B.E. PROJECT REPORT ON

CROSS PLATFORM APPLICATION FOR MAJOR PROJECT MANAGEMENT AND TRACKING

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY,
PUNE IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE
DEGREE OF

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

SUBMITTED BY

STUDENT NAME:	**Exam No:**
NAMDAS PRANOTI ANKUSH	**B400360699**
NIRMAL ADITYA SANJAY	**B400360707**
PATIL OMKAR SARJERAO	**B400360727**
PATIL YASH NILKANTH	**B400360734**

Sinhgad Institutes

DEPARTMENT OF COMPUTER ENGINEERING

STES'S SMT. KASHIBAI NAVALE COLLEGE OF ENGINEERING

VADGAON BK, OFF SINHGAD ROAD, PUNE - 411041

SAVITRIBAI PHULE PUNE UNIVERSITY

2024-2025

Sinhgad Institutes

CERTIFICATE

This is to certify that the project report entitles

"CROSS PLATFORM APPLICATION FOR MAJOR PROJECT MANAGEMENT AND TRACKING"

Submitted by

STUDENT NAME	Exam No:
NAMDAS PRANOTI ANKUSH	**B400360699**
NIRMAL ADITYA SANJAY	**B400360707**
PATIL OMKAR SARJERAO	**B400360727**
PATIL YASH NILKANTH	**B400360734**

is a bonafide student of this institute and the work has been carried out by her under the supervision of **Prof. Vrushali Paithankar** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Engineering** (Computer Engineering).

(Prof. Vrushali Paithankar)
Guide
Department of Computer Engineering

External Guide

(Prof. R. H. Borhade)
Head,
Department of Computer Engineering

(Dr. A. V. Deshpande)
Principal,
Smt. Kashibai Navale College Of Engineering, Pune - 41

Date:

Place:

ACKNOWLEDGEMENT

The present world of competition there is a race of existence in which those are having will to come forward succeed. Project is like a bridge between theoretical and practical working. With this willing we joined this particular project. First of all, we would like to thank the supreme power the Almighty God who is obviously the one has always guided us to work on the right path of life.

We sincerely thank Prof. R. H. Borhade Sir, Head of the Department of Computer Science of Smt. Kashibai Navale college of engineering, for all the facilities provided to us in the pursuit of this project. We are indebted to our project guide Prof Vrushali Paithankar, Department of Computer Engineering of Smt. Kashibai Navale college of engineering. We feel it's a pleasure to be indebted to our guide for his valuable support, advice and encouragement and we think him for his superb and constant guidance towards this project.

We are deeply grateful to all the staff members of Computer department, for supporting us in all aspects. We acknowledge our deep sense of gratitude to our loving parents for being a constant source of inspiration and motivation.

NAME OF THE STUDENTS
Namdas Pranoti Ankush
Nirmal Aditya Sanjay
Patil Omkar Sarjerao
Patil Yash Nilkanth

ABSTRACT

Conventional management tools frequently fail to provide real-time responsiveness, effective resource allocation, and seamless cross-platform support in the context of large-scale project execution. A Cross- Platform Application for Major Project Management and Tracking, created with Flutter, is presented in this paper as a comprehensive solution that guarantees device-agnostic access on desktops, tablets, and smartphones. The system solves important issues with current platforms, such as platform dependency, disjointed team communication, and inefficient task monitoring. The suggested approach uses a hierarchical task delegation model in which project managers give leads tasks to complete and then divide those tasks into smaller, deadline-driven tasks. Team members are always informed thanks to Firebase's real-time updates and notifications, which cut down on misunderstandings and delays. Better tracking of project milestones and the identification of workflow bottlenecks are made possible by the application's sophisticated data visualizations and user-friendly interface. The system maximizes resource utilization, facilitates well-informed decision-making, and improves project workflow transparency through integrated analytics and stringent data security procedures. In order to establish this cross-platform tool as a scalable, effective, and safe substitute for traditional project management services in contemporary, fast-paced settings, this study examines its conception, deployment, and expected effects.

Keywords: Flutter Development, Firebase, Cloud, SQL lite, Operations Management, Real-Time Collaboration, Database, Resource Allocation, Team Communication.

TABLE OF CONTENTS

LIST OF ABBREVATIONS

ABBREVIATION	ILLUSTRATION
API	Application Programming Interface
CPU	Central Processing Unit
ERP	Enterprise Resource Planning
GPU	Graphics Processing Unit
RAM	Random Access Memory
SQL	Structured Query Language
SSD	Solid State Drive
UI	User Interface

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

In an era characterized by rapid technological advancements and dynamic market conditions, effective project management has emerged as a crucial determinant of organizational success. Large-scale projects, often complex in nature and involving multiple teams, present significant challenges that traditional project management tools struggle to address. This limitation often results in inefficiencies, communication breakdowns, and missed deadlines, hindering overall productivity.

To overcome these challenges, we propose the development of a Cross-Platform Application for Major Project Management and Tracking, designed using Flutter. This application aims to provide a streamlined and intuitive solution for managing projects across various devices, including desktops, tablets, and smartphones. By enabling real-time updates, seamless task allocation, and enhanced collaboration among team members, the application seeks to foster a more efficient and responsive work environment.

The application will cater to the needs of project managers, team leads, and employees, ensuring a cohesive and transparent approach to project management. By integrating advanced features such as reporting, analytics, and user-friendly interfaces, this cross-platform application is poised to revolutionize the way organizations manage projects, ultimately driving greater agility and competitiveness in today's fast-paced business landscape.

This application provides a user-friendly interface and powerful visualization capabilities to track the project’s milestones and identify bottlenecks, enabling detailed reports. The integrated analytics offer actionable insights into how the team is performing, resource allocation, and project timeline in order to enhance strategic planning and reduce risks. The communication tools provided within the application minimize time spent on other external platforms, as everything about the project is contained in one place. Data security also forms a paramount aspect of this solution, ensuring that all sensitive project information is protected from unauthorized access.

1.2 MOTIVATION

- **Complex Projects**: Managing large-scale projects requires an efficient tool to handle multiple teams and tight deadlines.

- **Limitations of Existing Tools**: Traditional project management solutions often lack flexibility and real-time capabilities, leading to inefficiencies.

- **Real-Time Collaboration**: Seamless interaction among team members is critical for project success.

- **Cross-Platform Accessibility**: Users need access to management tools across various devices to enhance productivity.

- **User-Friendly Design**: An intuitive interface can improve user adoption and satisfaction.

- **Informed Decision-Making**: Analytics and reporting features provide better visibility into project progress.

- **Scalability:** Organizations need solutions that can grow with them without significant costs.

- **Competitive Edge:** Innovative project management solutions enable organizations to respond effectively to market demands..

- **Manual Tracking and Communication Gaps**: Traditional project management often involves manual tracking, leading to miscommunication and time loss.

- **Challenges in Monitoring and Control :** Lack of structured role-based control makes it hard to monitor performance and project progress.

- **Importance of Mobile Accessibility:** There's a need for a mobile-first solution to manage tasks anytime, anywhere.

- **Promoting Organizational Efficiency:** This system helps organizations maintain transparency, accountability, and smooth communication across all project stakeholders.

.

1.3 PROBLEM DEFINITION AND OBJECTIVE OF THE PROJECT

1.3.1 PROBLEM DEFINITION

In businesses and organizations, effectively managing large projects across multiple teams presents critical challenges. Traditional methods often fall short in providing the visibility, control, and accountability needed for efficient project execution. Project managers face difficulties in overseeing timelines, coordinating with team leads, and monitoring task progress, leading to risks of miscommunication, resource inefficiencies, and project delays.

Team leads require a comprehensive solution to break down projects into actionable tasks, assign these tasks to team members, and monitor task progress, ensuring accountability and alignment with organizational goals. Employees need an accessible way to understand task requirements, deadlines, and update their progress, reducing bottlenecks and maintaining clarity throughout the project lifecycle.

The "Cross-platform application for Major Project Management and Tracking" aims to address these issues by providing an organized, cross-platform solution designed to enhance communication, accountability, and visibility across business and organizational projects, supporting effective workflow management and timely project completion.

1.3.2 OBJECTIVE OF THE PROJECT

The main objectives of the model are: -

- **Streamlined Task Management**: To provide a structured framework for assigning, tracking, and updating tasks among team members, ensuring clarity in responsibilities and deadlines.

- **Real-Time Collaboration**: To facilitate seamless communication among project stakeholders through integrated messaging and notification systems, enabling timely updates and coordination.

- **Cross-Platform Compatibility**: To develop a user-friendly application that functions consistently across various devices (desktops, tablets, smartphones), allowing users to access project information anytime and anywhere.

- **Enhanced Reporting and Analytics**: To incorporate robust analytics and reporting features that provide insights into project progress, resource allocation, helping teams make and optimize project outcomes.

1.4 PROJECT SCOPE AND LIMITATIONS

1.4.1 Project Scope

1. **User Management :** The application will support user registration, authentication, and role-based access, allowing project managers, team leads, and employees to interact based on their specific responsibilities

2. **Project and Task Lifecycle :** The scope includes creating, editing, and deleting projects and tasks. Users will be able to assign tasks to team members with detailed descriptions, priorities, and deadlines, facilitating effective project execution.

3. **Real-Time Updates :** The application will provide real-time updates for task assignments, status changes, and approaching deadlines, ensuring that all team members are informed and can respond promptly.

4. **Reporting and Analytics Tools :** The project will include features for generating reports and visualizing project data, enabling teams to track milestones, identify bottlenecks, and assess performance metrics for informed decision-making.

1.4.2 Limitations

1. **Dependency on Internet Connectivity :** The application requires a stable internet connection for real-time collaboration and updates, which may hinder access in areas with poor connectivity.

2. **Device Compatibility Issues :** While designed for cross-platform use, variations in device capabilities and screen sizes may lead to inconsistent user experiences or functionality.

3. **User Adoption Challenges :** Transitioning to a new project management tool may encounter resistance from team members accustomed to existing workflows, requiring additional training and support to ensure successful adoption.

1.5 METHODOLOGIES OF PROBLEM SOLVING

1. **Requirement Gathering and Analysis:** We identified the core requirements of the application by understanding the roles and responsibilities of managers, team leads, and employees in project management environments.

2. **Problem Decomposition:** The entire system was divided into smaller modules such as authentication, role-based dashboards, project assignment, task management, and progress tracking for better clarity and modular development.

3. **Role-Based Access Design:** The application logic was structured around different user roles, each with a specific set of functionalities and restricted access to certain modules to ensure proper authorization and task segregation.

4. **Top-Down Design Approach:** A high-level system design was first created, followed by the detailed design and implementation of each individual module. This ensured a clear development roadmap.

5. **Algorithmic Thinking:** Logical functions such as progress percentage calculation, task assignment, and completion status updates were carefully designed to provide accurate and dynamic feedback in real-time.

6. **Agile Development Methodology:** The development process followed an iterative approach, where the application was built in sprints. Each sprint focused on completing and testing a core feature of the system.

7. **Database Modeling:** A structured Firestore database schema was designed to manage user data, project details, task assignments, and real-time progress updates with efficient read/write operations.

8. **Testing and Debugging:** Unit and integration testing were conducted throughout development. Errors were diagnosed using logs and root cause analysis to ensure functional accuracy and system stability.

9. **User Interface Prototyping:** UI designs were planned and developed to offer a responsive and user-friendly experience across multiple devices, ensuring that each role has a clear and simple interaction flow.

10. **Deployment and Maintenance Planning:** Firebase was used for hosting and backend services. Deployment procedures and post-launch maintenance strategies were defined to ensure the system remains scalable and up-to-date.

CHAPTER 2
LITERATURE SURVEY

1. **Paper name**: project management in the cloud: benefits, risks and challenges
   Authors: Alena k. Kozak, Iryna d. Morozova, Yuliya v. Ryzhkova
   Published in: journal of management information systems

Abstarct:

This paper explores the integration of cloud computing in project management, focusing on the benefits, risks, and challenges associated with its adoption. The authors conduct a comprehensive literature review and empirical analysis to highlight how cloud-based tools enhance collaboration and efficiency in project execution. The findings indicate that while cloud solutions offer significant advantages such as reduced costs and improved accessibility, organizations must also address concerns related to data security and user adoption to fully leverage these technologies.

2. **Paper name**: Cloud Erp: a review of the adoption challenges and benefits
   Authors: E. Tongsuksai, P. L. Tongsuksai, T. Suksaeng
   Published in: international journal of information management, 2023

Abstract:

This paper reviews the current landscape of cloud ERP systems, examining the key challenges and benefits associated with their implementation in organizations. Through an analysis of case studies and surveys, the authors identify critical success factors that influence the adoption of cloud ERP, including organizational culture, technology readiness, and security considerations. The paper concludes that while cloud ERP offers scalability and flexibility, organizations must develop comprehensive strategies to overcome implementation hurdles and maximize their investment.

3. **Paper name**: the impact of cloud computing on project management: a case study approach
   Authors: v. Mokar, r. Jain, s. D. Patel
   Published in: international journal of project management, 2023

Abstarct:

This study investigates the impact of cloud computing technologies on project management practices through a series of case studies across various industries. The authors analyze how cloud-based tools facilitate project planning, execution, and monitoring. The results reveal that cloud computing enhances collaboration, reduces project cycle times, and improves resource allocation. However, the paper also discusses the potential risks associated with cloud dependency and the need for robust data management strategies to mitigate these risks.

4. **Paper name**: challenges and opportunities in implementing cloud erp systems
   Authors: S. Appandairajan, r. Prasad, m. K. Verma
   Published in: journal of business research, 2024

Abstract:

This paper examines the challenges and opportunities presented by cloud ERP systems in modern businesses. Through a mixed-methods approach, the authors identify the key barriers to implementation, including integration with legacy systems, user training, and data security concerns. The findings suggest that organizations that proactively address these challenges can harness the benefits of cloud ERP, such as improved agility and reduced operational costs. The paper emphasizes the importance of strategic planning and stakeholder engagement in successful cloud ERP adoption.

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATION

3.1 ASSUMPTIONS AND DEPENDENCIES

- **User Accessibility**: Users will have access to compatible devices (desktops, tablets, smartphones) and a stable internet connection.
- **Data Accuracy**: Users will provide accurate and timely information regarding tasks, deadlines, and project updates.
- **Security Compliance**: The application will adhere to relevant data protection regulations and standards.
- **Stakeholder Engagement**: All stakeholders will actively participate and provide feedback during the implementation process.
- **Development Framework**: The application's performance depends on the Flutter framework.

3.2 FUNCTIONAL REQUIREMENTS

1. **User Registration**:
   - Users should be able to create an account with a username, password, and email.
   - It should support role-based access, enabling different functionalities for project managers, team leads, and employees.
2. **Project and Task Management**:
   - Users should be able to create, edit, and delete projects and tasks.
   - The application must allow project managers to assign tasks to team leads, who can further subdivide them for team members with defined deadlines and priorities.
3. **Cross-Platform Functionality**:
   - The application must work seamlessly across desktops, tablets, and smartphones.
4. **Data Security**:
   - The application must implement data protection measures, including encryption and regular backups.

3.3 EXTERNAL INTERFACE REQUIREMENTS

3.3.1 USER INTERFACE

- The application should have a clean and intuitive user interface with easy navigation.
- The UI must be responsive and adaptable for desktops, tablets, and smartphones, ensuring a consistent experience across devices.
- It should include user-friendly forms for task and project management, notifications, and reporting.

3.3.2 HARDWARE INTERFACE

- The application should be compatible with standard hardware configurations, including desktops, laptops, tablets, and smartphones
- It must support touch input for mobile devices and responsive design for various screen resolutions.

3.3.3 SOFTWARE INTERFACE

- **Platform Required:** Windows / Linux
- **IDE:** VS Code, Android Studio
- **Programming Language:** Dart
- **Database:** Firestore, SQLite

3.4 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements cover all the remaining aspects of a software system that are not addressed by the functional requirements. While functional requirements define what the system should do such as features, tasks, or actions non-functional requirements specify how well those functions must be performed. They set the criteria that judge the quality and operation of a system rather than its specific behaviors. These requirements are essential to ensure the overall usability, performance, and effectiveness of the application under real-world conditions.They specify criteria that judge the operation of a system, rather than specific behaviors, for example: Some typical non-functional requirements are

3.4.1 PERFORMANCES REQUIREMENTS :

1. **System Response Time:** The application must ensure that user actions (such as loading projects, task creation, and updates) are completed within 2 seconds under normal operating conditions.

