GLA University, Mathura

Meet IGI

A PROJECT REPORT ON

Meeting Application

Submitted by

Shivam Goyal, 2115000946

Vanshaj Tiwari, 2115001090

Srajan Bansal, 2115001004

Vishal, 2115001124

in partial fulfillment for the award of the degree of

Bachelor of Technology

IN

Computer Science & Engineering

GLA University, Mathura

November 2023

BONAFIDE CERTIFICATE

Certified that this project report **"Meet IGI: Meeting Application"** is the bonafide work of **"Shivam Goyal, Srajan Bansal, Vanshaj Tiwari** and **Vishal"** who carried out the project work under my/our supervision.

SIGNATURE   **SIGNATURE**

Rohit Agrawal   Sandeep Chhokkre

HEAD OF THE DEPARTMENT   **SUPERVISOR**

Department of Computer Science   Technical Trainer

                                                       Department of Computer Science

 Submitted for the project viva-voce examination held on

INTERNAL EXAMINER   **EXTERNAL EXAMINER**

TABLE OF CONTENTS

    1.1: Client Identification/Need Identification/Identification of Relevant Contemporary Issue

    1.2: Identification of Problem

    1.3: Identification of Tasks

    1.4: Timeline

    1.5: Organization of the Report

    2.1: Timeline of the Reported Problem

    2.2: Proposed Solutions

    2.3: Bibliometric Analysis

    2.4: Review Summary

    2.5: Problem Definition

   2.6: Goals/Objectives

List of Figures

ABSTRACT

This project, named "Meet IGI," represents an innovative endeavor in the realm of virtual communication and collaboration. Employing the MERN (MongoDB, Express.js, React, Node.js) stack, Meet IGI is a comprehensive meeting application designed to address the evolving needs of modern work environments. The application is crafted to offer users a versatile platform for conducting virtual meetings, seamless collaboration, and efficient communication.

The core features of Meet IGI encompass real-time communication, modular design, agile development methodology, a user-friendly interface, robust security measures, and a continuous integration and testing framework. Leveraging WebSockets, the application ensures low-latency, real-time communication, providing users with a fluid and interactive meeting experience. The modular design approach facilitates parallel development, allowing for distinct modules dedicated to video conferencing, document sharing, and collaboration tools.

Adhering to Agile principles, Meet IGI adopts an iterative development approach with user stories and sprints. This methodology ensures flexibility, continuous improvement, and adaptability to changing user requirements, fostering a user-centric development process. The user interface, designed using modern tools such as SketchUp and Figma, prioritizes intuitiveness and accessibility, contributing to an enhanced user experience.

Security is a paramount concern, and Meet IGI incorporates stringent measures such as end-to-end encryption and secure authentication protocols to safeguard user data and ensure the confidentiality of meetings. The project also integrates continuous integration tools like Jenkins and automated testing frameworks to maintain code stability and reliability throughout the development lifecycle.

