GECTFMA:Facility Management App

Basila S Nazer(TCR21CS016)
Anagha Jayan(TCR21CS009)
Aparna Krishnan(TCR21CS015)
Jithin P Kumar (TCR21CS027)

Department of Computer Science & Engineering
Government Engineering College, Thrissur

April, 2024

1. Problem Description

2. Project Objectives

3. Solution Strategy

4. Design Components

5. Implementation

6. Result Analysis

7. Future Plan

1 Problem Description

2 Project Objectives

3 Solution Strategy

4 Design Components

5 Implementation

6 Result Analysis

7 Future Plan

The Problem

- Managing plumbing and electrical complaints on college campuses presents persistent challenges, hindering facility operations.
- Manual processes and ineffective communication channels contribute to significant delays in addressing maintenance issues.
- Miscommunication among stakeholders results in misunderstandings and a lack of transparency regarding complaint status.
- There is an urgent need for a comprehensive solution to revolutionize maintenance complaint management within college campuses.

The Problem

- Managing plumbing and electrical complaints on college campuses presents persistent challenges, hindering facility operations.
- Manual processes and ineffective communication channels contribute to significant delays in addressing maintenance issues.
- Miscommunication among stakeholders results in misunderstandings and a lack of transparency regarding complaint status.
- There is an urgent need for a comprehensive solution to revolutionize maintenance complaint management within college campuses.

The Problem

- Managing plumbing and electrical complaints on college campuses presents persistent challenges, hindering facility operations.
- Manual processes and ineffective communication channels contribute to significant delays in addressing maintenance issues.
- Miscommunication among stakeholders results in misunderstandings and a lack of transparency regarding complaint status.
- There is an urgent need for a comprehensive solution to revolutionize maintenance complaint management within college campuses.

The Problem

- Managing plumbing and electrical complaints on college campuses presents persistent challenges, hindering facility operations.
- Manual processes and ineffective communication channels contribute to significant delays in addressing maintenance issues.
- Miscommunication among stakeholders results in misunderstandings and a lack of transparency regarding complaint status.
- There is an urgent need for a comprehensive solution to revolutionize maintenance complaint management within college campuses.

1. Problem Description
2. **Project Objectives**
3. Solution Strategy
4. Design Components
5. Implementation
6. Result Analysis
7. Future Plan

Objectives

- Streamline Complaint Management: Develop a system to efficiently report, track, and resolve plumbing and electrical complaints within college campuses.
- Ensure Security and Privacy: Implement robust authentication mechanisms to safeguard user data and ensure the security of information shared within the application.
- Improve User Experience: Design an intuitive mobile application interface to streamline complaint submission, tracking, and interaction for all users, enhancing overall user experience and satisfaction.
- Enhance Accountability: Establish mechanisms to track the progress of complaints and assign responsibilities to relevant personnel, promoting accountability and transparency in the resolution process.

Objectives

- Streamline Complaint Management: Develop a system to efficiently report, track, and resolve plumbing and electrical complaints within college campuses.
- Ensure Security and Privacy: Implement robust authentication mechanisms to safeguard user data and ensure the security of information shared within the application.
- Improve User Experience: Design an intuitive mobile application interface to streamline complaint submission, tracking, and interaction for all users, enhancing overall user experience and satisfaction.
- Enhance Accountability: Establish mechanisms to track the progress of complaints and assign responsibilities to relevant personnel, promoting accountability and transparency in the resolution process.

Objectives

- Streamline Complaint Management: Develop a system to efficiently report, track, and resolve plumbing and electrical complaints within college campuses.
- Ensure Security and Privacy: Implement robust authentication mechanisms to safeguard user data and ensure the security of information shared within the application.
- Improve User Experience: Design an intuitive mobile application interface to streamline complaint submission, tracking, and interaction for all users, enhancing overall user experience and satisfaction.
- Enhance Accountability: Establish mechanisms to track the progress of complaints and assign responsibilities to relevant personnel, promoting accountability and transparency in the resolution process.

Objectives

- Streamline Complaint Management: Develop a system to efficiently report, track, and resolve plumbing and electrical complaints within college campuses.
- Ensure Security and Privacy: Implement robust authentication mechanisms to safeguard user data and ensure the security of information shared within the application.
- Improve User Experience: Design an intuitive mobile application interface to streamline complaint submission, tracking, and interaction for all users, enhancing overall user experience and satisfaction.
- Enhance Accountability: Establish mechanisms to track the progress of complaints and assign responsibilities to relevant personnel, promoting accountability and transparency in the resolution process.

