Major Project Report

On

QR CODE BASED GATEPASS MANAGEMENT SYSTEM

Submitted in partial fulfillment of the requirements for the award of the degree

BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY

Submitted By

B. Akash 20EG112305
G. Manideep 20EG112306
P. Praneeth 20EG112331

Under the guidance of
Dr. D. Lakshmi Padmaja
Associate Professor

Department of Information Technology

ANURAG UNIVERSITY

(Approved by AICTE and NBA Accredited)

Venkatapur (V), Ghatkesar (M), Medchal district, Hyderabad, Telangana,500088

2020-2024

ANURAG UNIVERSITY

(Approved by AICTE and NBA Accredited)

Venkatapur (V), Ghatkesar (M), Medchal district, Hyderabad, Telangana,500088

Department of Information Technology

CERTIFICATE

This is to certify that the major project report entitled **QR CODE BASED GATEPASS MANAGEMENT SYSTEM** is a Bonafide work done and submitted **by B. Akash (20EG112305), G. Manideep (20EG112306), P. Praneeth (20EG112331)** in partial fulfillment of the requirements for the award of the degree of B. Tech in Information Technology from Anurag University, Hyderabad during the academic year 2023-2024 and the Bonafide work has not been submitted elsewhere for the award of any other degree.

Internal Guide
Dr. D. Lakshmi Padmaja
Associate Professor
Department of IT

H.O.D
Dr. K. S. Reddy
Professor
Dean, Academics and Planning

External Examiner

ACKNOWLEDGEMENT

We would like to express our sincere thanks to **Dr. K. S. Reddy**, Dean, Academics and Planning, Head of the Department of Information Technology, Anurag University, Ghatkesar, whose motivation in the field of software development has made us to overcome all hardships during study and successful completion of project.

We would like to express our profound sense of gratitude to all for having helped us in completing the dissertation. We would like to express our deep-felt gratitude and sincere thanks to our guide, **Dr. D. Lakshmi Padmaja**, Associate Professor, Department of Information Technology, Anurag University, Ghatkesar for her skillful guidance, timely suggestions and encouragement in completing this project.

We extend our sincere thanks to **Dr. V. Vijaya Kumar**, Dean, School of Engineering, **Anurag University**, Venkatapur (V), Ghatkesar (M), R. R. Dist, for their encouragement and constant help. Finally, we would like to express our heartfelt thanks to our parents who were very supportive both financially and mentally and for their encouragement to achieve our set goals.

B. Akash (20EG112305)

G. Manideep (20EG112306)

P. Praneeth (20EG112331)

DECLARATION

This is to Certify that the project work entitled **QR CODE BASED GATEPASS MANAGEMENT SYSTEM** submitted to **Anurag University** in partial fulfilment of the requirement for the award of the Degree of Bachelor of Technology (B.Tech), is an original work carried out by **B. Akash (20EG112305), G. Manideep (20EG112306), P. Praneeth (20EG112331)** under the guidance of **Dr. D. Lakshmi Padmaja**, Associate Professor, Department of Information Technology. This matter embodied in this project is a genuine work, done by the students and has not been submitted whether the university or to any other university/Institute for the fulfilment of the requirement of any course of study.

B. Akash (20EG112305)

G. Manideep (20EG112306)

P. Praneeth (20EG112331)

ABSTRACT

The surge in school and college enrollments has elevated gate pass management to a critical activity. Traditional methods of requesting an out pass involve manual filling of details and waiting for higher authorities to grant permission, resulting in time-consuming processes, especially during peak periods. Automated systems are imperative for efficient management of gate-related activities. The QR Code Based Gatepass Management System is a software application designed to streamline this process by digitizing out pass generation. This not only reduces paper usage but also expedites the request process directly from the browser. The system tracks students and employees in real-time, following a systematic flow from student to faculty/HOD, culminating in the electronic generation of gate passes. Current applications suffer from limited dashboards, single-admin monitoring, and lack of standard authentication mechanisms, leading to delays. Our solution entails building a structured, user-friendly application with interactive dashboards, efficient data flow between students and faculty, prompt email notifications, text-processing, sentiment analysis, and robust security gate authentication mechanisms. This comprehensive system aims to address the challenges faced by students and faculty in gate pass requests and approvals.

TABLE OF CONTENT

LIST OF FIGURES

1. INTRODUCTION

In an era defined by digital advancements, the traditional paradigms of university gate pass management systems are undergoing a transformative shift. Our project, "The Implementation of a QR Code Based Gatepass Management System in Universities," represents a pioneering leap towards redefining campus security and operational efficiency. Recognizing the limitations inherent in manual, paper-based processes, our innovative solution leverages the ubiquitous presence of smartphones and the power of QR codes to revolutionize the way access is granted within educational institutions. The prevailing reliance on paperwork and manual verification often results in time-consuming procedures prone to errors. To address these challenges, our system introduces a seamless, paperless process where students, faculty, and staff can effortlessly generate QR code gate passes using their smartphones. This not only aligns with contemporary technological trends but also significantly enhances the convenience for individuals navigating the dynamic campus environment. At the heart of our QR code-based system is a commitment to bolstering security measures. The QR code serves as a digital token, encapsulating encrypted information about the individual's identity and their specific access permissions. This innovative approach not only minimizes the risk of unauthorized access but also facilitates swift and foolproof verification at entry and exit points.

The integration of smartphones for gate pass management marks a departure from cumbersome verification queues. Our system enables individuals to present their QR codes for rapid scanning, either through dedicated scanners or existing smartphones within the app. The result is a substantial reduction in wait times, fostering a more efficient and fluid movement of people in and out of the university premises. Beyond the immediate benefits of security and efficiency, our QR code gate pass system seamlessly integrates with the university's database. This integration provides real-time insights into entry and exit data, empowering security personnel to identify patterns, manage crowds effectively, and respond proactively to any security concerns. Moreover, our commitment extends to environmental sustainability. By eliminating excessive paperwork, we contribute to a greener campus, aligning with the global imperative to reduce our ecological footprint. In essence, the implementation of this QR Code Based Gatepass Management System heralds a new era for university campuses. By marrying the convenience of smartphones with advanced digital security measures, we not only address the shortcomings of traditional systems but also enhance the overall campus experience for students, faculty, and staff. Welcome to a future where access is not just granted but granted intelligently, efficiently, and securely..

1.1 OBJECTIVES

The QR Code Based Gatepass Management System is driven by a set of well-defined objectives aimed at addressing various challenges in the existing gate pass processes. The foremost objective is to enhance security within the campus environment by implementing a QR code-based system, leveraging the inherent tamper-proof nature of QR codes to deter unauthorized access. Simultaneously, the project seeks to streamline access processes and reduce the environmental impact associated with traditional paper-based gate passes, fostering a more sustainable and efficient approach to campus management. Standardizing authentication mechanisms, such as QR codes, is a pivotal objective to eradicate potential security vulnerabilities present in the current system. This initiative aims to ensure that the digital representation of gate passes is resistant to forgery or duplication, thereby fortifying the overall security framework. The project also prioritizes the facilitation of mobile-based access, allowing students, faculty, and staff to effortlessly generate and present QR codes using their smartphones. This not only aligns with contemporary technological trends but also contributes to the convenience and accessibility of gate pass management within the university community.

Real-time monitoring of entry and exit data is another key objective, driven by the integration of the gate pass system with a cloud-based database. This feature empowers security personnel with valuable insights, enabling them to identify patterns, manage crowds effectively, and respond promptly to any security concerns. Faculty workflow optimization is achieved through the digitization of approval processes and the generation of QR codes, reducing administrative burdens and expediting gate pass issuance. Furthermore, the incorporation of Natural Language Processing (NLP) adds an advanced layer of intelligence to the system. This objective involves analyzing the textual content of gate pass requests to automatically assign priority levels based on the perceived urgency and significance. This not only enhances the efficiency of faculty decision-making but also provides a nuanced understanding of the reasons behind each gate pass request. In terms of technology, the project aims to create a scalable solution by utilizing Flask, MongoDB Atlas, and Python. These technologies form the backbone of the system, ensuring a responsive and efficient user experience while also accommodating potential future expansions. Overall, the project's objectives converge to create a holistic and innovative QR Code Based Gatepass Management System, poised to revolutionize access control and security in various environments.

1.2 SCOPE

The QR Code Based Gatepass Management System's scope is ambitious, encompassing a campus-wide transformation to redefine access control and bolster security measures. With a focus on students, faculty, and staff members, the project seeks to standardize and streamline gate pass procedures through the implementation of a QR code-based system. The paramount objective is to fortify campus security by replacing conventional paper-based gate passes with tamper-proof digital QR codes, minimizing the risk of unauthorized access and the vulnerabilities associated with forgery or duplication. Efficiency improvement is central to the project's scope, aiming to expedite gate pass issuance and verification processes. By adopting QR codes, the system enhances accessibility and minimizes wait times at entry and exit points, contributing to a more fluid and organized campus flow. Embracing environmental sustainability, the project promotes a paperless approach, reducing the environmental impact of gate pass management. The scope extends to real-time monitoring capabilities, enabling security personnel to access comprehensive logs for effective crowd management and timely responses to security concerns. Faculty workflow optimization, mobile accessibility, and the integration of Natural Language Processing further enrich the project's scope, ensuring a holistic and innovative approach to access control within the evolving landscape of campus environments.

1.3 APPLICATIONS

- Educational Institutions
- Events and Conferences

1.4 ADVANTAGES

- Enhanced Security
- Streamline Control
- Paperless Approach

1.5 SYSTEM REQUIREMENTS

1.5.1 Hardware Requirements

- Processor : Intel i3 and above
- RAM : 4GB and Higher
- Hard Disk : 500GB Minimum

1.5.2 Software Requirements

- Programming Language / Platform : Python
- IDE : Vscode
- Web Framework : Flask
- UI Technologies : Jinja 2.0
- Database : MongoDB Atlas

2. LITERATURE SURVEY

The literature survey for the "QR Code Based Gatepass Management System" project involves a comprehensive exploration of existing research to establish a theoretical framework and inform the project's methodologies. The survey begins by scrutinizing the challenges associated with traditional paper-based gate pass systems within educational institutions, emphasizing delays, inefficiencies, and security vulnerabilities. The literature aims to identify gaps in these systems, particularly in the context of university environments, where a high volume of gate pass requests and varied interactions among students, faculty, and security personnel pose unique challenges.