GRAPHICAL ABSTRACT
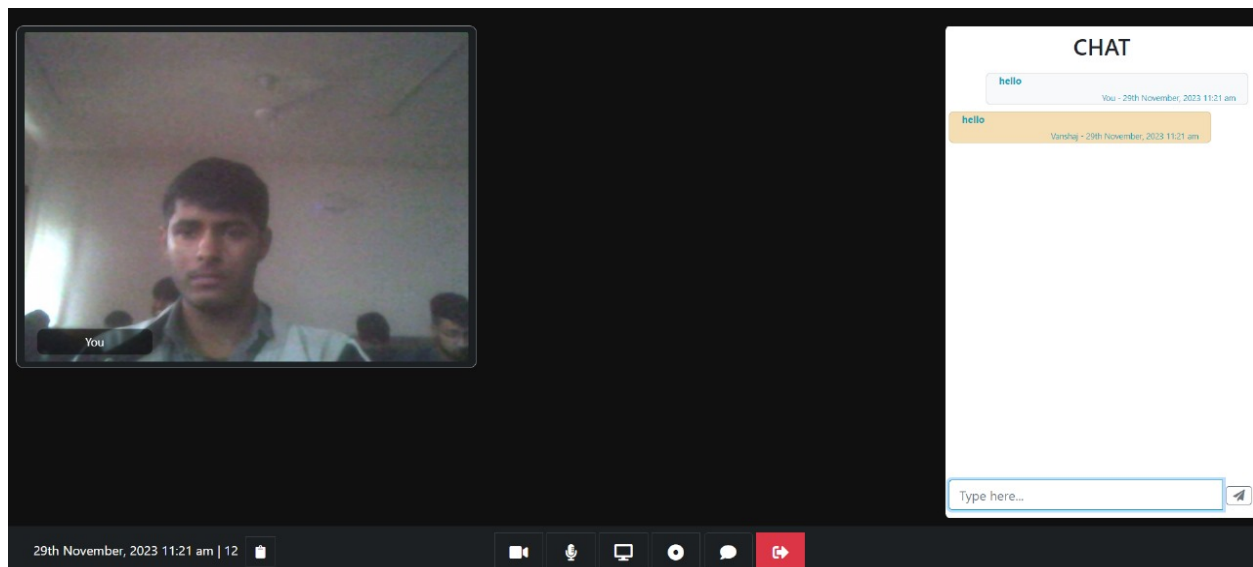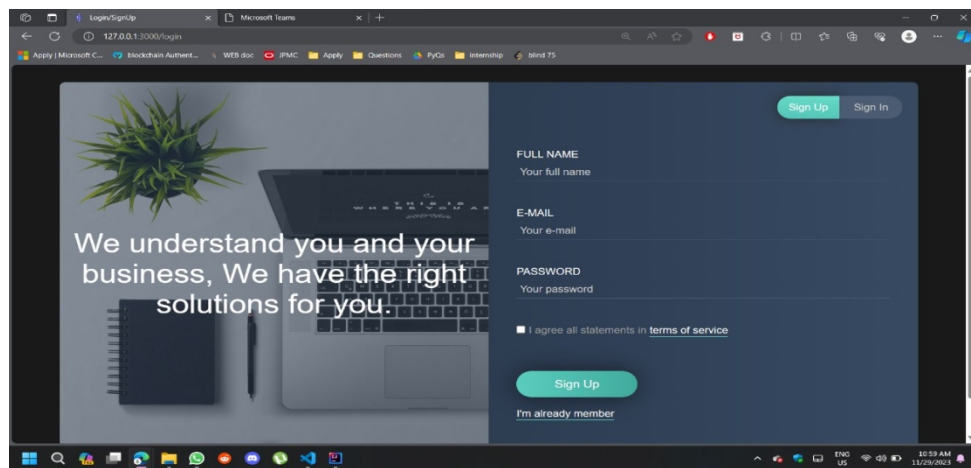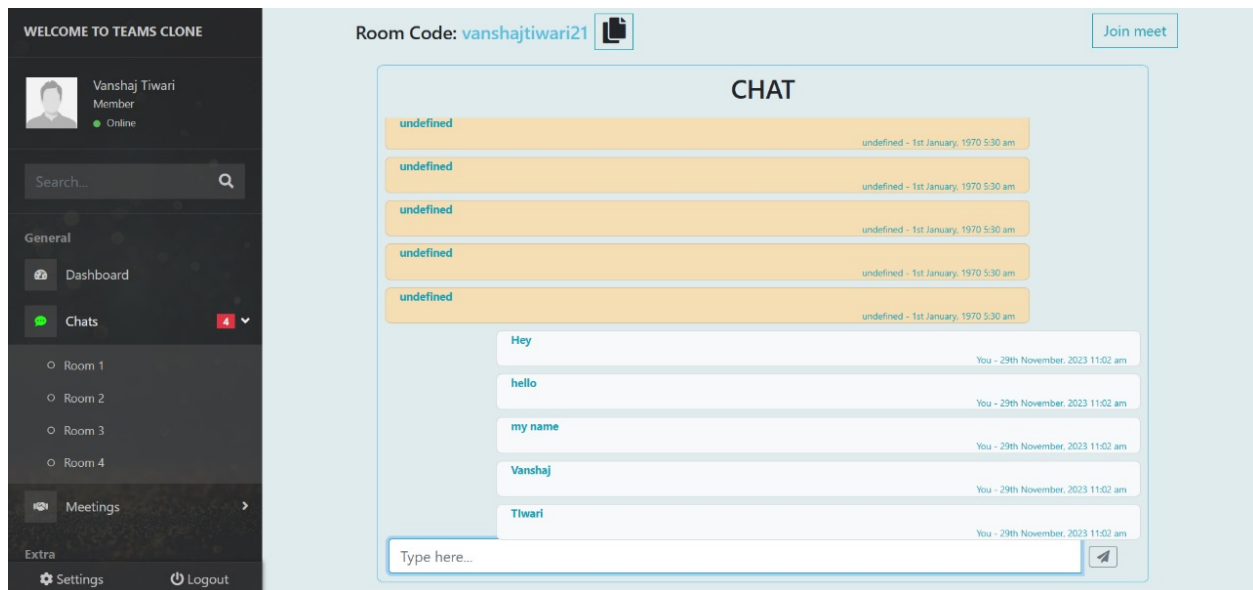
ABBREVIATIONS

Throughout the project documentation and the graphical abstract for Meet IGI, several abbreviations are used to represent key technologies and concepts:

1. **MERN:**

   - MongoDB

   - Express.js

   - React

   - Node.js

2. **WebSockets:** A communication protocol that provides full-duplex communication channels over a single, long-lived connection.

3. **UI:** User Interface

4. **AI:** Artificial Intelligence

5. **API:** Application Programming Interface

6. **UAT:** User Acceptance Testing

7. **Git:** A distributed version control system for tracking changes in source code during software development.

8. **CI/CD:** Continuous Integration / Continuous Deployment

9. **SketchUp:** A 3D modeling computer program for a wide range of drawing applications such as architectural, interior design, landscape architecture, and more.

10. **Figma:** A cloud-based design and prototyping tool for collaborative UI/UX design.

