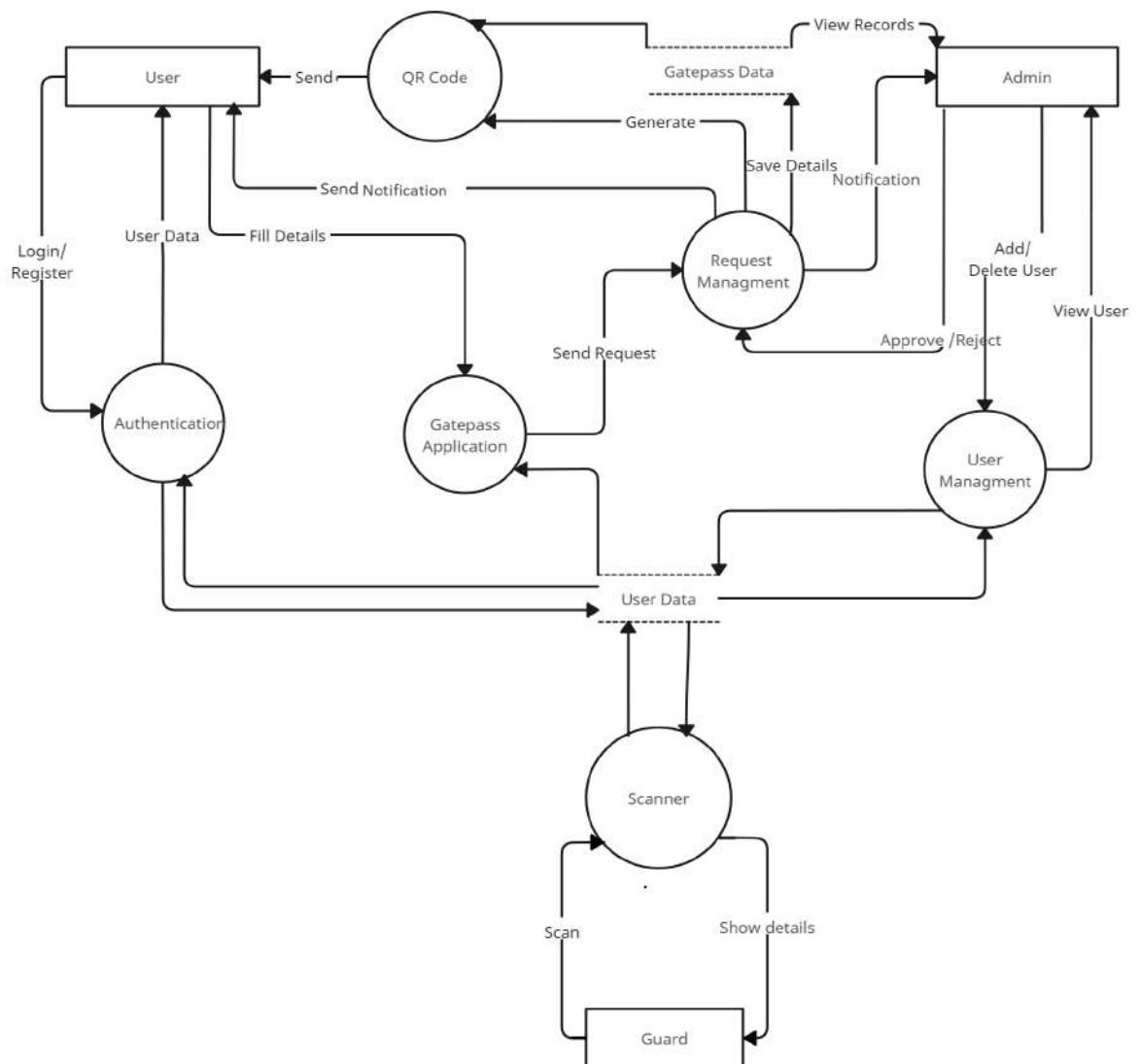


Figure 3.2: DFD Level 1

DFD Level 2

A DFD level 2 takes a level 1 process and dives deeper, showing its internal sub-processes like request submission, notification, approval/rejection, and gate pass generation. The user requests a gate pass. The system sends a notification to the admin. The admin can either accept or reject the request. If request is accepted, the system generates a QR code and sends a notification to the user. If rejected, the system sends a rejection notification to the user.

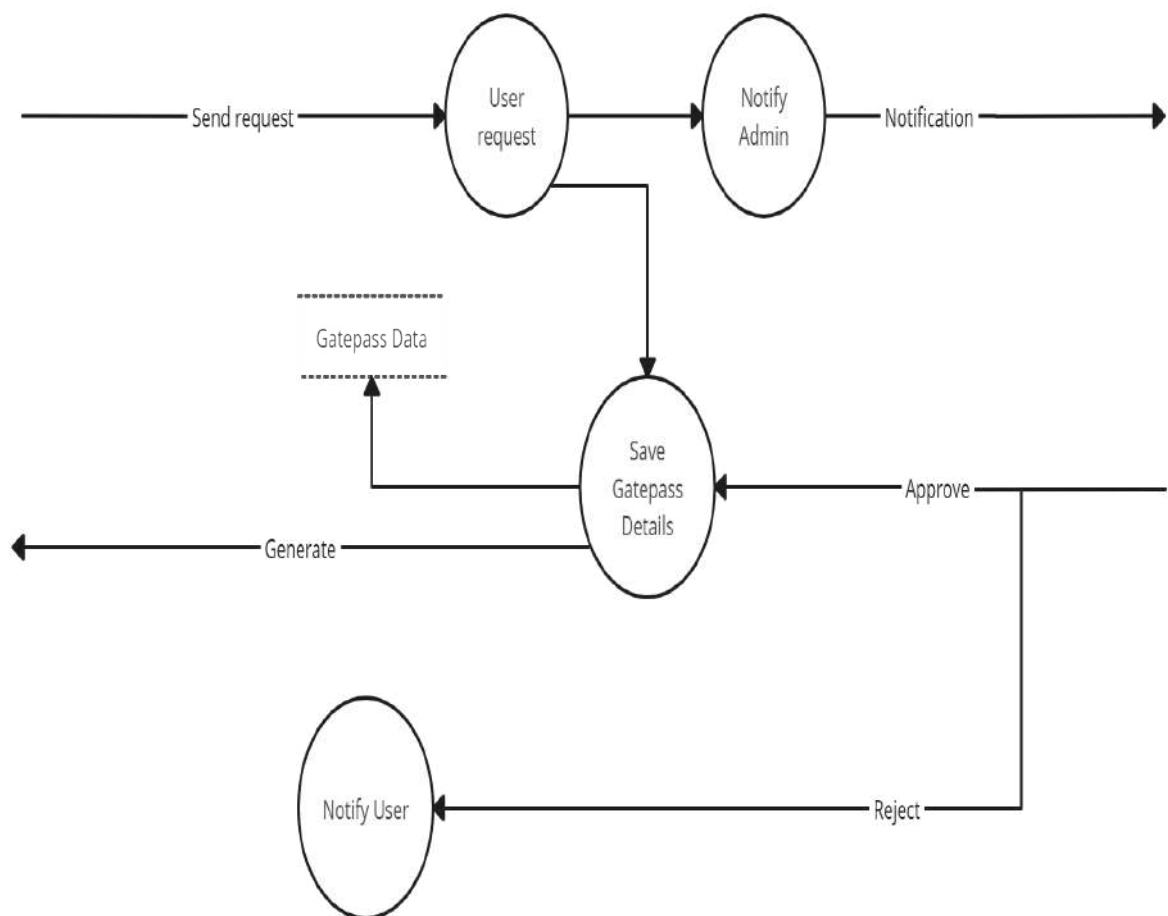


Figure 3.3: DFD Level 2

3.6.2 Class Diagram

The class diagram defines key elements like users, gate passes, and login functionality, showing how they interact like users request gate passes, admins approve/reject, and guards potentially verify details. User class represents different user types such as Student, Staff, Admin, and Security Guard. Each user has login credentials and basic details like name and contact. Gate pass class represents a gate pass record with details like user information, purpose, timestamps (date, leaving time, return time), and approval information (who permitted it). The Login class handles login functionality and authenticates users based on credentials and user type.

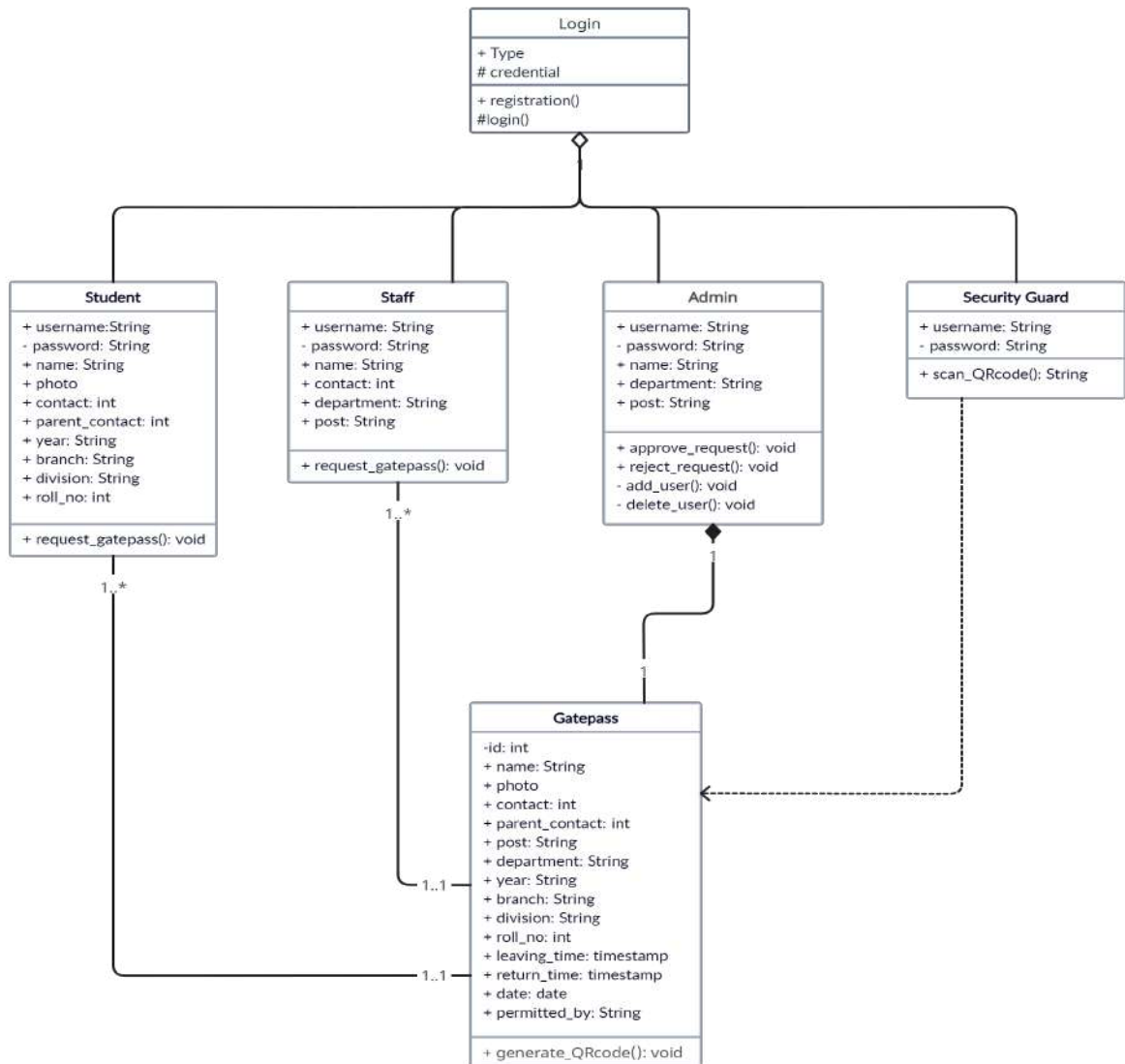


Figure 3.4: Class Diagram

3.6.3 State Machine Diagram

The state machine diagram you sent depicts a process for granting or rejecting a gate pass request. The process starts with the user being either logged in or needing to log in/sign up. If the user is logged in, they can then initiate a request for a gate pass. Once the request is submitted, it goes through a validity check to Admin. If the Request is rejected by the admin, then a request is rejected notification will be sent to the user. If a request is accepted then a QR code will generated and a notification will be sent to the user. The security guard will validate the QR code by scanning it. If QR is valid then the user will be allowed to exit from campus otherwise block the exit.

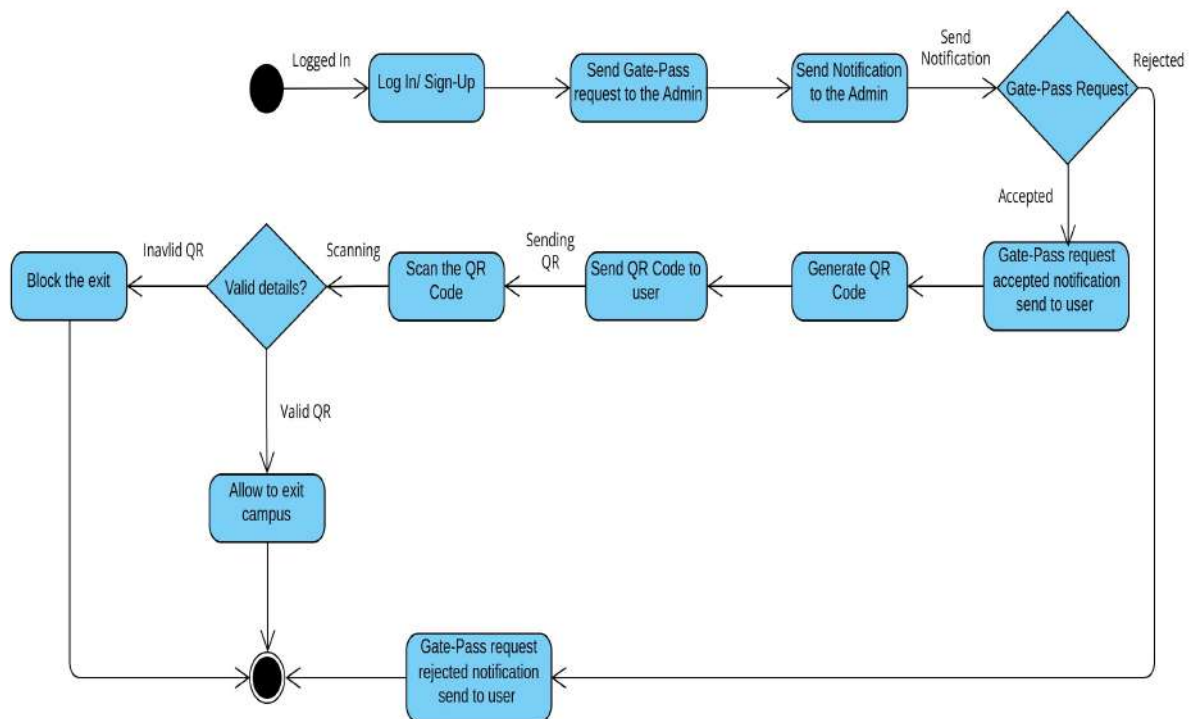


Figure 3.5: State Machine Diagram

3.7 System Implementation Plan

3.7.1 Implementation Plan

T1	Topic Finalization
T2	Requirement specification
T3	Technology Familiarization
T4	System Set up
T5	Concept Review Study
T6	Study of Android Technology
T7	Designing System Architecture
T8	Designing User Interface for Dashboard
T9	Implementation of Student and Staff Module
T10	Firebase Connectivity
T11	Implementation of Admin and Guard Module
T12	Implementation of Notification System
T13	Implementation of QR code generation
T14	Testing
T15	Documentation Preparation
T16	Maintenance

Table 3.1: Implementation Plan

Chapter 4

System Design

4.1 System Architecture

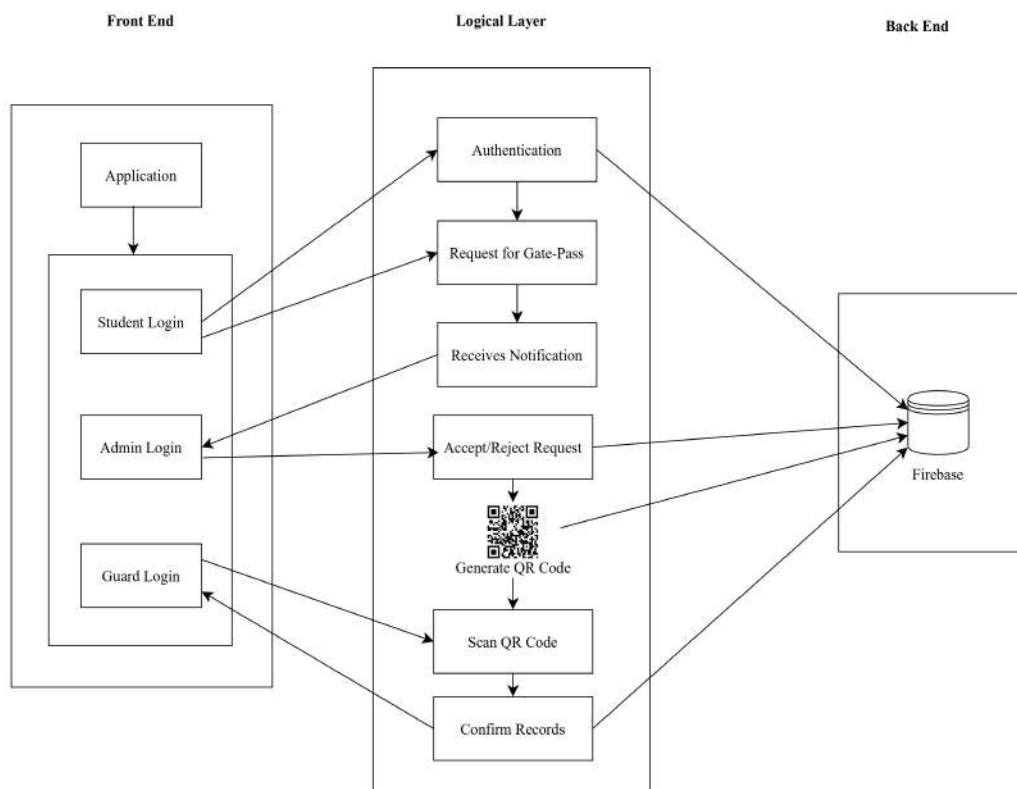


Figure 4.1: System Architecture

Here there are mainly 3 layers.