Moving forward, the survey explores technological advancements and authentication mechanisms adopted in similar systems. Notably, the focus is on modern solutions such as QR codes, RFID, and OTP. The goal is to understand how these technologies contribute to enhancing security, streamlining processes, and reducing manual errors in gate pass management. The literature survey delves into the experiences and findings of previous researchers who have explored the integration of digital technologies for efficient gate pass issuance and verification. A significant aspect of the literature survey involves an examination of studies related to automation, visitor, and leave management systems. These studies provide insights into how institutions have approached the management of visitors and leave requests, shedding light on potential solutions and challenges faced. The survey aims to distill valuable lessons and best practices from these existing systems, offering guidance on how the proposed QR Code Based Gatepass Management System can address similar scenarios.

Incorporating Natural Language Processing (NLP) into gate pass management represents a novel aspect of the project. The literature survey, therefore, includes a thorough investigation of studies where NLP has been applied to understand user inputs. This involves analyzing the reasons provided by students for gate pass requests and how NLP algorithms interpret the content. Insights from these studies inform the design of an intelligent layer within the proposed system, allowing for automatic prioritization of gate pass requests based on the context, sentiments, and keywords in the provided reasons. Furthermore, the survey expands its focus to the technological stack proposed for the project, including Flask, MongoDB, and Python. This exploration aims to understand how these technologies contribute to scalability, flexibility, and the overall efficiency of gate pass management systems. The cloud-based MongoDB Atlas is particularly emphasized for its role in storing and retrieving data, providing scalability, and facilitating real-time monitoring.

Considering user experience and environmental impact, the literature survey includes studies discussing the shift to paperless systems and the benefits of digital gate pass management. It explores how the proposed project aligns with principles of sustainability and user-friendly design, contributing to a positive campus experience for students, faculty, and staff.

The literature survey is a multifaceted exploration encompassing traditional gate pass challenges, technological advancements, authentication mechanisms, automation, visitor and leave management, NLP applications, and the proposed technological stack. The synthesis of insights from these diverse areas provides a solid theoretical foundation for the project, guiding its goals, methodologies, and potential contributions to the field of access control and security within educational institutions.

Reference 1:

Title: Gate-pass Management System
Author : Archana et al.
Description : This study proposes a paperless approach for visitor gate pass management, reducing manual effort and facilitating request verification through a single host. However, it highlights challenges when dealing with a high volume of requests and lacks details regarding the authentication technique.

Reference 2 :

Title : Student Leave Management System
Author : Kaushik
Description : The authors focus on building an automated Leave Management System, emphasizing student attendance management. While addressing the paper-based approach, the study lacks details on faculty leave management and the authentication mechanisms.

Reference 3 :

Title : Visitor Gate Pass Management System
Author : Harish Rapartiwar
Description : The development of a Visitor Gate Pass Management System is explored, emphasizing efficient visitor maintenance in an organizational context. The study includes details on visitor classification and types but does not delve into authentication mechanisms like QR codes or RFID.

Reference 4 :

Title : E-Gatepass System

Author : Chaitanya Lengure

Description : he study presents an application using the MVC pattern for gate pass management. It covers various modules but lacks discussion on authentication mechanisms such as QR codes, RFID, or OTP.

Reference 5 :

Title : Online Gate Pass Application form for Hostel Students

Author : S. Venkatesa Perumal

Description : This study proposes an online gate pass application form for hostel students, emphasizing notifications through emails. While involving a flow from student to security guard, the study lacks detailed dashboards and discussion on advanced authentication mechanisms.

3. ANALYSIS

3.1 EXISTING SYSTEMS

The current gate pass management system within the educational institution endeavors to transition away from traditional paper-based methods, embracing a digital approach for handling gate passes, visitor entries, and leave requests. Despite this effort, the existing system reveals shortcomings in both efficiency and security. The system is designed to handle diverse interactions among students, faculty, and security, as well as interactions involving visitors, employees, and security personnel. However, notable deficiencies emerge, such as limited dashboard functionalities and a singular administrator role, typically held by higher authorities like HODs, Principals, or Faculty members. This centralized administrative structure poses challenges in terms of scalability and adaptability to various operational needs. Moreover, a critical gap in the existing system lies in the absence of standardized authentication mechanisms like OTPs, RFID, or QR codes. This absence not only impedes the system's ability to establish a robust security framework but also exposes potential vulnerabilities, leaving the entire campus at risk. To address these limitations and fortify the security and efficiency of gate pass management processes, our project proposes the implementation of a comprehensive QR code-based system. This innovative solution aims to establish a secure and streamlined approach, eliminating the drawbacks associated with the current system and enhancing the overall experience for students, faculty, and staff within the educational institution.

3.1.1 DISADVANTAGES OF EXISTING SYSTEM

Existing gate pass systems in educational institutions, particularly those relying on traditional paper-based methods, suffer from inefficiency, manual errors, and limited security features. The centralized administrative structure and lack of real-time monitoring pose scalability challenges and hinder prompt responses to security concerns. Limited dashboard functionalities further constrain user access. The absence of standardized authentication mechanisms leaves the system vulnerable to unauthorized access. In response, the proposed QR code-based gate pass management system seeks to address these drawbacks, introducing advanced security measures, streamlined processes, and improved efficiency within educational institutions. This innovative solution aims to provide a secure, user-friendly experience, significantly reducing processing times and enhancing overall campus security.

3.2 PROPOSED SYSTEM

The proposed QR Code-based Gate Pass Management System aims to revolutionize the existing gate pass procedures within educational institutions by leveraging advanced technology to enhance security, efficiency, and user experience. At its core, the system introduces a paradigm shift from traditional paper-based methods to a digital approach, utilizing QR codes as the primary authentication and identification mechanism. The central theory of the proposed system revolves around the use of QR codes as tamper-proof digital representations of gate passes. Each student, faculty member, or authorized personnel is assigned a unique QR code containing encrypted information about their identity and authorization levels. This QR code becomes their digital gate pass, replacing the conventional paper passes prone to forgery and duplication.

The implementation involves three key user roles: students, faculty, and security personnel. Students initiate gate pass requests through the application, generating QR codes for quick and secure exits. Faculty members play a pivotal role in the approval process, reviewing and approving requests while generating QR codes for authorized exits. Security personnel, equipped with QR code scanners, verify the authenticity of passes at exit points, ensuring only authorized individuals can exit the premises.

Incorporating Natural Language Processing (NLP) adds an intelligent layer to the system. NLP analyzes the reasons provided by students for gate pass requests, automatically assigning priorities based on the content's urgency and significance. This advanced feature streamlines faculty decision-making, allowing them to address high-priority requests promptly.

The choice of technologies, including Flask for web development, MongoDB Atlas for cloud-based data storage, and Python for server-side logic, ensures a responsive and efficient system. GitHub serves as the collaborative platform for version control, facilitating effective teamwork and continuous improvement.

Overall, the proposed system aims to create a secure, streamlined, and user-friendly environment, reducing processing times, minimizing errors, and enhancing the overall gate pass management experience within educational institutions.

4. DESIGN

4.1. UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common languagefor creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of
software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

• It provides users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.

• It provides extendibility and specialization mechanisms to extend the core concepts.

• Be independent of particular programming languages and development process.

• It provides a formal basis for understanding the modeling language.

• Encourage the growth of OO tools market.

• Support higher level development concepts such as collaborations, frameworks, patterns and components.

• Integrate best practices.

4.1.1 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

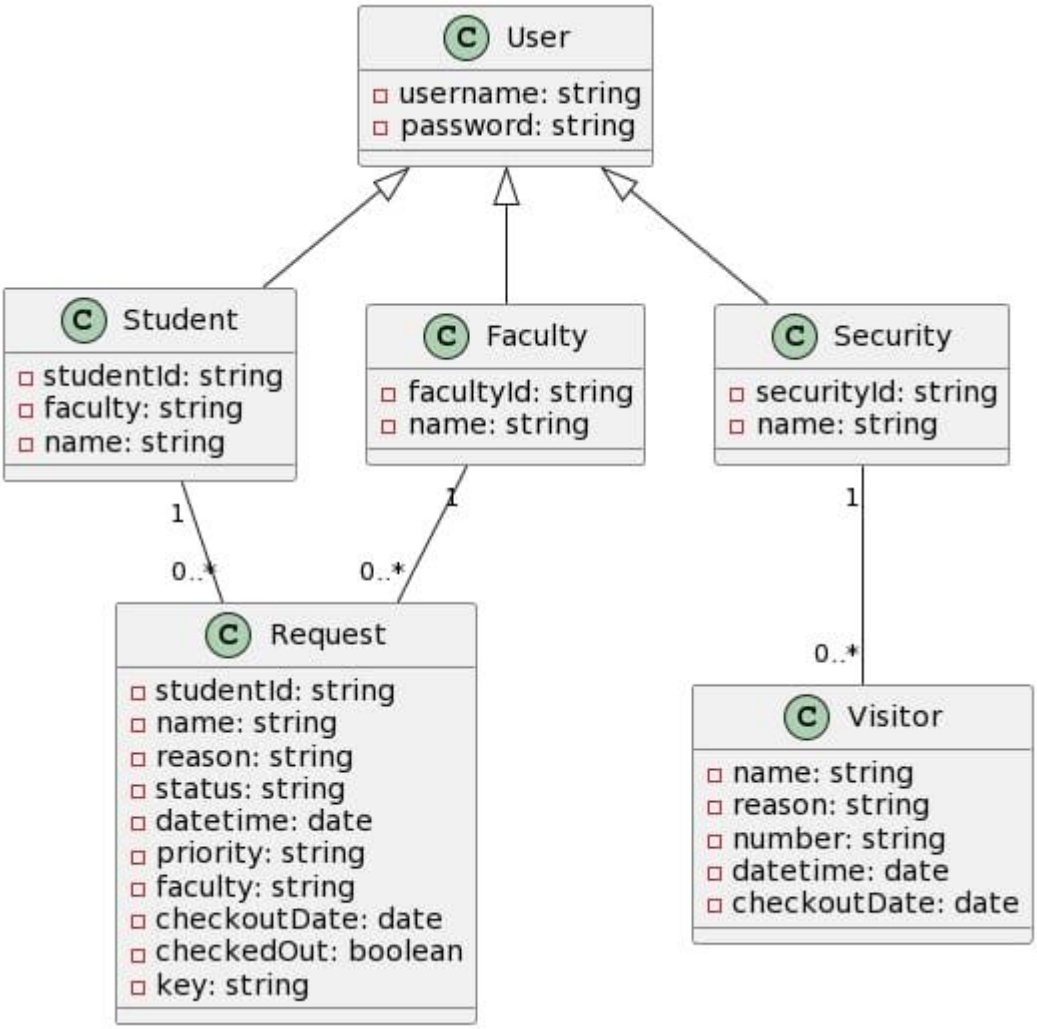

Fig 1 : Use Case Diagram

Here in this Fig 1 it represents the dynamic behaviour of a system and encapsulates the system's functionality by use cases like user, student, faculty and security.