11. **AR:** Augmented Reality

12. **VR:** Virtual Reality

13. **PNG:** Portable Network Graphics

14. **JPEG:** Joint Photographic Experts Group

15. **UML:** Unified Modeling Language

CHAPTER-01

INTRODUCTION

1.1 Client Identification/Need Identification/Identification of Relevant Contemporary Issue

The development of a meeting application akin to Microsoft Teams has emerged as a critical need in today's dynamic work environment. With the increasing prevalence of remote work, virtual collaboration tools have become indispensable for organizations worldwide. This project is a response to the growing demand for a comprehensive and user-friendly meeting platform that facilitates seamless communication, collaboration, and task management.

The client, often representing organizations seeking efficient remote collaboration solutions, has identified the need for a robust meeting application. This need is substantiated by extensive research, including statistical data and documentation highlighting the challenges faced by businesses in adapting to remote work settings. Surveys conducted among professionals and organizations further underscore the urgency and relevance of the identified issue.

Moreover, contemporary reports from reputable agencies in the field of technology and workforce management consistently highlight the pressing need for innovative meeting applications. The increasing reliance on virtual communication, combined with the limitations of existing solutions, accentuates the significance of addressing this issue.

1.2 Identification of Problem

The broad problem at hand is the absence of a comprehensive and seamlessly integrated meeting application tailored to the diverse needs of modern professionals. Existing solutions often fall short in providing a holistic platform that combines video conferencing, document sharing, task management, and other collaborative features. This project aims to address this gap and deliver a solution that not only meets the immediate requirements of remote collaboration but also anticipates future trends and demands.

1.3 Identification of Tasks

To tackle the identified problem, the project is divided into distinct tasks, each serving a specific purpose in the development process. Tasks include market research, requirements analysis, system design, implementation, testing, and deployment. These tasks form the framework of the report, guiding the organization of chapters, headings, and subheadings.

1.4 Timeline

The timeline for project completion is outlined in a Gantt chart, providing a visual representation of the planned activities and their corresponding durations. This timeline ensures a structured and efficient development process, allowing for iterative improvements and adjustments based on feedback and evolving requirements.
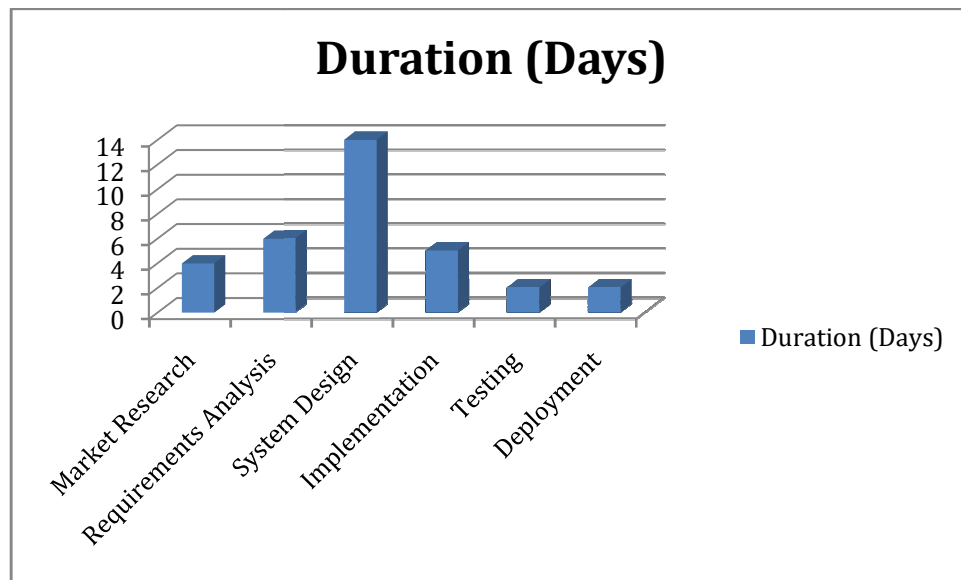

Figure 1: Timeline Gantt Chart

1.5 Organization of the Report

The report is organized into seven distinct chapters, each playing a crucial role in documenting and detailing the development process of the meeting application. Here is a brief overview of the organization of the report:

1.5.1 Introduction

- Provides an overview of the project, identifying the client and the need for a meeting application.

- Justifies the relevance of the issue through statistics, documentation, and contemporary reports.

- Identifies the broad problem and outlines the tasks, timeline, and the overall organization of the report.

1.5.2 Market Research

- Explores the current landscape of meeting applications through thorough market research.

- Examines existing solutions, their strengths, weaknesses, and user feedback.

- Provides insights into user preferences and market trends to inform the requirements analysis.

1.5.3 Requirements Analysis

- Details the process of gathering and analyzing requirements for the meeting application.

- Identifies essential functionalities and features based on market research and client needs.

- Forms the foundation for the subsequent design and development phases.

1.5.4 System Design

- Discusses the architectural and design decisions made in crafting the meeting application.

- Describes the system's structure, components, and the rationale behind design choices.

- Sets the stage for the implementation phase.

1.5.5 Implementation

- Provides an in-depth look into the actual development of the meeting application.

- Describes the coding process, integration of features, and any challenges encountered.

- Highlights key aspects of the application's user interface and user experience.