1. **Front End:** The front-end layer consists of the user interfaces that users interact with to request gate access. This includes the interfaces for Students, Staff, Admin, and Guard. Students/Staff can log in and request a gate pass. Admin can accept or reject gate pass requests and potentially send notifications to the user. Guard can verify QR codes and grant or deny entry.
2. **Logical Layer:** The logical layer, also referred to as the business logic layer, is responsible for processing the gate-pass requests received from the front-end layer and applying the business rules. Authentication involves verifying the identity of users logging into the system. Authorization involves determining whether a user has the necessary permissions to act, such as requesting a gate pass or approving/rejecting a request. Generating QR codes involves creating a unique QR code after approving the gate pass request from the admin.
3. **Back End:** The back-end layer consists of the data storage and server-side functionality that support the application. This layer uses Google Firebase, a cloud database. It stores Student/Staff data, gate pass requests, and potential approval logs. The back end also includes server-side functions that manage tasks such as authentication, QR code generation, and communication between the different layers.

4.1.1 Working:

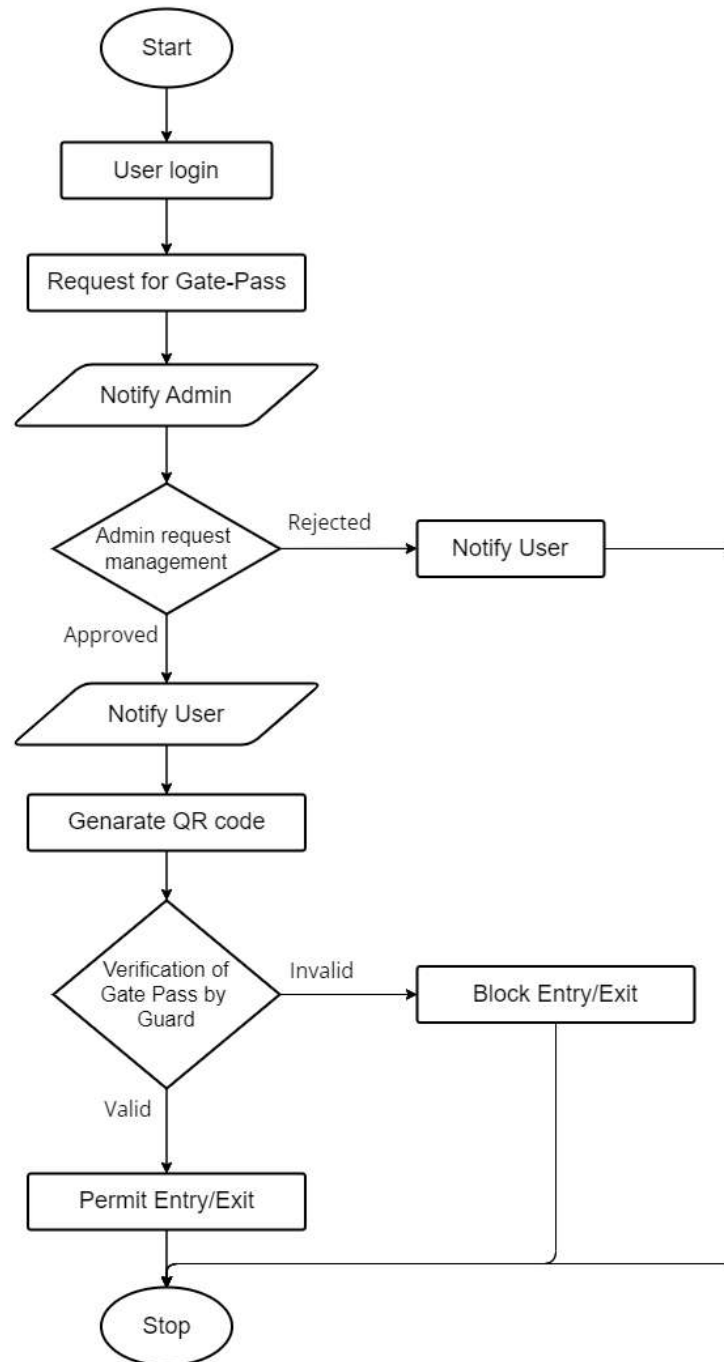


Figure 4.2: Proces flow of DiG-Pass App

Students and staff will sign up for the app by filling in details like name, roll number, department, email, etc. The details will be stored in the Firebase real-time database. After registering in the app, they will sign in using a logging email and password. After logging in to the application, the user can send a gate pass request by filling out the reason for the gate pass, leaving time, and vehicle number, if applicable. The other details in the gate pass request, like name, email, branch, year, and department will be fetched automatically in the form. Users don't need to fill in those details again.

When the user sends the gate-pass request, the admin will receive a notification of every new gate-pass request. For students, the notification of a gate-pass request will be sent to the class teacher. They can review the gate pass request. According to the details, the class teacher will accept or reject the request. The request will be sent to the HOD for review. They can accept or reject the request, and then it goes to the principal for review. If it is accepted, then a QR code will be generated and a notification will be sent to the students.

When the staff sends the gate pass, the notification will be sent to the HOD. They can review the gate pass request. According to the details, the HOD will accept or reject the request. If the request is accepted by HOD, then it goes to the principal for review. If it is accepted, then a QR code will be generated and a notification will be sent to the staff.

At last, the security guard will scan the QR code that is received by the user. After scanning the QR code, details like leaving time, reason, and other details will be fetched and displayed. If the QR code is valid, then it allows users to exit from the campus; otherwise, block the exit.

4.2 UML Diagrams

4.2.1 Entity Relationship Diagram

The ER diagram defines key entities like users, departments, and gate passes, showing how they interact. User entity represents users of the system and has attributes like username, password, name, contact, and photo. Department entities represent departments within the system. The Gatepass entity represents a gate pass record with details like requesting user, purpose, timestamps, and approval information. The Request entity represents the request for a gate pass by a user.

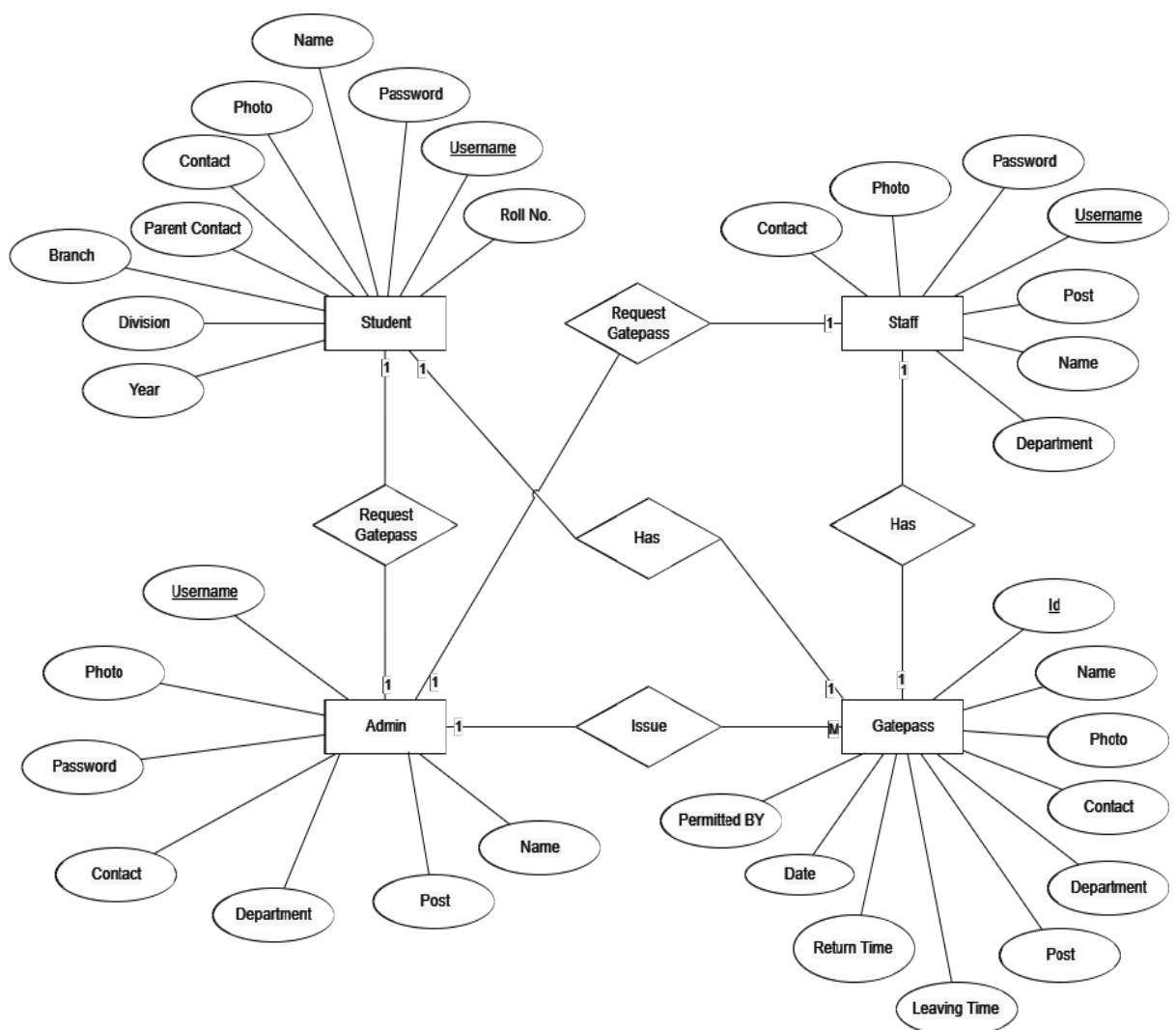


Figure 4.3: ER Diagram

4.2.2 Activity Diagram

The user logs into the system. The system validates the user credentials. If valid, the user is directed to a user interface where they can request a gate pass. The user fills out a gate pass request form with details like purpose and duration. The system submits the request to the admin for approval. The admin receives a notification about the request. The admin reviews the request details and can either approve or reject it. If approved, the system generates a QR code for the gate pass. If rejected, the system sends a notification to the user informing them of the rejection. The user (if approved) receives a notification with the QR code.

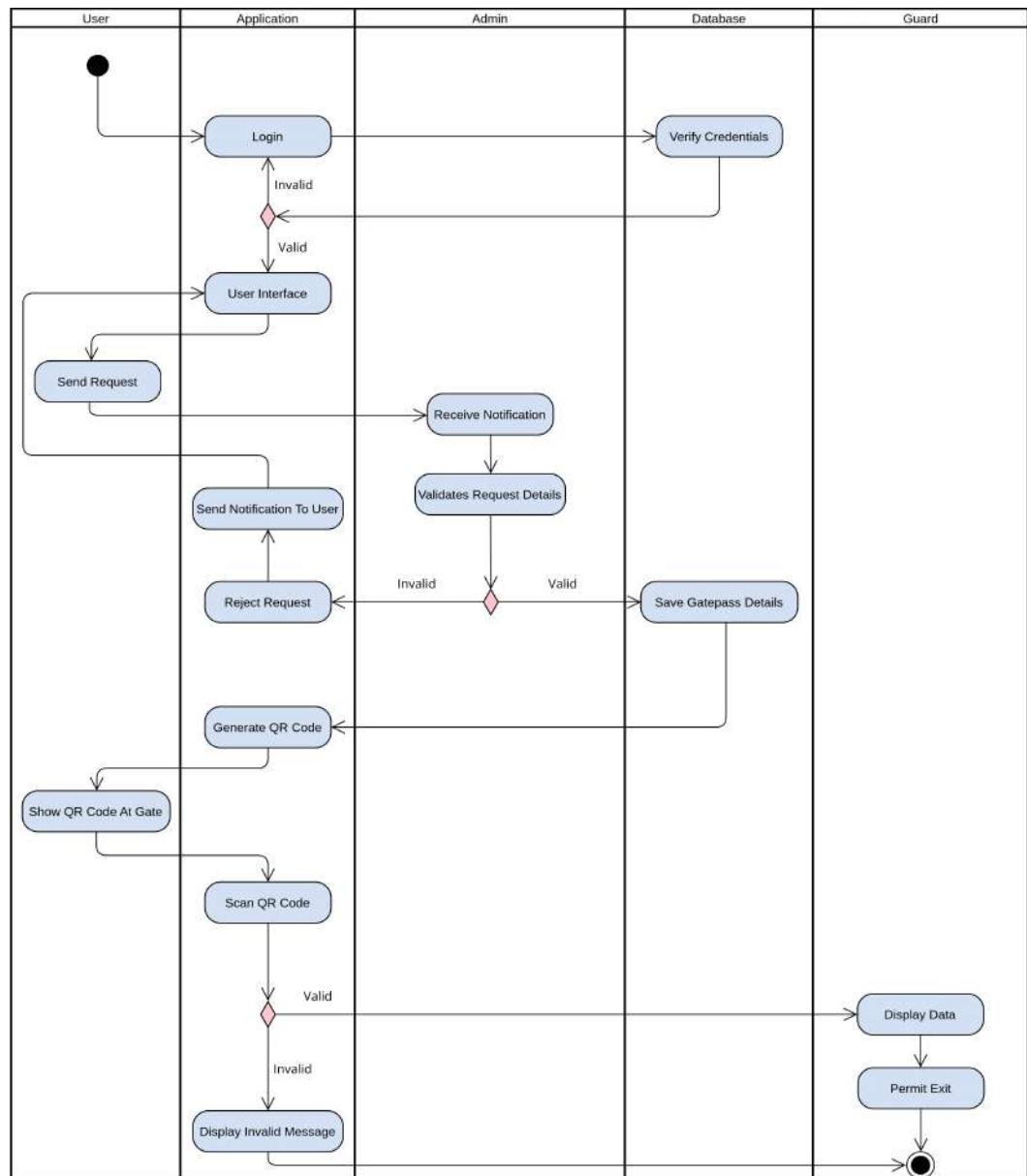


Figure 4.4: Activity Diagram

4.2.3 Use Case Diagram

The use case diagram shows a system for managing gate passes. It depicts actors such as students, admins, and security guards interacting with the system. Students or Staff request a gate pass through the system. Admin can view gate passes and manage requests. They can approve or reject requests. Guard can validate gate passes using a QR code. The diagram uses extensions and inclusions to show optional and mandatory functionalities. For instance, generating a QR code for a gate pass is included in approving a gate pass request.

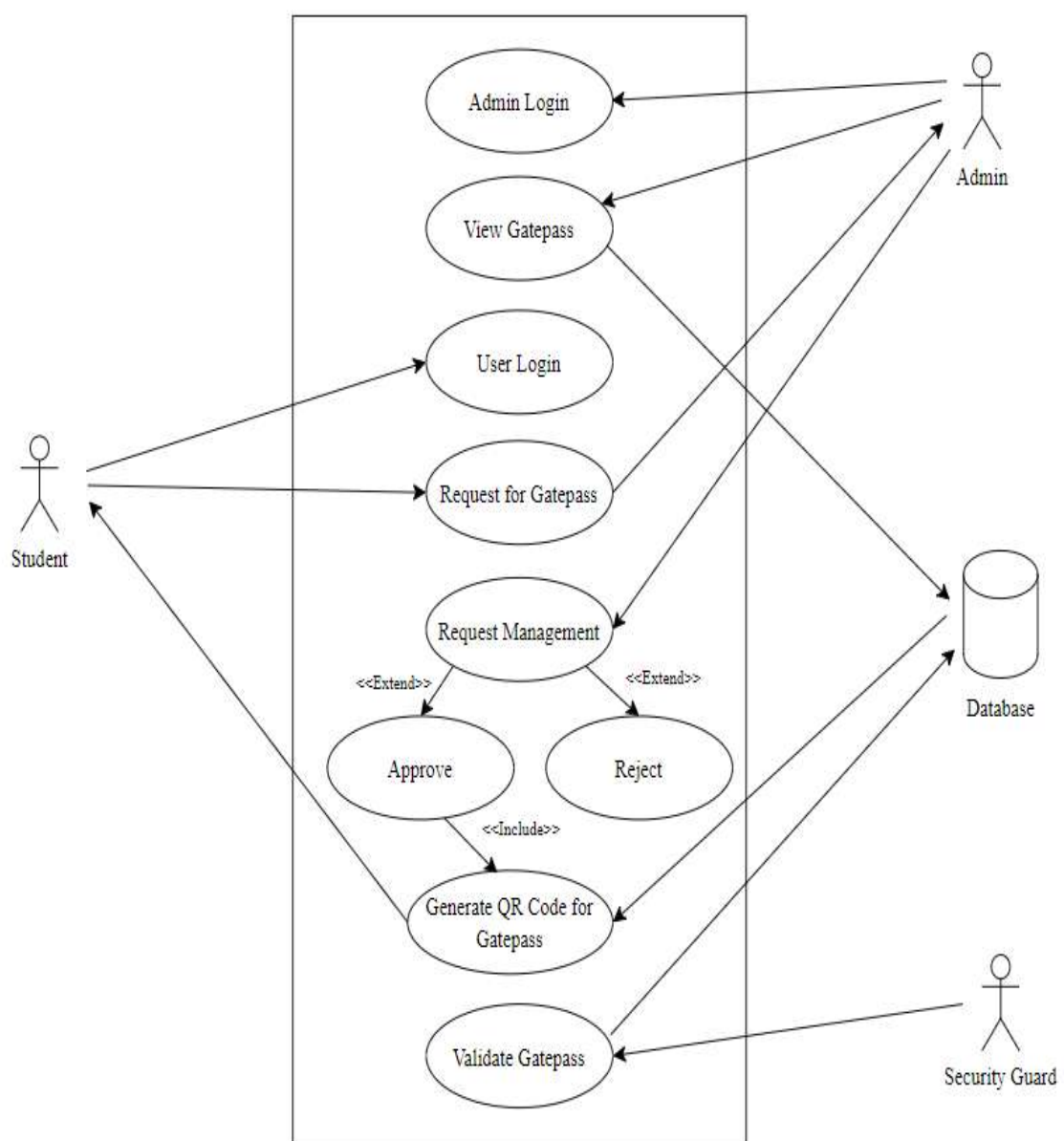


Figure 4.5: Use Case Diagram

4.2.4 Communication Diagram

This communication diagram depicts the process flow for a DiG-Pass system. The user (A1) logs in or signs up and submits a gate pass request. The DiG-Pass system checks authentication and forwards the request to the admin (A2). The admin receives the notification, reviews the request, and either accepts or rejects it, sending a notification back to the user. Finally, the guard (A3) scans the user's QR code at the exit point, verifies the details, and permits or denies exit based on the admin's approval.