2. **Scalability:** The system should be able to handle at least 100 concurrent users without any degradation in performance.

3. **Maintainability:** The codebase should be modular and well-documented, allowing for easy updates and modifications.

4. **Usability:** The application must have an intuitive and visually appealing interface that facilitates easy navigation for all user roles.

3.4.2 SAFETY REQUIRMENT

1. **Data Integrity:** The system ensures that all project and task data is accurately stored and not altered or deleted by unauthorized users.

2. **Access Control:** Role-based access restricts users from performing unauthorized actions, ensuring safe handling of sensitive operations.

3. **Error Handling:** Proper error messages and fallback mechanisms are implemented to prevent system crashes and guide users during unexpected inputs or failures.

4. **Data Backup:** Firebase Firestore automatically manages data replication and backup, reducing the risk of data loss.

5. **Session Safety:** Secure session management is implemented to prevent unauthorized use, especially on shared or public devices.

3.4.3 SECURITY REQUIREMENT

1. **Authentication and Authorization:** The system uses Firebase Authentication to ensure that only registered users can log in. Role-based access control is implemented to restrict functionalities based on user roles (Manager, Team Lead, Employee).

2. **Secure Communication:** All data transmission between the client application and Firebase is encrypted using HTTPS to prevent interception or tampering.

3. **Data Privacy:** User-specific data is protected by Firebase Security Rules, ensuring that users can only access data relevant to their roles and projects.

4. **Input Validation:** All input fields are validated on both client and server sides to prevent injection attacks or invalid data entry.

3.4.4 SOFTWARE QUALITY ATTRIBUTES

1. **Usability:** The application is designed with a clean and intuitive user interface using Flutter, ensuring ease of navigation and interaction for users with different technical backgrounds.

2. **Reliability:** The system consistently performs its intended functions, such as task updates and progress tracking, without failure under normal operating conditions.

3. **Scalability:** By using Firebase as a backend, the system can scale to support more users and projects without major architectural changes.

4. **Maintainability:** Modular code structure and proper documentation ensure that future updates or bug fixes can be made efficiently.

5. **Performance Efficiency:** Real-time updates using Firestore ensure minimal latency in data retrieval and synchronization, leading to faster performance.

3.5 SYSTEM REQUIREMENTS

3.5.1 DATABASE REQUIRMENT

- The application uses Firebase Fire store, a cloud-based NoSQL database that enables real-time data storage and synchronization. It organizes data into collections and documents, supports role-based access control, and scales automatically with growing user and data needs.

3.5.2 SOFTWARE

- **Operating System:** Windows 10 (Recommended: Windows 11).
- **CPU:** Intel or AMD processor with 64-bit support (Recommended: 2.8 GHz or faster).
- **RAM:** Minimum 8 GB (Recommended: 16 GB).
- **GPU:** Integrated graphics or NVIDIA GeForce GTX 1050 (Recommended: NVIDIA GeForce GT 1660 or Quadro T1000)
- **Internet:** Stable internet connection required.
- **Platform:** Flutter SDK.
- **Development Environment:** Android Studio or Visual Studio Code.
- **Programming Language:** Dart.

- **Database:** MongoDB or Firebase.
- **Framework:** Flutter for cross-platform application development.
- **Version Control:** Git for source code management.

3.5.3 HARDWARE

Some products include both minimum and recommended system requirements. A video game, for instance, may function with the minimum required CPU and GPU, but it will perform better with the recommended hardware. A More powerful processor and graphics card may produce improved graphics and faster frame rates. Some system requirements are not flexible, such as the operating system(s) and disk space required for software installation. Others, such as CPU, GPU, and RAM requirements may vary significantly between the minimum and recommended requirements. When buying or upgrading a software program, it is often wise to make sure your system has close to the recommended requirements to ensure a good user experience.

- **Processor:**
  Recommended: Quad-core processor for smooth performance during multi-user access

- **RAM**:
  Recommended: 16 GB to handle larger projects and complex operations efficiently

- **Storage**:
  Recommended: 256 GB SSD for faster data access and improved application responsiveness

- **Graphics Card**:
  Recommended: NVIDIA GeForce GTX 1050 for better visualization features

- **Display**:
  Recommended: 1920 x 1080 resolution for clearer graphics and more workspace

- **Network Adapter**: Wi-Fi capability for essential internet connectivity and real-time collaboration.

3.6 SDLC MODEL

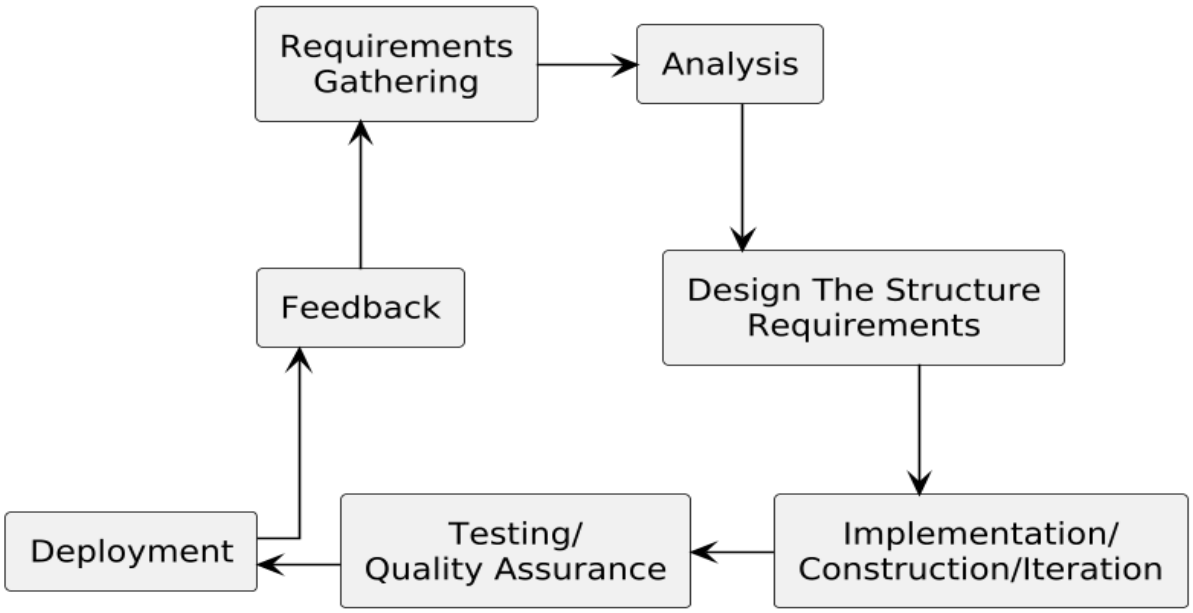


Fig. 3.1 Agile Model

The Agile model promotes iterative development and continuous feedback. The following are the sequential phases commonly followed in an Agile development cycle:

1. Requirement Gathering - In this phase, the development team interacts with stakeholders to gather functional and non-functional requirements. The team also estimates the time, cost, and effort required to implement the proposed features. Technical and economic feasibility studies are conducted to ensure the viability of the project.
2. Design the Requirement - This phase involves translating the gathered requirements into design documents. High-level UML diagrams or user-flow diagrams are created to visualize how new features will integrate with the existing system. Wireframes and UI mockups are developed to guide the implementation of user interfaces.
3. Construction / Iteration - During this phase, actual development begins. The team follows an iterative approach, implementing and delivering working features in short development cycles (sprints). The goal is to have a potentially shippable product at the end of each iteration.
4. Testing / Quality Assurance - This phase ensures the product's reliability, functionality, and performance through various testing methods:

    - **Unit Testing:** Tests individual components or functions to verify correctness in isolation.
    - **Integration Testing:** Ensures that different modules work together as intended.

Example: Verifying that the task assignment module integrates properly with the notification system.

- **System Testing:** Evaluates the complete system to ensure it meets all specified requirements.
  Example: Checking the end-to-end process from project creation to report generation.
- **User Acceptance Testing (UAT):** Confirms that the application meets user expectations and real-world scenarios. Example: Project managers and team leads verify dashboard usability and report accuracy.
- **Cross-Platform Testing:** Ensures consistent functionality and user experience across multiple devices. Example: Verifying UI on Android phones, tablets, and desktops.
- **Performance Testing:** Measures system speed, responsiveness, and stability under load. Example: Ensuring task updates are processed within 2 seconds, even with 100 users.
- **Security Testing:** Assesses the system's ability to protect data and enforce access control.
  Example: Confirming that unauthorized users cannot access restricted project data.

- **Unit Testing**: Unit testing is the process of checking small pieces of code to ensure that the individual parts of a program work properly on their own. Unit testing is used to test individual blocks (units) of code.

- **Integration Testing**: Integration testing is used to identify and resolve any issues that may arise when different units of the software are combined.

- **System Testing**: Goal is to ensure that the software meets the requirements of the users and that it works correctly in all possible scenarios.

5. Deployment: In this step, the development team will deploy the working project to end users.

6. Feedback: This is the last step of the Agile Model. In this, the team receives feedback about the product and works on correcting bugs based on feedback provided by the customer.

CHAPTER 4
SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

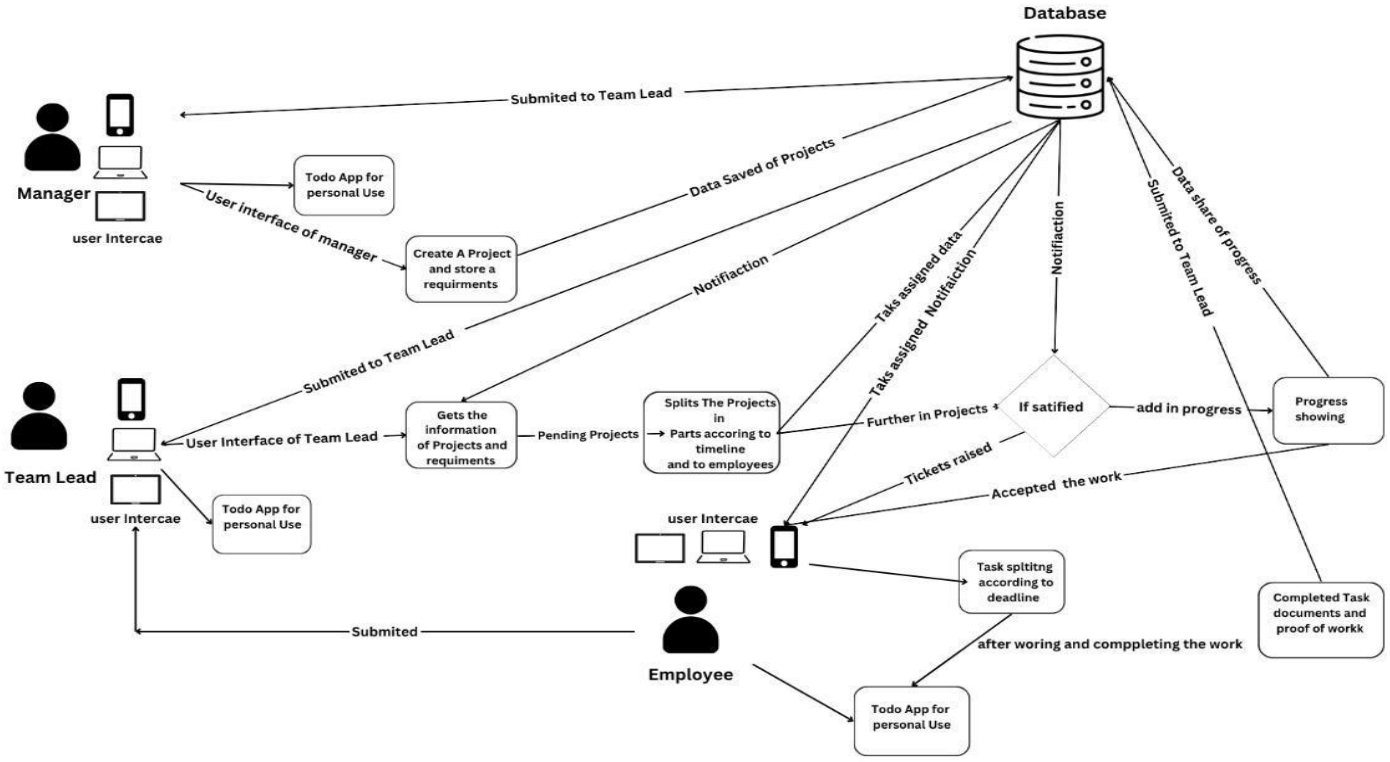


Fig 4.1 System Architecture

This system architecture diagram outlines the workflow of an application that involves multiple users, modules, and data interactions. Here's a breakdown of its components and their interactions:

1. **User Interaction:**
   - Users interact with the system in different roles (e.g., as teachers or project supervisors). The user can input or retrieve information from the system.

2. **Input & Authentication (User Side):**
   - **Login/Authentication:** Users must authenticate themselves through a login system before accessing various modules. This ensures secure access and user-specific content management.

3. **Data Handling and Modules (System Side):**

   - **Attendance:** Tracks attendance of students, linked to specific projects or sessions.
   - **Assessment:** Handles the evaluation of student projects. This module integrates with databases to store grades and feedback.

- **Project/Group Management:** Facilitates the creation and management of student project groups, assigning students to different groups under specific supervisors.

- **Guide Assignment:** Assigns guides (teachers or mentors) to student projects, facilitating supervision throughout the project lifecycle.

4. **Centralized Database:**

   - All modules interact with a centralized database for data storage and retrieval. The database stores information on student groups, projects, grades, and user details.

5. **Evaluation & Feedback:**

   - The **Evaluation** component interacts with both the database and the project/group management system to pull relevant project data for grading. Teachers or supervisors input assessments, which are stored in the database.

6. **Output (User Side):**

   - The system generates outputs based on the data collected and processed by the modules (e.g., attendance records, project assessments).

This architecture represents an integrated system designed to manage various aspects of a student project lifecycle, including group management, evaluation, and reporting.