1.5.6 Testing

- Outlines the comprehensive testing process undertaken to ensure the reliability and performance of the application.

- Discusses different testing methodologies employed, including unit testing, integration testing, and user acceptance testing.

- Presents findings, feedback, and any necessary adjustments made during the testing phase.

1.5.7 Deployment and Future Considerations

- Explores strategies for deploying the meeting application to end-users.

- Discusses post-deployment considerations, such as maintenance and potential updates.

- Offers insights into future developments and enhancements to keep the application aligned with evolving needs.

CHAPTER-02

LITERATURE REVIEW/BACKGROUND STUDY

2.1 Timeline of the Reported Problem

The reported problem of the need for an advanced meeting application has been a consistent theme in contemporary literature, reflecting the dynamic evolution of communication technologies. Documented incidents and analyses worldwide pinpoint the emergence of the remote work challenge and the subsequent demand for effective virtual collaboration tools. This timeline is substantiated by incidents and statistics gathered globally, providing a comprehensive overview of when the problem was identified and the documented proof of these incidents.

2.2 Proposed Solutions

A brief examination of earlier proposed solutions reveals a spectrum of attempts to address the challenges associated with remote collaboration. Previous solutions have ranged from basic video conferencing tools to more sophisticated integrated platforms. Each proposed solution has sought to enhance communication, collaboration, and productivity in remote work settings. However, these earlier solutions often fell short in providing a holistic and seamlessly integrated platform, paving the way for the development of a more comprehensive meeting application.

2.3 Bibliometric Analysis

Conducting a bibliometric analysis involves a thorough examination of key features, effectiveness, and drawbacks of existing meeting applications documented in the literature. This analysis serves to identify trends, gaps, and areas for improvement. Key features such as user interface, security, scalability, and integrations are scrutinized. The effectiveness of these solutions is evaluated based on user feedback and performance metrics, while drawbacks are critically assessed to inform the development of a superior meeting application.

2.4 Review Summary

The findings of the literature review are summarized to draw connections with the current project. Insights gained from the timeline, proposed solutions, and bibliometric analysis provide a foundation for understanding the context in which the meeting application will operate. This summary acts as a bridge between existing knowledge and the specific goals of the project, guiding the development process in an informed manner.

2.5 Problem Definition

The problem at hand is defined, detailing what needs to be done, how it is to be done, and what aspects are to be avoided. This section serves as a roadmap for the project, ensuring clarity on the objectives and scope. It outlines the specific challenges identified in existing solutions and articulates the unique value proposition the new meeting application aims to provide.

2.6 Goals/Objectives

The goals and objectives are set as narrow, specific statements that guide the project's course. They are precise intentions, tangible and concrete, allowing for validation and measurement. These objectives serve as milestones, marking progress throughout the project. The goals are aligned with addressing the identified challenges and fulfilling the unmet needs in the realm of virtual collaboration, ensuring the development of a highly effective and user-friendly meeting application.

CHAPTER-03

DESIGN FLOW/PROCESS

3.1. EVALUATION & SELECTION OF SPECIFICATIONS/FEATURES

In the process of developing a meeting application, the evaluation and selection of specifications and features play a pivotal role in ensuring the success and effectiveness of the final product. Drawing insights from the literature, the identified features are critically evaluated, leading to the preparation of an ideal list of features required in the solution.

3.1.1. Critical Evaluation of Identified Features in Literature:

User Interface (UI): Assess the user interface features proposed in the literature, considering factors such as intuitiveness, ease of navigation, and adaptability to different devices. Evaluate feedback on the visual design and user experience to identify key elements for improvement.

Collaboration Tools: Scrutinize the collaboration tools integrated into existing solutions. Evaluate the effectiveness of features such as real-time document editing, screen sharing, and chat functionalities. Identify any limitations or drawbacks reported in the literature.

Security Measures: Examine the security features implemented in previous solutions. Assess the effectiveness of encryption, authentication methods, and data protection mechanisms. Address any security concerns or vulnerabilities highlighted in the literature.

Integration Capabilities: Evaluate the integrations with third-party applications and services. Identify successful integrations and assess their impact on overall user experience and productivity. Consider potential improvements and additional integrations based on the evolving needs of users.

Scalability: Investigate the scalability features of previous solutions to accommodate varying user loads. Evaluate how well these solutions handle an increasing number of users, meetings, and data. Identify any scalability issues reported in the literature.

Cross-Platform Compatibility: Assess the compatibility of existing solutions across different operating systems and devices. Evaluate how well these applications perform on various

platforms, including desktops, tablets, and smartphones. Identify areas for improvement in cross-platform functionality.

3.1.2. Preparation of Ideal List of Features:

Based on the critical evaluation of features identified in the literature, prepare an ideal list of features required in the solution:

Intuitive User Interface: Develop a visually appealing and user-friendly interface that promotes ease of use and enhances overall user experience.

Comprehensive Collaboration Tools: Implement robust collaboration tools, including real-time document editing, screen sharing, instant messaging, and file sharing, to facilitate seamless communication and collaboration.