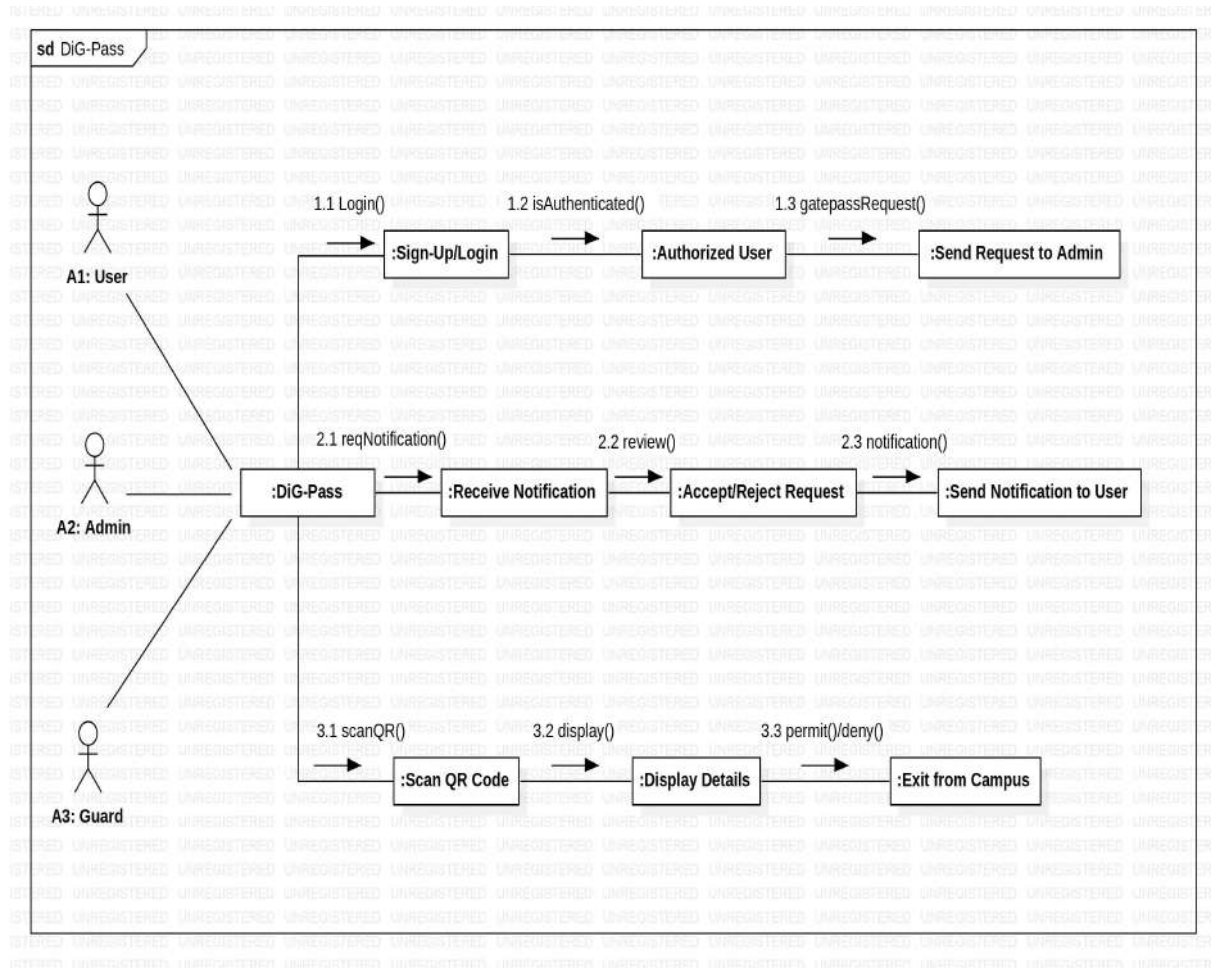


Figure 4.6: Communication Diagram

4.2.5 Sequence Diagram

The user logs in to the application. The application verifies the user's credentials. If the credentials are valid, the user can request a gate pass. The application sends a notification to the admin to approve or reject the request. If the admin approves the request, the application generates a QR code and saves the gate pass details. The user shows the QR code to the guard. The guard scans the QR code. The application fetches data and verifies if it's within the valid date. If the data is valid, the guard permits the user's exit. Otherwise, the exit is restricted.

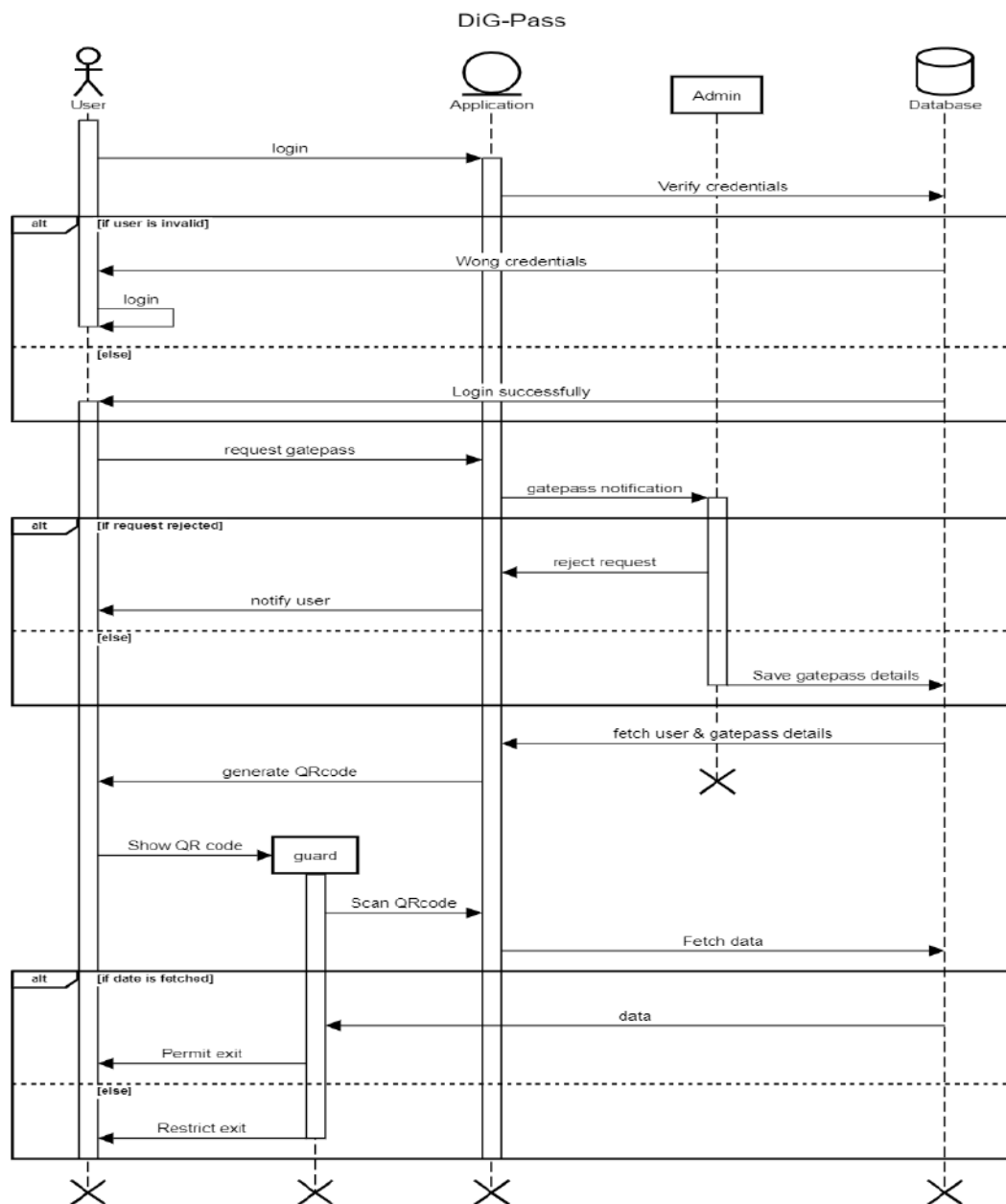


Figure 4.7: Sequence Diagram

4.2.6 Component Diagram

The sequence diagram shows the interaction between a user interface, an application component, and an admin interface. The user interface allows users to interact with the application. When a user requests a gate pass, the application sends a notification to the admin for approval. If the admin approves, the application generates a QR code for the user. The user shows the QR code to a guard, who scans it and verifies its validity with the application. If the QR code is valid, the guard grants the user access.

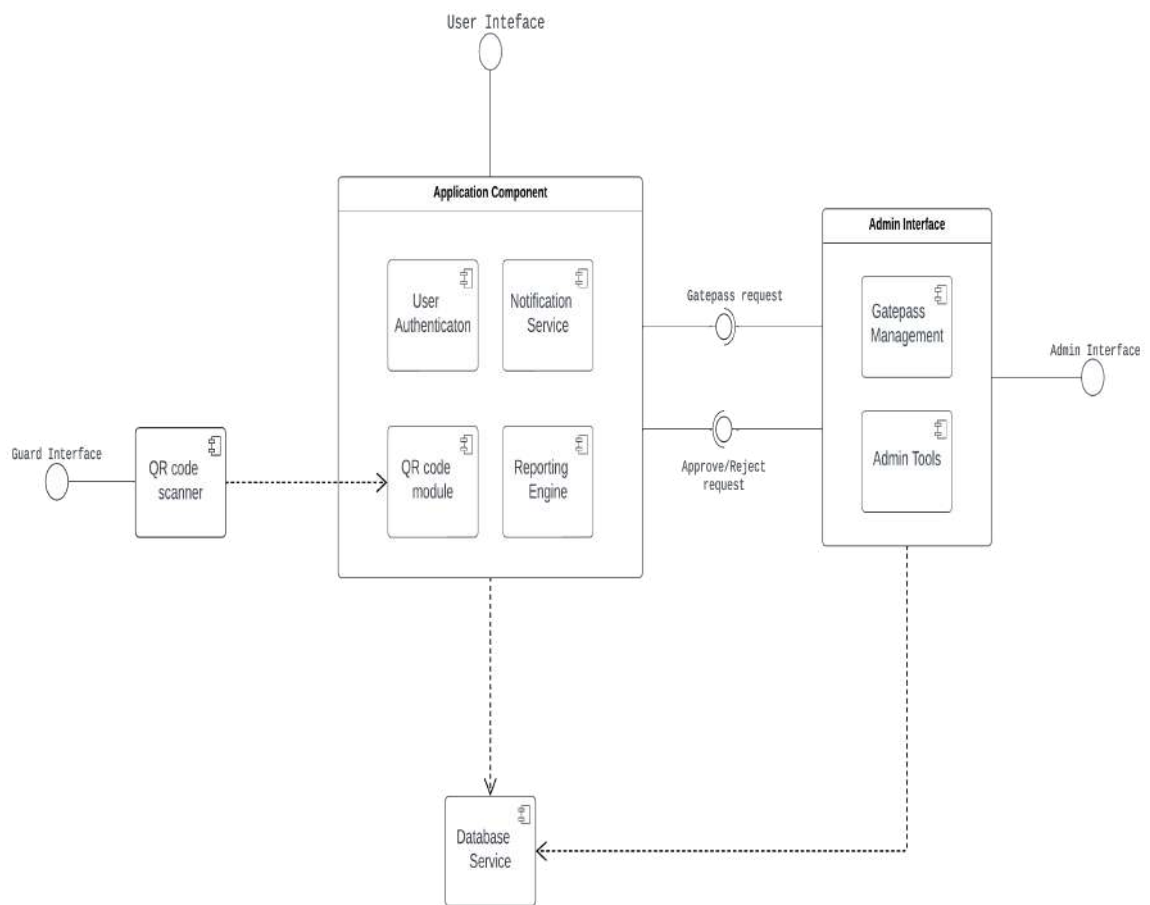


Figure 4.8: Component Diagram

4.2.7 Deployment Diagram

The deployment diagram shows a typical deployment of a mobile application with a backend server. Mobile devices are smartphones with installed application. There are three types of users in the diagram: regular users, admins, and guards. Each user type has their own smartphone app. The Application server stores and manages the application code. It communicates with the database server and sends responses to user requests. The Database server stores the application's data, such as user information and gate pass details. Firebase is a backend cloud service that provides features like authentication, notifications, and databases for mobile applications.

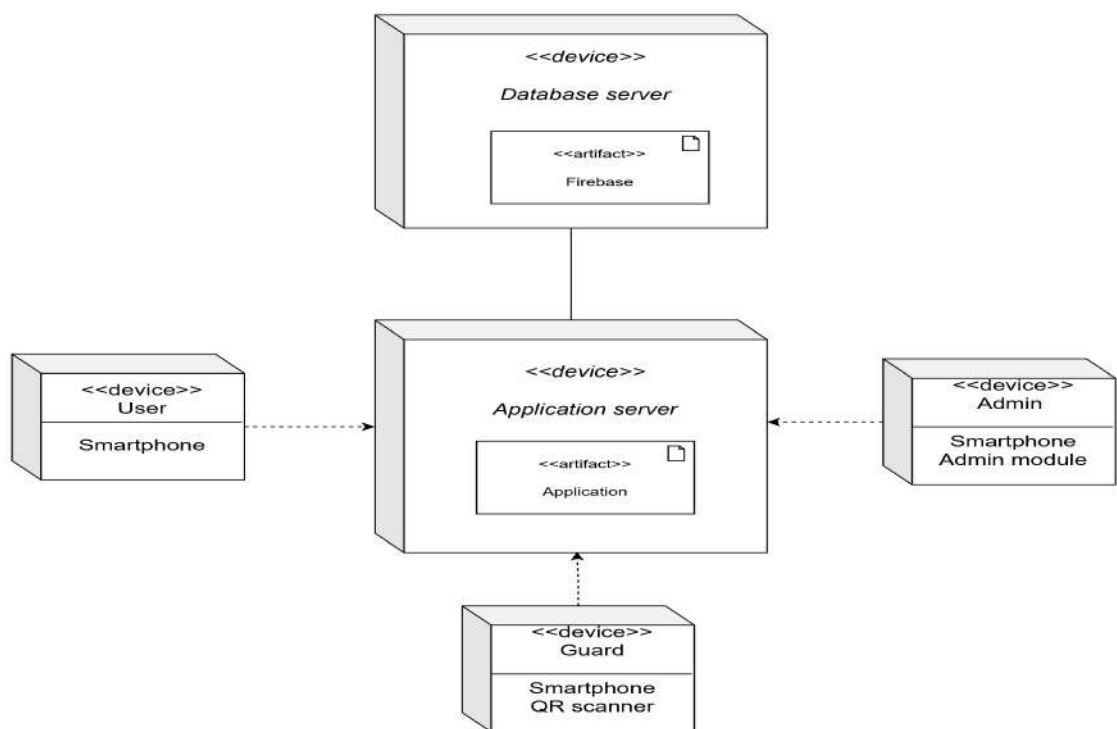


Figure 4.9: Deployment Diagram

Chapter 5

Technical Specifications

5.1 Technology details used in the project

1. **Android:** Android development refers to the process of creating applications for devices running the Android operating system. The app would be developed using the concept of Java programming language using Android Studio IDE.
2. **Firebase:** A backend platform called Firebase can be used to create iOS, Android, and Web applications. It provides hosting, various APIs, authentication methods, and real-time databases. This tutorial serves as an introduction to the Firebase platform, explaining its fundamentals and how to work with its different parts and sub-parts.
3. **Java:** Android Studio helps to create new Java classes; enumeration and singleton classes; and interface and annotation types based on file templates to create a new Java class or type, follow these steps: In the Project window, right-click a Java file or folder, and select New Java Class.
4. **JDK:** The Java Development Kit (JDK) is a cross-platform software development environment that offers a collection of tools and libraries necessary for developing Java-based software applications and applets.
5. **Zxing:** Zxing stands for Zebra Crossing, it is one of the most popular open-source APIs for integrating QR(Quick Response) Code processing. It is a barcode image processing library implemented in Java, with ports to other languages. It has support for the 1D product, 1D industrial, and 2D barcodes. Google uses ZXing by web search to obtain millions of barcodes on the web indexable.

Chapter 6

Project Estimation Schedule and Team Structure

6.1 Project Estimate

COCOMO Model A popular method for estimating software costs is called the Constructive Cost Model (COCOMO), which was created by Barry Boehm. It offers an organized method for determining the amount of work, time, and money needed to develop software projects. To determine the effort and cost estimations, COCOMO takes into account a variety of variables and project features. COCOMO is available in three versions: Basic, Intermediate, and Advanced. An overview of each version is given below:

1. **COCOMO Basic:** Based on the project size, expressed in lines of code (LOC), the COCOMO Basic model calculates the software development effort. It takes into account the formula:

$$Effort = a * (KLOC)^b$$

where,

- The total development effort is measured in person-months as effort.
 - KLOC stands for a thousand lines of code, which is a measure of the software's projected size.
 - The constants a and b depend on the experience of the development team and the type of project (such as organic, semi-detached, or embedded).
2. **COCOMO Intermediate:** The COCOMO Intermediate model builds on the Basic model by including extra project aspects such as product qualities, hardware

limitations, human resource capacity, and development flexibility. It takes into account 15 various cost factors that affect the calculation of overall effort and cost. Four categories have been established for these cost factors: product, platform, staff, and project.

3. **COCOMO Advanced:** The COCOMO Advanced model improves the estimation by taking into account more elements including multi-site development, software reuse, and software stability. It considers elements including the degree of software reuse, the complexity of the dependability requirements for the software, and the effect of geographically dispersed development teams.

There are three modes of development.

Development mode	Size	Innovation	Deadline	Dev. Environment
Organic	Small	little	not light	Stable
Semi-detached	Medium	Medium	Medium	Medium
Embedded	Large	Greater	Tight	Complex hardware

Table 6.1: Modes of development

Here are the coefficients related to the development mode for the intermediate model.