1 Problem Description

2 Project Objectives

3 Solution Strategy

4 Design Components

5 Implementation

6 Result Analysis

7 Future Plan

Solution Strategy

- GECTFMA Mobile App: Our full-stack mobile application serves as the central platform for managing maintenance complaints.
- Complaint Registration and Worker Assignment: The app allows HODs of all departments to register complaints and it would be verified by the Plumbing in-charge and electrical HOD and later the sergent can assign workers.
- HODs can submit maintenance complaints directly through the app. These complaints are then processed and managed within the system, ensuring timely resolution.
- Real-time Feedback and Analysis: The app provides real-time feedback mechanisms, allowing HODs to track the status of their complaints and receive updates on their resolution progress.

Solution Strategy

- GECTFMA Mobile App: Our full-stack mobile application serves as the central platform for managing maintenance complaints.
- Complaint Registration and Worker Assignment: The app allows HODs of all departments to register complaints and it would be verified by the Plumbing in-charge and electrical HOD and later the sergent can assign workers.
- HODs can submit maintenance complaints directly through the app. These complaints are then processed and managed within the system, ensuring timely resolution.
- Real-time Feedback and Analysis: The app provides real-time feedback mechanisms, allowing HODs to track the status of their complaints and receive updates on their resolution progress.

Solution Strategy

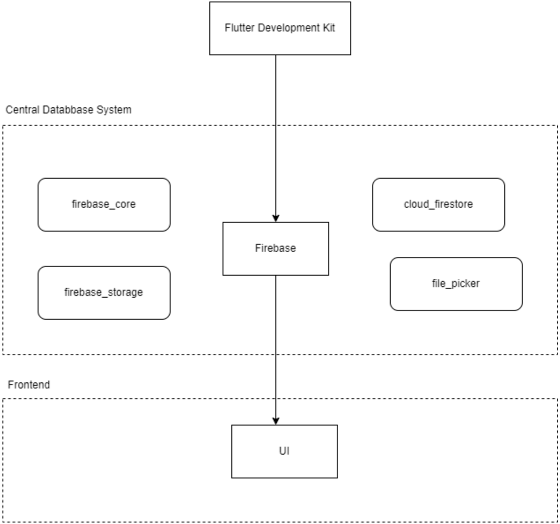
- GECTFMA Mobile App: Our full-stack mobile application serves as the central platform for managing maintenance complaints.
- Complaint Registration and Worker Assignment: The app allows HODs of all departments to register complaints and it would be verified by the Plumbing in-charge and electrical HOD and later the sergent can assign workers.
- HODs can submit maintenance complaints directly through the app. These complaints are then processed and managed within the system, ensuring timely resolution.
- Real-time Feedback and Analysis: The app provides real-time feedback mechanisms, allowing HODs to track the status of their complaints and receive updates on their resolution progress.

Solution Strategy

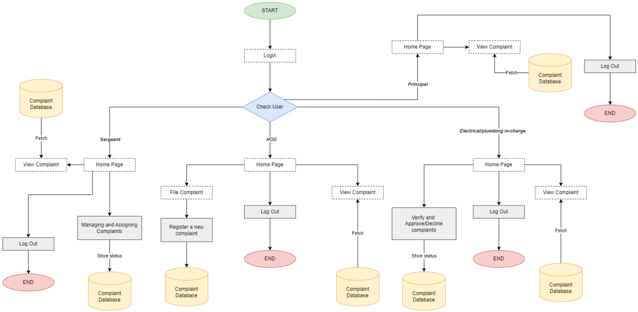
- GECTFMA Mobile App: Our full-stack mobile application serves as the central platform for managing maintenance complaints.
- Complaint Registration and Worker Assignment: The app allows HODs of all departments to register complaints and it would be verified by the Plumbing in-charge and electrical HOD and later the sergent can assign workers.
- HODs can submit maintenance complaints directly through the app. These complaints are then processed and managed within the system, ensuring timely resolution.
- Real-time Feedback and Analysis: The app provides real-time feedback mechanisms, allowing HODs to track the status of their complaints and receive updates on their resolution progress.