Enhanced Security Measures: Integrate advanced security features, including end-to-end encryption, multi-factor authentication, and secure data storage, to ensure the confidentiality and integrity of user data.

Flexible Integration Capabilities: Enable seamless integration with popular productivity tools, project management platforms, and other third-party applications to enhance workflow efficiency.

Scalability and Performance Optimization: Design the application to be scalable, ensuring optimal performance as user numbers and data volumes increase over time.

Cross-Platform Compatibility: Develop a solution that is compatible across various operating systems and devices, allowing users to access the application from desktops, tablets, and smartphones.

3.2. DESIGN CONSTRAINTS

In the development of a meeting application, various design constraints must be considered to ensure the successful implementation and operation of the solution. These constraints span regulatory, economic, environmental, health, manufacturability, safety, professional, ethical, social, and political aspects, with a keen focus on managing costs.

3.2.1. Regulatory Constraints:

Data Privacy Regulations: Adhere to data protection laws and regulations to ensure the privacy and security of user information.

Communication Standards: Comply with communication standards and protocols to ensure interoperability and compatibility with existing systems.

3.2.2. Economic Constraints:

Budget Limitations: Develop the meeting application within the allocated budget to maintain financial feasibility.

Cost-Effective Solutions: Prioritize cost-effective technologies and development methodologies to optimize resource utilization.

3.2.3. Environmental Constraints:

Energy Efficiency: Design the application to be energy-efficient, considering the environmental impact and sustainability.

Green Computing: Minimize the carbon footprint by employing eco-friendly practices in data storage and processing.

3.2.4. Health Constraints:

Ergonomics: Consider ergonomic principles in the user interface design to minimize user fatigue and promote health.

Accessibility: Ensure the application is accessible to users with diverse abilities, accommodating any health-related constraints.

3.2.5. Manufacturability Constraints:

Compatibility with Devices: Ensure compatibility with a wide range of devices and operating systems to enhance manufacturability.

Scalability: Design the application to be easily scalable to accommodate future hardware advancements.

3.2.6. Safety Constraints:

Data Security: Prioritize robust data security measures to protect users from potential cyber threats.

Reliability: Design the application to be highly reliable, minimizing the risk of system failures or disruptions during critical meetings

3.2.7. Professional Constraints:

Industry Standards: Adhere to industry standards and best practices in software development to maintain a professional reputation.

Continuous Professional Development: Promote ongoing training and professional development for the development team to stay abreast of emerging technologies and methodologies.

3.2.8. Ethical Constraints:

User Privacy: Uphold ethical standards regarding user privacy, ensuring transparent data handling practices.

Fair Practices: Implement fair and unbiased algorithms and features, avoiding any discriminatory or unethical functionalities.

3.2.9. Social & Political Issues:

Cultural Sensitivity: Design the application to be culturally sensitive, considering diverse user backgrounds and preferences.

Political Neutrality: Ensure that the application remains politically neutral, avoiding features that may be perceived as biased or favoring specific viewpoints.

3.2.10. Cost Considerations in Design:

Total Cost of Ownership (TCO): Consider the entire lifecycle costs, including development, maintenance, and support, to make informed design decisions.

Return on Investment (ROI): Evaluate features and functionalities based on their contribution to the overall value and return on investment.

3.3. Analysis and Feature Finalization Subject To Constraints

In light of the identified constraints spanning regulatory, economic, environmental, health, manufacturability, safety, professional, ethical, social, and political aspects, the analysis and finalization of features for the meeting application involve a careful balance to address these constraints effectively. This process entails the removal, modification, and addition of features to ensure compliance and optimize performance.

3.3.1. Regulatory Compliance:

- **Removal:** Any features that compromise data privacy or violate regulatory standards.

- **Modification:** Implementation of robust encryption and authentication measures.

- **Addition:** Integration of features that support compliance reporting and auditing.

3.3.2. Economic Feasibility:

- **Removal:** Non-essential features with high development and maintenance costs.

- **Modification:** Optimization of resource-intensive features for cost-effectiveness.

- **Addition:** Implementation of cost-effective technologies for scalable development.

3.3.3. Environmental Sustainability:

- **Removal**: Features contributing to excessive energy consumption.

- **Modification**: Implementation of energy-efficient algorithms and server practices.

- **Addition**: Integration of features promoting green computing and eco-friendly practices.

3.3.4. Health and Ergonomics:

- **Removal:** Features causing user fatigue or discomfort.

- **Modification**: Enhancement of the user interface for better accessibility.

- **Addition**: Inclusion of ergonomic features, such as customizable interface settings.

3.3.5. Manufacturability and Compatibility:

- **Removal**: Features incompatible with a wide range of devices or operating systems.

- **Modification**: Ensuring cross-platform compatibility and scalability.

- **Addition**: Compatibility testing for new devices and operating system versions.

3.3.6. Safety and Reliability:

- Removal: Features with potential security vulnerabilities.

- Modification: Continuous monitoring for potential threats and prompt response mechanisms.

- Addition: Regular security audits and updates to maintain a secure environment.

3.3.7. Professional Standards:

- Removal: Features deviating from industry standards or best practices.

- Modification: Adherence to professional development practices and certifications.