4.1.2 SEQUENCE DIAGRAM

Sequence Diagrams Represent the objects participating the interaction horizontally and time vertically. A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants that the subject can perform in collaboration with one or more actors. Use cases define the offered behavior of the subject without reference to its internal structure. These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment.

In this section, we delve into the theoretical foundations and concepts surrounding Denial of Service (DoS) attacks. Explore key theories, frameworks, and models that underpin the understanding of DoS attacks, their evolution, and the principles guiding their mitigation.

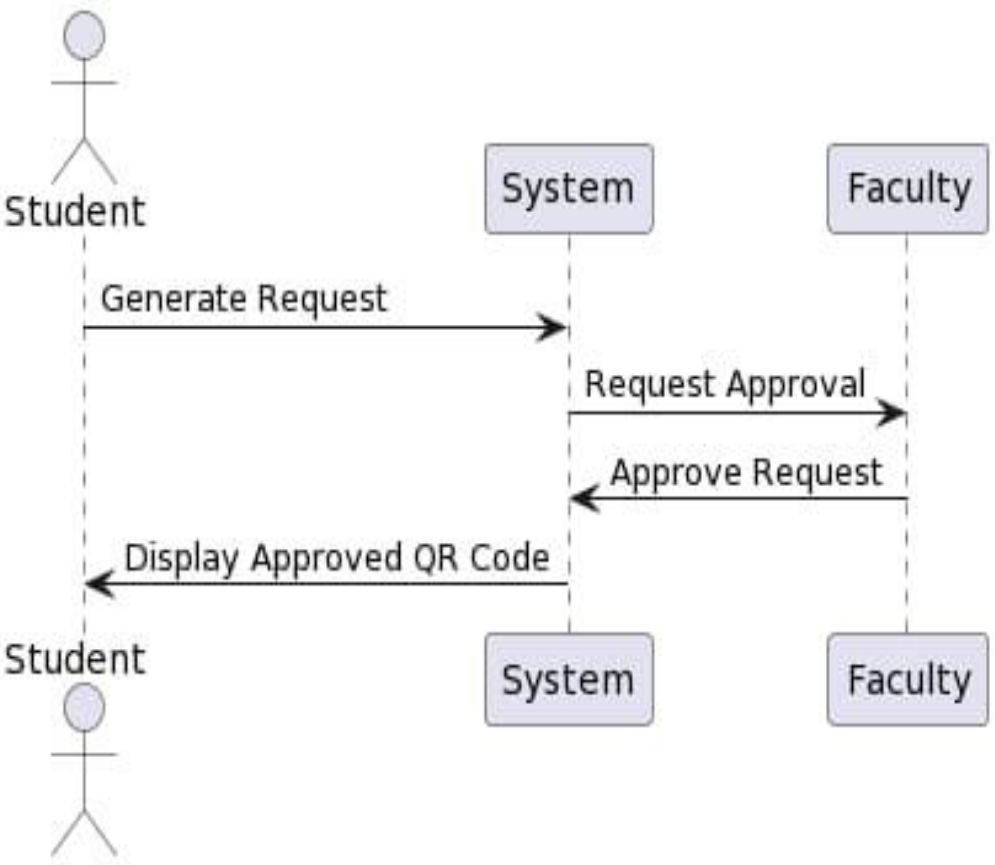

Fig 2 : Student Sequence Diagram

Here in this Fig 2 it represents How the flow of requests work for student in QR code based gatepass management system.

Sequence diagram for security scanning QR code :

The sequence diagram for security scanning QR codes in the Gate Pass Management System delineates the procedural steps involved in verifying individuals' exits using QR codes. Initiated by the security personnel, the process begins with the presentation of the QR code by the pass holder, received upon approval of their gate pass request. The security personnel then scans the QR code using a designated device, triggering the decoding process to extract encrypted information. The system subsequently verifies the authenticity and validity of the QR code, cross-referencing it with authorization data stored in the database to confirm the pass holder's identity and access permissions. Upon successful verification, the system grants exit access, logging relevant data for monitoring and auditing purposes. Finally, the system provides feedback to the pass holder, ensuring a seamless and secure exit experience. This sequence diagram elucidates the dynamic interactions between stakeholders and the system, elucidating the security protocol within the Gate Pass Management System.

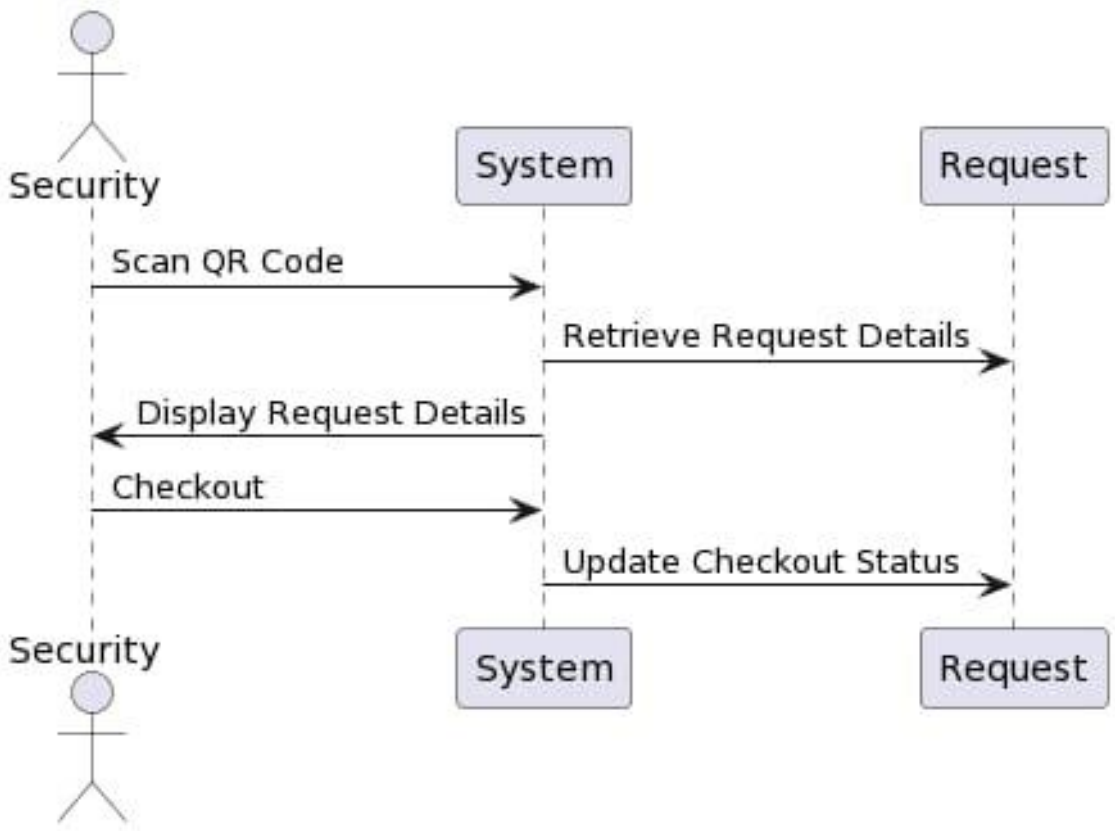

Fig 3 : Security Sequence Diagram

Here in this Fig 3 it represents How the flow of requests work for Security in QR code based gatepass management system.

4.1.3 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

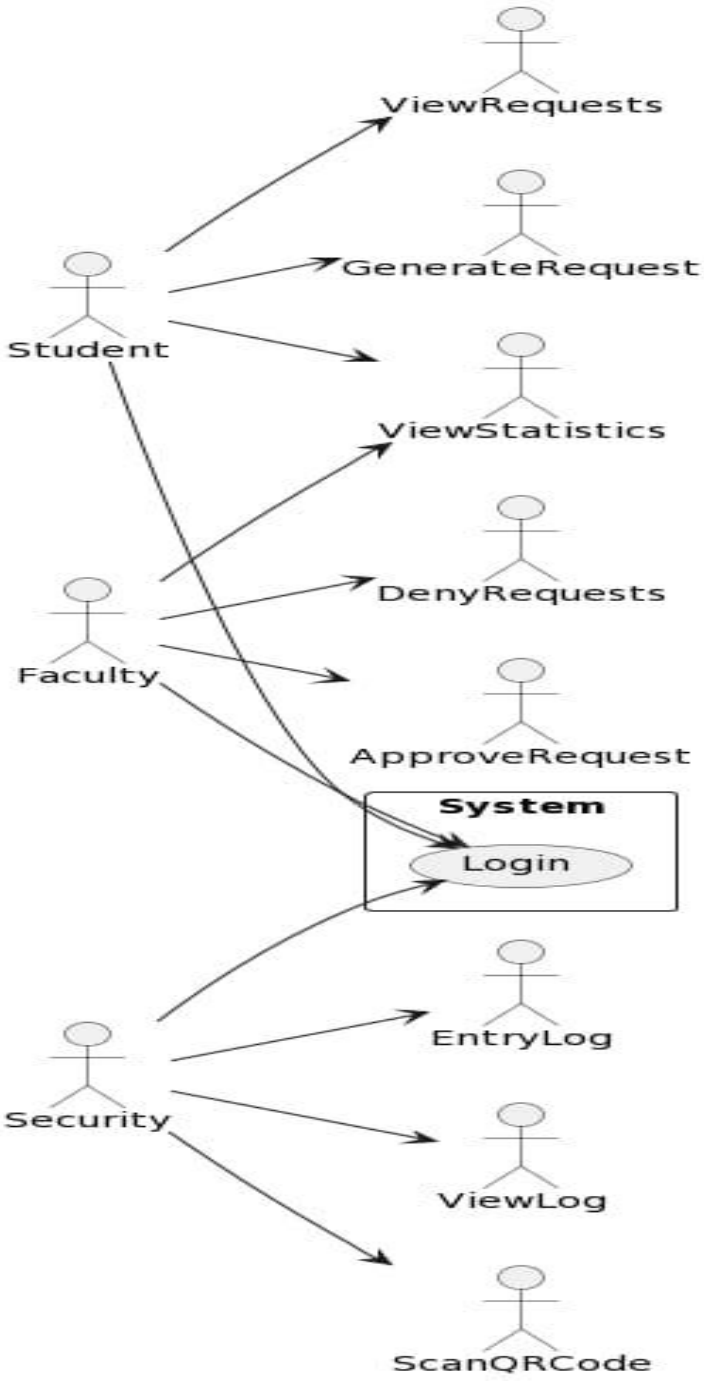

Fig 4 : Use Class Diagram

Here in this Fig 4 it represents the static view of an application. It representing the types of objects residing in the system and the relationships between them.