4.2 DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

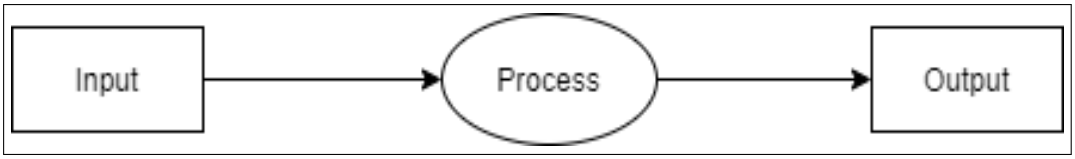


Fig 4.2 Data Flow Diagram-0

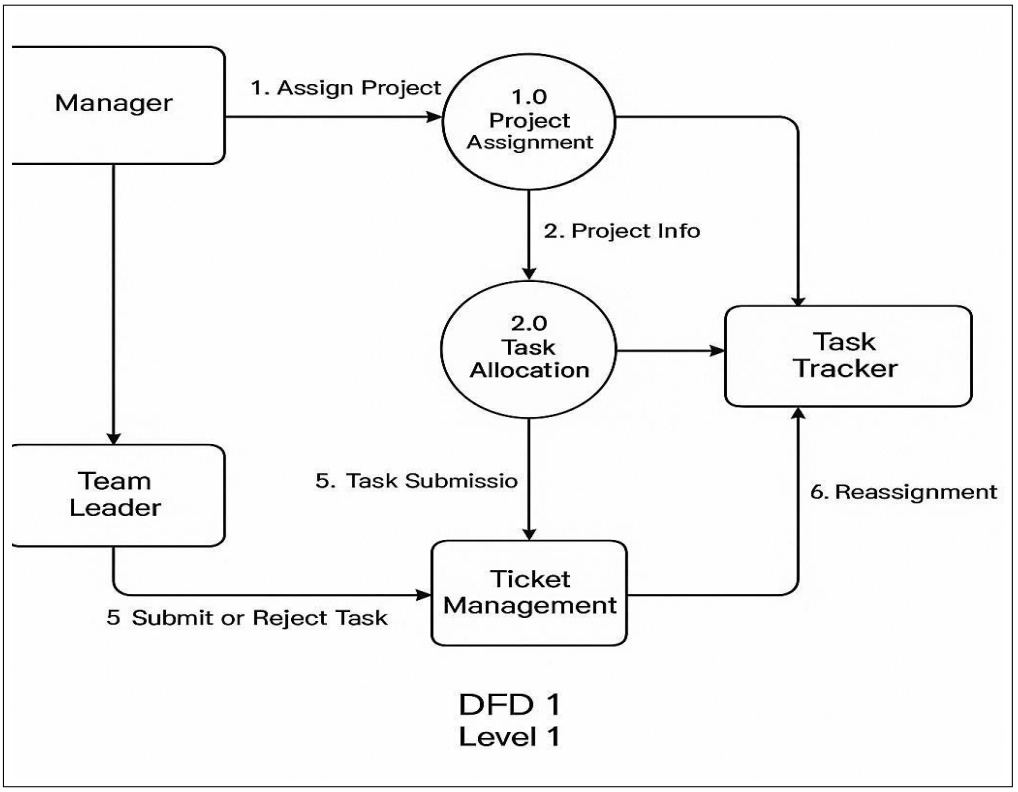


Fig 4.3 Data Flow Diagram-1

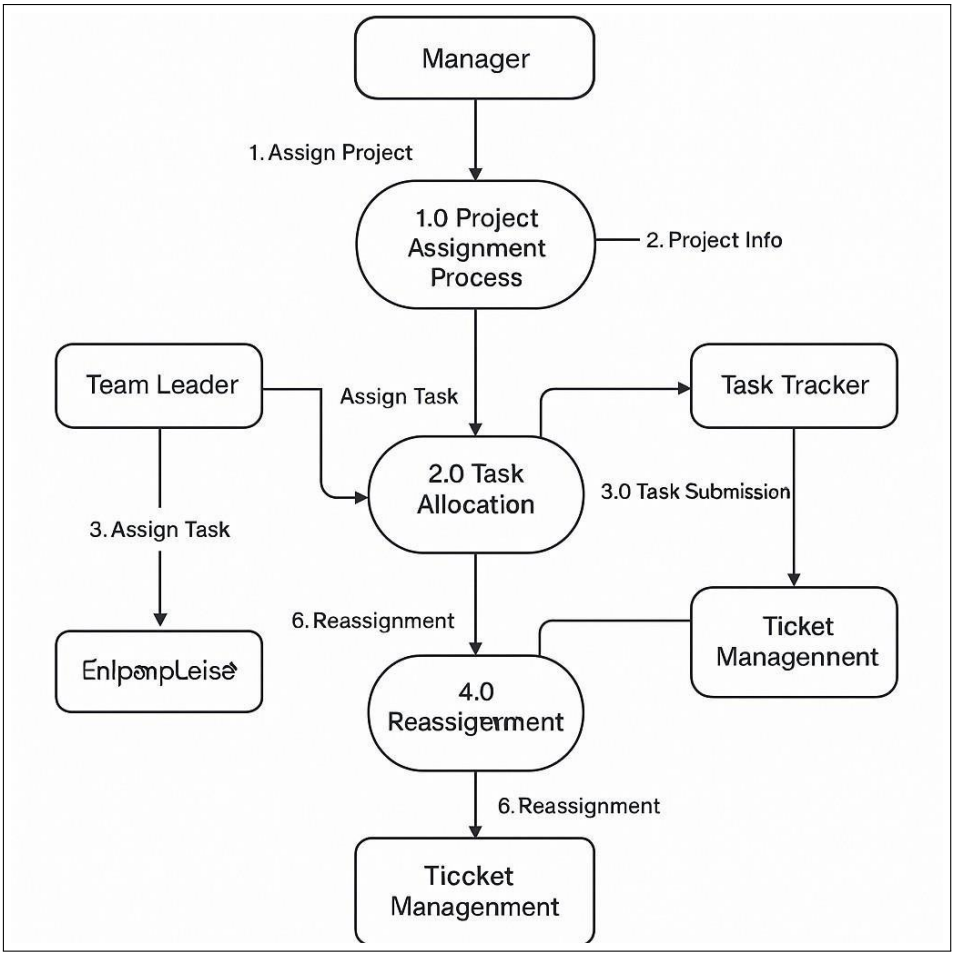


Fig 4.4 Data Flow Diagram-2

4.3 ENTITY RELATIONSHIP DIAGRAM

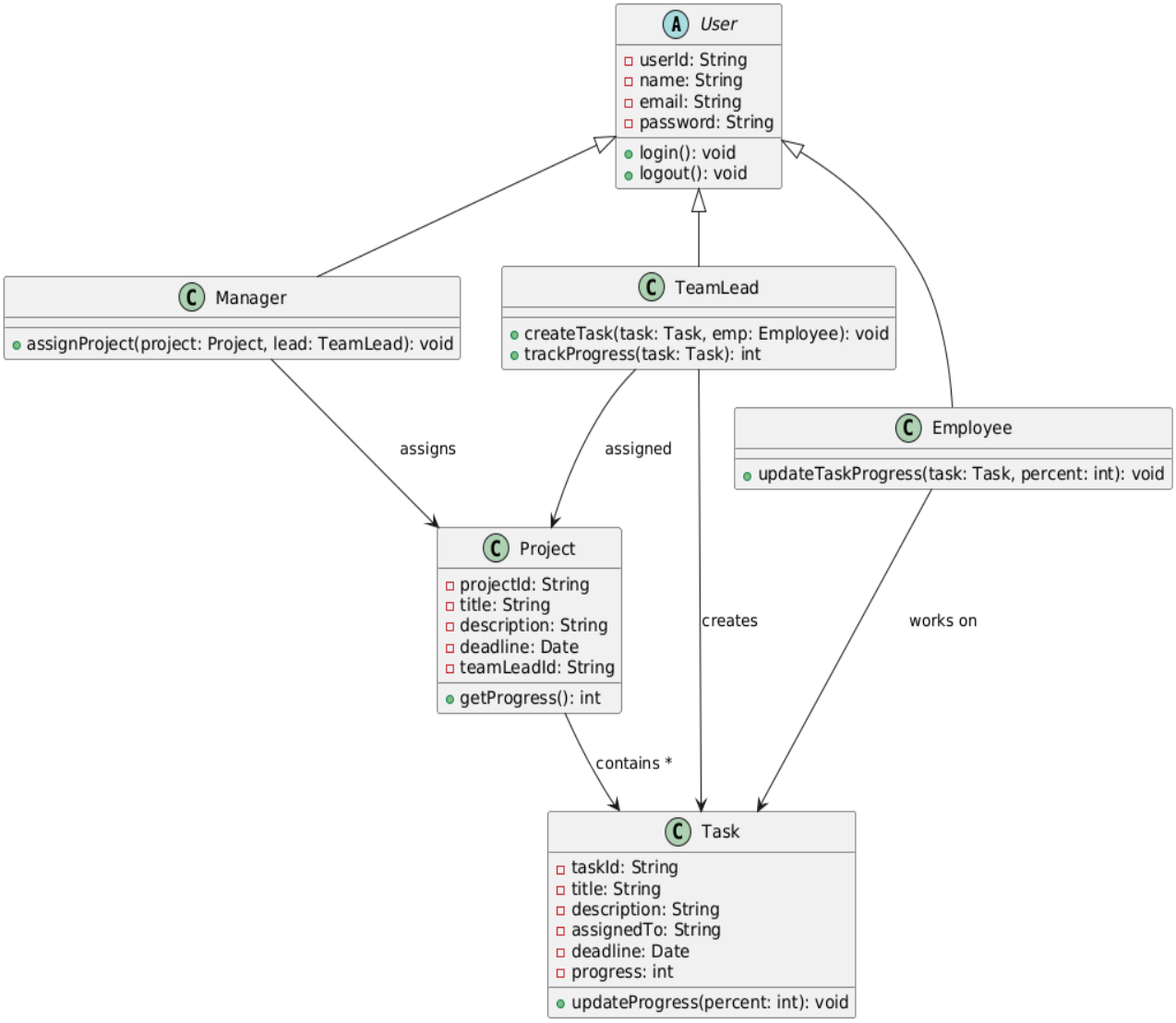


Fig 4.5 ER Diagram

4.4 UML DIAGRAMS

4.4.1 Component Diagram

A Component Diagram displays the structural relationship of components of a software system. These are mostly used when working with complex systems that have many components. Components communicate with each other using interfaces. The interfaces are linked using connectors**.**

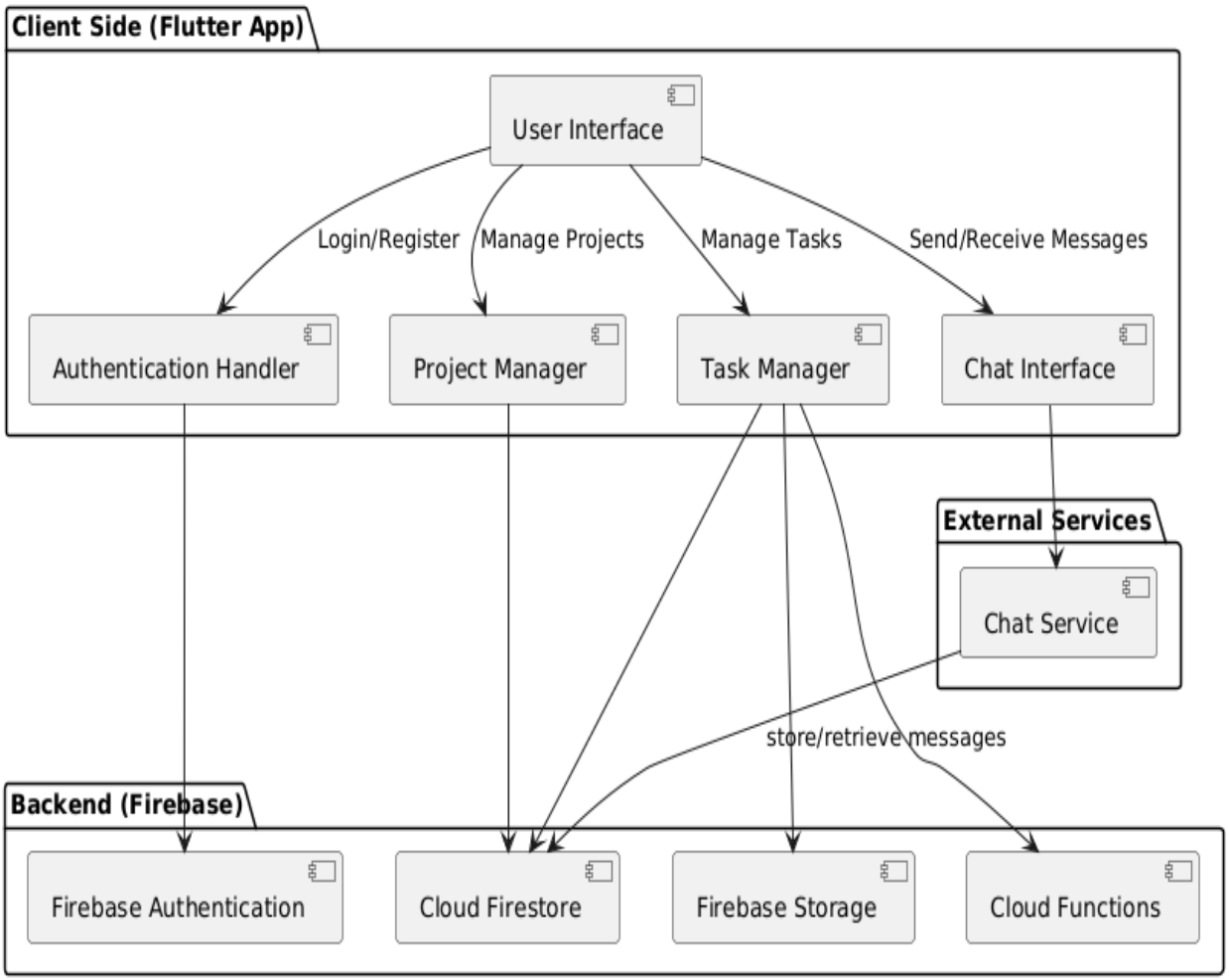


Fig 4.6 Component Diagram

4.4.2 Activity Diagram

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modelling.