Development mode	a	b	c	d
Organic	2.8	1.05	2.5	0.38

Table 6.2: Coefficients related to development modes for intermediate model

6.1.1 Equations:

$$E = a * (KLOC)^b$$

where,

$a = 2.8$, $b = 1.05$, for an organic project.

E = Efforts in person month

6.1.2 Organic project:

Project of moderate size and complexity, where teams with mixed experience levels must meet mixed rigid and less than rigid requirements (project midway between embedded and organic types).

Number of People:

Equation for calculation of the number of people required for completion of the project, using the COCOMO model is: $N=E/D$

where,

N = Number of people required

E = Efforts in person-month

D = Duration of project in months

Cost of Project:

The equation for the calculation of the cost of the project, using the COCOMO model is:

$$C = E * Cp$$

where,

C = Cost of project

E = Efforts

Cp = Cost incurred per person-month

6.1.3 Calculation

Efforts:

$$E = a * (KLOC)^b$$

$$E = 2.4 * (12.300)^{1.05}$$

E = 33.466 person-months

A total of 33.466 person-months are required to complete the project successfully.

Duration of Project:

$$D = c * (E)^d$$

$$D = 2.5 * (33.466)^{0.38}$$

D = 9 months

The approximate duration of the project is 9 months.

Number of people required for the project:

$$N = E/D$$

$$N=33.466/9$$

$$N=3.71$$

$$N=4 \text{ people}$$

Therefore 4 people are required to complete the project on schedule successfully.

Cost of Project:

$$C = E * Cp$$

$$C=33.466*700$$

Therefore, the cost of the project is 23,426/- (approx)

6.2 Project Schedule and Team Structure

All our project tasks are divided as shown in the table

Task No.	Task Title
T1	Topic Finalization
T2	Requirement specification
T3	Technology Familiarization
T4	System Set up
T5	Concept Review Study
T6	Study of Android Technology
T7	Designing System Architecture
T8	Designing User Interface for Dashboard
T9	Implementation of Student and Staff Module
T10	Firebase Connectivity
T11	Implementation of Admin and Guard Module
T12	Implementation of Notification System
T13	Implementation of QR code generation
T14	Testing
T15	Documentation Preparation
T16	Maintenance

Table 6.3: Lists Of Tasks

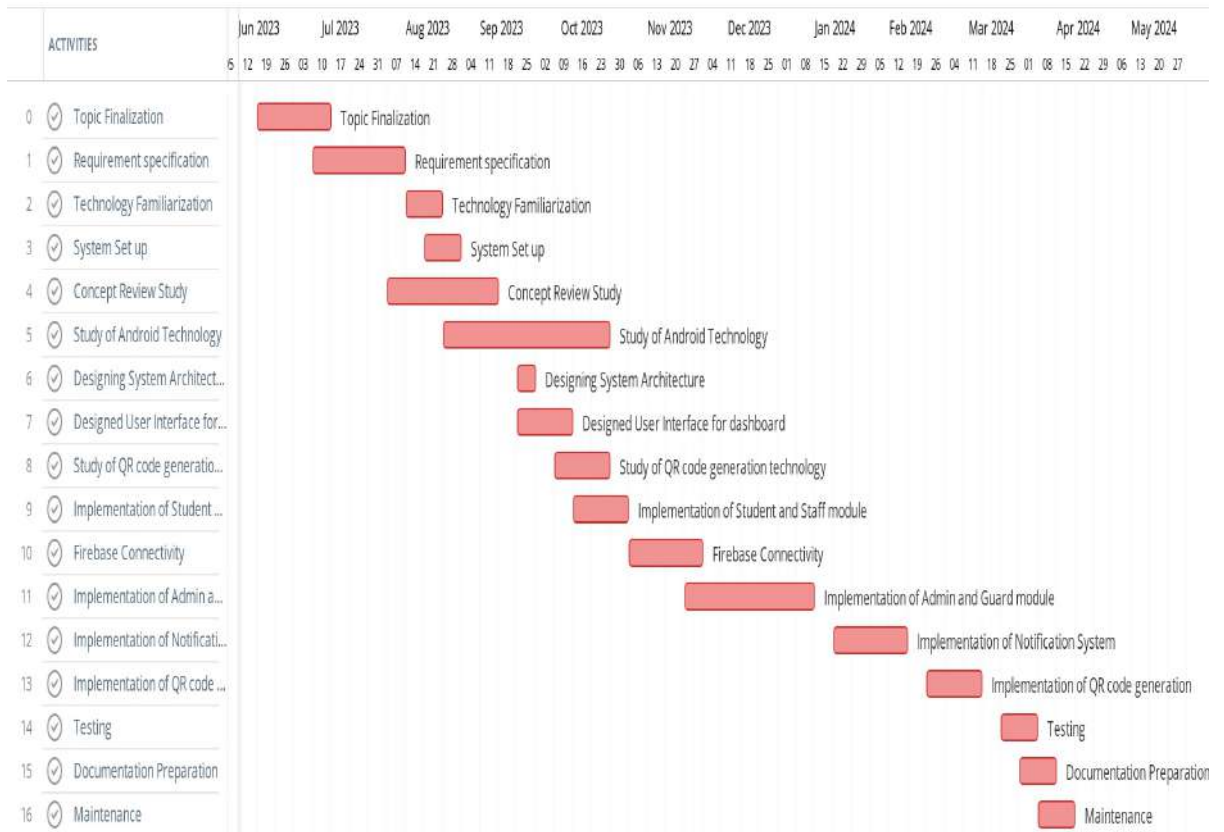


Figure 6.1: Timeline Diagram

Each task is assigned to one or more team members as shown in Fig:

Developer ID	Developer Name
D1	Omkar Jadhav
D2	Parag Shirsat
D3	Abhishek Soundankar
D4	Om Suryawanshi

Table 6.4: Lists of Developers

Chapter 7

Software Implementation

7.1 Introduction

To implement the Android app "DiG-Pass: Enhance and Secure Solution for Gate-Pass Using QR Code," we will break down the software implementation into several components:

1. **User Authentication:** Implement user authentication using the Firebase Authentication feature to allow users to sign up, and sign in the app securely.
2. **Gate-Pass Request Submission:** Allow Students as well as Staff to submit gate pass requests by entering reason, and leaving time through the app. Users can upload documents such as PDFs or images containing necessary information.
3. **Notification System:** Implement a notification system to notify users about the status of their gate pass requests. Notifications can include QR code attachments for approved requests or rejection notifications to users for denied requests.
4. **Gate-Pass Request Management:** Provide Admin with a separate user interface to manage gate pass requests. They can view, approve, or reject requests based on predefined criteria.
5. **QR Code Generation:** Generate a unique QR code after approving the gate pass request. The QR code should contain essential information such as user details, reason, leaving time, and a unique pass number.

7.2 Databases

Firebase Database is a real-time cloud-hosted NoSQL database offered by Google, commonly utilized for building mobile and web applications. In the context of the Android app DiG-Pass, which aims to enhance and secure gate-pass management using QR code technology, the Firebase Database serves as a crucial backend component. It facilitates seamless storage and retrieval of data related to gate passes, users, and their associated information. With Firebase's real-time synchronization capabilities, any updates made to the database are instantly reflected across all connected clients, ensuring consistency and reliability in data access. This feature is particularly advantageous for DiG-Pass, enabling quick verification of gate passes by scanning QR codes, as the database updates in real-time with each pass issuance or validation. Moreover, Firebase offers robust security features, including authentication and access control mechanisms, safeguarding sensitive information, and ensuring only authorized personnel can manage gate-pass data within the app. Overall, the Firebase Database plays a pivotal role in powering the functionality of DiG-Pass, offering scalability, real-time updates, and robust security for an efficient and secure gate-pass management solution.

7.3 Important module and algorithms

7.3.1 Modules

1. **Login Page:**

The login page provides a secure entry point for registered users to access the application by entering their credentials, such as email address and password. It authenticates users against the stored credentials in the Firebase Database, granting access upon successful verification.

2. **Sign-Up Page:**

The sign-up page allows new users to create an account within the "DiG-Pass" system. Users can input their personal details, such as photo, name, department, mobile no, email, and password, which are then stored securely in the Firebase Database. After successful registration, users can log in to the application.

3. **Manage User:**

This module enables the admin to manage user accounts and access permissions within the application. Admins can add new users, deactivate accounts, or update user information.

4. Manage Gate-Pass Requests:

The manage gate-pass requests module allows administrators to review and process gate-pass requests submitted by students and staff. Admins can review requests, approve or reject them based on predefined criteria, and reasons, and communicate the decision to users. This module streamlines the gate-pass issuance process and ensures efficient management of access permissions.

5. Gate-Pass History:

Gate-pass history provides users with a comprehensive record of their past gate-pass activities, including issued passes, access granted, and access denied instances. Users can view details such as gate pass numbers, date, and access logs, facilitating transparency and accountability in gate pass usage.

6. Edit profile:

The edit profile module enables users to update their personal information stored in the application's database. Users can modify details such as contact information, profile picture, or password, ensuring accurate and up-to-date user profiles within the Firebase database.

7. Gate-Pass Request Page:

The gate-pass request page provides students and staff with a user-friendly UI to submit requests for gate passes. Users can specify details such as the purpose of the pass, the time of leaving, and the file as proof of the purpose of the gate pass. After submission, the notification of the new gate pass will be sent to the admin. The request is forwarded to the admin for review and processing.

7.3.2 Algorithm

1. QR Code Generation Algorithm:

This algorithm generates unique QR codes for each gate pass issued by the app. It may involve encoding relevant information such as Gate-Pass Number, User details, the purpose of the gate pass, and access permissions into the QR code in a secure and efficient manner.

2. Encryption and Decryption Algorithm:

To enhance security, sensitive information stored within the app's database, such as User details and access logs, can be encrypted using encryption algorithms like AES (Advanced Encryption Standard). Decryption algorithms are then used to decrypt this information only when necessary, ensuring data confidentiality.

3. Data Synchronization Algorithm:

The app employs algorithms for synchronizing gate-pass data between the local device and the cloud database in real-time. This ensures that any updates or changes made to gate passes are propagated across all devices and users seamlessly.

7.4 Business logic

DiG-Pass project aims to revolutionize gate-pass management with its enhanced security features and seamless user experience. The core business logic of the Android app revolves around generating unique QR codes and scanning QR codes for gate-pass authentication. Upon user request, after the approval of the admin, the app generates a unique QR code containing encrypted information such as user identification and user details. This QR code serves as the digital pass for accessing designated areas. On the receiving end, a security guard equipped with the app scans the QR code to verify the QR codes and grant access accordingly. The app's backend system securely stores user data, access permissions, and pass history, ensuring accountability and traceability. Additionally, the app includes features for administrators to manage user accounts, and permissions, and generate comprehensive reports. By leveraging QR technology and robust security measures, the DiG-Pass project provides a reliable solution for enhancing gate-pass management while prioritizing user convenience and data protection.

Chapter 8

Software Testing

8.1 Introduction

Software testing must be started as early in the software development process as possible, and it must be integrated into the process of determining requirements. One stage of a lifecycle is testing. The lifecycle of software development is one in which we identify a requirement, write some code to address it, and then assess if we have satisfied the stakeholders, including the users, owners, and other parties with an interest in the product's functionality.

8.1.1 Test cases for Student Login Page

No	Behaviour Description	Property
1	Unique Test case ID	TC001
2	Test Case Name	Successful Registration
3	Prerequisites	Firebase Authentication initialized
4	Test Case Description	Enter valid Email and Password.
5	Input	Valid Email and Password
6	Expected Result	Successfully logged in and redirected to the dashboard
7	Actual Result	Successfully logged in and redirected to the dashboard
8	Pass/Fail	Pass

Table 8.1: Test Case for successful Login

No	Behaviour Description	Property
1	Unique Test case ID	TC002
2	Test Case Name	Unsuccessful Student Login with invalid Email
3	Prerequisites	Firestore Authentication initialized
4	Test Case Description	Enter invalid Email and valid Password
5	Input	Invalid Email
6	Expected Result	Error message indicating the invalid Email.
7	Actual Result	Error message indicating the invalid Email.
8	Pass/Fail	Pass

Table 8.2: Test case for Student Login with invalid Email

No	Behaviour Description	Property
1	Unique Test case ID	TC003
2	Test Case Name	Student Login with invalid Password
3	Prerequisites	Firestore Authentication initialized
4	Test Case Description	valid Email and invalid Password
5	Input	Invalid Password
6	Expected Result	error message indicating the invalid Password
7	Actual Result	Error message indicating the invalid Password
8	Pass/Fail	Pass

Table 8.3: Test case for invalid Password

No	Behaviour Description	Property
1	Unique Test case ID	TC004
2	Test Case Name	Student Login with empty fields
3	Prerequisites	Firestore Authentication initialized
4	Test Case Description	Leave Email and Password
5	Input	None
6	Expected Result	Error message to enter valid Email and Password.
7	Actual Result	Error message to enter valid Email and Password
8	Pass/Fail	Pass

Table 8.4: Test case for Student Login with empty fields

No	Behaviour Description	Property
1	Unique Test case ID	TC005
2	Test Case Name	Student Login with invalid credentials
3	Prerequisites	Firestore Authentication initialized
4	Test Case Description	login with invalid Email and Password
5	Input	Invalid Email and Password
6	Expected Result	Error message with invalid credentials.
7	Actual Result	Error message with invalid credentials.
8	Pass/Fail	Pass

Table 8.5: Test cases for Student Login with invalid credentials

No	Behaviour Description	Property
1	Unique Test case ID	TC006
2	Test Case Name	Student Login with empty Email
3	Prerequisites	Firestore Authentication initialized
4	Test Case Description	Login without entering Email
5	Input	Empty Email
6	Expected Result	Error Message "Enter Email."
7	Actual Result	Error Message "Enter Email."
8	Pass/Fail	Pass

Table 8.6: Test cases for Student Login with empty Email

No	Behaviour Description	Property
1	Unique Test case ID	TC007
2	Test Case Name	Student Login with empty Password
3	Prerequisites	Firestore Authentication initialized
4	Test Case Description	Login without entering Password
5	Input	Empty Password
6	Expected Result	Error Message "Enter Password"
7	Actual Result	Error Message "Enter Password"
8	Pass/Fail	Pass

Table 8.7: Test cases for Student Login with empty Password

8.1.2 Test cases for Student Sign-Up Page

No	Behaviour Description	Property
1	Unique Test case ID	TC008
2	Test Case Name	Student Sign-Up with valid data
3	Prerequisites	Student is not signed in
4	Test Case Description	Student Sign-Up with valid data
5	Input	Valid data with a selected image
6	Expected Result	Successfully registered and redirected to Login screen.
7	Actual Result	Successfully registered and redirected to Login screen.
8	Pass/Fail	Pass

Table 8.8: Test cases for Student Sign-Up with valid data

No	Behaviour Description	Property
1	Unique Test case ID	TC009
2	Test Case Name	Invalid Email format
3	Prerequisites	Student is not signed in
4	Test Case Description	Sign Up with an invalid Email format
5	Input	Invalid Email
6	Expected Result	Error message "Invalid Email format"
7	Actual Result	Error message "Invalid Email format"
8	Pass/Fail	Pass

Table 8.9: Test cases for Invalid Email

No	Behaviour Description	Property
1	Unique Test case ID	TC010
2	Test Case Name	Required fields
3	Prerequisites	Student is not signed in
4	Test Case Description	Sign Up without filling in all required fields
5	Input	Incomplete Details
6	Expected Result	Error messages for each missing field.
7	Actual Result	Error messages for each missing field.
8	Pass/Fail	Pass

Table 8.10: Test cases for Required fields

No	Behaviour Description	Property
1	Unique Test case ID	TC011
2	Test Case Name	Password length less than 6 characters
3	Prerequisites	Student is not signed in
4	Test Case Description	Sign Up with a Password less than 6 characters
5	Input	Password less than 6 characters
6	Expected Result	"Password should be at least 6 characters long"
7	Actual Result	"Password should be at least 6 characters long"
8	Pass/Fail	Pass

Table 8.11: Test cases for Password

No	Behaviour Description	Property
1	Unique Test case ID	TC012
2	Test Case Name	Sign Up with an Email that is already registered
3	Prerequisites	Student is not signed in
4	Test Case Description	Sign Up with a email that already registered
5	Input	Email that already registered with valid data
6	Expected Result	Error message "Email already registered"
7	Actual Result	Error message "Email already registered"
8	Pass/Fail	Pass

Table 8.12: Test cases for Duplicate email

8.1.3 Test Cases For Student Edit Profile Page

No	Behaviour Description	Property
1	Unique Test case ID	TC013
2	Test Case Name	Edit Student Profile with valid data
3	Prerequisites	Student is logged in
4	Test Case Description	Update the profile details
5	Input	Valid student data
6	Expected Result	"Student details updated successfully" message
7	Actual Result	"Student details updated successfully" message
8	Pass/Fail	Pass

Table 8.13: Test cases for Updating Student Profile

No	Behaviour Description	Property
1	Unique Test case ID	TC014
2	Test Case Name	Handling empty fields
3	Prerequisites	Student is logged in
4	Test Case Description	Update Profile with empty fields
5	Input	Empty or null data for one or more required fields
6	Expected Result	"Please fill all fields" message
7	Actual Result	"Please fill all fields" message
8	Pass/Fail	Pass

Table 8.14: Test cases for Empty fields

No	Behaviour Description	Property
1	Unique Test case ID	TC015
2	Test Case Name	Updating profile picture
3	Prerequisites	Student is logged in
4	Test Case Description	Updating profile picture
5	Input	Valid Image
6	Expected Result	"Student details updated successfully" message
7	Actual Result	"Student details updated successfully" message
8	Pass/Fail	Pass

Table 8.15: Test cases for Updating Profile Image

No	Behaviour Description	Property
1	Unique Test case ID	TC016
2	Test Case Name	Retrieving student data from Firebase
3	Prerequisites	Student is logged in
4	Test Case Description	Ensure student data is retrieved from Firebase correctly
5	Input	Valid student data stored in Firebase
6	Expected Result	Details retrieved from Firebase should be displayed
7	Actual Result	Details retrieved from Firebase should be displayed
8	Pass/Fail	Pass