4.1.4 COMPONENT DIAGRAM

Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems. They are the special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system, but it describes the components used to make those functionalities.

Thus, from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc. Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment. A single component diagram cannot represent the entire system, but a collection of diagrams is used to represent the whole.

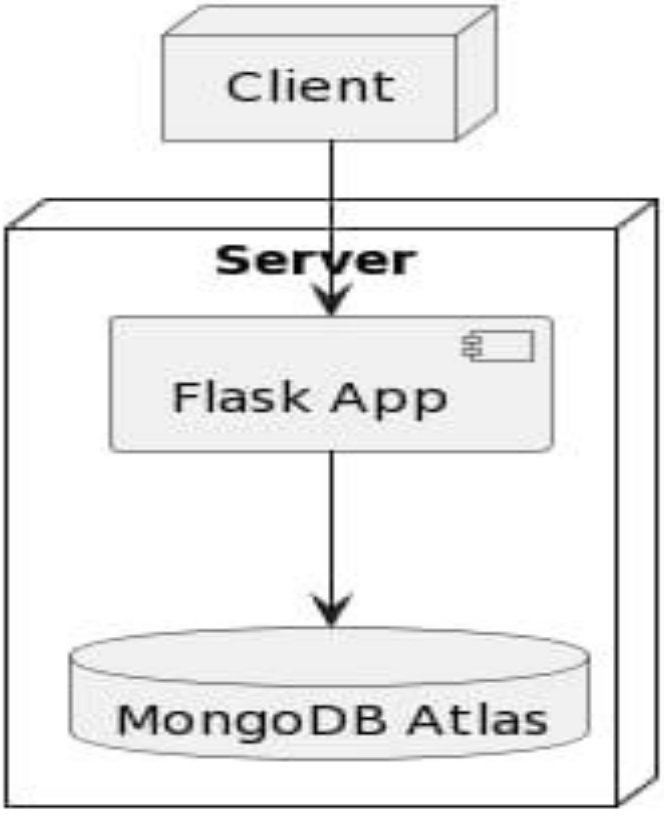

Fig 5 : Component Diagram

Here in this Fig 5 it represents the physical components of the system, it describes the components used to make those functionalities.

5. IMPLEMENTATION

5.1 MODULE DESCRIPTION:

Login of the user:

The login functionality within the QR Code Based Gatepass Management System serves as the initial point of interaction for users, ensuring secure access to the system's features. Users, including students, faculty, and security personnel, undergo a straightforward authentication process through a dedicated login page. This page prompts individuals to input their unique credentials, typically a combination of a username and password, ensuring that only authorized users gain entry. The login mechanism is designed to be user-friendly, with a responsive interface implemented using Flask, a lightweight web framework. Security is prioritized through robust authentication protocols, preventing unauthorized access to sensitive information. Successful login grants users access to personalized dashboards and functionalities based on their roles, facilitating a seamless and controlled experience tailored to the specific needs of students, faculty, or security personnel. The integration of MongoDB Atlas ensures secure data storage, and the collaborative use of GitHub for version control enhances transparency and stability throughout the login process and the entire project.

Requesting Gate Pass:

The gate pass request functionality within the QR Code Based Gatepass Management System provides students with a user-friendly and efficient process to seek permission for exiting the educational institution. Through their dedicated dashboards, students initiate gate pass requests, detailing essential information such as the purpose, date, and destination of their intended exit. The system's streamlined interface facilitates the generation of QR codes directly on students' smartphones, eliminating the need for cumbersome paperwork. Once a student submits a request, faculty members play a crucial role in the approval process. Upon faculty approval, the system dynamically generates a QR code, securely embedding the necessary information. This QR code becomes the digital representation of the gate pass, resistant to tampering or forgery.

The incorporation of Natural Language Processing (NLP) adds an intelligent layer to the system, analyzing the reasons provided by students and automatically assigning priority levels based on urgency. This advanced feature optimizes the decision-making process for faculty, allowing them to prioritize and address high-priority requests promptly. The seamless integration of Flask for web development and MongoDB Atlas for data storage ensures a secure and efficient gate pass issuance process. Upon successful approval, students can conveniently present and scan their QR codes at exit points, enabling security personnel to verify the authenticity of the passes efficiently. This innovative approach not only expedites gate pass issuance but also enhances the overall security and responsiveness of the system, contributing to a modern and effective gate pass management experience within the educational institution.

Generation of Gate pass:

The QR Code Generation feature in the QR Code Based Gatepass Management System is a pivotal component ensuring the secure and efficient creation of digital gate passes. Once a student's gate pass request is approved by the faculty, the system seamlessly generates a unique QR code associated with the specific exit authorization. This QR code becomes the digital counterpart of the traditional paper gate pass, embedding encrypted information about the individual, their authorization details, and the approved exit parameters. Leveraging the Flask framework and Python programming, the system ensures a responsive and streamlined generation process. The use of MongoDB Atlas for cloud-based data storage guarantees secure storage and retrieval of the generated QR codes.

The generated QR codes possess a tamper-proof nature, enhancing security measures against forgery or duplication. Each QR code is dynamically created to represent the authorized exit, contributing to a paperless and technologically advanced gate pass system. This approach not only reduces the environmental impact associated with traditional paperwork but also expedites the exit process for students, faculty, and staff. The successful integration of QR code technology elevates the overall efficiency and security of the gate pass management system, aligning with the project's commitment to leveraging advanced technology for a modernized campus experience.

Accepting or rejecting gate pass:

The gate pass acceptance and rejection process in the QR Code Based Gatepass Management System forms a critical phase overseen by faculty members. Upon a student's submission of a gate pass request, faculty members are entrusted with the responsibility of thoroughly reviewing the details, encompassing the purpose, date, and destination of the intended exit. Through their dedicated dashboards, faculty members have the authority to either approve or reject the request. If the faculty accepts the request, the system dynamically generates a QR code, serving as the digital representation of the approved gate pass. Notably, this QR code incorporates an expiration date, ensuring that the authorization is valid only for a specific timeframe. This feature enhances security and control over the exit authorization process. The integration of Flask for web development and MongoDB Atlas for data storage ensures a secure and responsive platform for faculty decision-making.

Conversely, in cases where the faculty decides to reject the request, the system refrains from generating a QR code. This efficient and transparent mechanism facilitates clear communication between faculty and students regarding the status of gate pass requests. The dynamic acceptance and rejection process, coupled with the incorporation of time limited QR codes, contributes to a sophisticated, secure, and streamlined gate pass management system within educational institutions, aligning with the project's goal of modernizing and enhancing campus security.

Verifying qr codes:

The QR Code verification process in the QR Code Based Gatepass Management System plays a crucial role in ensuring the authenticity of digital gate passes at exit points. Security personnel are equipped with QR code scanners, enabling them to swiftly and accurately verify the legitimacy of the presented QR codes. As students approach exit points, the security personnel scan the QR codes using dedicated scanners or even existing smartphones within the app. The system processes the scanned information, validating the digital identity and authorization details encrypted within the QR code.

The incorporation of QR code technology introduces a tamper-proof digital identity, significantly reducing the risk of unauthorized access or forgery. This streamlined verification process not only enhances security measures but also contributes to a more efficient flow of people in and out of the educational institution. The real-time verification at exit points ensures that only individuals with valid and current QR codes, approved by faculty members, are permitted to exit the premises. This integration of technology, security protocols, and user-friendly verification mechanisms aligns with the project's overarching goal of elevating the overall security and efficiency of gate pass management within educational institutions.

Invalid qr codes:

The QR Code Based Gatepass Management System incorporates robust measures to address and identify invalid QR codes, ensuring the integrity of the exit authorization process within the educational institution. When security personnel encounter a QR code that is either expired or does not align with the system's encryption parameters, the system categorizes it as invalid. The dynamic nature of QR codes, including built-in expiration dates, enables an automated check for their validity during the scanning process.

In the event of an invalid QR code, security personnel are promptly alerted, signaling a potential breach of the gate pass system. This immediate response mechanism enables security teams to take appropriate action, denying access to individuals with expired or unauthorized QR codes. The systematic handling of invalid QR codes enhances overall security, preventing unauthorized individuals from exiting the premises undetected.

The implementation of this feature underscores the system's commitment to maintaining a tamper-proof and secure gate pass management process. By promptly identifying and addressing invalid QR codes, the project ensures the reliability and effectiveness of the digital gate pass system, contributing to a safer and more secure campus environment.

Visitor entry:

The QR Code Based Gatepass Management System extends its capabilities to seamlessly handle visitor entries, ushering in a new era of efficiency and security for the educational institution. Visitors entering the campus are guided through a digital form within the system, where they input essential information such as their name, purpose of the visit, and contact details. This modernized approach replaces traditional paper-based methods, ensuring a standardized and user-friendly process for capturing visitor information.

A dedicated Visitor View feature empowers authorized personnel to access a comprehensive overview, facilitating the tracking and management of the number of visitors present on campus at any given time. This not only enhances security measures but also contributes to effective crowd management and quick response to any security concerns.

By introducing digital forms for visitor entries and a sophisticated Visitor View, the project transforms the visitor management process, aligning with the goal of modernizing gate pass systems and fortifying overall campus security. This innovative approach ensures a seamless and secure experience for visitors while providing valuable insights for security personnel to regulate access within the educational institution.

5.2. IMPLEMENTATION TECHNOLOGIES

Flask:

Flask, a micro web framework for Python, plays a pivotal role as the foundational web framework in the QR Code Based Gatepass Management System project. Its reputation for being lightweight, versatile, and user-friendly makes it an ideal choice for developing the core structure of the application. Flask excels in providing a minimalistic yet robust foundation, allowing developers to build scalable and efficient web applications with ease.

One of Flask's primary strengths lies in its simplicity. It doesn't come with an abundance of built-in features, which makes it highly customizable and adaptable to the specific needs of a project. This simplicity, however, doesn't compromise its capabilities. Flask efficiently handles crucial aspects of web application development, such as managing HTTP requests and routing, enabling the systematic handling of different types of user interactions.

In the context of the QR Code Based Gatepass Management System, Flask facilitates seamless communication between the user interface and the back-end logic. Its routing capabilities enable the application to respond appropriately to different URLs, ensuring that users are directed to the correct pages or actions based on their requests. This routing mechanism is crucial in managing various components of the gate pass system, including user authentication, gate pass requests, and QR code generation.