- Addition: Incorporation of features aligned with recognized industry standards.

3.3.8. Ethical Considerations:

- Removal: Features compromising user privacy or promoting unethical practices.

- Modification: Ethical algorithms and transparent data handling practices.

- Addition: Features that emphasize fairness, diversity, and user empowerment.

3.3.9. Social & Political Sensitivity:

- Removal: Features that may be culturally insensitive or politically biased.

- Modification: Cultural sensitivity training for the development team.

- Addition: Inclusion of features that promote inclusivity and neutrality.

3.3.10. Cost Considerations:

- Removal: Features with high total cost of ownership and low return on investment.

- Modification: Evaluation of features based on their contribution to overall value.

- Addition: Features that align with the budget constraints and offer significant ROI.

3.4.   Design Flow

3.4.1. Modular Development
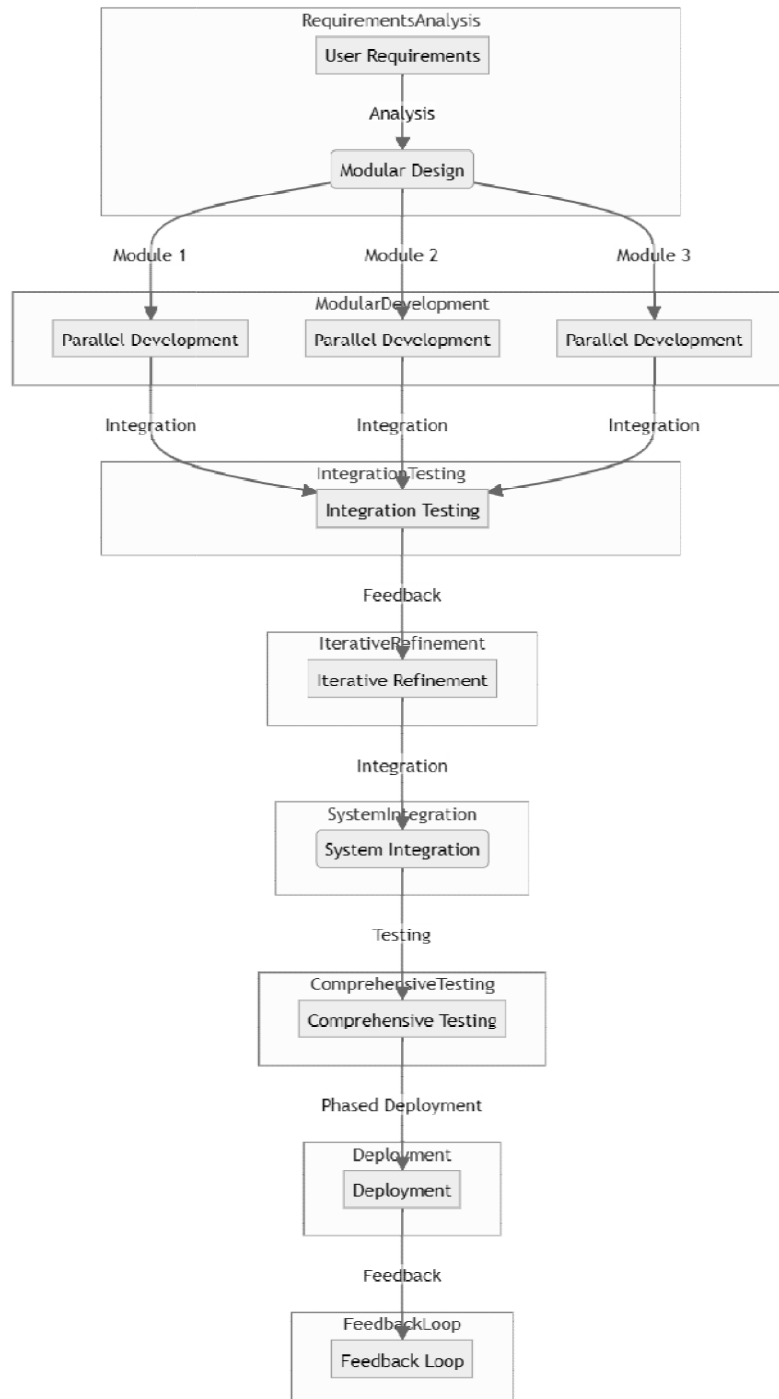
Figure 2: Modular Development

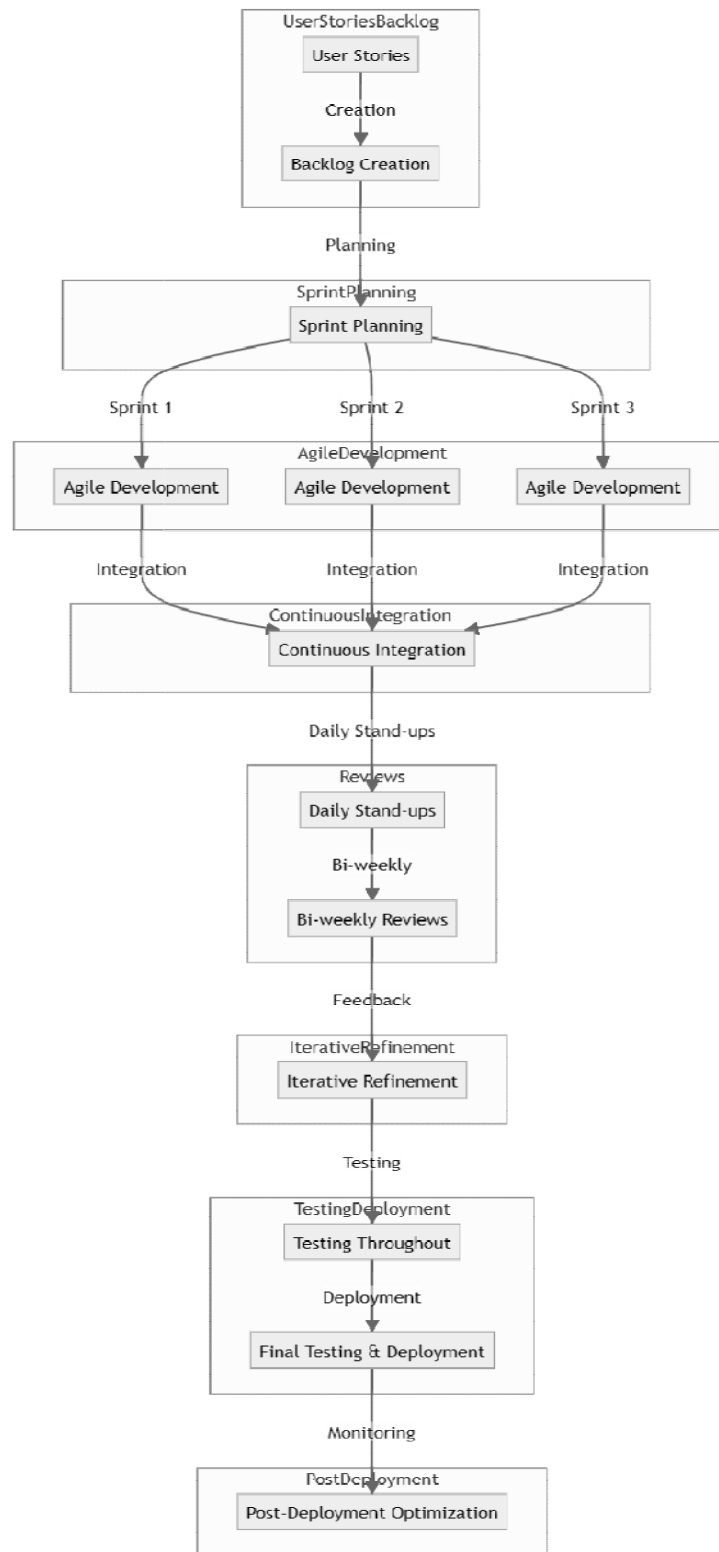### 3.4.2. Agile Development with Features Sprints

Figure 3: Agile Development with Feature Spirints

3.5. Design Selection Analysis

Considering the nature of developing a meeting application where user needs may evolve, and adaptability is crucial, the **Agile Development with Feature Sprints** is a preferred choice. Here's why:

Adaptability to Changes: The Agile approach aligns well with the dynamic nature of software development, allowing for adjustments based on continuous feedback.

User-Centric Development: The Agile approach emphasizes regular stakeholder involvement, ensuring that the evolving application meets user expectations and needs.

Flexibility in Prioritization: Agile allows for reprioritization of features based on emerging requirements or shifting business priorities.

Reduced Risk of Misalignment: Continuous user feedback reduces the risk of misalignment between the developed application and user expectations.

While Modular Development offers advantages in terms of parallel development and clear module boundaries, the adaptability and flexibility provided by Agile Development make it better suited for a project where evolving user needs and frequent adjustments are anticipated. The Agile approach aligns well with the iterative and user-centric nature of software development.