Table 8.16: Test cases for Retrieving student data from Firebase

No	Behaviour Description	Property
1	Unique Test case ID	TC017
2	Test Case Name	Handling failed image upload to Firebase
3	Prerequisites	Student is logged in
4	Test Case Description	Handle failure while uploading a profile picture
5	Input	Invalid or corrupted image
6	Expected Result	Error message "failure to upload the profile picture"
7	Actual Result	Error message "failure to upload the profile picture"
8	Pass/Fail	Pass

Table 8.17: Test cases for Handling failed image upload to Firebase

No	Behaviour Description	Property
1	Unique Test case ID	TC018
2	Test Case Name	Retrieving student data from Firebase
3	Prerequisites	Student is logged in
4	Test Case Description	Ensure student data is retrieved from Firebase correctly
5	Input	Valid student data stored in Firebase
6	Expected Result	Details retrieved from Firebase should be displayed
7	Actual Result	Details retrieved from Firebase should be displayed
8	Pass/Fail	Pass

Table 8.18: Test cases for Retrieving student data from Firebase

8.2 Test Cases For Student Gate Pass Request Page

No	Behaviour Description	Property
1	Unique Test case ID	TC019
2	Test Case Name	Submit the form without uploading a File
3	Prerequisites	All required fields are filled with valid data
4	Test Case Description	Fill all required fields with valid data
5	Input	Valid data for all fields except the File Upload
6	Expected Result	"Form submitted successfully" message
7	Actual Result	"Form submitted successfully" message
8	Pass/Fail	Pass

Table 8.19: Test cases for Submit the form without File

No	Behaviour Description	Property
1	Unique Test case ID	TC020
2	Test Case Name	Submit the form with uploading a File
3	Prerequisites	All required fields are filled with valid data
4	Test Case Description	Enter all required data and select and upload file
5	Input	Valid data for all fields including the File
6	Expected Result	"Form submitted successfully" message
7	Actual Result	"Form submitted successfully" message
8	Pass/Fail	Pass

Table 8.20: Test cases for Submit the form with file

No	Behaviour Description	Property
1	Unique Test case ID	TC021
2	Test Case Name	Submit the form without entering Leaving Time
3	Prerequisites	All required fields except the Leaving Time
4	Test Case Description	Submit the form without entering Leaving Time
5	Input	Valid data for all fields except Leaving Time
6	Expected Result	"Enter Leaving Time" message
7	Actual Result	"Enter Leaving Time" message
8	Pass/Fail	Pass

Table 8.21: Test cases for Submit the form without Leaving Time

No	Behaviour Description	Property
1	Unique Test case ID	TC022
2	Test Case Name	Select "Yes" and enter Vehicle Number
3	Prerequisites	All required fields are filled with valid data
4	Test Case Description	Enter all required data and select and upload file
5	Input	Select "Yes" Vehicle option and enter the Vehicle Number
6	Expected Result	Vehicle Number field is visible, form submission proceeds.
7	Actual Result	Vehicle Number field is visible, form submission proceeds
8	Pass/Fail	Pass

Table 8.22: Test cases for Vehicle Option

No	Behaviour Description	Property
1	Unique Test case ID	TC023
2	Test Case Name	Submitting form without Reason
3	Prerequisites	All required fields are filled with valid data
4	Test Case Description	Submitting form without Reason
5	Input	Valid student data for all fields except Reason
6	Expected Result	Error message "Enter a Reason"
7	Actual Result	Error message "Enter a Reason"
8	Pass/Fail	Pass

Table 8.23: Test cases for Reason

No	Behaviour Description	Property
1	Unique Test case ID	TC024
2	Test Case Name	Upload a file with an invalid format
3	Prerequisites	All required fields are filled with valid data
4	Test Case Description	Upload a file with an invalid format
5	Input	File with invalid format
6	Expected Result	"Failed to upload file" message
7	Actual Result	"Failed to upload file" message
8	Pass/Fail	Fail

Table 8.24: Test cases Invalid file format

No	Behaviour Description	Property
1	Unique Test case ID	TC025
2	Test Case Name	Send Notification to Admin
3	Prerequisites	All required fields are filled with valid data
4	Test Case Description	Send Notification to Admin
5	Input	All valid data
6	Expected Result	"New Request Received" Notification will sent to Admin
7	Actual Result	"New Request Received" Notification will sent to Admin
8	Pass/Fail	Pass

Table 8.25: Test cases for Sending Notification

8.2.1 Test cases for Staff Login Page

No	Behaviour Description	Property
1	Unique Test case ID	TC026
2	Test Case Name	Login with Enter valid Email and Password
3	Prerequisites	Staff exists in Firebase Authentication
4	Test Case Description	Login with Enter valid Email and Password
5	Input	Valid Email and Password
6	Expected Result	"Login Successful" message should be displayed.
7	Actual Result	"Login Successful" message should be displayed.
8	Pass/Fail	Pass

Table 8.26: Test cases for Staff Login

No	Behaviour Description	Property
1	Unique Test case ID	TC027
2	Test Case Name	Login with empty Email field
3	Prerequisites	Staff exists in Firebase Authentication
4	Test Case Description	Login with empty Email field
5	Input	Empty Email field
6	Expected Result	"Enter Email" message should be displayed.
7	Actual Result	"Enter Email" message should be displayed.
8	Pass/Fail	Pass

Table 8.27: Test cases for Empty email field

No	Behaviour Description	Property
1	Unique Test case ID	TC028
2	Test Case Name	Login with empty password field
3	Prerequisites	Staff exists in Firebase Authentication
4	Test Case Description	Login with empty passwordfield
5	Input	Empty password field
6	Expected Result	"Enter Password" message should be displayed.
7	Actual Result	"Enter Password" message should be displayed.
8	Pass/Fail	Pass

Table 8.28: Test cases for Empty Password field

No	Behaviour Description	Property
1	Unique Test case ID	TC029
2	Test Case Name	Login with an invalid Email format
3	Prerequisites	Staff exists in Firebase Authentication
4	Test Case Description	Login with invalid Email
5	Input	Invalid Email
6	Expected Result	"Invalid Email" message should be displayed.
7	Actual Result	"Invalid Email" message should be displayed.
8	Pass/Fail	Pass

Table 8.29: Test cases for Invalid Email

No	Behaviour Description	Property
1	Unique Test case ID	TC030
2	Test Case Name	Login with incorrect Email and Password
3	Prerequisites	Staff exists in Firebase Authentication
4	Test Case Description	Login with incorrect Email and Password
5	Input	Incorrect Email and Password
6	Expected Result	"Invalid Credentials" message should be displayed.
7	Actual Result	"Invalid Credentials" message should be displayed.
8	Pass/Fail	Pass

Table 8.30: Test cases for Incorrect Credentials

8.2.2 Test cases for Staff Sign-Up Page

No	Behaviour Description	Property
1	Unique Test case ID	TC031
2	Test Case Name	Staff Sign-Up with valid data
3	Prerequisites	Staff is not signed in
4	Test Case Description	Staff Sign-Up with valid data
5	Input	Valid data with a selected image
6	Expected Result	Successfully registered and redirected to Login screen.
7	Actual Result	Successfully registered and redirected to Login screen.
8	Pass/Fail	Pass

Table 8.31: Test cases for Staff Sign-Up with valid data

No	Behaviour Description	Property
1	Unique Test case ID	TC032
2	Test Case Name	Invalid Email format
3	Prerequisites	Staff is not signed in
4	Test Case Description	Sign Up with an invalid Email format
5	Input	Invalid Email
6	Expected Result	Error message "Invalid Email format"
7	Actual Result	Error message "Invalid Email format"
8	Pass/Fail	Pass

Table 8.32: Test cases for Invalid Email

No	Behaviour Description	Property
1	Unique Test case ID	TC033
2	Test Case Name	Required fields
3	Prerequisites	Staff is not signed in
4	Test Case Description	Sign Up without filling in all required fields
5	Input	Incomplete Details
6	Expected Result	Error messages for each missing field.
7	Actual Result	Error messages for each missing field.
8	Pass/Fail	Pass

Table 8.33: Test cases for Required fields

No	Behaviour Description	Property
1	Unique Test case ID	TC034
2	Test Case Name	Password length less than 6 characters
3	Prerequisites	Student is not signed in
4	Test Case Description	Sign Up with a Password less than 6 characters
5	Input	Password less than 6 characters
6	Expected Result	"Password should be at least 6 characters long"
7	Actual Result	"Password should be at least 6 characters long"
8	Pass/Fail	Pass

Table 8.34: Test cases for Password

No	Behaviour Description	Property
1	Unique Test case ID	TC035
2	Test Case Name	Sign Up with an Email that is already registered
3	Prerequisites	Staff is not signed in
4	Test Case Description	Sign Up with a email that already registered
5	Input	Email that already registered with valid data
6	Expected Result	Error message "Email already registered"
7	Actual Result	Error message "Email already registered"
8	Pass/Fail	Pass

Table 8.35: Test cases for Duplicate email

8.2.3 Test Cases For Staff Edit Profile Page

No	Behaviour Description	Property
1	Unique Test case ID	TC036
2	Test Case Name	Edit Staff Profile with valid data
3	Prerequisites	Staff is logged in
4	Test Case Description	Update the profile details
5	Input	Valid Staff data
6	Expected Result	"Staff details updated successfully" message
7	Actual Result	"Staff details updated successfully" message
8	Pass/Fail	Pass

Table 8.36: Test cases for Updating Staff Profile

No	Behaviour Description	Property
1	Unique Test case ID	TC037
2	Test Case Name	Handling empty fields
3	Prerequisites	Staff is logged in
4	Test Case Description	Update Profile with empty fields
5	Input	Empty or null data for one or more required fields
6	Expected Result	"Please fill all fields" message
7	Actual Result	"Please fill all fields" message
8	Pass/Fail	Pass

Table 8.37: Test cases for Empty fields

No	Behaviour Description	Property
1	Unique Test case ID	TC038
2	Test Case Name	Updating profile picture
3	Prerequisites	Staff is logged in
4	Test Case Description	Updating profile picture
5	Input	Valid Image
6	Expected Result	"Staff details updated successfully" message
7	Actual Result	"Staff details updated successfully" message
8	Pass/Fail	Pass

Table 8.38: Test cases for Updating Profile Image

No	Behaviour Description	Property
1	Unique Test case ID	TC039
2	Test Case Name	Retrieving Staff data from Firebase
3	Prerequisites	Staff is logged in
4	Test Case Description	Ensure Staff data is retrieved from Firebase correctly
5	Input	Valid Staff data stored in Firebase
6	Expected Result	Details retrieved from Firebase should be displayed
7	Actual Result	Details retrieved from Firebase should be displayed
8	Pass/Fail	Pass

Table 8.39: Test cases for Retrieving Staff data from Firebase

No	Behaviour Description	Property
1	Unique Test case ID	TC040
2	Test Case Name	Handling failed image upload to Firebase
3	Prerequisites	Staff is logged in
4	Test Case Description	Handle failure while uploading a profile picture
5	Input	Invalid or corrupted image
6	Expected Result	Error message "failure to upload the profile picture"
7	Actual Result	Error message "failure to upload the profile picture"
8	Pass/Fail	Pass

Table 8.40: Test cases for Handling failed image upload to Firebase

No	Behaviour Description	Property
1	Unique Test case ID	TC041
2	Test Case Name	Retrieving Staff data from Firebase
3	Prerequisites	Staff is logged in
4	Test Case Description	Ensure Staff data is retrieved from Firebase correctly
5	Input	Valid Staff data stored in Firebase
6	Expected Result	Details retrieved from Firebase should be displayed
7	Actual Result	Details retrieved from Firebase should be displayed
8	Pass/Fail	Pass

Table 8.41: Test cases for Retrieving Staff data from Firebase

8.2.4 Test Cases For Staff Gate-Pass Request Page

No	Behaviour Description	Property
1	Unique Test case ID	TC042
2	Test Case Name	Submit the form without uploading a File
3	Prerequisites	All required fields are filled with valid data
4	Test Case Description	Fill all required fields with valid data
5	Input	Valid data for all fields except the File Upload
6	Expected Result	"Form submitted successfully" message
7	Actual Result	"Form submitted successfully" message
8	Pass/Fail	Pass

Table 8.42: Test cases for Submit the form without File

No	Behaviour Description	Property
1	Unique Test case ID	TC043
2	Test Case Name	Submit the form with uploading a File
3	Prerequisites	All required fields are filled with valid data
4	Test Case Description	Enter all required data and select and upload file
5	Input	Valid data for all fields including the File
6	Expected Result	"Form submitted successfully" message
7	Actual Result	"Form submitted successfully" message
8	Pass/Fail	Pass

Table 8.43: Test cases for Submit the form with file

No	Behaviour Description	Property
1	Unique Test case ID	TC044
2	Test Case Name	Submit the form without entering Leaving Time
3	Prerequisites	All required fields except the Leaving Time
4	Test Case Description	Submit the form without entering Leaving Time
5	Input	Valid data for all fields except Leaving Time
6	Expected Result	"Enter Leaving Time" message
7	Actual Result	"Enter Leaving Time" message
8	Pass/Fail	Pass

Table 8.44: Test cases for Submit the form without Leaving Time

No	Behaviour Description	Property
1	Unique Test case ID	TC045
2	Test Case Name	Select "Yes" Vehicle option and enter Vehicle Number
3	Prerequisites	All required fields are filled with valid data
4	Test Case Description	Enter all required data and select and upload file
5	Input	Select "Yes" Vehicle option and enter the Vehicle Number
6	Expected Result	Vehicle Number field is visible, form submission proceeds.
7	Actual Result	Vehicle Number field is visible, form submission proceeds
8	Pass/Fail	Pass

Table 8.45: Test cases for Vehicle Option

No	Behaviour Description	Property
1	Unique Test case ID	TC046
2	Test Case Name	Submitting form without Reason
3	Prerequisites	All required fields are filled with valid data
4	Test Case Description	Submitting form without Reason
5	Input	Valid student data for all fields except Reason
6	Expected Result	Error message "Enter a Reason"
7	Actual Result	Error message "Enter a Reason"
8	Pass/Fail	Pass

Table 8.46: Test cases for Reason

No	Behaviour Description	Property
1	Unique Test case ID	TC047
2	Test Case Name	Upload a file with an invalid format
3	Prerequisites	All required fields are filled with valid data
4	Test Case Description	Upload a file with an invalid format
5	Input	File with invalid format
6	Expected Result	"Failed to upload file" message
7	Actual Result	"Failed to upload file" message
8	Pass/Fail	Fail

Table 8.47: Test cases Invalid file format

No	Behaviour Description	Property
1	Unique Test case ID	TC048
2	Test Case Name	Send Notification to Admin
3	Prerequisites	All required fields are filled with valid data
4	Test Case Description	Send Notification to Admin
5	Input	All valid data
6	Expected Result	"New Request Received" Notification will sent to Admin
7	Actual Result	"New Request Received" Notification will sent to Admin
8	Pass/Fail	Pass

Table 8.48: Test cases for Sending Notification

8.2.5 Test cases for Guard Login Page

No	Behaviour Description	Property
1	Unique Test case ID	TC049
2	Test Case Name	Login with Enter valid Email and Password
3	Prerequisites	None
4	Test Case Description	Login with Enter valid Email and Password
5	Input	Valid Email and Password
6	Expected Result	"Login Successful" message should be displayed.
7	Actual Result	"Login Successful" message should be displayed.
8	Pass/Fail	Pass

Table 8.49: Test cases for Guard Login

No	Behaviour Description	Property
1	Unique Test case ID	TC050
2	Test Case Name	Login with empty Email field
3	Prerequisites	None
4	Test Case Description	Login with empty Email field
5	Input	Empty Email field
6	Expected Result	"Enter Email" message should be displayed.
7	Actual Result	"Enter Email" message should be displayed.
8	Pass/Fail	Pass

Table 8.50: Test cases for Empty email field

No	Behaviour Description	Property
1	Unique Test case ID	TC051
2	Test Case Name	Login with empty password field
3	Prerequisites	None
4	Test Case Description	Login with empty passwordfield
5	Input	Empty password field
6	Expected Result	"Enter Password" message should be displayed.
7	Actual Result	"Enter Password" message should be displayed.
8	Pass/Fail	Pass

Table 8.51: Test cases for Empty Password field

No	Behaviour Description	Property
1	Unique Test case ID	TC052
2	Test Case Name	Login with an invalid Email format
3	Prerequisites	None
4	Test Case Description	Login with invalid Email
5	Input	Invalid Email
6	Expected Result	"Invalid Email" message should be displayed.
7	Actual Result	"Invalid Email" message should be displayed.
8	Pass/Fail	Pass

Table 8.52: Test cases for Invalid Email

No	Behaviour Description	Property
1	Unique Test case ID	TC053
2	Test Case Name	Login with incorrect Email and Password
3	Prerequisites	None
4	Test Case Description	Login with incorrect Email and Password
5	Input	Incorrect Email and Password
6	Expected Result	"Invalid Credentials" message should be displayed.
7	Actual Result	"Invalid Credentials" message should be displayed.
8	Pass/Fail	Pass

Table 8.53: Test cases for Incorrect Credentials

No	Behaviour Description	Property
1	Unique Test case ID	TC054
2	Test Case Name	Login with incorrect Email and Password
3	Prerequisites	None
4	Test Case Description	Login with incorrect Email and Password
5	Input	Incorrect Email and Password
6	Expected Result	"Invalid Credentials" message should be displayed.
7	Actual Result	"Invalid Credentials" message should be displayed.
8	Pass/Fail	Pass

Table 8.54: Test cases for Incorrect Credentials

8.2.6 Test cases for Admin Login Page

No	Behaviour Description	Property
1	Unique Test case ID	TC055
2	Test Case Name	Login with Enter valid Email and Password
3	Prerequisites	Admin exists in Firebase Authentication
4	Test Case Description	Login with Enter valid Email and Password
5	Input	Valid Email and Password
6	Expected Result	"Login Successful" message should be displayed.
7	Actual Result	"Login Successful" message should be displayed.
8	Pass/Fail	Pass

Table 8.55: Test cases for Admin Login

No	Behaviour Description	Property
1	Unique Test case ID	TC056
2	Test Case Name	Login with empty Email field
3	Prerequisites	Admin exists in Firebase Authentication
4	Test Case Description	Login with empty Email field
5	Input	Empty Email field
6	Expected Result	"Enter Email" message should be displayed.
7	Actual Result	"Enter Email" message should be displayed.
8	Pass/Fail	Pass