1. Problem Description
2. Project Objectives
3. Solution Strategy
4. Design Components
5. Implementation
6. Result Analysis
7. Future Plan

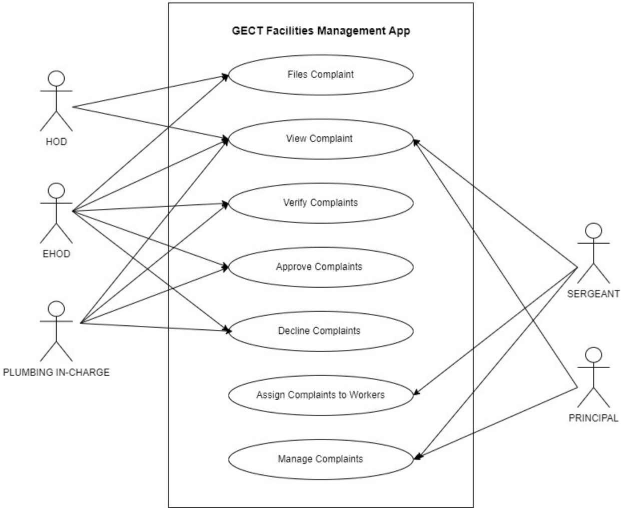
Architecture Diagram



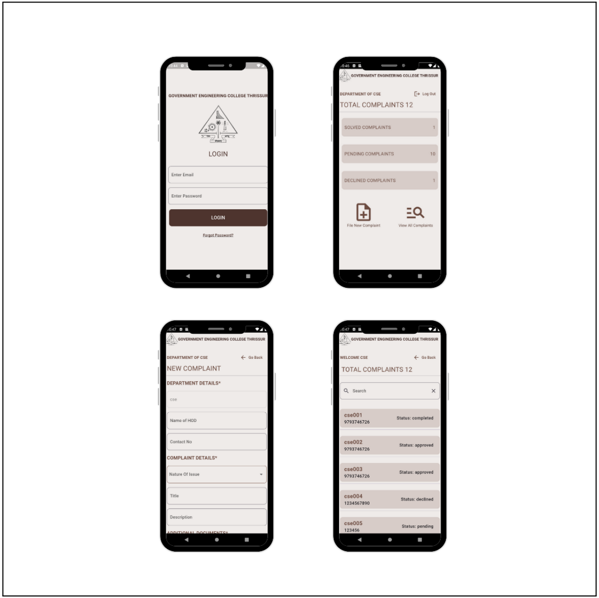
Workflow Diagram



Use Case Diagram



User Interface



GOVERNMENT ENGINEERING COLLEGE THRISSUR

LOGIN

Enter Email

Enter Password

LOGIN

Forgot Password?

GOVERNMENT ENGINEERING COLLEGE THRISSUR

DEPARTMENT OF CSE
Log Out

TOTAL COMPLAINTS 12

SOLVED COMPLAINTS	1
PENDING COMPLAINTS	10
DECLINED COMPLAINTS	1

File New Complaint
View All Complaints

GOVERNMENT ENGINEERING COLLEGE THRISSUR

DEPARTMENT OF CSE
Go Back

NEW COMPLAINT

DEPARTMENT DETAILS*

cse

Name of HOD

Contact No

COMPLAINT DETAILS*

Nature Of Issue

Title

Description

ADDITIONAL DOCUMENTS*

GOVERNMENT ENGINEERING COLLEGE THRISSUR

WELCOME CSE
Go Back

TOTAL COMPLAINTS 12

Search

cse001 9793746726	Status: completed
cse002 9793746726	Status: approved
cse003 9793746726	Status: approved
cse004 1234567890	Status: declined
cse005 123456	Status: pending

1 Problem Description

2 Project Objectives

3 Solution Strategy

4 Design Components

5 Implementation

6 Result Analysis

7 Future Plan

Authentication System

- Design: The design proposed Firebase Authentication for user authentication, ensuring secure access to the system.
- Implementation: The implementation confirms the utilization of Firebase Authentication, aligning with the design's choice for robust encryption protocols and seamless integration with both frontend and backend systems.

Authentication System

- Design: The design proposed Firebase Authentication for user authentication, ensuring secure access to the system.
- Implementation: The implementation confirms the utilization of Firebase Authentication, aligning with the design's choice for robust encryption protocols and seamless integration with both frontend and backend systems.

Database Management

- Design: The design suggested Firebase Firestore for the database system, providing scalability and real-time data synchronization.
- Implementation: The implementation effectively utilized Cloud Firestore, leveraging its NoSQL capabilities for efficient data storage and retrieval.