Flask's integration with the MongoDB database is particularly noteworthy for this project. MongoDB, being a NoSQL database, complements Flask's flexibility by allowing data storage in a JSON-like format. Flask's compatibility with MongoDB streamlines the handling of data related to gate passes, users, and other critical information in the project. The cloud-based version, MongoDB Atlas, adds scalability and accessibility to the database, aligning with the project's need for efficient data management. Flask's lightweight design, simplicity, and adept handling of web development fundamentals make it an excellent choice for developing the backbone of the QR Code Based Gatepass Management System. Its ability to manage HTTP requests, handle routing, and seamlessly interact with the MongoDB database ensures a responsive, efficient, and highly adaptable web framework for this innovative campus security project.

MongoDB Atlas:

MongoDB Atlas stands as a foundational element within the architecture of the QR Code Based Gatepass Management System, fulfilling a crucial role in the storage and retrieval of data. This cloud-based iteration of MongoDB, a prominent NoSQL database, is instrumental in enhancing the efficiency, flexibility, and accessibility of the gate pass management system.

At its core, MongoDB, and by extension MongoDB Atlas, operates as a NoSQL database, distinguishing itself from traditional relational databases. The system's flexibility is derived from its document-oriented, JSON-like data model. This allows data to be stored in a schema-less, dynamic structure, enabling the accommodation of diverse data types and formats. In the context of the gate pass management system, this flexibility proves invaluable, as it caters to the evolving and diverse data associated with gate passes, user profiles, and other essential information.

The decision to employ MongoDB Atlas as a cloud-based solution introduces several key advantages. One of its primary benefits is scalability. As a cloud-based service, MongoDB Atlas can seamlessly scale its resources to manage increasing volumes of data and accommodate growing user demands. This scalability is critical for a system like gate pass management, where the database needs to efficiently handle large datasets and a potentially high number of simultaneous user interactions.

Moreover, the cloud-based nature of MongoDB Atlas significantly enhances accessibility. Geographical constraints are eliminated, allowing authorized users to access and interact with the database from any location with an internet connection. This is particularly relevant for a gate pass management system within an educational institution, where administrators, faculty, and security personnel may need to retrieve or update data from various locations on the campus. In essence, MongoDB Atlas empowers the gate pass management system by combining the robust features of MongoDB's NoSQL database with the benefits of a cloud-based infrastructure. Its role in handling data efficiently, adapting to changing data requirements, and providing scalability and accessibility aligns seamlessly with the project's objectives, contributing to a responsive, adaptable, and efficient gate pass management solution.

Python:

Python, a versatile and dynamically typed programming language, plays a central and integral role in the development of the QR Code Based Gatepass Management System. Python is employed for server-side logic, data processing, and the seamless integration of various components within the application. Its versatility and ease of use make it a preferred choice for a wide range of applications, and in this context, it contributes to the robust functionality of the gate pass management system. One of Python's key strengths in this project is its compatibility with Flask, the web framework used for developing the application. Flask leverages Python's capabilities to handle crucial aspects of web application development, such as routing HTTP requests and facilitating interactions with the MongoDB database. Python, in conjunction with Flask, ensures a seamless connection between server-side functionalities and the web development framework, providing a solid foundation for the gate pass management system.

GitHub:

GitHub, a widely used platform, stands as a crucial element in the development workflow of the QR Code Based Gatepass Management System, serving as the designated version control platform for collaborative development. GitHub's role in version control enhances the efficiency, transparency, and collaborative aspects of the project, making it an invaluable tool for managing the codebase. As a version control platform, GitHub enables developers to host and manage repositories, which contain the entire codebase and associated project files. Hosting the project repository on GitHub provides a centralized location for collaborative development, allowing multiple developers to work on the same project simultaneously. This collaboration is made possible through the implementation of Git, a distributed version control system, underlying GitHub's functionalities.

Natural Language Processing (NLP):

Natural Language Processing (NLP) is a subfield of artificial intelligence (AI) that focuses on the interaction between computers and human language. Its primary goal is to enable machines to understand, interpret, and generate human-like language. NLP involves the development of algorithms and models that can process and analyze large volumes of natural language data, extracting meaningful insights and facilitating communication between humans and machines.

Applications of NLP:

1. Sentiment Analysis: NLP is extensively used for sentiment analysis, allowing businesses to understand and gauge public opinions, emotions, and attitudes expressed in text data, such as social media posts, reviews, or customer feedback.
2. Language Translation: NLP powers language translation services, enabling the translation of text or speech from one language to another. This application is crucial for breaking down language barriers and fostering global communication.
3. Chatbots and Virtual Assistants: NLP plays a pivotal role in the development of chatbots and virtual assistants, enabling these systems to understand user queries in their natural language and provide relevant responses. This enhances user interaction and customer support.
4. Information Extraction: NLP is employed to extract valuable information, entities, and relationships from unstructured text data. This is particularly useful for automating the process of extracting structured data from documents or articles.
5. Speech Recognition: NLP technologies are integral to speech recognition systems, allowing machines to convert spoken language into text. This has widespread applications in voice-activated systems and virtual assistants.

Advantages of NLP:

1. Efficiency: NLP automates language-related tasks, saving time and increasing efficiency in processing and analyzing large volumes of textual data. It streamlines tasks such as information extraction, sentiment analysis, and language translation.

2. Insight Generation: By extracting meaningful insights from unstructured data, NLP enables businesses and organizations to make informed decisions based on textual information. This can lead to a deeper understanding of customer preferences, market trends, and other crucial factors

Disadvantages of NLP:

1. Ambiguity: Natural language is inherently ambiguous, and NLP systems may struggle to accurately interpret context or resolve ambiguities. Ambiguities can arise from homonyms, synonyms, or the use of language with multiple meanings.
2. Data Limitations: The performance of NLP models heavily depends on the quality and quantity of training data. Insufficient or biased datasets may lead to poor performance or biased outcomes, impacting the reliability of NLP applications.

NLP in Gate Pass Management:

In the specific context of the Gate Pass Management System, incorporating NLP introduces an intelligent layer that goes beyond conventional request processing. By leveraging NLP, the system gains the ability to analyze the reasons provided by students when making gate pass requests, thereby introducing a more sophisticated and nuanced approach to request prioritization.

Request Analysis with NLP:

NLP processes the textual input provided by students, delving into the context, sentiments, and key keywords within the text to derive a nuanced understanding of the purpose behind each gate pass request. This involves training NLP algorithms to interpret the intricacies of natural language, enabling the system to recognize diverse reasons students may cite for their requests. Whether the reasons are related to personal matters, academic commitments, or emergencies, NLP aids in comprehending the underlying context.

Priority Assignment with NLP:

Based on the NLP analysis, the system automatically assigns one of three priority levels—Low, Medium, or High—to each gate pass request. This classification is determined by the perceived urgency and significance derived from the content of the student's reason. The advanced capabilities of NLP contribute to a more sophisticated and context-aware prioritization process.

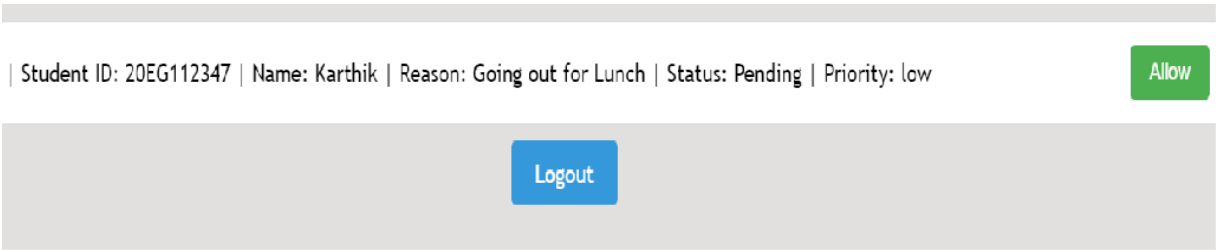

Fig 6 : Priority of the request.

Here in this Fig 6 it displays the priority of the requests sent by the students.

Efficient Faculty Decision-Making with NLP:

The automated priority assignment facilitated by NLP brings efficiency to the decision-making process for faculty members. They can readily observe the assigned priority level when reviewing requests. This classification aids faculty in quickly identifying and addressing high-priority requests, ensuring a timely response to critical situations. NLP enhances the gate pass management system's ability to efficiently process and prioritize requests, improving overall system effectiveness.

The incorporation of NLP into the Gate Pass Management System introduces a layer of intelligence that significantly enhances the system's ability to understand and prioritize requests based on the nuanced content provided by students. The applications, advantages, and disadvantages of NLP broadly apply to this specific context, contributing to a more sophisticated and efficient gate pass management process.

spaCy Integration in QR Code Management System

In our QR Code Management System, we have integrated the spaCy library to enhance the efficiency of processing user requests for gate passes. spaCy is a powerful natural language processing (NLP) library in Python that offers robust capabilities for text processing and understanding.

Request Prioritization

When a user submits a request for a gate pass by providing a reason, spaCy comes into play to extract keywords and analyze the request. These keywords are crucial in determining the priority of the request. By leveraging spaCy's linguistic models and algorithms, we assign a priority level (1, 2, or 3) to each request based on the significance of the extracted keywords.

Keyword Extraction

spaCy enables us to efficiently extract relevant keywords from the user-provided reason for requesting a gate pass. These keywords serve as indicators of the urgency, importance, or specific nature of the request. By accurately identifying and analyzing these keywords, we ensure that our prioritization algorithm makes informed decisions.

Priority Assignment

Once the keywords are extracted, our system employs a prioritization algorithm that takes into account various factors such as the frequency of keywords, their semantic relevance, and predefined criteria for priority assignment. The result is a transparent and systematic approach to prioritizing gate pass requests, ensuring that urgent and critical requests are addressed promptly.

Implementation

Integration with Existing Workflow

Integrating spaCy into our QR Code Management System was a seamless process. We leveraged spaCy's user-friendly API and extensive documentation to incorporate NLP capabilities into our existing workflow. The modular design of spaCy allowed us to tailor its functionality to suit our specific requirements, enabling smooth interaction with our application.

Customization and Fine-Tuning

To optimize the performance of our prioritization system, we fine-tuned spaCy's models and parameters to align with the domain-specific language and context of gate pass requests. This customization ensures that spaCy accurately captures the nuances of user input and delivers reliable keyword extraction and prioritization results.