Table 8.56: Test cases for Empty email field

No	Behaviour Description	Property
1	Unique Test case ID	TC057
2	Test Case Name	Login with empty password field
3	Prerequisites	Admin exists in Firebase Authentication
4	Test Case Description	Login with empty passwordfield
5	Input	Empty password field
6	Expected Result	"Enter Password" message should be displayed.
7	Actual Result	"Enter Password" message should be displayed.
8	Pass/Fail	Pass

Table 8.57: Test cases for Empty Password field

No	Behaviour Description	Property
1	Unique Test case ID	TC058
2	Test Case Name	Login with an invalid Email format
3	Prerequisites	Admin exists in Firebase Authentication
4	Test Case Description	Login with invalid Email
5	Input	Invalid Email
6	Expected Result	"Invalid Email" message should be displayed.
7	Actual Result	"Invalid Email" message should be displayed.
8	Pass/Fail	Pass

Table 8.58: Test cases for Invalid Email

No	Behaviour Description	Property
1	Unique Test case ID	TC059
2	Test Case Name	Login with incorrect Email and Password
3	Prerequisites	Admin exists in Firebase Authentication
4	Test Case Description	Login with incorrect Email and Password
5	Input	Incorrect Email and Password
6	Expected Result	"Invalid Credentials" message should be displayed.
7	Actual Result	"Invalid Credentials" message should be displayed.
8	Pass/Fail	Pass

Table 8.59: Test cases for Incorrect Credentials

8.2.7 Test cases for Manage Requests Page

No	Behaviour Description	Property
1	Unique Test case ID	TC060
2	Test Case Name	Fetch Requests Successfully
3	Prerequisites	Firebase containing details and requests
4	Test Case Description	Verify that requests are fetched successfully from Firebase
5	Input	None
6	Expected Result	Requests are fetched and displayed in the list
7	Actual Result	Requests are fetched and displayed in the list
8	Pass/Fail	Pass

Table 8.60: Test cases for Displaying the gate-pass requests

No	Behaviour Description	Property
1	Unique Test case ID	TC061
2	Test Case Name	Unable to Fetch Requests
3	Prerequisites	Firebase containing details and requests
4	Test Case Description	Verify handling of failure to fetch requests
5	Input	None
6	Expected Result	Toast message displays "Failed to retrieve data"
7	Actual Result	Toast message displays "Failed to retrieve data"
8	Pass/Fail	Fail

Table 8.61: Test cases for handling failure of fetching details

No	Behaviour Description	Property
1	Unique Test case ID	TC062
2	Test Case Name	Accept Gate-pass Request
3	Prerequisites	Gate pass request available in the list
4	Test Case Description	Verify acceptance of gate pass request
5	Input	Click on "Accept" button
6	Expected Result	"Your Request is accepted" Notification, QR sent to user
7	Actual Result	"Your Request is accepted" Notification , QR sent to user
8	Pass/Fail	Pass

Table 8.62: Test cases for Accepting the gate-pass request

No	Behaviour Description	Property
1	Unique Test case ID	TC063
2	Test Case Name	Reject Gate-pass Request
3	Prerequisites	Gate pass request available in the list
4	Test Case Description	Verify rejection of gate pass request
5	Input	Click on "Reject" button
6	Expected Result	Notification "Your Request is Rejected" is sent.
7	Actual Result	Notification "Your Request is Rejected" is sent.
8	Pass/Fail	Pass

Table 8.63: Test cases for Rejecting the gate-pass request

8.3 Snapshots of Test Cases

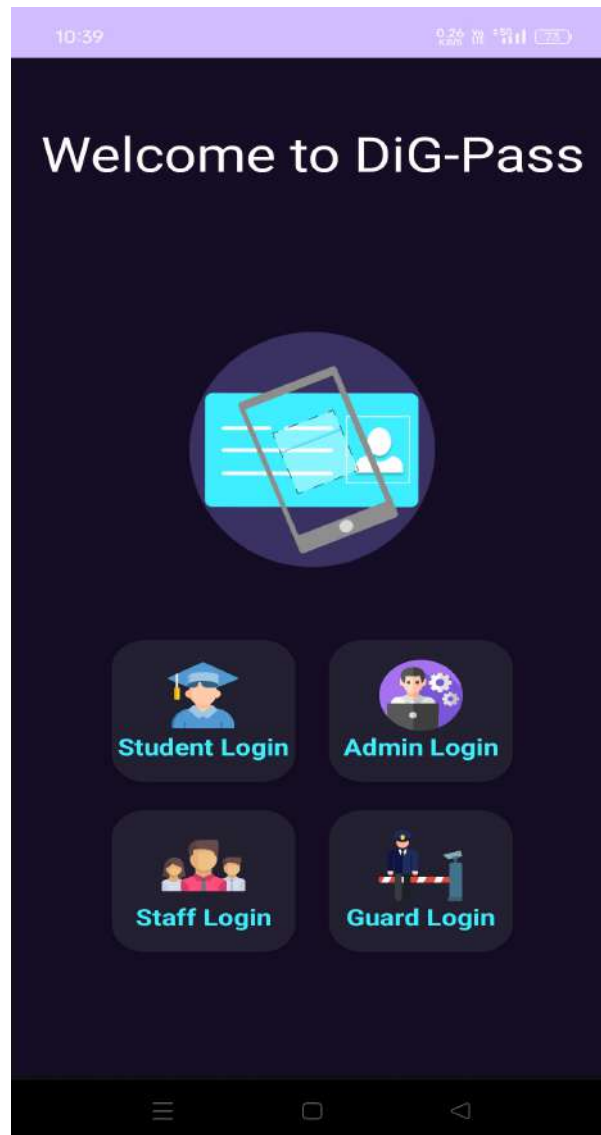


Figure 8.1: DiG-Pass Dashboard

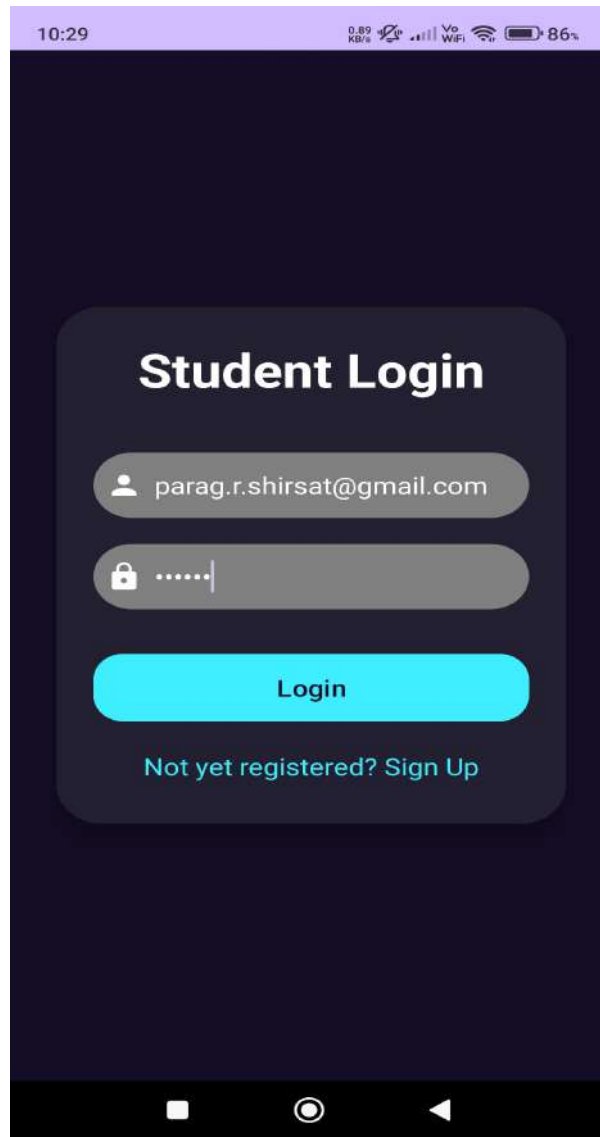


Figure 8.2: Student Login Page

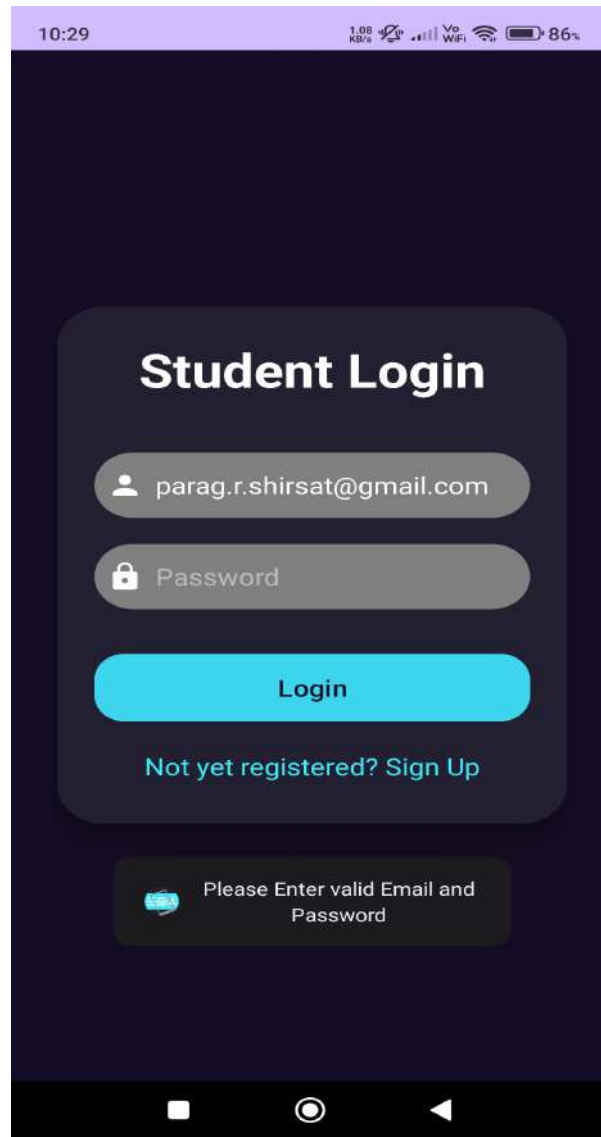





Figure 8.3: Student Login Page with Validation


11:40 0.00 KB/s 78%

Sign Up





 paragshirsat22@gmail.com



 Demo

46 B

BE CO

 7219129198  7219129198

Sign Up

[Already an user? Login](#)

Figure 8.4: Student Sign-Up Page

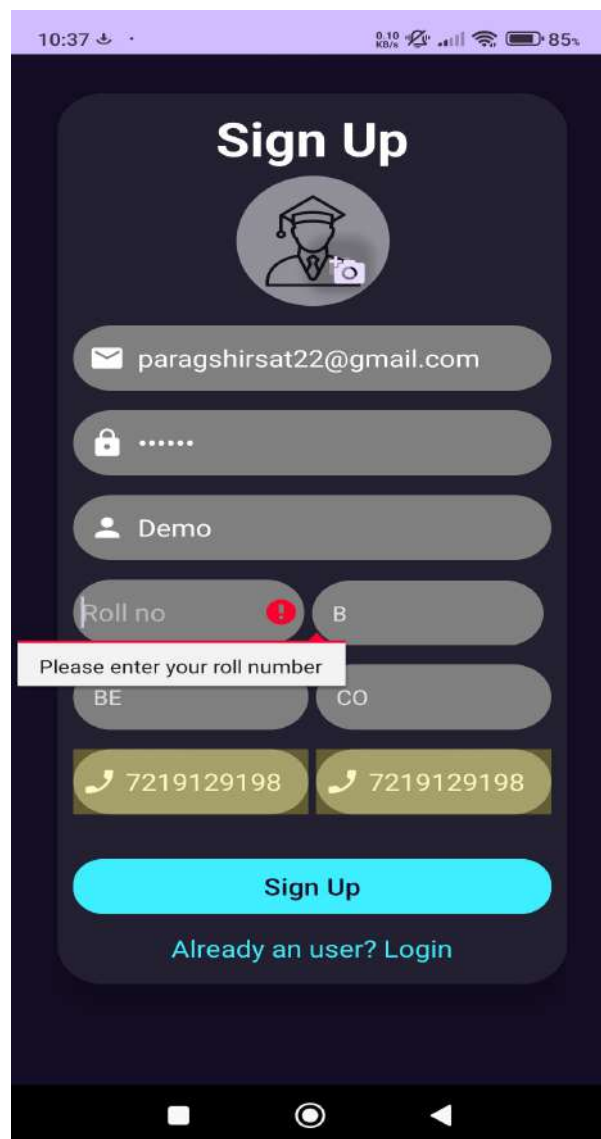





Figure 8.5: Student Sign-Up Page with Error Message



10:37 1.07 KB/s Wi-Fi 85%

Sign Up



 paragshirsat22@gmail.com





 Name 

Please enter your name

46 B

BE CO

 7219129198  7219129198

Sign Up

Already an user? [Login](#)