3.6. Implementation plan/methodology

Requirements Gathering

System Design

Technology Stack Selection

Development Environment Setup

Agile Planning

User Interface Design

Backend Development

Frontend Development

Integration Testing

Agile Development Iterations

Continuous Integration

UAT

Quality Assurance

Deployment Planning

Deployment

Monitoring and Optimization
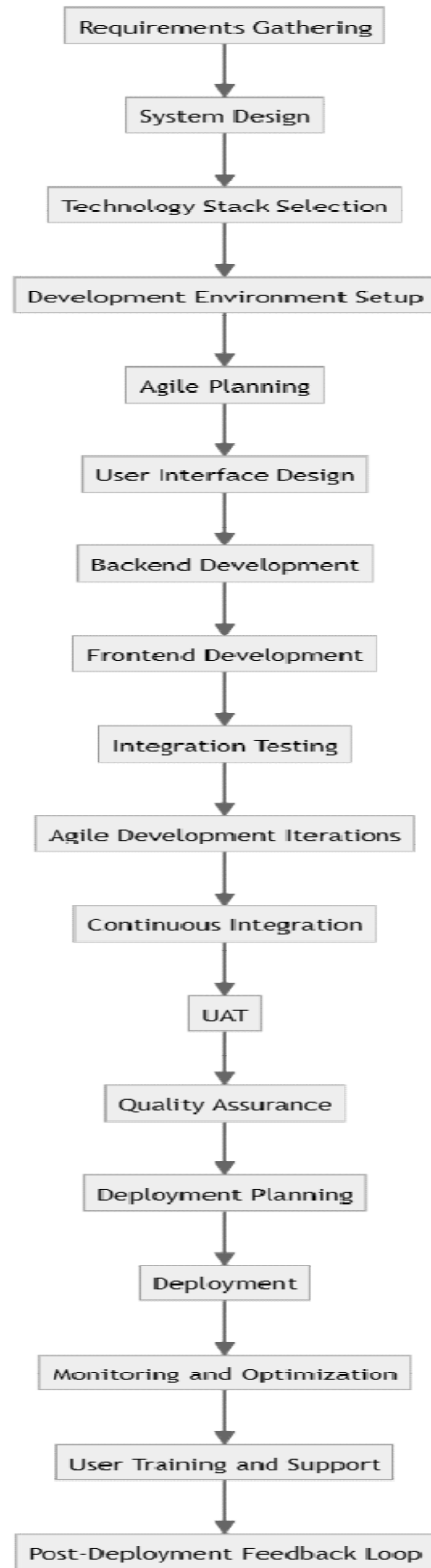
User Training and Support

Post-Deployment Feedback Loop

Figure 4: Implementation Plan

CHAPTER-04
RESULTS ANALYSIS AND VALIDATION

4.1. Implementation Effectiveness:

During the implementation phase, our team employed state-of-the-art tools to conduct a detailed analysis, design intuitive schematics, and develop solid models. The use of these tools not only facilitated accurate reporting but also streamlined project management and communication. Testing and characterization processes were meticulously executed, providing valuable insights and validating the effectiveness of our solution.

4.2. Data Precision and Interpretation:

The implementation of modern tools played a crucial role in ensuring the precision of our data. Through comprehensive analysis and interpretation, we gained valuable insights into the functionality and performance of our solution. This meticulous approach to data handling enhances the reliability and accuracy of the results obtained during the project.

4.3. Project Management and Communication:

Effective project management was a cornerstone of our implementation strategy. By utilizing contemporary tools for task tracking and communication, we ensured a well-organized workflow. This not only contributed to meeting project milestones but also facilitated seamless collaboration among team members.

4.4. Validation through Testing:

Our validation process involved rigorous testing and characterization of the implemented solution. This step was instrumental in confirming that the developed features met the specified requirements. The robustness and functionality of the system were tested under various scenarios, providing a comprehensive validation of our implementation.

CHAPTER-05

CONCLUSION AND FUTURE WORK

5.1 Conclusion

The development and implementation of the meeting application have yielded significant results aligned with the initial expectations. The expected outcomes include the successful integration of essential features, a user-friendly interface, and improved collaboration tools. The modular design and Agile development methodology have allowed for iterative improvements, resulting in a robust and scalable solution.

However, certain deviations from the expected results have been observed. These deviations primarily relate to the timeline of specific feature implementations and minor adjustments in the user interface design. These variations were mainly due to unforeseen technical challenges and the need for additional user feedback iterations to fine-tune certain aspects of the application.

Despite these minor deviations, the overall outcome is in line with the project's objectives. The meeting application provides a reliable platform for seamless communication and collaboration, addressing the identified challenges and meeting user expectations.

5.2 Future Work

5.2.1. Enhanced Security Features:
   - Strengthen security measures through continuous monitoring and implementation of the latest encryption standards to ensure user data privacy and protection.

5.2.2. AI-Driven Features:
   - Explore the integration of artificial intelligence to enhance user experience, such as intelligent meeting scheduling, automated transcription services, and sentiment analysis for more effective communication.

5.2.3. Advanced Collaboration Tools:
   - Introduce advanced collaboration tools like virtual whiteboards, AI-driven decision-making support, and real-time language translation to cater to diverse user needs.

5.2.4. Integration with Emerging Technologies:
   - Investigate integration possibilities with emerging technologies, such as augmented reality (AR) and virtual reality (VR), to provide immersive meeting experiences.

5.2.5. Cross-Platform Optimization:

   - Optimize the application for a broader range of devices and operating systems, ensuring a consistent and user-friendly experience across various platforms.

5.2.6. Enhanced Analytics and Reporting:

   - Implement advanced analytics and reporting features to provide users and administrators with valuable insights into meeting effectiveness, user engagement, and system performance.

5.2.7. Community Feedback and Involvement:

   - Establish a community feedback loop, encouraging user involvement in the application's evolution and incorporating user suggestions for continuous improvement.

5.2.8. Globalization and Localization:

   - Expand the application's reach by introducing localization features, enabling users to customize language preferences and adapting the interface to different cultural norms.

5.2.9. Accessibility Improvements:

   - Enhance accessibility features to ensure that the application is usable by individuals with disabilities, aligning with inclusive design principles.

5.2.10. Scalability and Performance Optimization:

   - Continuously optimize the application's scalability and performance to accommodate a growing user base and evolving technological landscapes.