Database Management

- Design: The design suggested Firebase Firestore for the database system, providing scalability and real-time data synchronization.
- Implementation: The implementation effectively utilized Cloud Firestore, leveraging its NoSQL capabilities for efficient data storage and retrieval.

UI/UX Enhancement

- Design: The design outlined intuitive UI/UX features and principles, including responsive design and clear navigation pathways, to enhance user experience.
- Implementation: During implementation, these design principles were put into practice, resulting in tangible enhancements to the interface. The implementation focused on optimizing layout designs, improving color schemes, and ensuring consistent UI elements across different screens, thereby enhancing user interaction and satisfaction.

UI/UX Enhancement

- Design: The design outlined intuitive UI/UX features and principles, including responsive design and clear navigation pathways, to enhance user experience.
- Implementation: During implementation, these design principles were put into practice, resulting in tangible enhancements to the interface. The implementation focused on optimizing layout designs, improving color schemes, and ensuring consistent UI elements across different screens, thereby enhancing user interaction and satisfaction.

1. Problem Description
2. Project Objectives
3. Solution Strategy
4. Design Components
5. Implementation
6. Result Analysis
7. Future Plan

Correctness Analysis

- Functionality Verification: Unit testing was utilized to verify the functionality of individual components and modules within the application, ensuring that all components function correctly and handle edge cases effectively.
- Integration Testing: It focused on evaluating interactions between different components to ensure smooth data flow and proper response handling between frontend and backend components.
- User Acceptance Testing(UAT): Real user testing validated the application against user requirements, ensuring correctness in terms of meeting user expectations and addressing any encountered issues.
- Performance Testing: It ensured correctness in terms of system response time, scalability , and resource usage, validating that the system performs as expected under various scenarios.
- Feedback Incorporation:Customer feedback collected during

Correctness Analysis

- Functionality Verification: Unit testing was utilized to verify the functionality of individual components and modules within the application, ensuring that all components function correctly and handle edge cases effectively.
- Integration Testing: It focused on evaluating interactions between different components to ensure smooth data flow and proper response handling between frontend and backend components.
- User Acceptance Testing(UAT): Real user testing validated the application against user requirements, ensuring correctness in terms of meeting user expectations and addressing any encountered issues.
- Performance Testing: It ensured correctness in terms of system response time, scalability , and resource usage, validating that the system performs as expected under various scenarios.
- Feedback Incorporation:Customer feedback collected during

Correctness Analysis

- Functionality Verification: Unit testing was utilized to verify the functionality of individual components and modules within the application, ensuring that all components function correctly and handle edge cases effectively.
- Integration Testing: It focused on evaluating interactions between different components to ensure smooth data flow and proper response handling between frontend and backend components.
- User Acceptance Testing(UAT): Real user testing validated the application against user requirements, ensuring correctness in terms of meeting user expectations and addressing any encountered issues.
- Performance Testing: It ensured correctness in terms of system response time, scalability , and resource usage, validating that the system performs as expected under various scenarios.
- Feedback Incorporation:Customer feedback collected during

Correctness Analysis

- Functionality Verification: Unit testing was utilized to verify the functionality of individual components and modules within the application, ensuring that all components function correctly and handle edge cases effectively.
- Integration Testing: It focused on evaluating interactions between different components to ensure smooth data flow and proper response handling between frontend and backend components.
- User Acceptance Testing(UAT): Real user testing validated the application against user requirements, ensuring correctness in terms of meeting user expectations and addressing any encountered issues.
- Performance Testing: It ensured correctness in terms of system response time, scalability , and resource usage, validating that the system performs as expected under various scenarios.
- Feedback Incorporation:Customer feedback collected during

Correctness Analysis

- Functionality Verification: Unit testing was utilized to verify the functionality of individual components and modules within the application, ensuring that all components function correctly and handle edge cases effectively.
- Integration Testing: It focused on evaluating interactions between different components to ensure smooth data flow and proper response handling between frontend and backend components.
- User Acceptance Testing(UAT): Real user testing validated the application against user requirements, ensuring correctness in terms of meeting user expectations and addressing any encountered issues.
- Performance Testing: It ensured correctness in terms of system response time, scalability , and resource usage, validating that the system performs as expected under various scenarios.
- Feedback Incorporation:Customer feedback collected during

Customer Feedback

- User Experience: Positive feedback on the user friendly interface, intuitive navigation and streamline workflows, indicating satisfaction with the overall user experience.
- Performance: Feedback on system responsiveness and performance under different scenarios, guiding optimizations efforts to enhance user satisfaction.
- Security: Feedback on the authentication system's robustness and data protection measures, ensuring user confidence in the security of their credentials and sensitive informations.