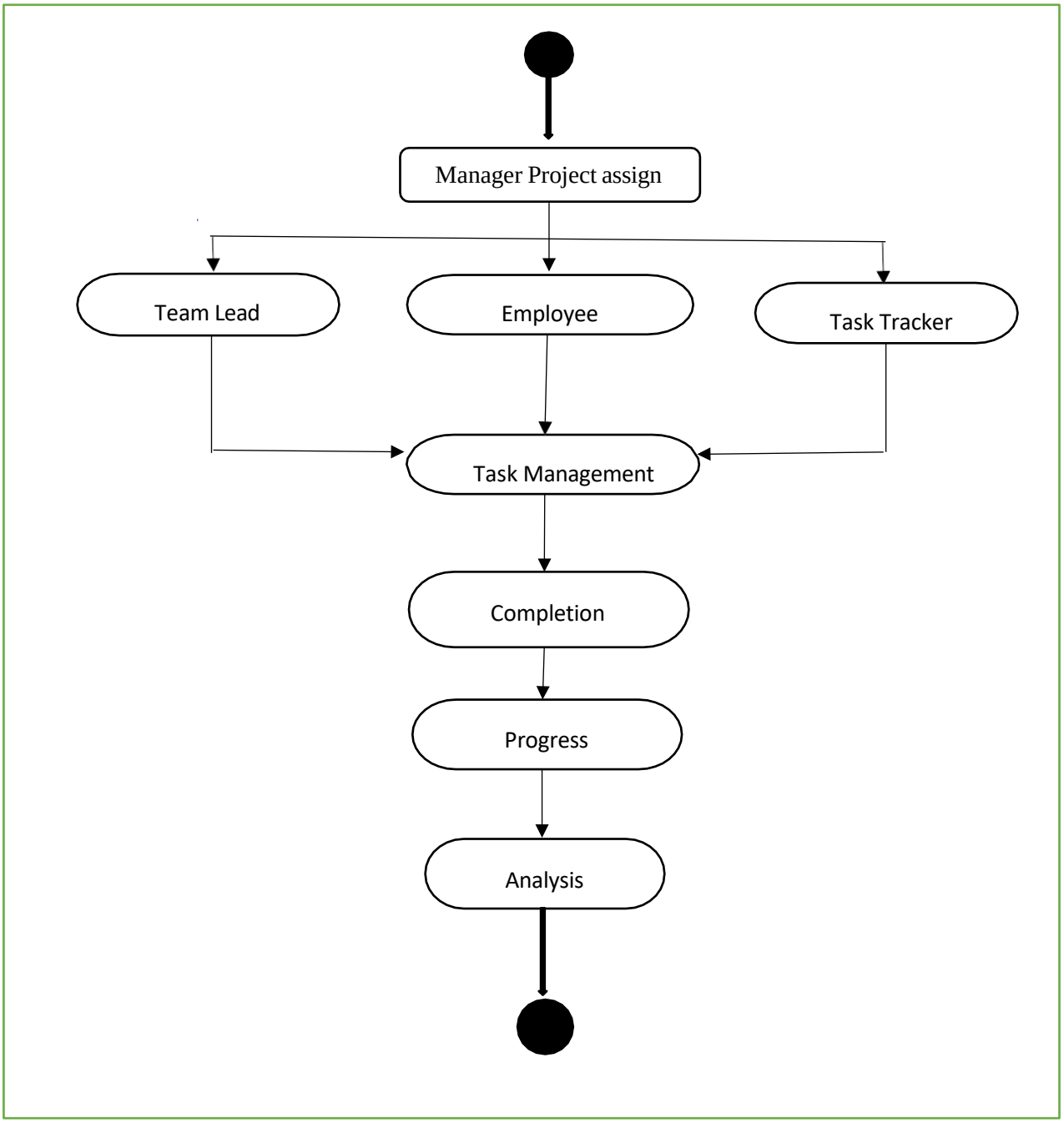


Fig 4.7 Activity Diagram

4.4.3 Sequence Diagram

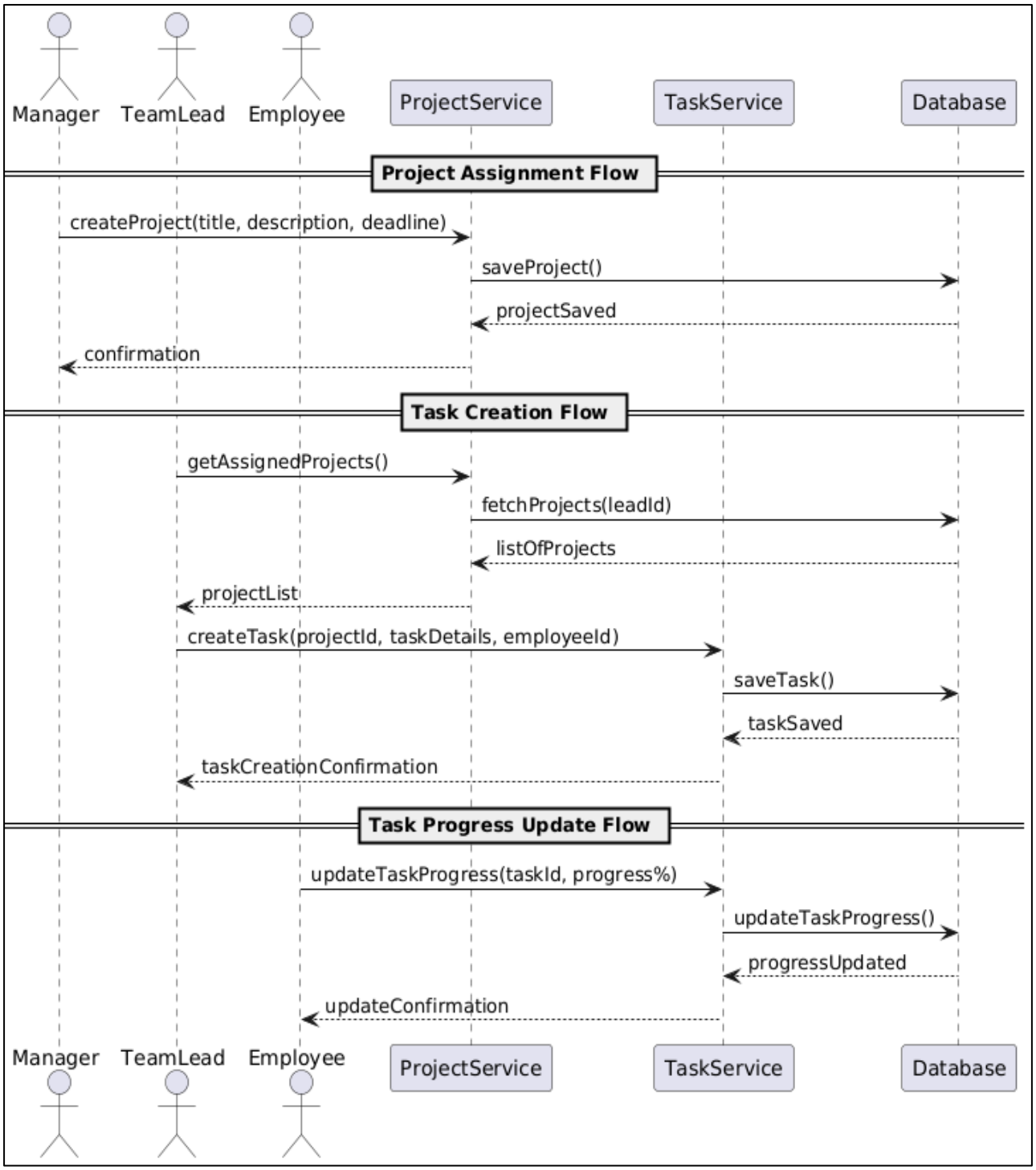


Fig 4.8 Sequence Diagram

4.4.4 Use Case Diagram

Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.

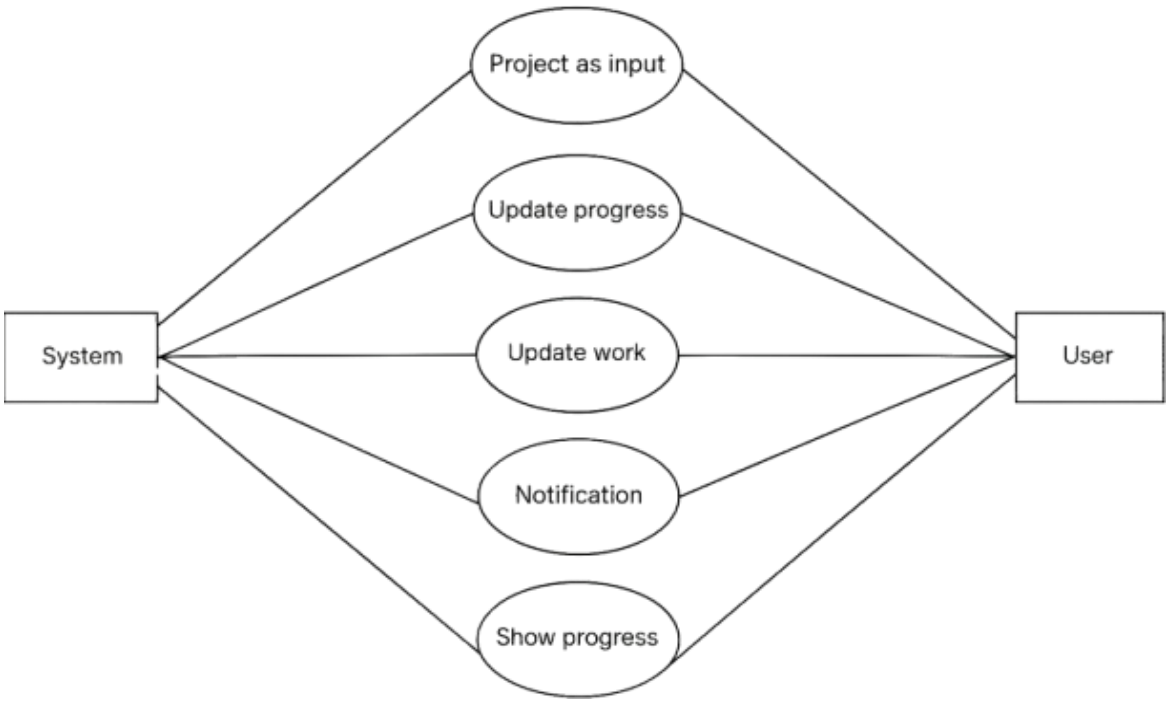


Fig 4.9 Use Case Diagram

4.4.5 Deployment Diagram

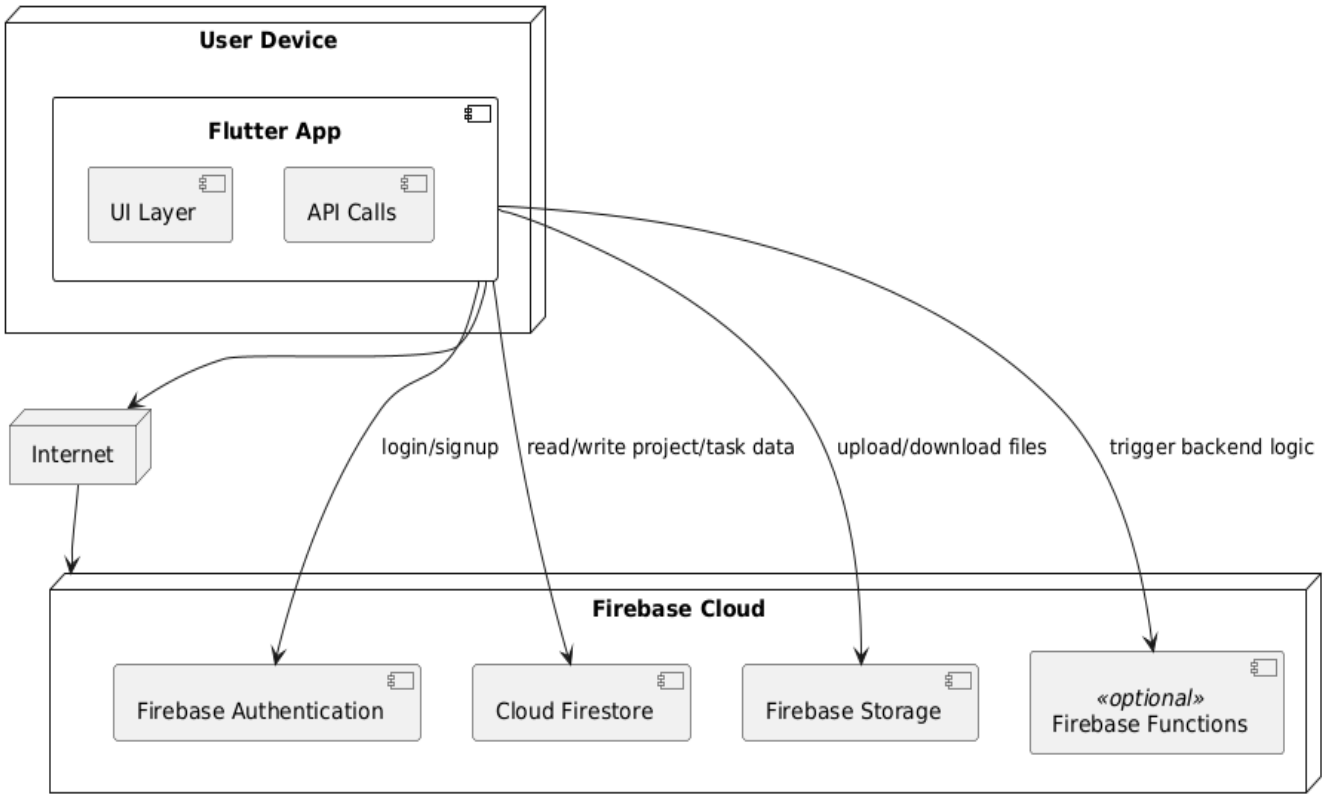


Fig. 4.10 Deployment Diagram

4.4.6 State Machine Diagram

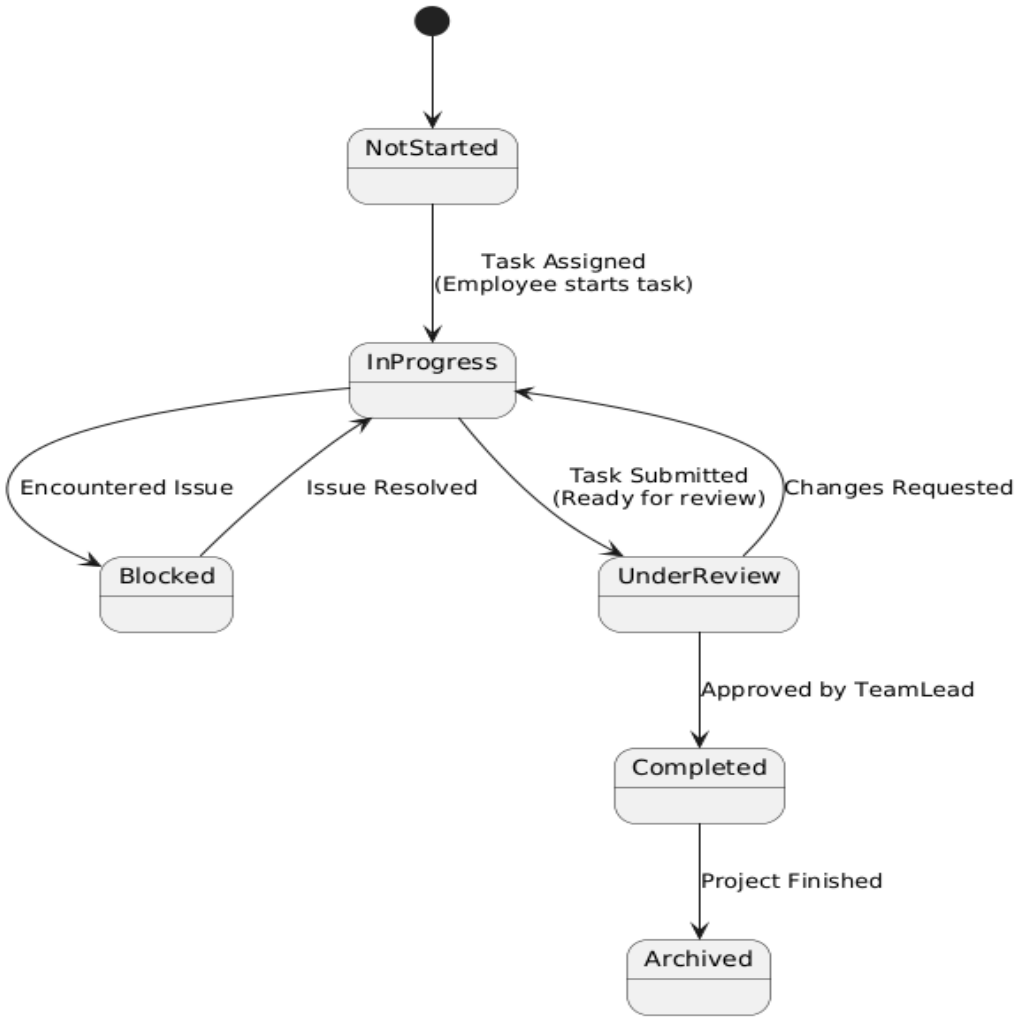


Fig. 4.11 State Machine Diagram

CHAPTER 5
PROJECT PLAN

5.1 PROJECT ESTIMATE

5.1.1 Reconciled Estimates

Cost Estimate

- Cost will estimate after completing the project that depend on time to complete the

  project. Also, efforts required to complete**.**

- WhatsApp messaging service charges are also included in it.

Time Estimates

- Time will depend on modules of project.
- Also, project plan of execution**.**
- Time can also depend on the donor search speed for requestor.

5.1.2 Project Resources

- **Hardware Resources Required:**
  - System: Quad-core processor, 16 GB RAM
  - Storage: Minimum 256 GB SSD
  - Device: Desktop /Laptop
- **Software Resources Required:**
  - Operating system: Microsoft Windows.
  - Coding Language: dart
  - IDE: Visual Studio
  - Code Platform: Flutter SDK

5.2 RISK MANAGEMENT

5.2.1 Risk Identification

- Assigning specific role to specified user and allow role base authentication and login to system.
- While accepting data from user there is risk cross site script injection on system.
- Providing secure and fast communication.
- Lack of data for training models.
- More time and resources are required for training.

5.2.2 Risk Analysis

- **RISK ID 1**:
  Firebase Authentication is used for role-based access (Manager, Team Lead, Employee). Misconfiguration in Firebase rules may allow unauthorized access.
- **RISK ID 2**:
  During user login or registration, form validation is enforced to ensure proper entries are submitted (e.g., valid email, password strength).
- **RISK ID 3**:
  A stable and fast internet connection is required as the app relies on Firebase services like Firestore, Storage, and Authentication.
- **RISK ID 4**:
  Firestore chat system, if not optimized (e.g., no pagination or message limits), can cause excessive document reads and degrade performance.
- **RISK ID 5**:
  To minimize time and resource constraints, we leveraged Firebase's integrated backend services (like Firestore and Storage) and prebuilt Flutter packages.

5.2.3 Overview of Risk Mitigation, Monitoring, Management

- **Identification of Risks :** Potential risks such as data inconsistency, authentication failures, cloud service outages, and role-based access errors were identified during the planning phase of the project.
- **Risk Assessment and Categorization :** Risks were assessed based on their likelihood and impact. They were categorized into technical, operational, and user-related risks to prioritize appropriate mitigation strategies.
- **Risk Mitigation Strategies :**
  - Implemented Firebase Authentication with secure login protocols to mitigate unauthorized access.

- Firestore rules and role-based access controls were enforced to prevent data leaks or misuse.
- Data is backed up and stored securely using Firebase's real-time sync and redundancy.
- Modular coding and testing help isolate failures without impacting the entire system.