Benefits

Enhanced Efficiency

By automating the process of request prioritization using spaCy, we have significantly improved the efficiency of our gate pass management system. The ability to quickly analyze and prioritize requests based on their content allows us to streamline the workflow and allocate resources more effectively.

Improved User Experience

With spaCy-powered prioritization, users experience faster response times and greater transparency in the handling of their gate pass requests. By assigning priority levels based on objective criteria derived from the request content, we ensure fairness and consistency in our decision-making process.

Scalability and Adaptability

The scalability and adaptability of spaCy make it an ideal choice for our dynamic QR Code Management System. As our system evolves and grows, spaCy provides a flexible framework that can accommodate new features, languages, and use cases with minimal overhead.

Integration of spaCy into our QR Code Management System has been instrumental in optimizing the process of request prioritization and enhancing the overall user experience. By harnessing the power of natural language processing, we have established a robust framework for efficient and intelligent handling of gate pass requests.

5.3 LIBRARIES USED

1. Flask (2.1.3):

Flask serves as the foundational web framework for the project. Known for its lightweight nature and user-friendly design, Flask handles crucial aspects of web application development, including managing HTTP requests, routing, and facilitating interactions with the MongoDB database. Its simplicity makes it an excellent choice for developing the backbone of the application, ensuring a responsive and efficient user experience.

2. Flask-PyMongo (2.3.0):

This extension seamlessly integrates MongoDB with Flask, allowing the Gate Pass Management System to store and retrieve data efficiently. Flask-PyMongo streamlines the interaction between the Flask application and the MongoDB database, providing a scalable and accessible solution for managing gate pass information.

3. qrcode[pil] (7.3):

The qrcode library, along with the Pillow (PIL) dependency, enables the generation of QR codes within the application. QR codes play a pivotal role in representing digital gate passes, containing encrypted information about individuals and their authorization to access specific areas or exit the premises.

4. gunicorn (21.2.0):

Gunicorn is a reliable WSGI HTTP server for running the Flask application in a production environment. It ensures the application's stability and performance, handling concurrent requests efficiently and providing a robust foundation for deployment.

5. Werkzeug (2.2.2):

Werkzeug is a WSGI utility library for Python, used in conjunction with Flask to handle various tasks such as routing, request handling, and HTTP utilities. It enhances the capabilities of Flask, contributing to the overall functionality and reliability of the web application.

6. pyzbar:

Pyzbar is a Python library that facilitates the decoding of QR codes. In the context of the Gate Pass Management System, pyzbar is employed to scan and interpret QR codes, ensuring a secure and efficient exit tracking process.

7. Flask-SocketIO (5.3.6):

Flask-SocketIO extends Flask to support WebSocket communication. This technology is crucial for enabling real-time features within the application, such as notifications or updates, enhancing the user experience.

8. spacy (2.2.4):

Spacy is a natural language processing (NLP) library that may be utilized for various language-related tasks. While the specific use of Spacy in the Gate Pass Management System is not explicitly detailed, its inclusion suggests the potential for advanced language processing capabilities within the application.

9. Flask-Session:

Flask-Session is an extension that adds support for server-side sessions in Flask applications. It allows the system to manage user sessions, providing a secure and efficient way to handle user-specific data and interactions.

10. matplotlib:

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. In the Gate Pass Management System, matplotlib may be employed for data visualization or graphical representation of information related to gate pass requests, approvals, or other relevant metrics.

11. flask_mail:

Flask-Mail is an extension for Flask that simplifies email integration. It enables the system to send email notifications, which can be a valuable feature for alerting users or administrators about important events or updates within the gate pass management process.

The technologies stack chosen for the Gate Pass Management System reflects a well-rounded selection of tools and libraries, each serving a specific purpose to ensure the system's functionality, security, and user experience. From web framework and database integration to QR code generation and real-time communication, these technologies collectively contribute to the success of the project.

5.4 Sample Code

```
from flask import *
from flask_pymongo import PyMongo
from flask_session import Session
import random
from bson import ObjectId
import qrcode
from io import BytesIO
from datetime import datetime
from pyzbar import *
from flask_socketio import SocketIO
import spacy
import matplotlib.pyplot as plt
from io import BytesIO
import base64
from werkzeug.utils import secure_filename
import os
from flask_mail import Mail, Message
import time


app = Flask(_name_)
socketio = SocketIO(app)
UPLOAD_FOLDER = 'photos'
ALLOWED_EXTENSIONS = {'jpg'}

app.config['MONGO_URI'] =
'mongodb+srv://pingalipraneeth1:DgCwSk9Cn9mTx32a@augatepass.1dvhlzv.mongodb.net/g
atepass_db?retryWrites=true&w=majority'
app.config['SECRET_KEY'] = 'your_secret_key'
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
mongo = PyMongo(app)
nlp = spacy.load("en_core_web_sm")

app.config["SESSION_PERMANENT"] = False
app.config["SESSION_TYPE"] = "filesystem"
Session(app)

app.config['MAIL_SERVER'] = 'smtp.gmail.com'
app.config['MAIL_PORT'] = 587
```

```
app.config['MAIL_USE_TLS'] = True
app.config['MAIL_USERNAME'] = 'poppingaming1@gmail.com'
app.config['MAIL_PASSWORD'] = 'atjj cynj vkwt ljkn'

mail = Mail(app)

@app.after_request
def add_cache_control(response):
    response.headers["Cache-Control"] = "no-cache, no-store, must-revalidate"
    response.headers["Pragma"] = "no-cache"
    response.headers["Expires"] = "0"
    return response

def prioritize_text(text):
    doc = nlp(text.lower())

    priority_keywords = {
        "urgent": 3,
        "health": 2,
        "emergency": 3,
        "family emergency": 2,
        "personal reasons": 1,
        "vacation": 1,
        "wedding": 1,
        "birth of a child": 1,
        "fever": 2,
        "headache": 2,
        "stomach pain": 2,
        "educational purposes": 1,
        "unplanned event": 1,
        "death": 3,
        "injury": 3,
        "exam": 3,
        "education": 3,
    }

    priority_labels = {
    3: "high",
    2: "medium",
    1: "low",
    }

    priority = 1

    for keyword, weight in priority_keywords.items():
        if keyword in doc.text:
            priority = max(priority, weight)

    return priority_labels[priority]
```

```
@app.route('/', methods=['GET', 'POST'])
def login():
    if session.get("login_type"):
        if (session["login_type"] == 'wrong'):
            return redirect('/wrong')
        if (session["login_type"] == "student"):
            return redirect(url_for('student'))
        if (session["login_type"] == "faculty"):
            return redirect(url_for('faculty'))
        if (session["login_type"] == "security"):
            return redirect(url_for('security'))

    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        login_type = request.form['login_type']
        session["username"] = request.form['username']
        session["login_type"] = request.form['login_type']

        user_data = None
        if login_type == 'student':
            user_data = mongo.db.studentdata.find_one({'username': username, 'password':
password})
            user = mongo.db.students.find_one({'username': username})
            if user_data:
                session['name'] = user.get('name', '')
                fac=mongo.db.students.find_one({'username': session['username']})
                session['mentor'] = fac.get("faculty")
        elif login_type == 'faculty':
            user_data = mongo.db.facultydata.find_one({'username': username, 'password':
password})
        elif login_type == 'security':
            user_data = mongo.db.securitydata.find_one({'username': username, 'password':
password})

        if user_data:
            session['login_type'] = login_type
            if login_type == 'student':
                return redirect(url_for('student'))
            elif login_type == 'faculty':
                return redirect(url_for('faculty'))
            elif login_type == 'security':
                return redirect(url_for('security'))
        else:
            session["username"] = None
            session["name"] = None
            session["login_type"] = 'wrong'
            return redirect("/")

    return render_template('index.html')
```

```
@app.route("/logout")
def logout():
    session["login_type"] = None
    session["username"] = None
    session["name"] = None

    return redirect("/")

def allowed_file(filename):
    return '.' in filename and \
           filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        login_type = request.form['login_type']
        photo = request.files['photo']

        # Check if the photo has been uploaded
        if photo.filename == '':
            return "No selected file"
        if photo and allowed_file(photo.filename):
            filename = secure_filename(photo.filename)
            # Rename the file to the username before saving
            filename = secure_filename(username) + os.path.splitext(filename)[1]
            photo.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
        else:
            return "Unsupported file format. Please upload an image."

        existing_user = None
        if login_type == 'student':
            existing_user = mongo.db.studentdata.find_one({'username': username})
        elif login_type == 'faculty':
            existing_user = mongo.db.facultydata.find_one({'username': username})
        elif login_type == 'security':
            existing_user = mongo.db.securitydata.find_one({'username': username})
        if existing_user:
            return "Username already exists. Please choose a different username."
        if login_type == 'student':
            mongo.db.studentdata.insert_one({'username': username, 'password': password, 'photo':
filename})
        elif login_type == 'faculty':
            mongo.db.facultydata.insert_one({'username': username, 'password': password, 'photo':
filename})
        elif login_type == 'security':
            mongo.db.securitydata.insert_one({'username': username, 'password': password,
'photo': filename})
        return "Registration successful. You can now log in."
```

```
    return render_template('register.html')

@app.route('/student', methods=['GET', 'POST'])
def student():
    if 'login_type' not in session or session['login_type'] != 'student':
        return redirect(url_for('login'))

    if request.method == 'POST':
        student_id = session['username']
        name = session['name']
        reason = request.form['reason']
        priority = prioritize_text(reason)
        current_date = datetime.now().date().strftime('%d-%m-%Y')
        fac=mongo.db.students.find_one({'username': session['username']})
        facc=fac.get("faculty")
        checkk="False"
        mongo.db.requests.insert_one({'student_id': student_id, 'name': name, 'reason': reason, 'status': 'Pending', 'datetime': current_date, 'priority': priority, 'faculty': facc, 'checkedout': checkk, 'checkouttime': "Null"})
        return redirect(url_for('student'))

    return render_template('student.html')

@app.route('/photos/<path:filename>')
def photos(filename):
    return send_from_directory('photos', filename)


@app.route('/faculty', methods=['GET', 'POST'])
def faculty():
    if 'login_type' not in session or session['login_type'] != 'faculty':
        return redirect(url_for('login'))

    if request.method == 'POST':
        request_id = request.form['request_id']
        action = request.form['action']

        if action == 'allow':
            random_key = str(random.randint(100000, 999999))
            mongo.db.requests.update_one({'_id': ObjectId(request_id)}, {'$set': {'status': 'Approved', 'key': random_key}})

        elif action == 'deny':
            mongo.db.requests.update_one({'_id': ObjectId(request_id)}, {'$set': {'status': 'Denied', 'key': None}})

        return redirect(url_for('faculty'))


    requests = mongo.db.requests.find({'status': 'Pending', 'faculty': session['username']})
```

```
    return render_template('faculty.html', requests=requests)

@app.route('/security', methods=['GET', 'POST'])
def security():
    if 'login_type' not in session or session['login_type'] not in ['student', 'faculty', 'security']:
        return redirect(url_for('login'))

    if request.method == 'POST':
        if request.form['action'] == 'entry':

            name = request.form['name']
            reason = request.form['reason']
            number = request.form['number']
            current_datetime = datetime.now().strftime('%d-%m-%Y %H:%M:%S')
            mongo.db.visitors.insert_one({'name': name, 'reason': reason, 'number': number,
'datetime': current_datetime, 'checkout': False})

            return redirect(url_for('security'))

    return render_template('security.html')
```

6. TESTING

6.1 Software Testing

Software testing is the process of evaluating a software item to detect differences between given input and expected output. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words, software testing is a verification and validation process.