Figure 8.6: Student Sign-Up Page with Error Message

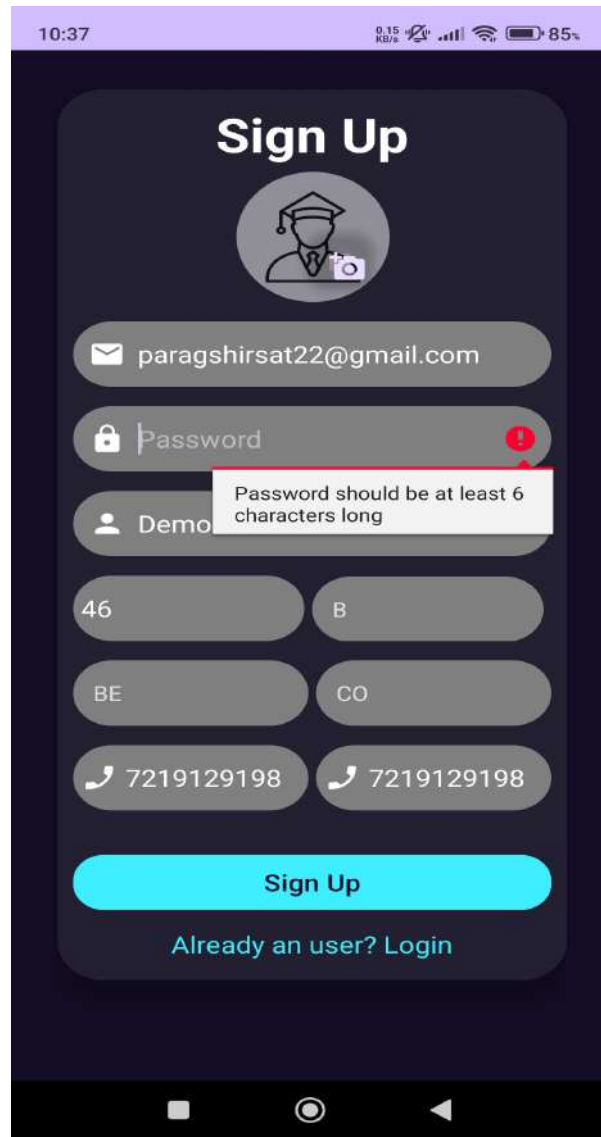


Figure 8.7: Student Sign-Up Page with Error Message

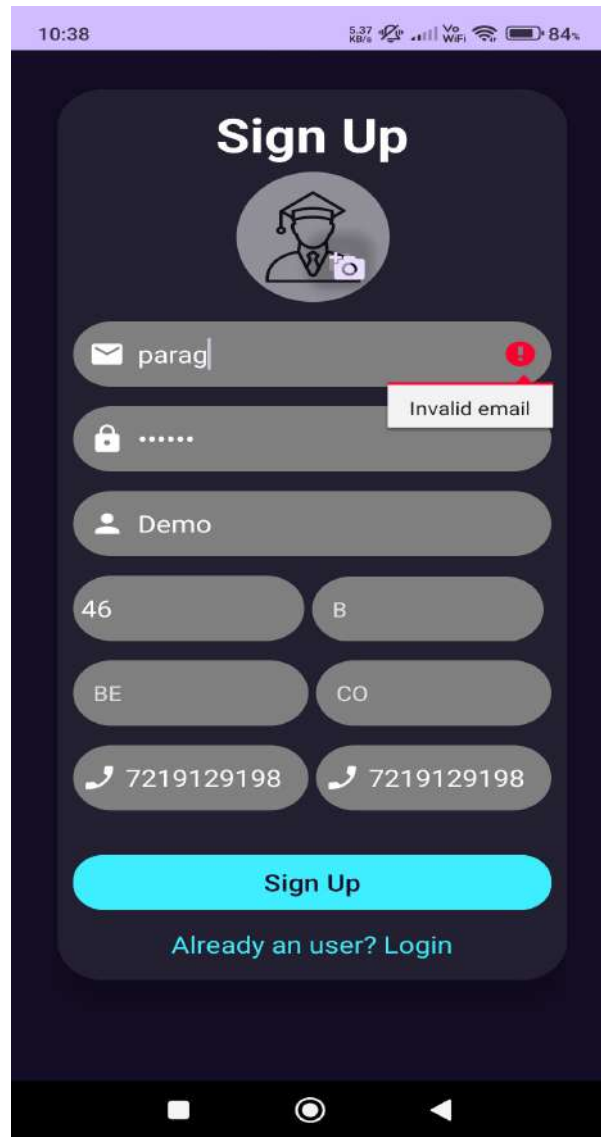


Figure 8.8: Student Sign-Up Page with Error Message

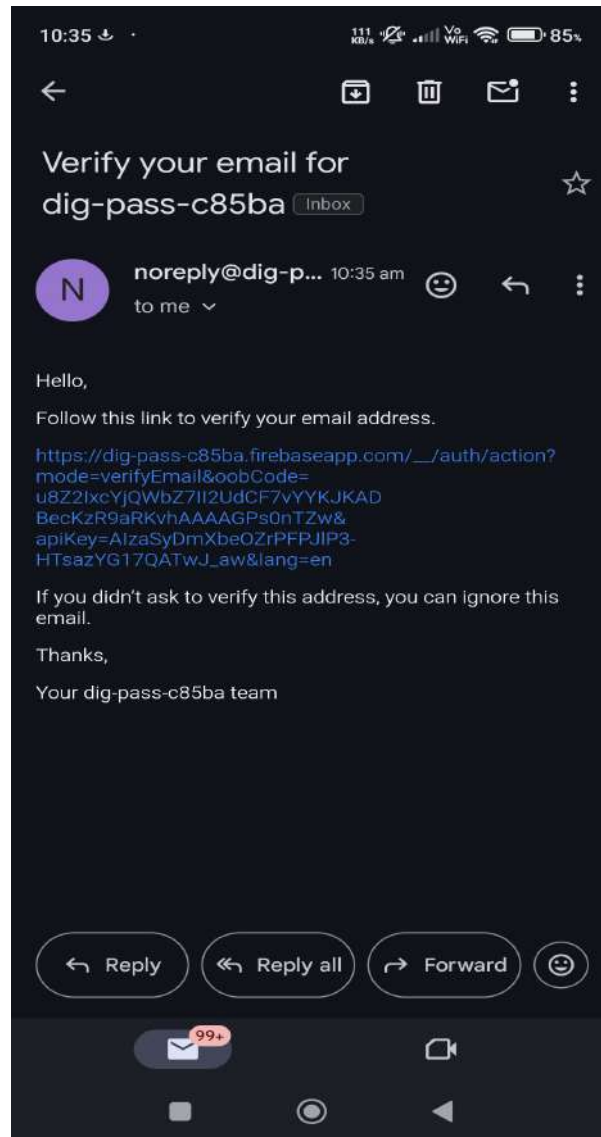


Figure 8.9: Verify User Email

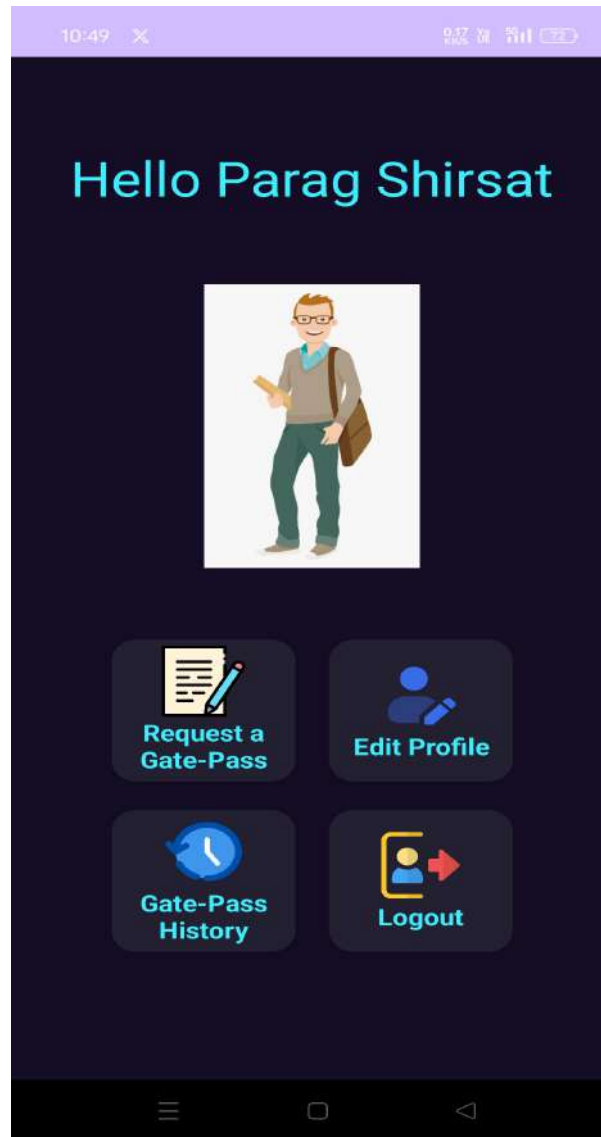


Figure 8.10: Student Dashboard

The screenshot displays a mobile application interface for requesting a student gate-pass. The app is titled "Student Gate-Pass" and shows the date as 26/05/2024 and the pass number as 5. The form includes the following fields and controls:

- Email: parag.r.shirsat@gmail.com
- Name: Parag Shirsat
- Roll Number: 46
- Branch: B
- Semester: BE
- Course: CO
- Phone Number: 7219129198
- Medical Emergency: Medical emergency |
- Time: 12:15
- Vehicle Status: Do you have a Vehicle? (Yes/No)
- Buttons: Upload File, Submit
- File Name: (empty)

Figure 8.11: Student Gate-Pass Request

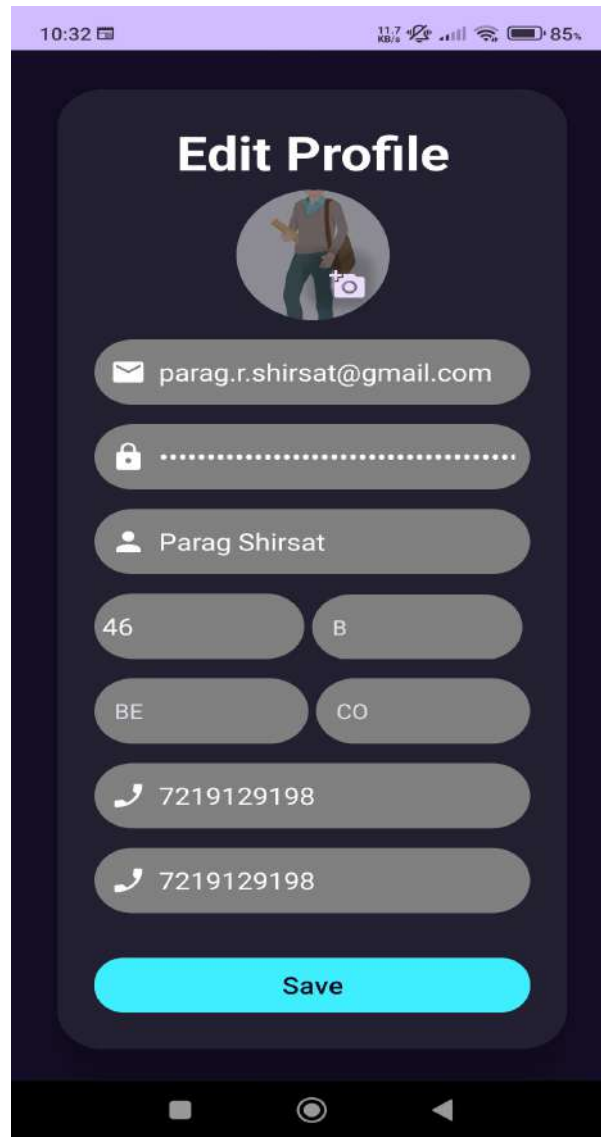


Figure 8.12: Edit Profile Page

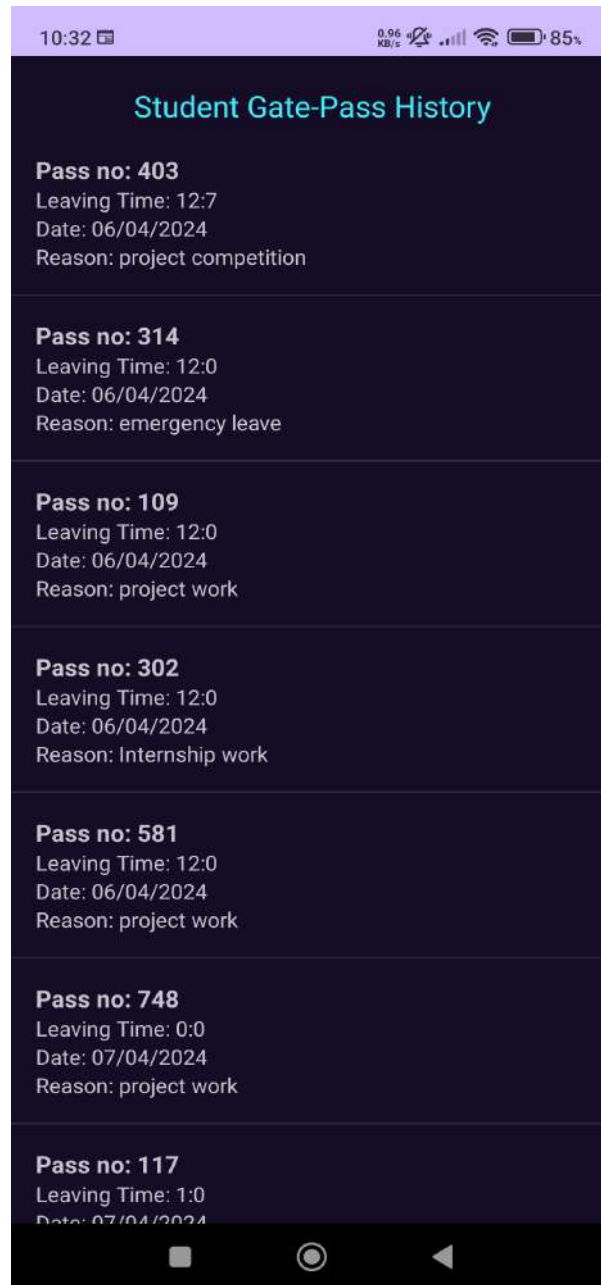


Figure 8.13: Gate-Pass History

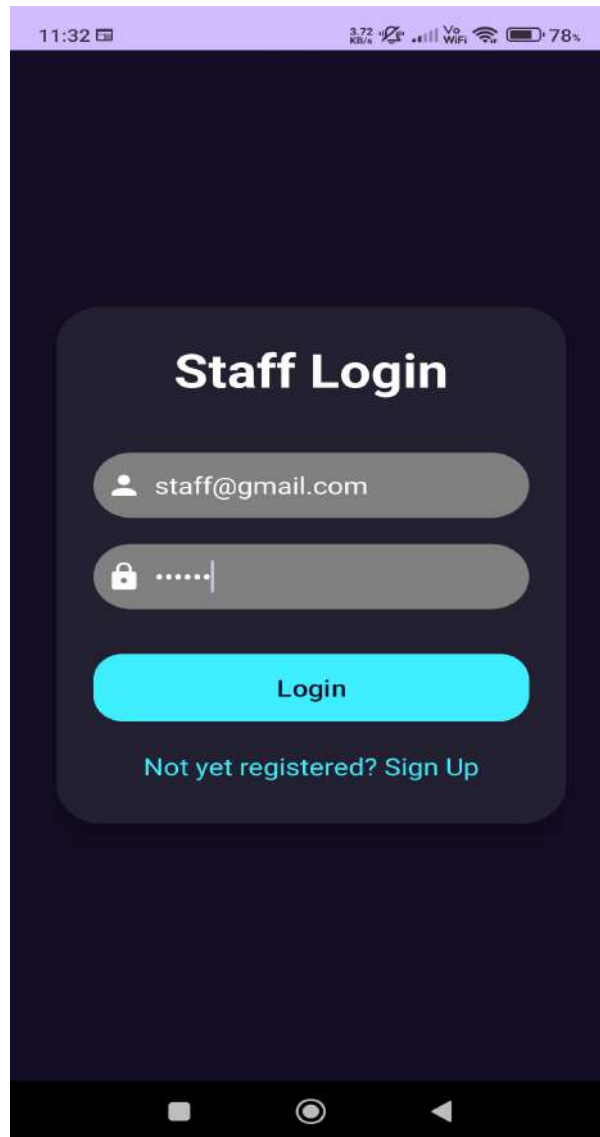


Figure 8.14: Staff Login Page

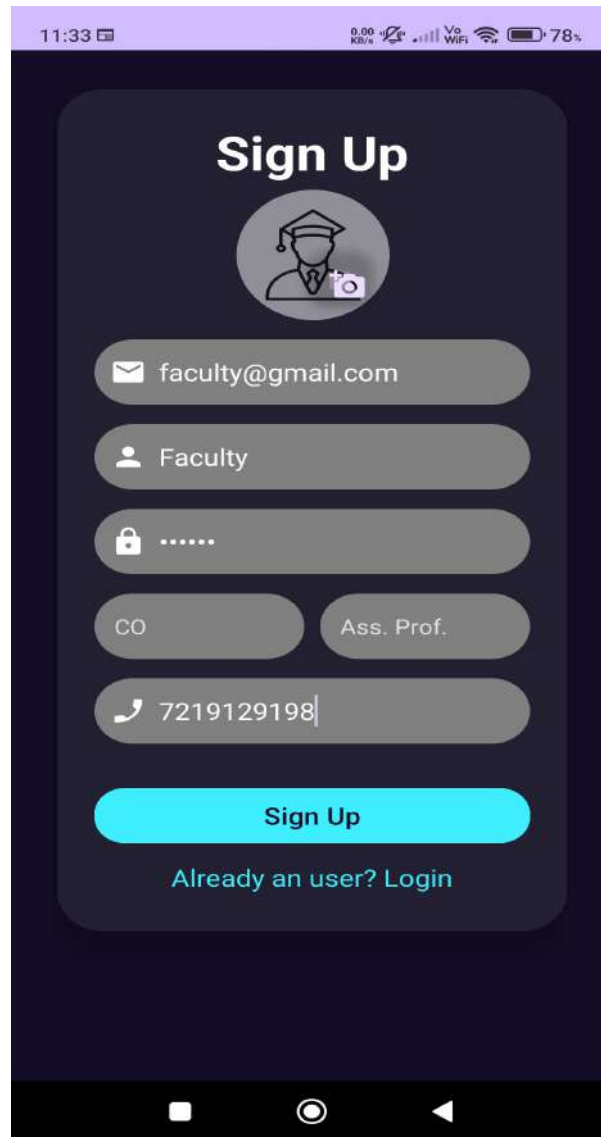


Figure 8.15: Staff Sign-Up Page

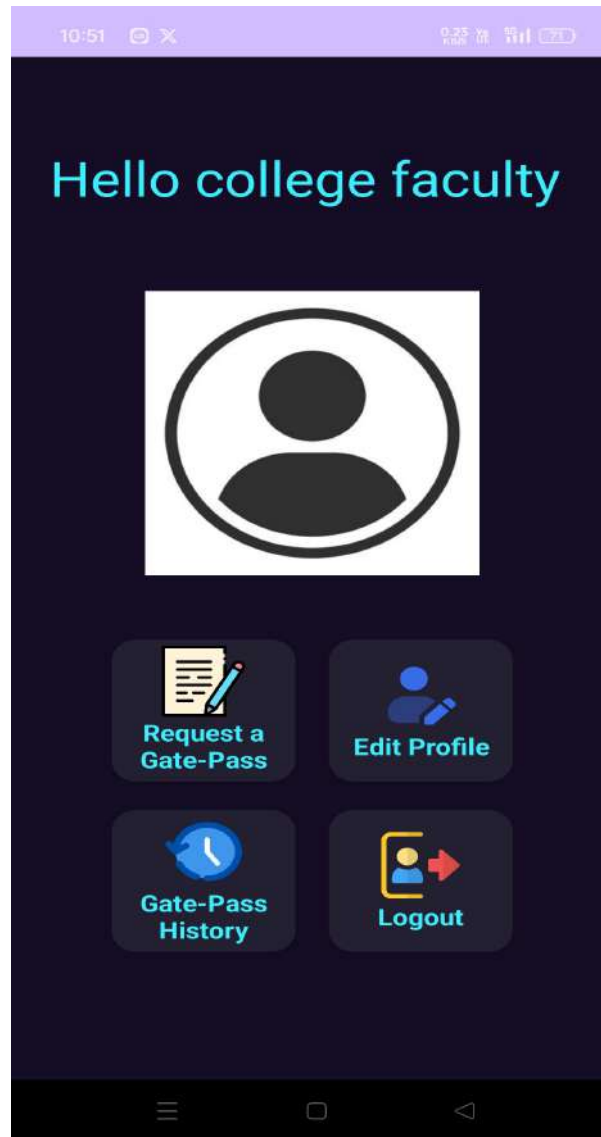


Figure 8.16: Staff Dashboard Page

The image shows a mobile application interface for requesting a staff gate-pass. The form is titled "Staff Gate-Pass" and includes the following fields and options:

- Date:** 26/05/2024
- Pass no:** 516
- Email:** staff@gmail.com
- Role:** college faculty
- Phone:** 7219129198
- Designation:** CO and Prof. (radio buttons)
- Reason:** Appointment in hospital (with a menu icon)
- Start Time:** 11:00 (with a "Set" button)
- End Time:** 15:00 (with a "Set" button)
- Vehicle:** Do you have a Vehicle (Yes/No radio buttons, with "No" selected)
- File Upload:** Upload File button (with a paperclip icon)
- Submit:** Submit button
- File Name:** (label below the upload button)