Customer Feedback

- User Experience: Positive feedback on the user friendly interface, intuitive navigation and streamline workflows, indicating satisfaction with the overall user experience.
- Performance: Feedback on system responsiveness and performance under different scenarios, guiding optimizations efforts to enhance user satisfaction.
- Security: Feedback on the authentication system's robustness and data protection measures, ensuring user confidence in the security of their credentials and sensitive informations.

Customer Feedback

- User Experience: Positive feedback on the user friendly interface, intuitive navigation and streamline workflows, indicating satisfaction with the overall user experience.
- Performance: Feedback on system responsiveness and performance under different scenarios, guiding optimizations efforts to enhance user satisfaction.
- Security: Feedback on the authentication system's robustness and data protection measures, ensuring user confidence in the security of their credentials and sensitive informations.

Innovations Made

- Comprehensive Testing Methodologies: Employing a range of comprehensive testing methodologies including Unit Testing, Integration Testing, User Acceptance Testing and Performance Testing to ensure the quality, reliability, and correctness of the application.
- UI/UX Focus: Prioritising user experience through intuitive navigation, streamlined workflows, enhancing overall user satisfaction and productivity.

Innovations Made

- Comprehensive Testing Methodologies: Employing a range of comprehensive testing methodologies including Unit Testing, Integration Testing, User Acceptance Testing and Performance Testing to ensure the quality, reliability, and correctness of the application.
- UI/UX Focus: Prioritising user experience through intuitive navigation, streamlined workflows, enhancing overall user satisfaction and productivity.

Publication Details

- This application will be available for android devices.It is planning to be released on google play store.

1. Problem Description
2. Project Objectives
3. Solution Strategy
4. Design Components
5. Implementation
6. Result Analysis
7. Future Plan

Reasons for continuing this project

- Expansion Potential: Extend project to other colleges within the university for broader impact.
- Scale and Reach: Increase reach by implementing the app across multiple campuses.
- Unified System: Create a standardized system for facility management across the university.

Reasons for continuing this project

- Expansion Potential: Extend project to other colleges within the university for broader impact.
- Scale and Reach: Increase reach by implementing the app across multiple campuses.
- Unified System: Create a standardized system for facility management across the university.

Reasons for continuing this project

- Expansion Potential: Extend project to other colleges within the university for broader impact.
- Scale and Reach: Increase reach by implementing the app across multiple campuses.
- Unified System: Create a standardized system for facility management across the university.

Reasons for switching to a different project

- Exploration and Growth: Seeking new challenges to expand our skill set and knowledge base.
- Fresh Perspectives: Embracing the opportunity to approach a different problem from diverse angles.
- Innovation Drive: Motivated to explore novel solutions and contribute to addressing varied issues.

Reasons for switching to a different project

- Exploration and Growth: Seeking new challenges to expand our skill set and knowledge base.
- Fresh Perspectives: Embracing the opportunity to approach a different problem from diverse angles.
- Innovation Drive: Motivated to explore novel solutions and contribute to addressing varied issues.

Reasons for switching to a different project

- Exploration and Growth: Seeking new challenges to expand our skill set and knowledge base.
- Fresh Perspectives: Embracing the opportunity to approach a different problem from diverse angles.
- Innovation Drive: Motivated to explore novel solutions and contribute to addressing varied issues.

Possible Extension

- Chat Functionality: Implement a chat feature allowing HODs to communicate directly with assigned workers regarding the current condition of maintenance tasks.
- Web Interface Development: Develop a web interface alongside the app for enhanced accessibility and usability.
- University-wide Implementation: Extend the app's functionality to cater to all colleges within the university, facilitating streamlined maintenance management across the entire institution.

Possible Extension

- Chat Functionality: Implement a chat feature allowing HODs to communicate directly with assigned workers regarding the current condition of maintenance tasks.
- Web Interface Development: Develop a web interface alongside the app for enhanced accessibility and usability.
- University-wide Implementation: Extend the app's functionality to cater to all colleges within the university, facilitating streamlined maintenance management across the entire institution.

Possible Extension

- Chat Functionality: Implement a chat feature allowing HODs to communicate directly with assigned workers regarding the current condition of maintenance tasks.
- Web Interface Development: Develop a web interface alongside the app for enhanced accessibility and usability.
- University-wide Implementation: Extend the app's functionality to cater to all colleges within the university, facilitating streamlined maintenance management across the entire institution.