Verification:

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

Validation:

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

Basics of software testing

There are two basics of software testing: Black box testing and white box testing.

Black box Testing

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

White box Testing

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing.

Black box testing is often used for validation and white box testing is often used for verification.

6.2. TYPES OF TESTING

There are different types of testing

- Unit Testing
- Integration Testing
- Functional Testing
- System Testing
- Regression Testing

Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

Functional Testing

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

System Testing

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

6.3 Test Cases

Tested	Test Cases	Input	Output	Result
1	Faculty Approval of a Request	Valid student username and password	Successful login, redirect to student dashboard.	Pass
2	Security Scanning a Valid QR Code	Faculty username and password, request ID to approve	Request status changes to "Approved", QR code generated, reflected in student's view requests.	Pass
3	Generating Request by Student	Valid QR code from an approved request	Request details displayed, option to mark as checked out.	Pass
4	Viewing Statistics by Faculty	Student username and password, reason for request	Request is added to the system with pending status, visible in student's request history.	Pass
5	Entry Log Management by Security	Faculty username and password	Statistical data displayed, such as number of requests per day in the present month.	Pass
6	Invalid Login Credentials	Security username and password	Ability to enter visitor data into the system, view visitor log, and mark visitors as checked out.	Pass

7	Denying a Request by Faculty	Invalid username and password combination for any user type	Login attempt fails, appropriate error message displayed.	Pass
8	Denying a Request by Faculty	Invalid username and password combination for any user type	Login attempt fails, appropriate error message displayed.	Pass
9	Simultaneous Request Generation by Multiple Students	QR code from a request that has already been checked out	Error message indicating the QR code has expired.	Pass

7. USER INTERFACE

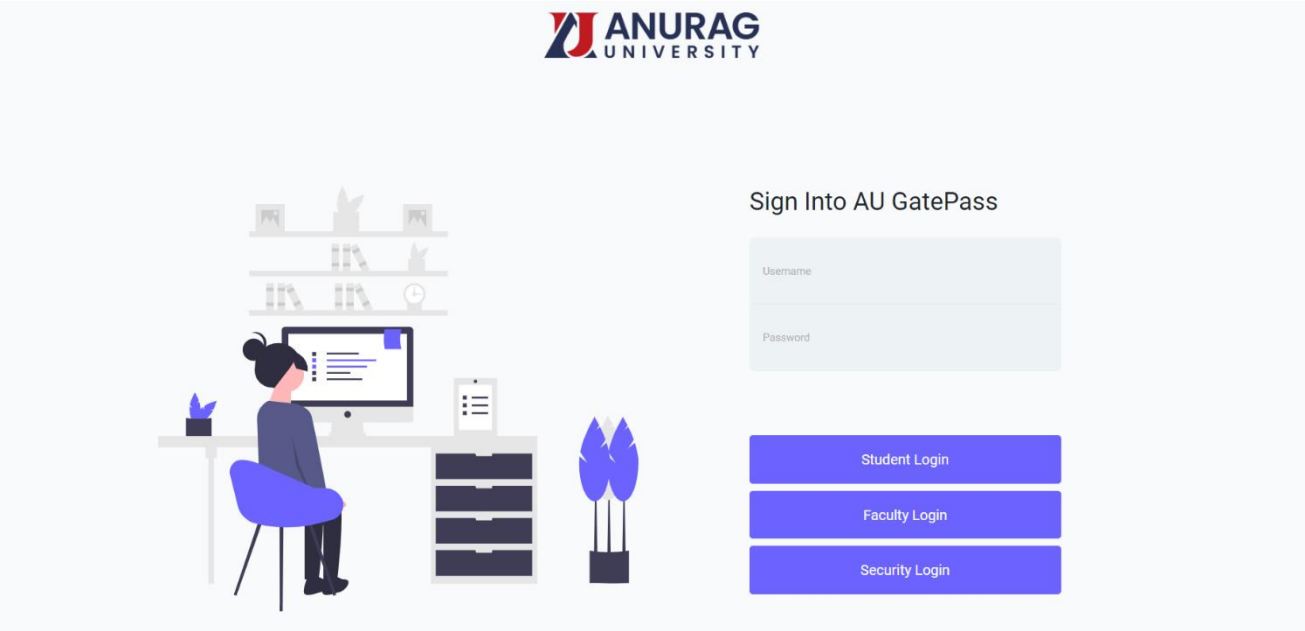

Fig 7 : Home Page of AU Gate Pass

Fig 7 displays the home page of QR Code Based Gate Pass Management System. It consists of three users to login

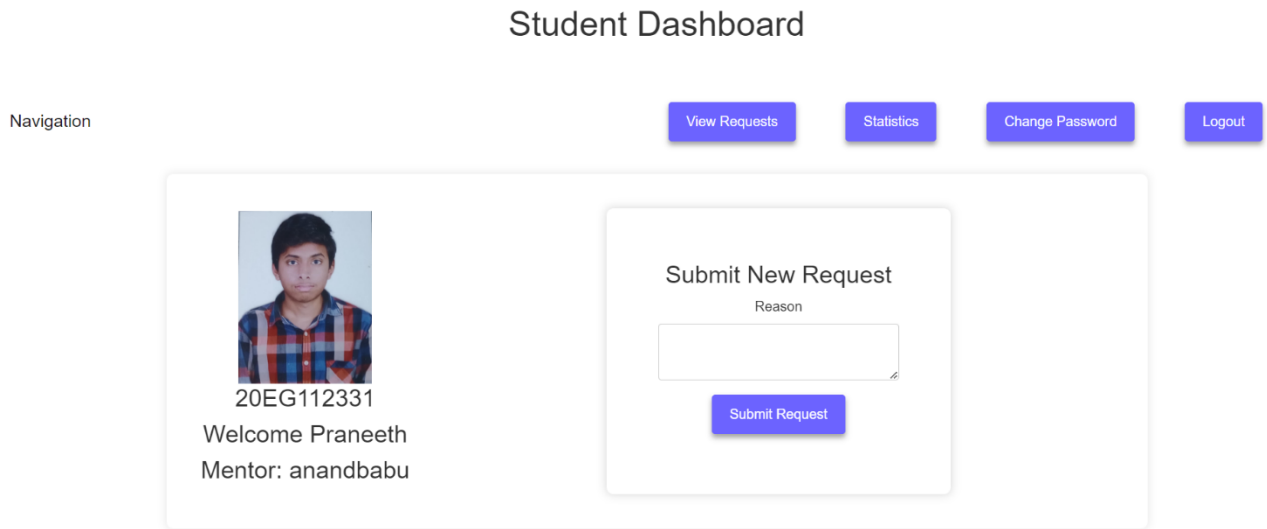

Fig 8 : Student Dashboard

Fig 8 displays the student dashboard of QR Code Based Gate Pass Management System. Students Dashboard allows students to request gate pass from the faculty. They can view the status of their requests and the statistics of their requests.

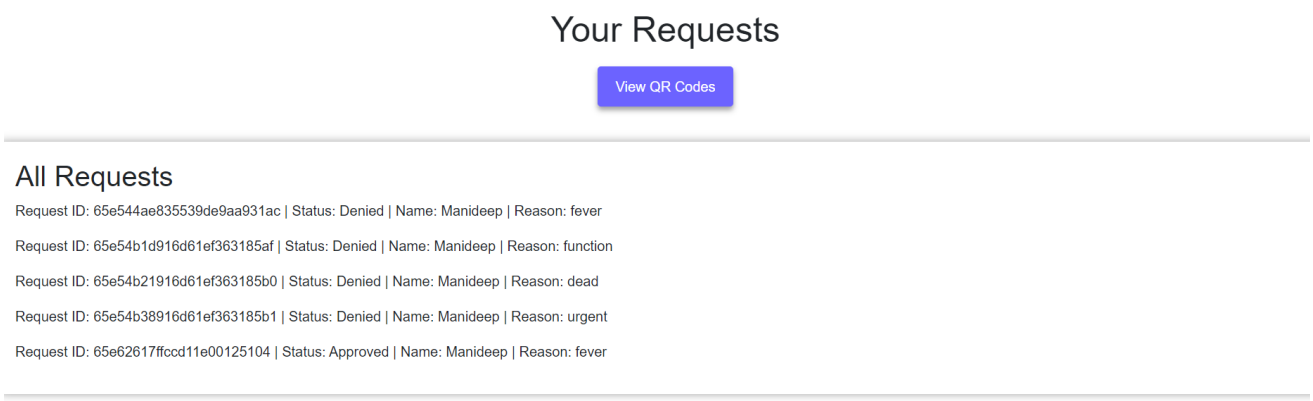

Fig 9 : Student Requests

Fig 9 displays the requested requests from a strudent, where we can see the reason and the status of the request.

Fig 10 : Faculty Dashboard

Fig 10 displays the faculty dashboard, where the faculty has the authority to accept and deny the requests of the students. They can see the reason and the priority of the request.

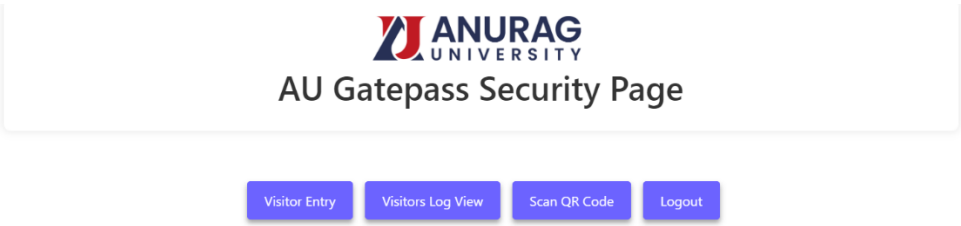

Fig 11 : Security Dashboard

Fig 11 displays the security dashboard, where it contains visitor entry for the visitors into the campus. There is visitor log for the security personnel has access to view the visitors who are in the campus and the main feature is to scan the qr code of the students for the exit from the campus.