- **Risk Monitoring :** Continuous monitoring was achieved through Firebase's built- in analytics and Crashlytics tools to detect errors and exceptions in real-time. These tools provide alerts and performance reports.
- **Contingency and Recovery Planning :** In case of critical service failures, fallback mechanisms such as retry logic, cached data access, and user alerts were designed. User progress data is frequently synced to prevent loss.
- **Ongoing Management :** Risk reviews were performed at the end of each development sprint. New risks identified during integration or user testing were added to a risk log and mitigated promptly.

5.3 PROJECT SCHEDULE

5.3.1 Project Task Set

Major Tasks in the Project stages are:

- **Task 1**: Requirement Analysis and Resource Gathering.
- **Task 2:** Technology Study (Flutter, Firebase, MongoDB).
- **Task 3**: Project Specification and Architecture Planning.
- **Task 4**: Design and Development of Prototype
- **Task 5**: Design Final User Interface for Cross-Platform Devices.
- **Task 6**: Coding and Implementation (Develop Manager, Team Lead, and Employee Modules).
- **Task 7:** Testing and Debugging (Module by Module).
- **Task 8**: Integration Testing and Final Deployment.

5.3.2 Project Timeline

Activity	July			August			September			
	week 1	week 2	week 3	week 1	week 2	week 3	week 1	week 2	week 3	week 4
Initiate the Project										
Communication										
Resource Gathering										
Study Technology										
Plan the Project										
Project Specification										
Design UML and ER diagram										
Design Test Plan										

Table 5.1 Timeline Chart Phase 1

Activity	Oct		Nov		Jan			Mar		Apr			May	
	week 1	week 2	week 1	week 2	week 1	week 2	week 3	week 1	week 2	week 1	week 2	week 3	week 1	Week 2
Execute the Project														
Design UI Prototype														
Build Final UI														
Build the Database														
Build and Test Login														
Integrate Database Service														
Build and Test User Service														
ChatApp														
Implement Algorithm														
Build and Test Security Feature														
Documentation														

Table 5.2 Timeline Chart Phase 2

5.4 TEAM ORGANIZATION

5.4.1 Team Structure

Responsibilities and duties were distributed among the four members according to their skills.

• One member was responsible for backend development, managing the server, database, and API integration.
• Another member focused on designing a responsive and user-friendly interface (frontend) for all devices.
• A third member handled creating test cases and conducting thorough testing of each module to ensure functionality and performance.
• The fourth member managed project coordination, including documentation, tracking progress, and ensuring smooth communication among the team. Management Reporting and Communication.

CHAPTER 6
PROJECT IMPLEMENTATION

6.1 OVERVIEW OF PROJECT MODULES

6.1.1 USER REGISTRATION AND AUTHENTICATION MODULE

Built using Flutter, this module ensures the application functions uniformly across different devices, including desktops, tablets, and smartphones. The responsive UI adapts automatically to different screen sizes and orientations.

6.1.2 Cross-Platform Compatibility Module

The admin module allows configuration of system settings, role management, and user permissions. It also supports maintaining system security, backups, and performance optimization.

6.1.3 Admin Control and Configuration Module

This module facilitates in-app communication through chat and comments, allowing seamless interaction between managers, leads, and employees. It eliminates dependency on third-party messaging apps and maintains all project discussions within the platform.

6.1.4 Communication Module

This module includes charts, graphs, and performance dashboards that help users track project milestones and identify potential bottlenecks. It provides insights into resource utilization and helps with informed decision-making.

6.1.5 Reporting and Analytics Module

This module provides real-time updates and alerts for task assignment, updates, and completion. It ensures that all team members stay informed, enhancing collaboration and reducing communication delays.

6.1.6 Real-Time Notification and Update Module

This is the core module where projects are created, and tasks are assigned hierarchically. Project managers assign tasks to team leads, who further assign subtasks to individual team members, along with deadlines and priorities. It ensures a smooth flow of responsibilities and tracks progress for each task.

6.1.7 Project and Task Management Module

This modules enables users to register and log in to the application. It supports role- based access control ensuring that project managers, team leads and employees access only the features assigned to their roles

6.2 TOOLS AND TECHNOLOGIES USED

Category	Technology Used	Purpose
Frontend	Flutter	Cross-platform UI development for Android, iOS, and Web
Backend & BaaS	Firebase (Firestore, Firebase Auth, Cloud Functions)	Real-time database, authentication, and serverless backend logic
Authentication	Firebase Authentication	Role-based secure login and user identity management
Database	Cloud Firestore	NoSQL database for storing projects, tasks, users, and progress data
Cloud Storage	Firebase Storage	Upload and manage documents or media associated with tasks or projects
Notification System	Firebase Cloud Messaging (FCM)	Real-time push notifications for task assignments and updates
Deployment & Hosting	Firebase Hosting	Hosting the application and related assets
Security	Firebase Rules, HTTPS,Role-based Access	Secure access control, encrypted data flow, and protection from abuse
Version Control	Git, GitHub	Code management, collaboration, and version tracking

Table 6.1 Tools and Technologies used

CHAPTER 7
SOFTWARE TESTING

7.1 TYPES OF TESTING

1. Unit Testing

Tests individual functions, components, or modules of the application to ensure that each part works as expected.
Example:
Testing the login function to verify if valid and invalid inputs return correct results.

2. Integration Testing

Verifies that different modules or services used by the application work well together.
Example:
Checking whether the task assignment module successfully integrates with the notification system.

3. System Testing

Evaluates the complete and integrated software system to ensure it meets the specified requirements.
Example:
Testing the full flow from project creation to task assignment, completion, and report generation.

4. User Acceptance Testing (UAT)

Ensures that the application works for end users and satisfies real-world use cases.
Example:
Team leads and project managers test the system to confirm that the dashboard, task tracking, and report sections meet their expectations.

5. Cross-Platform Testing

Checks that the application performs consistently across multiple platforms and devices.
Example:
Verifying proper UI display and functionality on Android phones, tablets, and Windows desktops.

6. Performance Testing

Measures responsiveness, stability, and speed under specific conditions.
Example:
Testing if task creation and updates occur within 2 seconds, even with 100 active users.

7. Security Testing

Evaluates the system's ability to protect data and maintain functionality as intended.
Example:
Ensuring role-based access is enforced and unauthorized users cannot access project data.

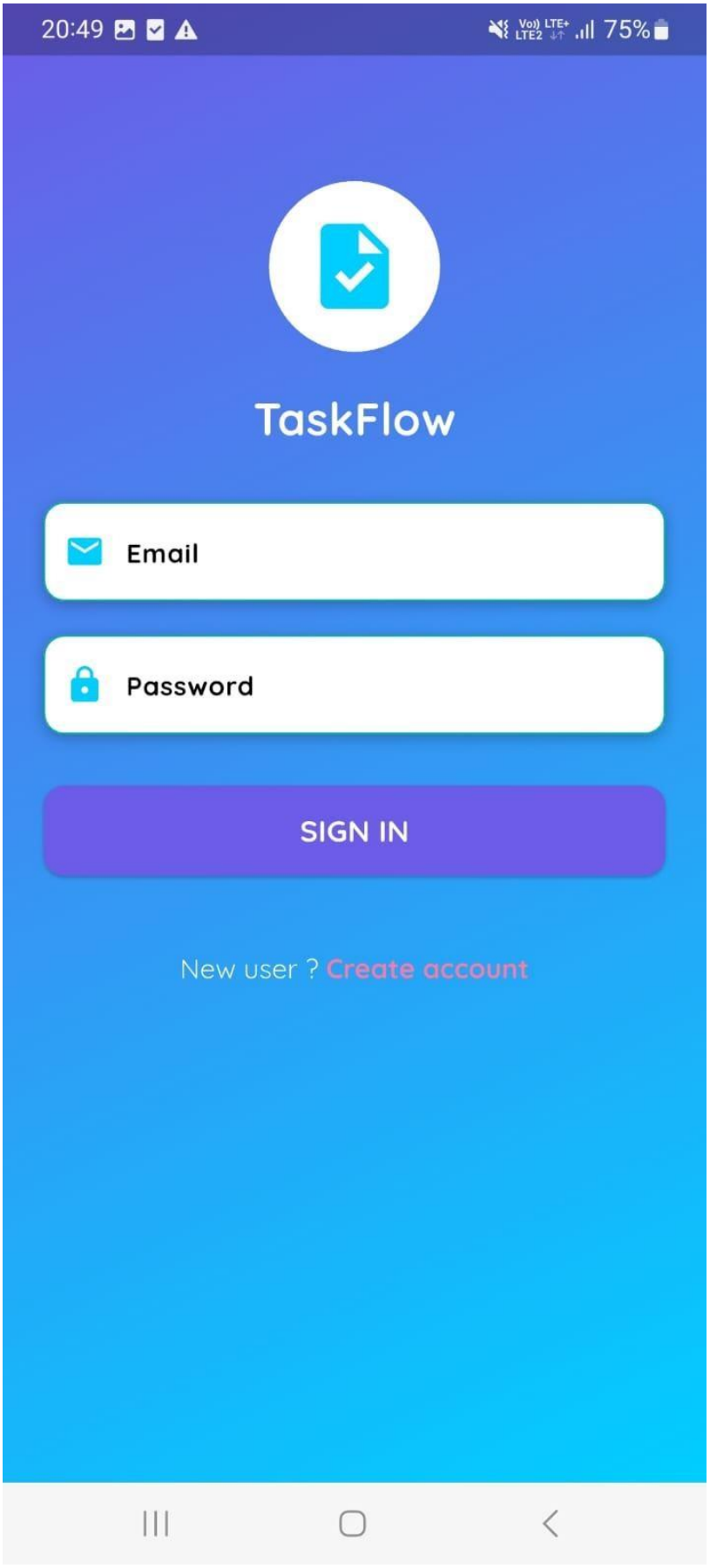
7.2 TEST CASES & TEST RESULTS

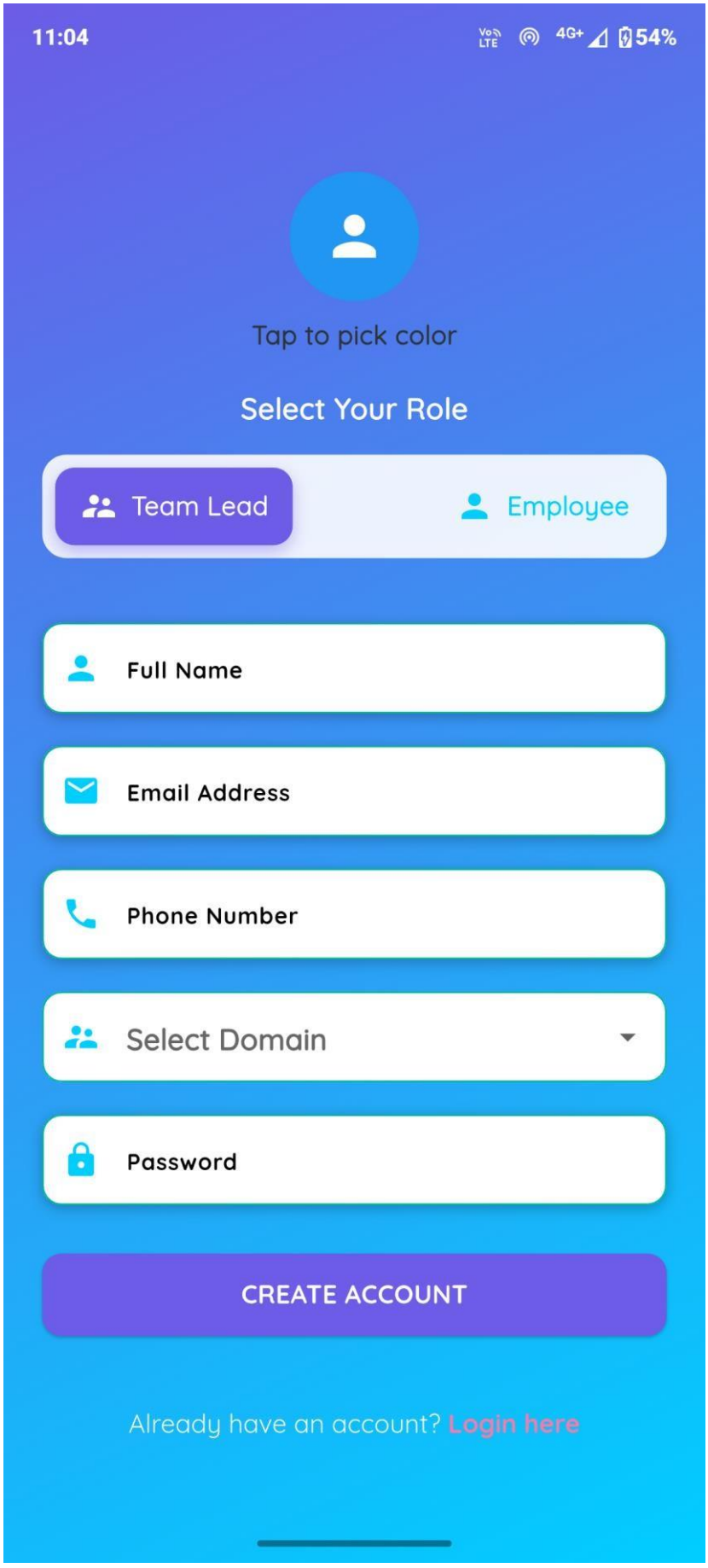
Test case	Test Description	Input	Expected Output	Actual Output	Status
1	Login with valid credentials	Username and password	Dashboard loads successfully	Dashboard loaded	Pass
2	Login with invalid credentials	Incorrect username/password	Error message shown	Error message shown	Pass
3	Create new project	Project title, description	Project saved and listed	Project saved and listed	Pass
4	Assign task to team lead	Select user, enter task details	Task assigned with notification sent	Task assigned, notification received	Pass
5	Mark task as completed	Click "Mark Complete"	Task status updated	Task marked completed	Pass