Figure 8.17: Staff Gate-Pass Request

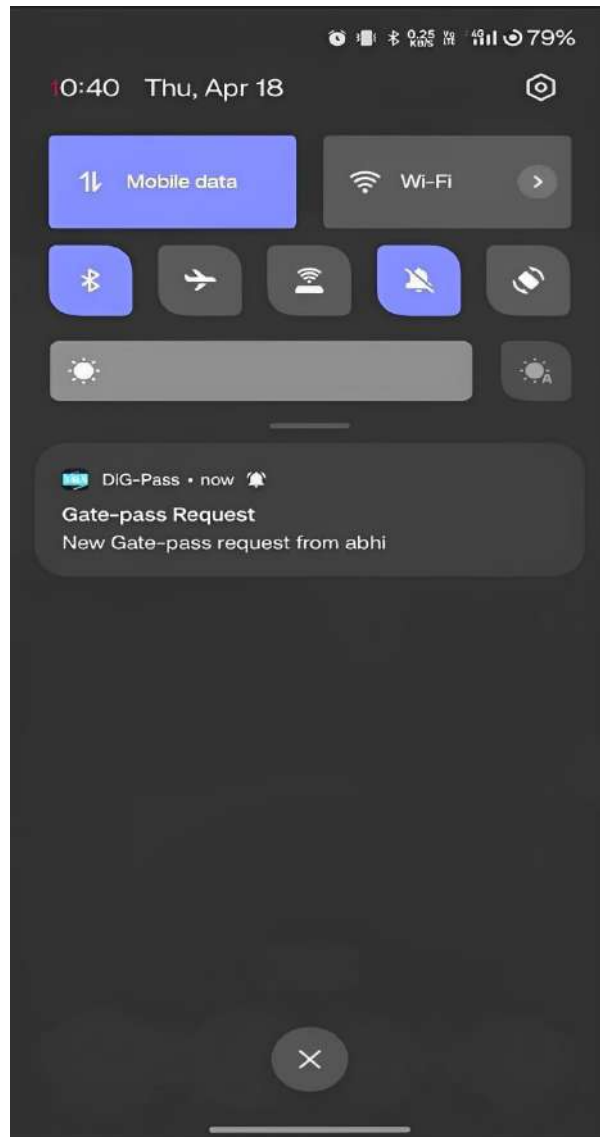


Figure 8.18: Notification to Admin



Figure 8.19: Request Details



Figure 8.20: QR Code Generation

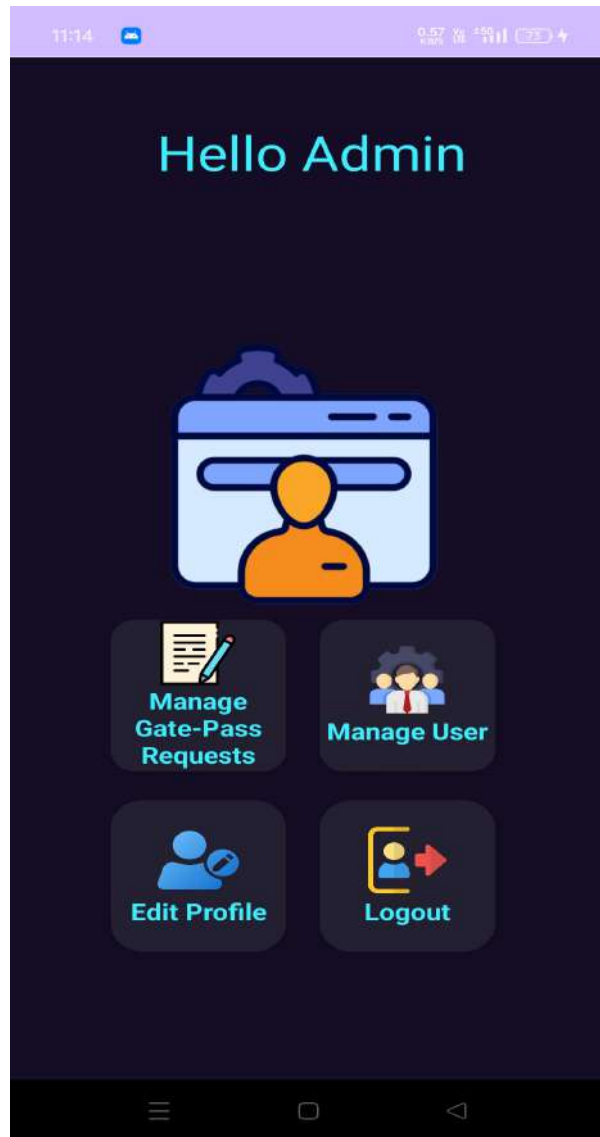





Figure 8.21: Admin Dashboard


10:43 0.18 KB/s 84%

Create Admin




 admin@gmail.com

 Admin



HOD

CE

 7219129198

Create

Figure 8.22: Create Admin Page

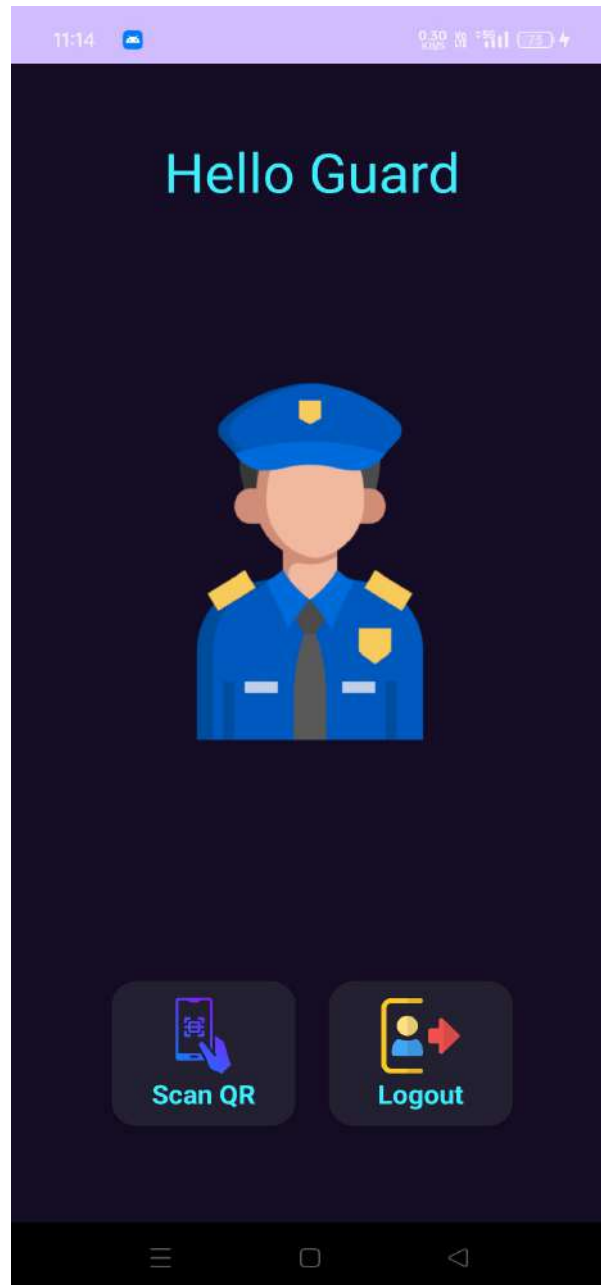


Figure 8.23: Guard Dashboard



Figure 8.24: Scan QR Code

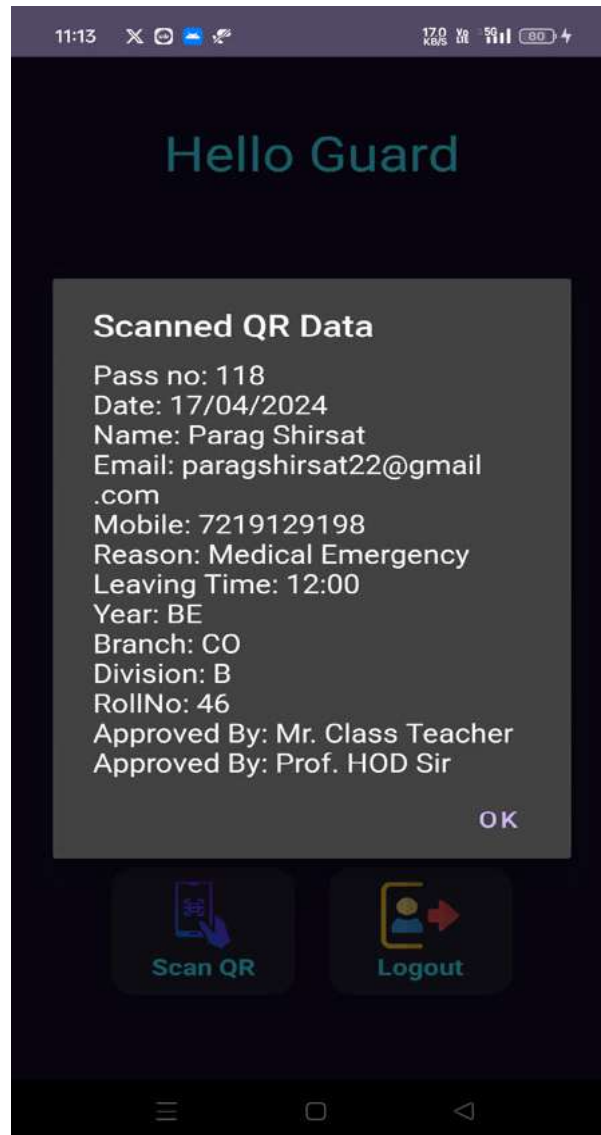


Figure 8.25: QR Code Details

Chapter 9

Result

9.1 Result

The outcomes demonstrate the efficacy and efficiency of the functionality of the Android app "DiG-Pass" in gate-pass management using QR codes within the framework of the tasks and time allocations given. Users may easily register and utilize the site's features because the Sign-Up process is quick. To guarantee seamless user interaction and administrative control, gate-pass requests, user authentication, and admin processes' acceptance or rejection of the requests are all completed in a timely manner. Additionally, it improves user experience because QR codes are generated quickly and validated. While quick QR code validation occurs during a guard's departure from campus, the quick creation of unique QR codes upon request approval guarantees that users can access their gate passes. Conversations about these findings might center on how well the app works to improve security, shorten wait times, and streamline gate-pass management procedures. The app's efficiency in operation is demonstrated by the speed at which tasks like creating QR codes and sending notifications are completed. It also highlights the app's dedication to user convenience without sacrificing security how quickly user authentication and QR code validation can be completed. Thus, the "DiG-Pass" system's overall performance fulfills its promise of giving users an improved and safe way to manage their gate passes via QR codes.

Time taken for Gate-Pass Process:

Task	Time Taken in sec
Sign Up	20
User Authentication	5
Make a Gate-Pass Request	10
Send Notification to Admin	5
Accept/Reject the Request	20
Generate the QR code	5
Send Notification to User	5
QR Code Validation	10

Table 9.1: Performance Matrix

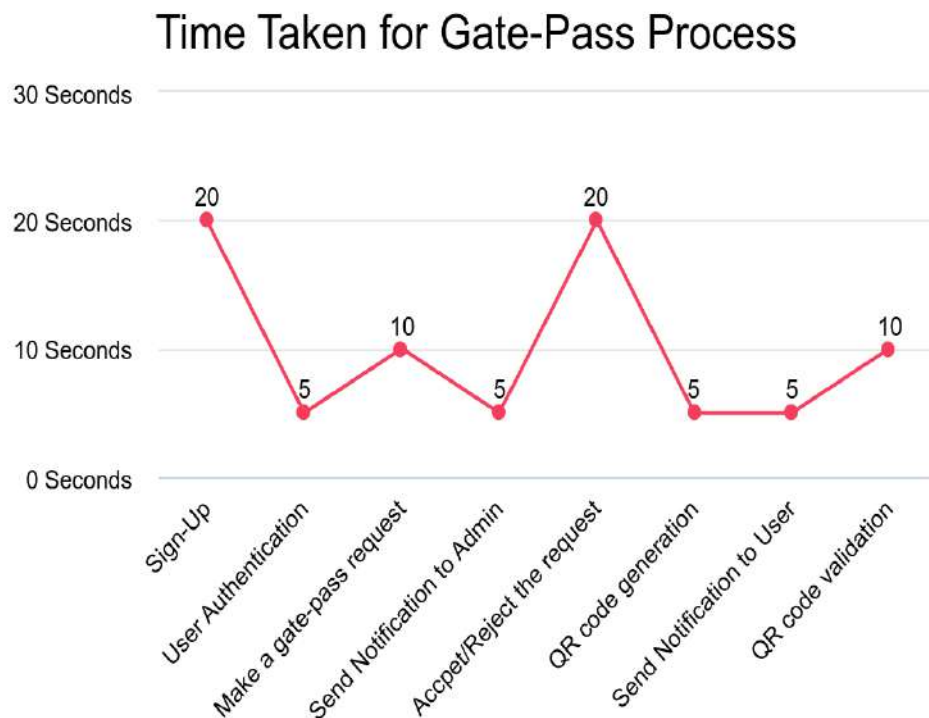


Figure 9.1: Speed Visualization

Chapter 10

Deployment and Maintenance

10.1 Deployment and Maintenance

10.1.1 Installation and un-installation

Installing the DiG-Pass app on your Android device is a straightforward process designed to enhance and secure the management of gate passes through the use of QR codes. To begin, navigate to the Google Play Store on your Android smartphone. In the search bar, type "DiG-Pass" and select the app from the search results. Tap the "Install" button, and the app will download and install automatically on your device. Once the installation is complete, open the app and follow the on-screen instructions to set up your account. This may involve entering your personal details and any required security information. DiG-Pass employs advanced QR code technology to ensure that each gate pass is unique and tamper-proof, enhancing the security of entry and exit points. The app streamlines the gate-pass process, reducing paperwork and the risk of unauthorized access, providing a modern, efficient solution for managing secure entry systems.

10.1.2 Maintenance

Maintaining the DiG-Pass app will involve regular updates for both the user-facing app and the administrative backend. On the user side, this likely includes bug fixes, security patches to address vulnerabilities in the QR code system or the app itself, and compatibility updates to ensure the app functions smoothly on the latest Android operating systems. For administrators, maintenance might involve adding new features for managing gate passes, improving reporting functionalities, and ensuring the system stays integrated with any relevant security infrastructure.

Conclusion and Future Scope

Conclusion

DiG-Pass will lessen the administrative load and manual paperwork by offering a quicker and more effective gate pass issuance system via the app. This system will take the place of the drawn-out gate-pass procedure. This system will guarantee that the college campus can only be exited by those who are authorized. DiG-Pass will offer a single-use QR code generator as a means of gate pass validation. Students are unable to create duplicate gate passes by using a QR code. Every student will receive a unique gate pass. Unlike the conventional gate pass procedure, no one is able to add a name to the gate pass. With ease, staff members and students can use their smartphones to request a gate pass. The app tracks access points and offers real-time data on exits. Administrators can quickly review the history of gate passes because it keeps track of them in the database according to students.

Future Scope

The future scope of the project entails several avenues for expansion and enhancement. Firstly, integrating advanced encryption methods to further secure the QR code data could be explored, ensuring the utmost protection of sensitive information. Additionally, incorporating machine learning algorithms for predictive analytics could optimize gate-pass management by anticipating and mitigating potential bottlenecks or security breaches. Furthermore, extending the application beyond gate-pass management to encompass broader access control systems in various sectors like transportation, healthcare, and education could significantly broaden its utility and impact. Finally, continuous updates and improvements to the user interface and user experience based on feedback and technological advancements would ensure the project remains cutting-edge and user-friendly.

References

- [1] Ms. Ashwini Jarali, Ms. Snehal Kodilkar, Mr. Siddharth Patel, Mr. Shubham Tondare, Mr. Ganesh Kudale, "DiGinty-Securing gated premises using QR-code", Intelligent Computing and Control Systems (ICICCS 2019).
- [2] Akshay ET, Afsal M, Abhinav R, Rahul C, Professor Mohammed Malik CK, Associate Professor Haseena M. "Authenticated Gate-Pass-Generating Application Using QR-Code", International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET), Volume: 12, Issue: 4, April 2023.
- [3] Deepanshu Jaiswal, Devansh Singh, Ms. Aarushi Thusu, "Implementation of Smart and Secure Gate Pass System using QR Code", International Journal of Trend in Scientific Research and Development (IJTSRD), Volume: 7, Issue: 1, January-February 2023.
- [4] Dr. Sunil Bhutada, Dr. Sreenivas Mekala, Mayukhi Gandham, Rishika Bhat, Ruchitha Upadhyayula, "Face Recognition Based Gate Pass System", International Journal of Scientific Research in Science, Engineering and Technology, Print ISSN: 2395-1990.
- [5] Abhijit Alane (Leader), Shrinivas Chalikwar (member), Ganesh Pekam (member), Padmavati Sarode (Mentor), Pranav Pekam(member), "Gatepass Generation and Management System Using QR Code", JETIR May 2022, Volume: 9, Issue: 5.
- [6] V. Sellam, Medha Shree, Shreya Chopdar, Shambhavi, "Gate Pass System", International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958 (Online), Volume:9, Issue:2, December 2019.
- [7] Chaitanya Lengure, Laxmikant Kakde, Mamta Bargat, Saachi Jambhulkar, Prof. Ashish Palandurkar, Prof. Hemant Wade, "E-Gatepass System", International Research Journal of Engineering and Technology (IRJET), Volume: 05, Issue: 03, Mar-2018.

Plagiarism Report

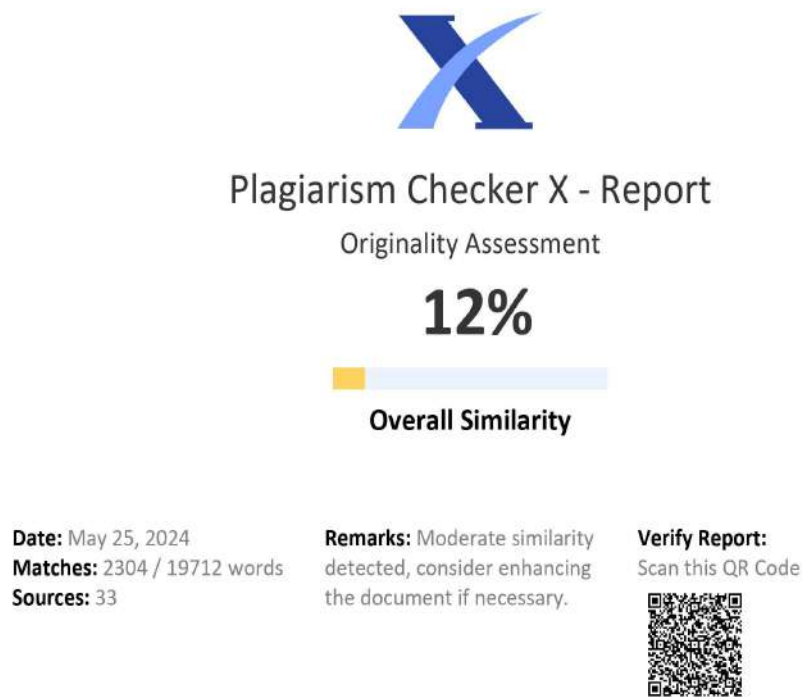


Figure 10.1: Plagiarism Report

Paper Publication and Certificate Details

Published Paper 1

Published paper in International Research Journal of Modernization in Engineering Technology and Science on "DIG-PASS: ENHANCE AND SECURE SOLUTION FOR GATE-PASS USING QR CODE", ISSN 2582-5208, Volume 05 Issue 12.







Published Paper 2

Published paper in International Journal of Emerging Technologies and Innovative Research on "DIG-PASS: ENHANCE AND SECURE SOLUTION FOR GATE-PASS USING QR CODE", ISSN 2349-5162, Volume 11 Issue 5.







Project Competition

Participated in “Kaushalya 2024” National Level Competition organized by Guru Gobind Singh College of Engineering and Research Centre, Nashik.