Fig 12 : Visitor Entry Form

Fig 12 displays the visitor entry form where the visitor needs to enter their basic details and a authentic reason.

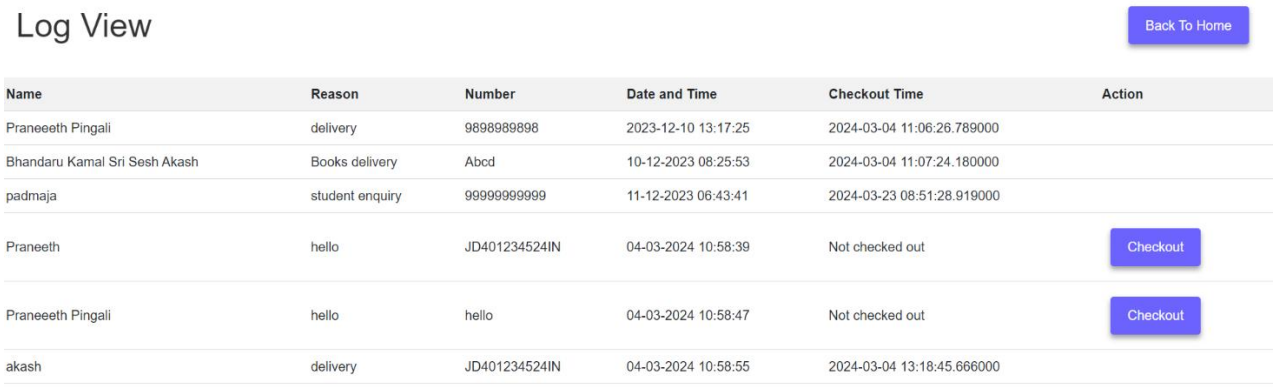

Fig 13 : Visitor Log View

Fig 13 displays the visitor log view where the security personnel has access of looking to the visitors status in the campus.

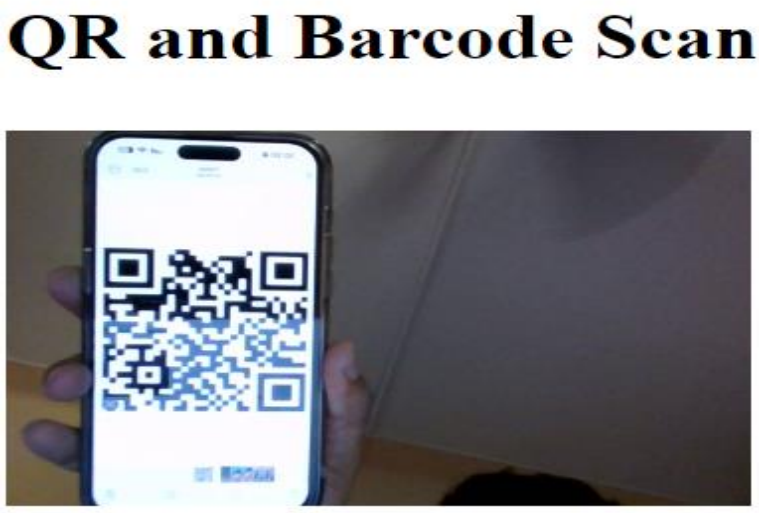

Fig 14 : QR and Barcode Scanner

Fig 14 displays the qr code scanner where the smartphone consists of a valid authentic qr code which is scanned by the security to get exit from the campus.

Requests Received by Date for Student ID:

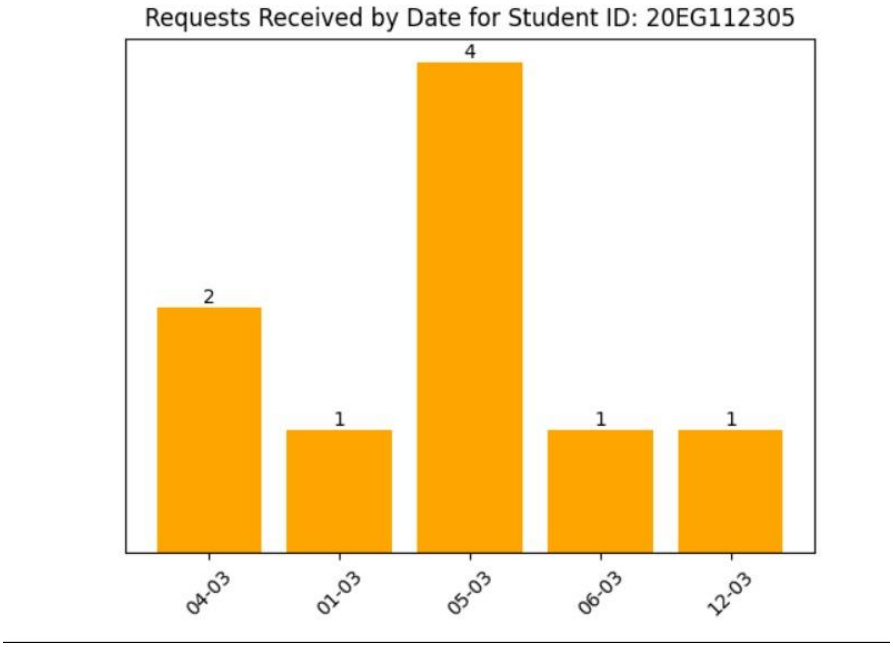

Fig 15 : Bar chart of the requests received by date for student id using matplotlib.

Here in the Fig 15 the bar chart shows the requests received by the user 20EG112305 on different days .

All Requests Received by Date

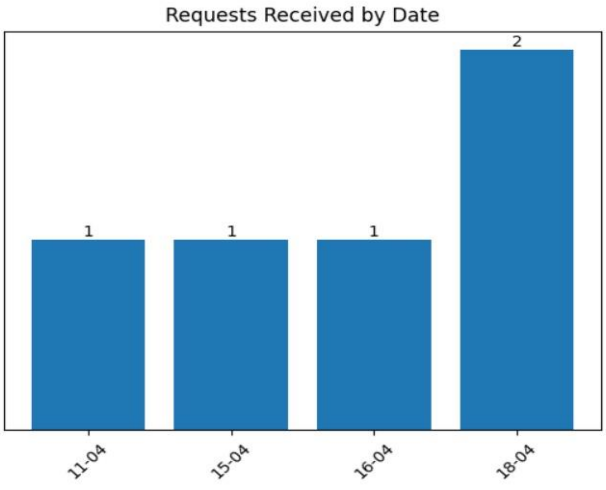

Fig 16: Bar chart of the all requests received by date using matplotlib.

Here in the Fig 16 the bar chart displays the stats of the requests received by all the al the students date wise.

8. CONCLUSION AND FUTURE ENHANCEMENT

8.1 CONCLUSION

In conclusion, the AU Gate Pass presents a comprehensive and innovative solution to enhance security, efficiency, and user experience within university campuses. By transitioning from traditional paper-based gate pass systems to a digital framework centered around QR codes, the project introduces a streamlined approach to access control. The system addresses the shortcomings of existing methods, such as time-consuming manual processes, susceptibility to errors, and the ease of forgery in paper passes.

The integration of Flask as the web framework, MongoDB Atlas as the cloud-based database, and various Python libraries underscores the technological robustness of the solution. The addition of Natural Language Processing (NLP) introduces an intelligent layer for prioritizing gate pass requests based on the articulated reasons, facilitating efficient decision-making for faculty members. Furthermore, the project's scope extends beyond students to include faculty and security personnel, fostering a holistic approach to gate pass management. Real-time monitoring, visitor logs, and the ability to generate QR codes through the application streamline processes and contribute to a more organized campus environment.

The sequence diagram illustrates the meticulous process of security scanning QR codes, emphasizing the secure and efficient exit verification procedures. The technologies utilized, including Flask, MongoDB Atlas, and NLP, collectively contribute to the success of the project, ensuring a responsive, scalable, and secure system. In essence, the QR Code Based Gatepass Management System not only modernizes access control within educational institutions but also sets a benchmark for security and efficiency. The implementation of QR codes not only reduces the environmental impact of excessive paperwork but also aligns with the growing reliance on digital solutions. Overall, this project signifies a significant leap forward in optimizing gate pass management processes, offering a model that can be adapted and implemented across various educational institutions.

8.2 FUTURE WORKS

The AU Gate Pass lays the foundation for several future advancements and expansions. The project's modular architecture and integration of cutting-edge technologies provide ample opportunities for growth and refinement. Here are some potential future scopes for the system:

1. Integration of Biometric Authentication: Enhancing security measures by incorporating biometric authentication methods, such as fingerprint or facial recognition, can add an extra layer of identity verification, making the system even more robust and secure.

2. Mobile App Development: Developing dedicated mobile applications for Android and iOS platforms can extend the accessibility of the gate pass system. Mobile apps would enable users to manage gate pass requests, view status, and generate QR codes conveniently from their smartphones.

3. Machine Learning for Predictive Analytics: Implementing machine learning algorithms can help in predictive analytics for anticipating peak exit times, identifying patterns in gate pass requests, and optimizing the allocation of security personnel resources based on historical data.

4. Enhanced Reporting and Analytics: Integrating advanced reporting and analytics features can provide in-depth insights into gate pass trends, security-related incidents, and overall system performance. Customizable dashboards can be designed for different user roles, including administrators, faculty, and security personnel.

5. Multi-Factor Authentication (MFA): Strengthening security by incorporating multi-factor authentication methods, such as combining QR codes with one-time passwords (OTPs) sent to users' mobile devices, can further fortify the authentication process.

9. REFERENCES

1. Prof. Archana S. Banait, Ms. Neha Buwa, Ms. Varsha Jadhav, Ms. Pooja Ganate, Ms. Shubhangi Dagale. "Gate-pass Management System," International Journal of Science & Engineering Development Research.
2. Bose, Dr. Rajesh and Aithal, P. S. and Roy, Sandip. Survey of Twitter Viewpoint on Application of Drugs by VADER Sentiment Analysis among Distinct Countries.
3. Kaushik, Vikrant Kumar, A. Gupta, A. Kumar and Abhishek Prasad. "Student Leave Management System," Read Paper
4. Harish Rapartiwar, Pushpanjali Shivratri, Omkar Sonakul, Prof. Ashwini Bhugul. "Visitor Gate Pass Management System