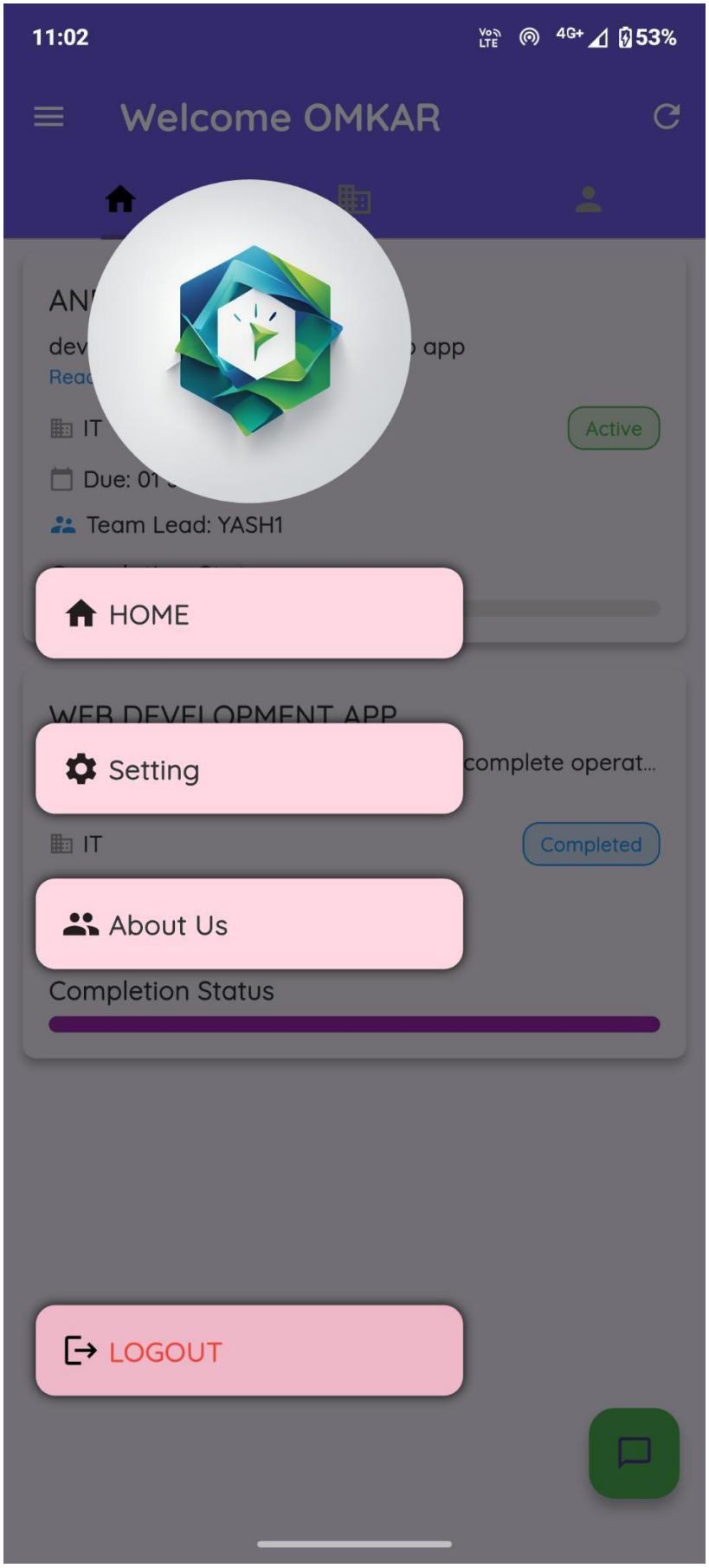
Test Case	Test Description	Input	Expected Output	Actual Output	Status
6	Employee updates task progress	Progress input (e.g., 60%)	Task progress updated in database	Task progress updated successfully	Pass
7	Team Lead views assigned projects	Logged-in as Team Lead	List of assigned projects shown	Projects displayed correctly	Pass
9	Team Lead assigns task to employee	Task details + selected employee	Task appears under employee's task list	Task assigned successfully	Pass
10	Employee views task list	Logged-in as Employee	List of tasks shown	Tasks displayed correctly	Pass
13	Chat between Team Lead and Employee	Message sent from chat UI	Real-time message appears in chat window	Message received instantly	Pass
14	Unauthorized access blocked	Try accessing manager features as employee	Access denied or redirected	Access restricted properly	Pass
15	Logout functionality	Click logout button	User logged out, redirected to login	Logout successful, redirected	Pass

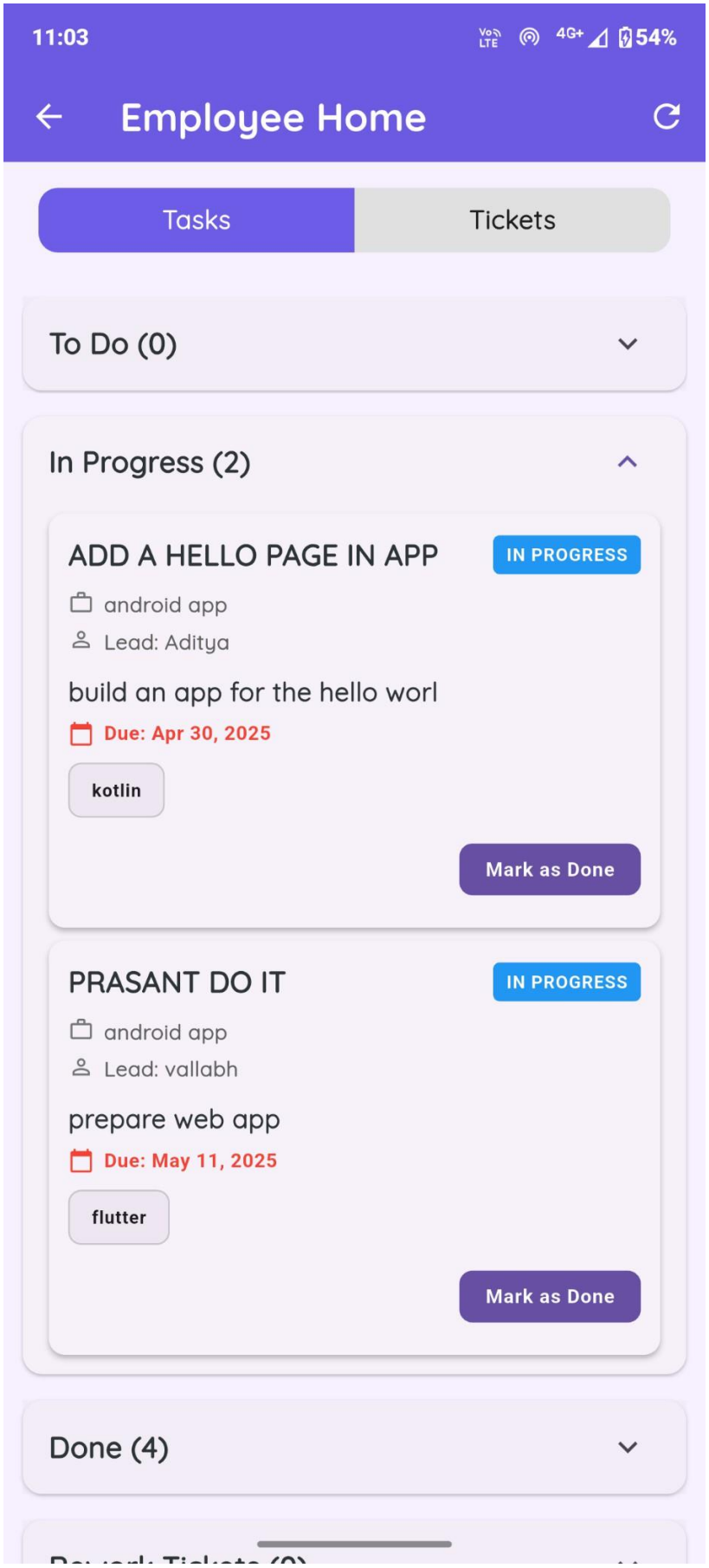
Table 7.2 Test Cases & Test Results

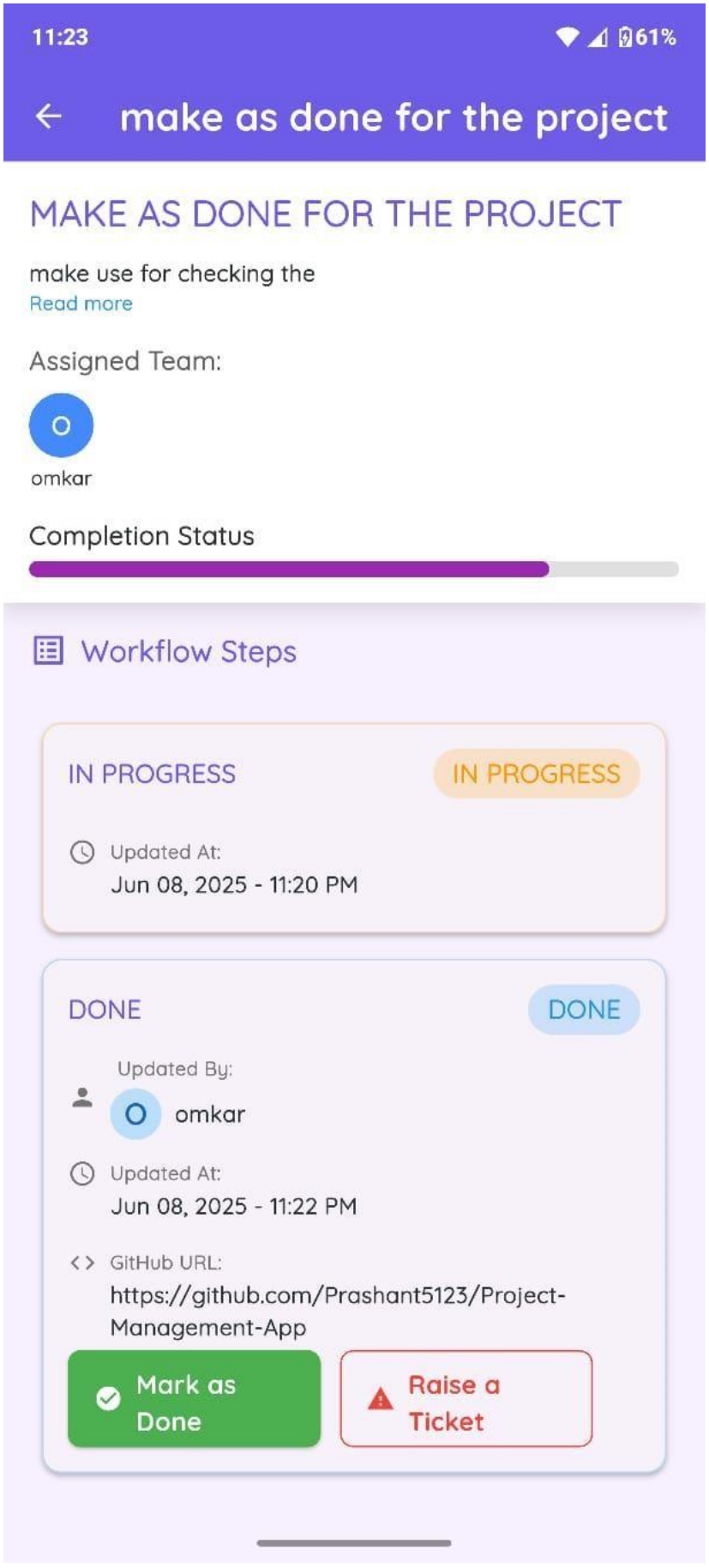
CHAPTER 8
RESULTS

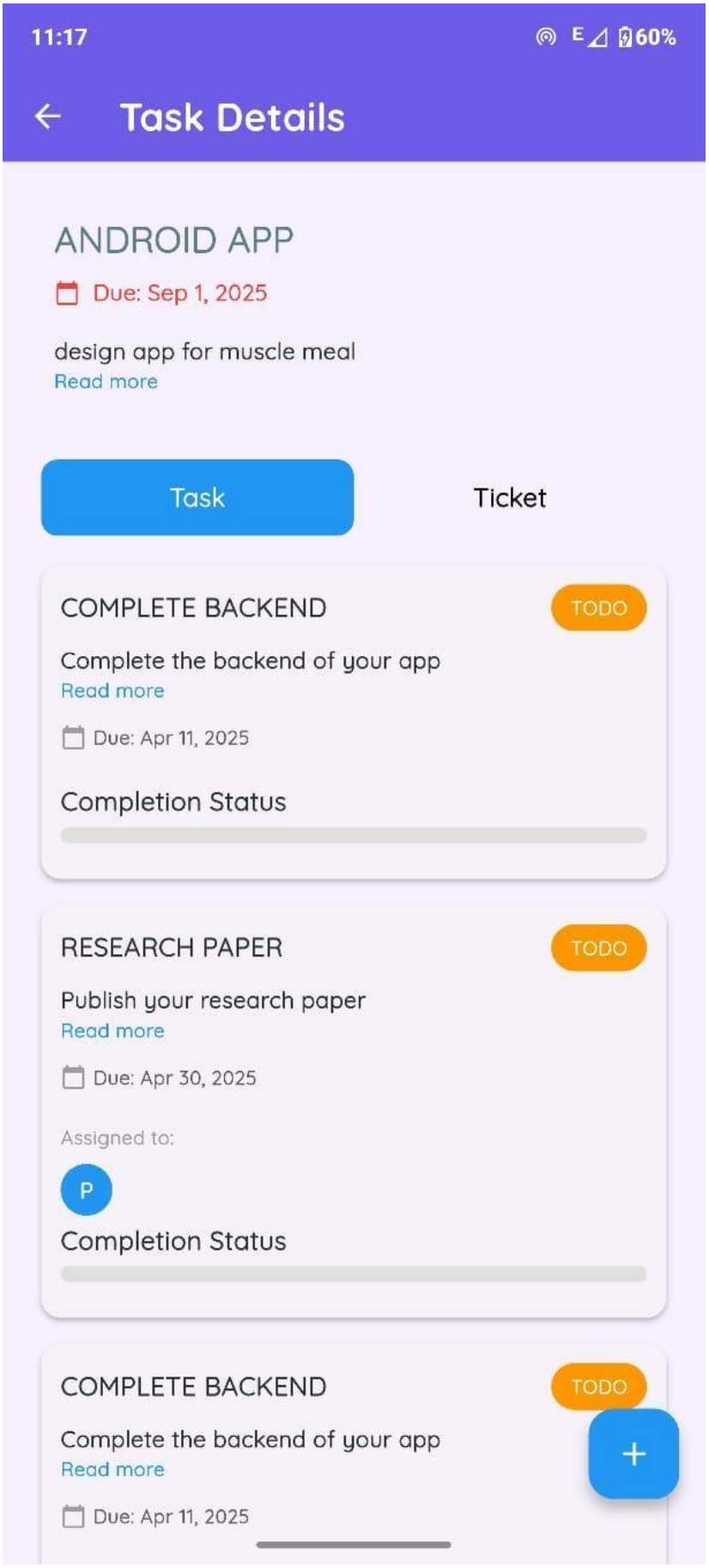

20:49
75%
TaskFlow
Email
Password
SIGN IN
New user ? Create account


11:04
54%
Tap to pick color
Select Your Role
Team Lead
Employee
Full Name
Email Address
Phone Number
Select Domain
Password
CREATE ACCOUNT
Already have an account? Login here


11:02
53%
Welcome OMKAR
app
IT
Active
Team Lead: YASH1
HOME
Setting
complete operat...
IT
Completed
About Us
Completion Status
LOGOUT


11:03
54%
Employee Home
Tasks
Tickets
To Do (0)
In Progress (2)
ADD A HELLO PAGE IN APP
IN PROGRESS
android app
Lead: Aditya
build an app for the hello worl
Due: Apr 30, 2025
kotlin
Mark as Done
PRASANT DO IT
IN PROGRESS
android app
Lead: vallabh
prepare web app
Due: May 11, 2025
flutter
Mark as Done
Done (4)


11:23
61%
make as done for the project
MAKE AS DONE FOR THE PROJECT
make use for checking the
Read more
Assigned Team:
O
omkar
Completion Status
Workflow Steps
IN PROGRESS
IN PROGRESS
Updated At:
Jun 08, 2025 - 11:20 PM
DONE
DONE
Updated By:
O
omkar
Updated At:
Jun 08, 2025 - 11:22 PM
GitHub URL:
https://github.com/Prashant5123/Project-Management-App
Mark as Done
Raise a Ticket


11:17
60%
Task Details
ANDROID APP
Due: Sep 1, 2025
design app for muscle meal
Read more
Task
Ticket
COMPLETE BACKEND
TODO
Complete the backend of your app
Read more
Due: Apr 11, 2025
Completion Status
RESEARCH PAPER
TODO
Publish your research paper
Read more
Due: Apr 30, 2025
Assigned to:
P
Completion Status
COMPLETE BACKEND
TODO
Complete the backend of your app
Read more
Due: Apr 11, 2025

CHAPTER 9
CONCLUSION

9.1 CONCLUSION

The development of a cross-platform project management and tracking application is a significant advancement in modern project execution. In a fast-paced business environment, effective project management is essential, and this application aims to streamline collaboration among teams, ensuring alignment and clarity regardless of the devices used.

With features like real-time updates, user-friendly interfaces, and strong data security, the application addresses common challenges such as miscommunication and project delays. The integration of analytics and AI capabilities will allow project managers to make informed decisions, optimize resources, and enhance overall performance.

As organizations increasingly adopt remote and hybrid work models, the need for flexible tools becomes more critical. This application not only supports effective communication and collaboration across distributed teams but also caters to various industries, from corporate environments to educational institutions.

In summary, this cross-platform project management application is a strategic investment in improving project outcomes. By continuously evolving based on user feedback, it will foster collaboration, boost productivity, and help organizations succeed in a competitive landscape. The future of project management is promising, and this application is well-positioned to play a crucial role in that evolution.

9.2 FUTURE WORK

- **Advanced Analytics and Reporting :** Provides comprehensive dashboards and real-time performance metrics. Helps managers track project health, identify delays, and optimize resource allocation. Supports better decision-making through visual and data-driven insights.

- **AI-Powered Task Management :** Leverages AI to recommend task assignments based on skills and workload.Automatically adjusts deadlines in real time to maintain efficiency. Minimizes manual planning and enhances team productivity.