Alternative domains that can absorb the same solution

- Property Management: Extend the solution to property management companies or real estate firms for efficient handling of maintenance requests and facility management.
- Hospitality Industry: Implement the solution in hotels or resorts to manage maintenance tasks for various facilities, rooms, and amenities.
- Corporate Offices: Adapt the solution for corporate office buildings to streamline maintenance operations and improve facility management efficiency.

Alternative domains that can absorb the same solution

- Property Management: Extend the solution to property management companies or real estate firms for efficient handling of maintenance requests and facility management.
- Hospitality Industry: Implement the solution in hotels or resorts to manage maintenance tasks for various facilities, rooms, and amenities.
- Corporate Offices: Adapt the solution for corporate office buildings to streamline maintenance operations and improve facility management efficiency.

Alternative domains that can absorb the same solution

- Property Management: Extend the solution to property management companies or real estate firms for efficient handling of maintenance requests and facility management.
- Hospitality Industry: Implement the solution in hotels or resorts to manage maintenance tasks for various facilities, rooms, and amenities.
- Corporate Offices: Adapt the solution for corporate office buildings to streamline maintenance operations and improve facility management efficiency.

Possible performance optimizations

- Caching Mechanisms: Implement caching strategies to store frequently accessed data, reducing the need for repeated database queries and improving response times.
- Load Balancing: Distribute incoming traffic across multiple servers to prevent overloading and ensure optimal performance during peak usage periods.
- Code Optimization: Review and optimize codebase for efficiency, including minimizing database queries, reducing redundant calculations, and optimizing algorithms.
- Database Indexing: Utilize database indexing to improve query performance, especially for frequently accessed fields or columns.
- Asynchronous Processing: Implement asynchronous processing for non-blocking operations, such as sending notifications or processing background tasks, to enhance system responsiveness.

Possible performance optimizations

- Caching Mechanisms: Implement caching strategies to store frequently accessed data, reducing the need for repeated database queries and improving response times.
- Load Balancing: Distribute incoming traffic across multiple servers to prevent overloading and ensure optimal performance during peak usage periods.
- Code Optimization: Review and optimize codebase for efficiency, including minimizing database queries, reducing redundant calculations, and optimizing algorithms.
- Database Indexing: Utilize database indexing to improve query performance, especially for frequently accessed fields or columns.
- Asynchronous Processing: Implement asynchronous processing for non-blocking operations, such as sending notifications or processing background tasks, to enhance system responsiveness.

Possible performance optimizations

- Caching Mechanisms: Implement caching strategies to store frequently accessed data, reducing the need for repeated database queries and improving response times.
- Load Balancing: Distribute incoming traffic across multiple servers to prevent overloading and ensure optimal performance during peak usage periods.
- Code Optimization: Review and optimize codebase for efficiency, including minimizing database queries, reducing redundant calculations, and optimizing algorithms.
- Database Indexing: Utilize database indexing to improve query performance, especially for frequently accessed fields or columns.
- Asynchronous Processing: Implement asynchronous processing for non-blocking operations, such as sending notifications or processing background tasks, to enhance system responsiveness.

Possible performance optimizations

- Caching Mechanisms: Implement caching strategies to store frequently accessed data, reducing the need for repeated database queries and improving response times.
- Load Balancing: Distribute incoming traffic across multiple servers to prevent overloading and ensure optimal performance during peak usage periods.
- Code Optimization: Review and optimize codebase for efficiency, including minimizing database queries, reducing redundant calculations, and optimizing algorithms.
- Database Indexing: Utilize database indexing to improve query performance, especially for frequently accessed fields or columns.
- Asynchronous Processing: Implement asynchronous processing for non-blocking operations, such as sending notifications or processing background tasks, to enhance system responsiveness.

Possible performance optimizations

- Caching Mechanisms: Implement caching strategies to store frequently accessed data, reducing the need for repeated database queries and improving response times.
- Load Balancing: Distribute incoming traffic across multiple servers to prevent overloading and ensure optimal performance during peak usage periods.
- Code Optimization: Review and optimize codebase for efficiency, including minimizing database queries, reducing redundant calculations, and optimizing algorithms.
- Database Indexing: Utilize database indexing to improve query performance, especially for frequently accessed fields or columns.
- Asynchronous Processing: Implement asynchronous processing for non-blocking operations, such as sending notifications or processing background tasks, to enhance system responsiveness.

Thank you