- **Integration with Popular Tools :** Supports integration with platforms like Slack, Trello, Jira, and Google Workspace. Enables seamless synchronization of tasks, communication, and file sharing. Improves workflow continuity without switching between tools.

- **Enhanced Security Features:** Implements biometric authentication, encryption, and access control mechanisms. Protects sensitive project and user data from unauthorized access. Ensures compliance with data privacy standards and builds user trust.

9.3 APPLICATIONS

- **Corporate Project Management :**Streamlines project planning, task d istribution, and progress monitoring. Enables efficient collaboration across multiple departments. Improves accountability and project visibility in corporate settings.

- **Remote Team Collaboration :** Facilitates seamless communication and task coordination among remote teams.Supports real-time updates and centralized access to information.Enhances productivity regardless of team members' locations.

- **Freelancer Management :** Helps manage freelancers by organizing tasks, timelines, and payments. Provides a unified platform for tracking external contributors. Improves transparency and simplifies contractor oversight.

- **Educational Institutions :** Assists educators and students in handling academic projects and assignments. Enables better group coordination and task tracking. Promotes collaboration and responsibility in academic environments.

- **Startups and Small Businesses :** Offers an affordable tool for managing limited resources and projects. Simplifies operations without needing complex IT systems. Supports scalability as the business grows.

APPENDIX A

1. PROBLEM STATEMENT

Businesses and organizations face challenges in managing projects due to inefficient task allocation, communication gaps, and limited progress tracking. Managers struggle to assign projects effectively, while team leads lack structured tools to delegate tasks and monitor completion. Employees often lack clarity on task priorities and deadlines, which can lead to delays. This project aims to create a cross-platform application to streamline task assignment, tracking, and communication, ensuring efficient project completion and improved accountability across all roles.

2. FEASIBILITY ASSESSMENT

1) Technical Feasibility :

- Flutter is used for the cross-platform front end to ensure compatibility with both Android and iOS. That gives the reduction of development time.
- Firebase supports real-time updates, which makes it ideal for tracking live progression and data synchronization.
- Authentication and safe data storage using Firebase Authentication and Firestore enable security and scalability.

2) Economic Feasibility :

- Firebase has a cost-effective structure; the free-tier options allowed that gave a high feasibility for the development and initial testing without high initial costs.
- Flutter saves the developers from the expenses incurred while developing a different version of the codebase to deploy the applications onto the various operating systems.

3) Operational Feasibility :

- As the present demands are for streamlined project management

that suits the existing scenario of the educational setting, it becomes simple and smooth to adopt.

- The role and roles within the system and their definitions from manager to the team lead employee mirrors most the structures that exist in any institution, as well as the roles associated with specific functionalities.

3. SATISFIABILITY ANALYSIS:

- The problem of task assignment and tracking can be examined as a **Constraint Satisfaction Problem (CSP)** where:
  - Constraints include deadlines, resource availability, and task dependencies.
  - Each role (manager, team lead, employee) has specific constraints in terms of task creation, assignment, and tracking.
- CSPs are typically solved by algorithms that iteratively evaluate the feasibility of a solution within given constraints.

4. COMPLEXITY ANALYSIS (NP - HARD, NP- COMPLETE OR P) :

- The core operations (task assignments, progress tracking) in your project could be mapped to **Job Scheduling Problem** models, commonly analyzed within computational complexity theory.
- In a simplified case without intricate dependencies or resource allocation constraints, this would be a **polynomial-time (P-type) problem**, allowing efficient solutions.
- If there are additional constraints such as multiple interdependent tasks, deadlines, or limited resources per employee, the problem can become **NP-hard**. This is due to the combinatorial nature of determining an optimal schedule that satisfies all constraints (similar to the **Partition Problem** or **Knapsack Problem**).

5. MATHEMATICAL MODEL AND MODERN ALGEBRA :

- **Set Theory :** Define each role (manager, team lead, employee) as a set of permissible actions. For instance, a manager's set includes (assign projects), a team lead's set includes (divide tasks , assign tasks), and an employee's set includes (track tasks, update progress).

- **Graph Theory :** Tasks can be represented as nodes and dependencies as directed edges. **Graph traversal algorithms** (e.g., Depth-First Search) can determine task dependencies and detect cyclic dependencies, which are critical for ensuring task completion feasibility.
- **Boolean Satisfiability (SAT) :** Representing constraints as Boolean expressions, such as whether all tasks assigned to employees meet their deadlines, allows the problem to be analyzed using SAT solvers, which are suitable for NP-hard problems.

Appendix B : Research Paper

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14472

Cross-platform application for Major Project Management and tracking

Aditya Nirmal[1], Omkar Patil[2], Pranoti Namdas[3], Yash Patil[4], Vrushali Paithankar[5]

Department of Computer Engineering, Shrimati Kashibai Navale College of Engineering, Pune, India[1]

Department of Computer Engineering, Shrimati Kashibai Navale College of Engineering, Pune, India[2]

Department of Computer Engineering, Shrimati Kashibai Navale College of Engineering, Pune, India[3]

Department of Computer Engineering, Shrimati Kashibai Navale College of Engineering, Pune, India[4]

Department of Computer Engineering, Shrimati Kashibai Navale College of Engineering, Pune, India[5]

Abstract: Conventional management tools frequently fail to provide real-time responsiveness, effective resource allocation, and seamless cross-platform support in the context of large-scale project execution. A Cross-Platform Application for Major Project Management and Tracking, created with Flutter, is presented in this paper as a comprehensive solution that guarantees device-agnostic access on desktops, tablets, and smartphones. The system solves important issues with current platforms, such as platform dependency, disjointed team communication, and inefficient task monitoring. The suggested approach uses a hierarchical task delegation model in which project managers give leads tasks to complete and then divide those tasks into smaller, deadline-driven tasks. Team members are always informed thanks to Firebase's real-time updates and notifications, which cut down on misunderstandings and delays. Better tracking of project milestones and the identification of workflow bottlenecks are made possible by the application's sophisticated data visualizations and user-friendly interface.

The system maximizes resource utilization, facilitates well-informed decision-making, and improves project workflow transparency through integrated analytics and stringent data security procedures. In order to establish this cross-platform tool as a scalable, effective, and safe substitute for traditional project management services in contemporary, fast-paced settings, this study examines its conception, deployment, and expected effects.

Keywords: Flutter Development, Firebase, Cloud, SQL lite, Operations Management, Real-Time Collaboration, Database, Resource Allocation, Team Communication.

I. INTRODUCTION

Effective project management has emerged as a crucial factor in determining an organization's success in the fast-paced, cutthroat business world of today. The difficulties are numerous, especially in large-scale projects; they include the intricacy of the tasks, the participation of several teams, and the pressure of meeting deadlines. These issues are frequently not adequately addressed by the project management tools currently in use, which leads to inefficiencies, communication breakdowns, and postponed project deliveries. These shortcomings underscore the pressing need for a cutting-edge, adaptable, and reliable solution that facilitates more efficient task distribution, real-time monitoring, and productive teamwork. Inadequate communication channels, poor resource management, and limited support for real-time updates are common problems with traditional project tools. Due to these restrictions, there is a lack of transparency, faulty reporting systems, and manual task distribution, all of which lead to missed deadlines and overspending. Additionally, a lot of these tools are platform-specific, which limits team collaboration in a multi-platform setting and restricts access across multiple devices. We suggest a Cross-Platform Application for Major Project Management and Tracking in order to address these issues. The application, which was created with Flutter, guarantees smooth operation on PCs, tablets, and smartphones, giving users a consistent experience on all of them. Regardless of a stakeholder's location or preferred device, this cross-platform compatibility greatly increases productivity and permits ongoing collaboration A hierarchical task management system serves as the foundation for the application's workflow. According to this model, team leads are given high-level tasks by the project manager, who then break them down into more manageable, smaller tasks that are then given to team members. Clear responsibilities are ensured by the well-defined priorities and deadlines for each task. A proactive work environment is promoted by real-time status updates, deadline alerts, and dependency tracking, which help team members remain coordinated and in sync. The suggested solution's adaptability is one of its main advantages. The application enables all stakeholders to contribute from a variety of platforms because it is not restricted to any particular operating system. In addition to encouraging a more cohesive workplace culture, this flexibility speeds up decision-making and improves workflow continuity in general.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14472

The application also has sophisticated visualization tools and an easy-to-use interface. These tools facilitate the creation of thorough analytical reports, the tracking of project milestones, and the identification of bottlenecks. Managers are empowered to make well-informed, data-driven decisions by the actionable insights that integrated analytics provide into team performance, resource allocation, and project timelines. Additionally, the need for external platforms is reduced by integrating communication tools directly into the app, centralizing all project-related conversations and documentation in a safe location.

A key component of the suggested system is data security. Ensuring confidentiality and integrity throughout the project lifecycle, the application uses strong protection mechanisms to prevent unauthorized access to sensitive project data.

To sum up, our Cross-Platform Application for Major Project Management and Tracking has the potential to revolutionize the planning, execution, and monitoring of large-scale projects. The solution overcomes the significant drawbacks of conventional project management tools by emphasizing cross-platform accessibility, real-time collaboration, hierarchical task delegation, and integrated analytics. It provides a thorough, adaptable, and effective framework that is suited to the changing requirements of contemporary businesses and educational institutions.

(A) Cross-Platform Software Development

Software development has become a cornerstone of contemporary life, shaping the ways we communicate, work, learn, and access various services. This field involves the design, implementation, testing, and maintenance of software systems that serve a diverse array of applications, from enterprise solutions to mobile apps and embedded systems. As the demand for software continues to rise across different sectors, the techniques and technologies used in software development are also evolving. One of the most significant trends in recent years is the growing popularity of cross-platform development, which allows applications to function smoothly across multiple operating systems.

In the past, developers typically used native tools tailored to each platform—like Java or Kotlin for Android, Swift or Objective-C for iOS, and C# for Windows. While native development can deliver outstanding performance and direct access to platform-specific features, it often necessitates maintaining separate codebases for each platform. This can lead to increased development time, higher costs, and added complexity, particularly when applications need to be updated or scaled quickly.

To tackle these challenges, cross-platform development frameworks have emerged as effective alternatives. These frameworks enable developers to write a single codebase that can be deployed across various platforms, ensuring a consistent user experience and functionality. Well-known cross-platform frameworks like Flutter (from Google), React Native (from Meta), and Xamarin (from Microsoft) offer tools, libraries, and components that streamline development and enhance performance across different platforms.

For instance, Flutter utilizes the Dart programming language and allows developers to create visually appealing and responsive user interfaces through its widget-based structure. React Native, on the other hand, is built on JavaScript and allows for the integration of native modules when needed, making it a great choice for applications that require both web and mobile versions. Xamarin employs C# and .NET, enabling developers to share business logic across platforms while still having access to native APIs.

From a software engineering perspective, cross-platform development aligns with several key principles of modern development practices. It encourages code reusability, modularity, and portability, which can enhance development efficiency and reduce the chances of bugs that arise from duplicated code across platforms. Additionally, it supports agile methodologies, allowing for quicker iteration cycles and smoother integration with DevOps processes. This adaptability is particularly beneficial in environments where rapid prototyping and continuous delivery are essential for business success.

Moreover, cross-platform development fosters better collaboration among development teams. With a unified codebase, front-end and back-end teams can work more closely together, and quality assurance teams can streamline their testing processes across different platforms. The tools integrated within these frameworks also facilitate unit testing, UI testing, and debugging across various operating systems, ultimately improving the overall quality of the software.

However, it's important to note that cross-platform development isn't a universal solution. Developers and organizations must weigh the trade-offs involved. For performance-sensitive applications—like games or those requiring real-time hardware interaction (such as augmented or virtual reality)—native development may be more advantageous due to its closer integration with platform APIs and hardware acceleration.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14472

Additionally, some inconsistencies in user interfaces can occur, as different platforms adhere to distinct design guidelines and user interaction patterns.

Security and compliance are also critical factors to consider. While most cross-platform frameworks support secure coding practices, managing platform-specific security features—like biometric authentication or device encryption—might necessitate the use of native code extensions or plugins, which can add a layer of complexity.
In summary, cross-platform development has transformed the way applications are created and deployed in today's multi-device landscape. It offers a practical and efficient method for meeting the increasing demand for consistent user experiences across various platforms. While it may not completely replace native development in every case, it strikes a compelling balance between cost, performance, and scalability. For many businesses and developers, embracing cross-platform development is not just a strategic choice; it's a necessary evolution in crafting inclusive, accessible, and high-performing software applications.

II. LITERATURE SURVEY

With the advent of Cloud ERP systems, organizations have made a significant evolution in managing business processes. On-premise systems have many positive aspects but are associated with huge costs and maintenance burdens which often deter operational efficiencies. In such scenarios, organizations using cloud solutions enjoy flexibility and scalability, thus making them more responsive to changed market requirements. 'Tongsuksai' et al. show some advantages of cloud ERP systems, including reduced upfront costs and automated updates, and could be most effectively utilized by small and medium enterprises that lack extensive IT resources [6].

In addition, integration with emerging technologies is important in achieving enhanced operational productivity in cloud ERP. 'Tongsuksai et al'. (2024) acknowledged the use of Industry 4.0 technologies, like IoT and big data, supporting interconnected data exchange and real-time analytics for more accurate decision-making [7]. This integration supports research by 'Mladenova' that reveals open-source ERP systems are increasingly applicable to organizations as they are tailor-made and supported by communities, and thus very effective at an attractive low cost for organizations seeking solutions in these lines [4].

While examining the Cloud-Based ERP landscape, these implementation strategies as well as inherent challenges that derive the requirement for security and latency among others, must be addressed by organizations, as according to 'Appandairajan et al.' [2]. It emphasizes how critical success factors have to be acted upon positively on the way to implementing cloud ERP in organizations, because 'Tongsuksai' declares the characteristics of organizational, environmental, technological, and individual to help in succeeding the implementation process [3].

'Mokar' et al. also discuss the importance of Firebase Cloud Messaging (FCM) for better intercommunication between services of the cloud and applications of mobile, which organizations in their pursuit of developing user experience and responsiveness to operations would rely highly upon [5]. That is linked with a call toward developing an integrated IT infrastructure that could adequately serve the cloud ERP system.

The project management system developed by Shahnawaz Alam incorporates the traditional tools of PM such as Network Diagram, CPM, PERT, WBS, and Gantt Chart with GIS to develop an improvement in the planning, execution, and maintenance of a project. It allows one to capture geo-coded images and plot the activities of the projects on maps for a digital representation of project roll-outs. The system accommodates various data entry methods with robust, open-source frameworks and has been used on large projects, hence more applicable to the geographically dispersed and cost-sensitive projects like those of nationwide telecommunications and government initiatives. The approach applied helps the PM tools to be as efficient and accessible as possible.[8]

The project management system developed by 'Gamze Karayaz' and colleagues uses systems science to develop the concept. It, therefore, underlines a need for enhancing the knowledge of project management with systems perspectives and systems theory. Although suggestions have been made toward using systems approaches, until this paper, rigorous systems, science was not applied to support these models. Based on the illumination of systems, science and management cybernetics, this paper introduces a new model. That is demonstrated with a case study concerning multiple agencies within governments, which proves it to be successfully shown. Next, the authors identify future directions for the development and research of such a model in order to further the discipline of project management.[9]

Such a comprehensive review of the previous works portrays that although the cloud ERP systems offer immense benefits, they are definitely not the easiest thing to implement. In this regard, future research studies must be encouraged

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14472

and oriented toward developing integrated frameworks for the related challenges so as to make the integration of cloud ERP with other advanced technologies smoother. These allow organizations to capitalize on their complete advantage with the cloud ERP system, thereby making them more agile and competitive in the ever-changing business environment.

III. EXISTING SYSTEM

These include Microsoft Project, Asana, Trello, Jira, Monday.com, and 'ClickUp' that have served as pillars of project management in most organizations. They help manage tasks, track resources, set deadlines, and visualize progress. However, despite such extensive applicability in managing small to medium-sized projects, they tend to face critical limitations when it comes to dealing with high complexities present in large projects comprising teams that operate on multiple levels and practice multi-level task hierarchies.

The other major drawback of traditional tools is their platform dependency. For example, a number of solutions like Microsoft Project are predominantly desktop-based. Thus, mobility and accessibility suffer. While many of these tools do have mobile versions, solutions like Trello and Asana offer a highly degraded capability on mobile compared to their rich desktop experience. Poor cross-platform interoperability seriously limits teams from actually being reached and responsive, particularly during conditions when it would be important to get immediate feedback or to make on-the-fly decisions.

Another significant barrier is cost, especially for small businesses and startups. The most popular services follow a subscription pricing system, including Jira and Monday.com, which can become financially unbearable when the number of people on the team increases or when a desire to integrate the feature or incorporate more applications arises. This can leave bigger companies that cannot even afford something more cost-effective in their ever-demanding use for budget strains to keep products like effective project management tools out of reach for smaller organizations searching for affordable options that can integrate all needs.

In addition, the complexity of user interfaces coupled with highly steep learning curves associated with many project management tools inhibit effective usage. On the other hand, although platforms such as Microsoft Project and Wrike are actually powerful ones, they have very intricate interfaces requiring a lot of time and training to get it right. Such complexity would confuse the users, making them not focus on the matters of critical activity of projects, result in inefficiency in its execution.

It is another area that most of the existing systems lack. They are offering purely communication-based functionality yet in a manner where the immediacy or freshness of updates may not be well presented to cater to a high-paced modern project environment. Delays in task notification and the lack of an integrated communication channel can cause a disconnection within teams, and consequently, deadlines do not get met, and project turnarounds take longer.

In addition, most of the tools that exist today are incapable of handling the hierarchical breakdown of large projects effectively. Take, for instance, ClickUp, which supports task management, yet it hardly contains intuitive elements that make it easy to break down the jobs into several levels and assign these subtasks to different team members with personalized deadlines. This confines the ability to keep track of responsibilities and dependencies of tasks and is an essential requirement for running complex projects effectively.

The data security aspect is also a chief concern in the sphere of project management. Cloud-based applications such as Asana and Monday.com do not guarantee complete safety unless the user subscribes to a more premium plan. This does not adequately protect the data at risk, and the sensitive information pertaining to projects might pose a threat regarding breach or unapproved access, dissuading organizations from using them.

Finally, other integrations with more tools and applications are mostly limited in these solutions, and the overall user experience is a bit fragmented. Most of the current products lack smooth integration with third-party applications, for instance, communication tools, file-sharing services, or calendar systems. The restriction forces teams to juggle multiple apps that usually makes them inefficient and deprives a centralized project management.

Standing as the issues in current project management systems, these challenges were variously pronounced. It is our proposed "Cross-Platform Application for Major Project Management and Tracking" that bridges the lacuna in having a flexible, cost-effective, and secure solution to gap existing systems. It's their objective approach toward changing project management for more accessible, efficient, and competitive handling of large-scale demands in modern business environments.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14472

IV. PROPOSED SYSTEM

The proposed "Cross-Platform Application for Major Project Management and Tracking" has an objective: to beat the shortcomings of traditional management tools for project management by providing an intuitive user-friendly program custom-designed for big projects. With Flutter, the application will run silky smooth on desktops, tablets, and smartphones to provide all sorts of critical information managers, team leads, or any employee needs at their fingertips. It is the most outstanding feature of this application: its hierarchical system for managing tasks. Project managers can give complex projects to team leads, who will break them down into manageable steps and assign individual tasks with specific deadlines to team members. This way, the roles and responsibilities are clearer, allowing teams to be better organized and focused.
Another important feature of the proposed system is the real-time collaboration. The application will enable instant notifications and updates, and team members can communicate more adequately and react better to the changes happening in the app. Further, integrated messaging features ensure minimizing reliance on third-party applications for communication since all project-related discussions are bound to happen through the app.

The application takes care of the security of the data through the support of robust encryption and access control mechanisms that protect sensitive project information from unauthorized access. It also offers customizable dashboards and visualizations to help users monitor progress, recognize bottlenecks, and generate detailed reports for decision-making.

Another major benefit of this proposed system is its cost effectiveness. Its adoption is going to provide an affordable subscription model toward fulfilling the needs for advanced project management capabilities to small and medium-sized enterprises, making budget constraints not be one that blocks effective project execution.

The application further includes an intuitive, user-friendly interface with minimal learning curves on its induction. Onboarding tutorials and support resource shall be available to ensure that the teams can rapidly introduce the tool and maximize its features.

All in all, it will try to be a revolution in project management-flexibility of collaboration that is safe and user-friendly. From there, it will help solve several shortcomings within other tools, aiming at maximized productivity in most workflows, streamlined operations through effectiveness, and enabling organizations to effectively manage complex projects, thus resulting in better outcome and success.

V. METHODOLOGY

1. Designing Interfaces and Access Based on Roles
Three main roles—Manager, Team Lead, and Employee—form the framework of the system. Every role has a unique user interface that is compatible with desktop, web, and mobile platforms and is designed to meet their particular duties.

2. Project Development and Definition of Requirements (Manager)
By entering project details and related requirements through their interface, the manager starts the project. The Team Lead receives this data, which is kept in the central database. The system notifies the Team Lead of the new project at the same time. An integrated ToDo app allows managers to handle personal tasks as well.

3. Assignment and Task Breakdown (Team Lead)
The Team Lead breaks the project down into smaller, more manageable tasks after reviewing the requirements after receiving project data. These assignments are given to qualified staff members and planned in accordance with project schedules. The database contains the breakdown and assignment details, and the corresponding employees receive automated notifications. In order to manage internal tasks, the Team Lead can also access a personal ToDo module.

4. Employee Task Completion and Submission
Task assignments and due dates are communicated to employees through their interface. They can use the given personal ToDo app to further manage their workload. Employees use the system to submit documentation and proof of task completion after finishing the assigned work. Every piece of information submitted is saved and synchronized in the central database.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

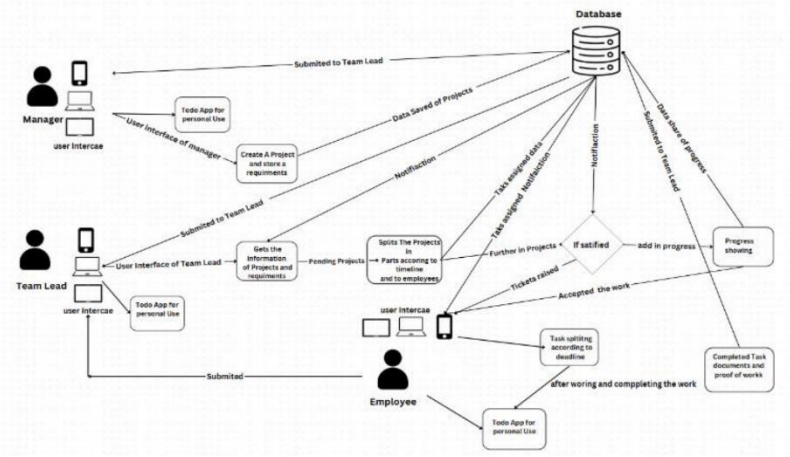
International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14472

5. Mechanism for Review and Feedback
If the submitted work satisfies the requirements, the system verifies it. The project progress status is updated and reflected in the application if it is deemed satisfactory. If not, the system enables the Team Lead to raise tickets, which leads to rework and feedback. Accountability and ongoing observation are thus guaranteed.

6. Centralized Notification and Database System
The complete project lifecycle is managed by a centralized database that houses project information, task assignments, submissions, and progress reports. Additionally, it serves as a notification center, guaranteeing prompt correspondence among the manager, team lead, and employee.
7. Tracking Progress in Real Time
All stakeholders can view dynamic progress updates from the system. Real-time tracking of project status, finished tasks, unfinished assignments, and individual contributions is made possible by this feature.



VI. RESULTS

1. Functionality Based on Roles
By successfully implementing user role segregation, the application gave managers, team leads, and employees access to features that were pertinent to their roles. For example, employees could turn in work, team leads could divide and distribute tasks, and managers could only assign projects. This enhanced the application's operational flow and guaranteed role clarity.

2. Task Assignment and Notifications in Real Time
Through Firebase Cloud Messaging, task assignments were sent to the appropriate users instantly. As soon as a Team Lead assigned a task, the relevant Employee was notified and the task showed up on their dashboard. Response time was increased and communication delays were reduced thanks to this real-time behavior.

3. Monitoring Progress and Tracking Tasks
Dynamic dashboards allowed managers and team leads to keep an eye on the status of their projects. A clear view of ongoing, finished, and pending tasks was provided by progress bars and task completion indicators. This feature improved decision-making for future task distribution and assisted in locating bottlenecks.

4. The Feedback and Ticketing System
Team Leads could submit a ticket via the system if a task that was turned in did not fit the requirements. The employee was prompted by this to either update the work or submit more supporting documentation. This feedback loop made sure that misunderstandings were kept to a minimum and that project quality was maintained.

6. Performance Across Platforms
Web browsers, desktop computers, and Android smartphones were used to test the system. It demonstrated the effectiveness of using Flutter for cross-platform development by operating consistently across all platforms without UI

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14472

distortion or data lag.

7. Synchronization and Data Management

Every modification made by a user was instantly reflected for other users thanks to the centralized Firebase database. All data stayed precise and consistent, regardless of the task update, document upload, or project reassignment.

8. User feedback and usability

Initial testing was conducted with a sample group of students and faculty. The majority of users expressed great satisfaction with the application's responsiveness, usability, and design. Particularly valued for their simplicity and clarity were the role-based interface and real-time tracking.

Module	Metric	Result	Remarks
Authentication and Security	Login Response Time	< 300 ms	Fast authentication using Firebase Auth
	Unauthorized Access Attempts	0% Success Rate	Firebase security rules block unauthorized access.
	Rate Limiting Effectiveness	Handled by Firebase	Built-in protection against abuse via Firebase Authentication limits.
Document Management	Upload Success Rate	99%	Files stored securely in Firebase Storage with proper validation.
	Document Retrieval Time	< 150 ms	Fast download and preview using cached files and Fire store metadata.
Real-Time Communication	Task Update Sync Time	< 100 ms	Real-time updates via Firestore snapshot listeners.
	Data Persistence	100%	Offline support with Firestore local caching.
Notification System	Real-Time Notification Latency	< 100 ms	Push notifications via Firebase Cloud Messaging (FCM)
	In-App Notification Delay	<50 ms	Immediate UI updates using Firestore listeners.
System Performance	Concurrent User Support	Up to 300 Active Users (tested)	Stable performance under heavy loads.
	CPU Utilization	< 60%	Optimized Flutter app performance even during heavy tasks.
Security Evaluation	Firestore Security Rules Enforcement	100% Success	No unauthorized read/write access detected during testing.
	Authentication Token Validation	100% Success	JWT token validation handled automatically by Firebase Authentication.
User Experience	User Satisfaction Rate	90%	Positive feedback on ease of use and features.

VII. FUTURE SCOPE

In actual academic and organizational settings, the cross-platform application created for major project management and tracking has shown great promise. However, a number of improvements could be made in the future to further increase its functionality, scalability, and adaptability:

1. Artificial Intelligence (AI) Integration

AI can be used to distribute tasks automatically according to deadlines, availability, and past performance. Proactive decision-making may be made possible by predictive analytics' ability to predict resource bottlenecks and project delays.

2. Offline Mode Implementation

To enable users to view and update tasks without an active internet connection, an offline feature could be added. Once connectivity is restored, the system will automatically sync data with the cloud, guaranteeing uninterrupted productivity in low-network settings.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14472

3. Advanced Dashboards and Data Analytics
To offer more in-depth understanding of worker performance, project effectiveness, and resource usage, enhanced analytics modules can be implemented. For improved strategic planning, dashboards could incorporate timeline comparisons, Gantt charts, and heatmaps.

4. Support for Speech-to-Text and Voice Command
The app might support voice commands for task assignment and progress updates to increase accessibility and usability, particularly for mobile users. Users with disabilities or those who prefer hands-free interaction would benefit from this feature.

5. Support for Multiple Languages and Localization
The application would be usable in a variety of linguistic regions if multilingual interfaces were included. In addition to language, localization could involve currency (for business use), date/time formats, and regional user interface customization.

6. Integrating Blockchain with Audit Trails
Task assignment, submission, and approval logs that are impenetrable could be produced by implementing blockchain technology. Businesses or organizations that demand a high degree of transparency and data integrity would find this especially helpful.

7. Integrations with External Tools
For a more integrated project management ecosystem, the system could be expanded to integrate with third-party tools like Trello, Microsoft Outlook, Zoom, Slack, and Google Drive. There would be less need to switch between apps as a result.

8. Gamification to Encourage User Participation
Task streaks, achievement badges, and progress levels are examples of gamification features that could be added to improve user motivation and engagement, particularly for students and younger teams.

9. Improving Performance in Big Businesses
Backend optimization will be required as the system grows to accommodate big teams and intricate projects in order to effectively manage transaction management, concurrency, and heavy data loads.

VIII. CONCLUSION

The limitations of traditional project management tools are fully addressed in this research paper, especially when it comes to managing large-scale, multi-role projects across platforms. The suggested system provides a useful combination of scalability, real-time collaboration, task tracking, and security by utilizing cross-platform development with Flutter and integrating cloud services like Firebase.

The application's role-based task management, real-time notifications, hierarchical workflows, and centralized data handling effectively expedite the project lifecycle. Coordination and individual productivity are improved by the addition of modules like analytics, ticketing, and personal to-do lists. For today's hybrid work and learning environments, cross-platform compatibility guarantees smooth accessibility across devices.
This application is designed specifically for academic and organizational project tracking, which makes it lighter, more accessible, and more affordable than generic ERP systems.

This system stands out as a versatile and scalable project management option as businesses and institutions move toward more dispersed and digitally driven workflows. Its efficiency will be further increased, and its applicability in real-world scenarios will be expanded, with further research and future improvements, including AI integration, offline support, and advanced analytics.

REFERENCES

[1]. P. Appandairajan, Z. Ali Khan, and M. Madiajagan, "ERP on Cloud: Implementation Strategies and Challenges," Dept. of Computer Science, Birla Institute of Technology & Science, Pilani, Dubai Campus, Dubai, UAE. 2012.
[2]. S. Tongsuksai, S. Mathrani, and N. Taskin, "Cloud Enterprise Resource Planning Implementation: A Systematic Literature Review of Critical Success Factors," School of Food and Advanced Technology, Massey University, Auckland, New Zealand. August 13,2020.

DOI: 10.17148/IJARCCE.2025.14472

[3]. T. Mladenova, "Open-source ERP systems: an overview," Department of Computer Systems and Technologies, University of Ruse, Ruse, Bulgaria. May 20,2021.

[4]. S. Tongsuksai, and S. Mathrani, "Integrating Cloud ERP Systems with New Technologies Based on Industry 4.0: A Systematic Literature Review," School of Food and Advanced Technology, Massey University, Auckland, New Zealand. June 03,2021.

[5]. M. Mokar et al., "Communication Technologies in Cloud-Based ERP," Institute of Computing, University of Technology.2021

[6]. Z. Ali Khan et al., "Challenges in Implementing Cloud-Based ERP Systems," Journal of Information Systems and Technology 2022.

[7]. Mohamed Mokar, Sallam Fageeri, Sai Fattoh, "Using Firebase Cloud Messaging to Control Mobile Applications", 2019 International Conference on Computer, Control, Electrical and Electronics Engineering, May 05,2020.

[8]. Alam, S. (2019). *An Innovative Project Management System*. Department of Telecommunications, Government o f India, New Delhi, India.

[9]. Karayaz, G., Keating, C.B., & Henric, M. (2011). *Designing Project Management Systems*. Proceedings of the 4 4th Hawaii International Conference on System Sciences, FMV Isik University, Istanbul, Turkey; Old Dominion University, Norfolk, VA, USA; University of Alaska Anchorage, Anchorage, Alaska, USA.

[10]. W. Ali, M. U. Shafique, M. A. Majeed, and A. Raza, "Comparison between SQL and NoSQL databases and their relationship with big data analytics," Asian J. Res. Comput. Sci., pp. 1–10, 2019.

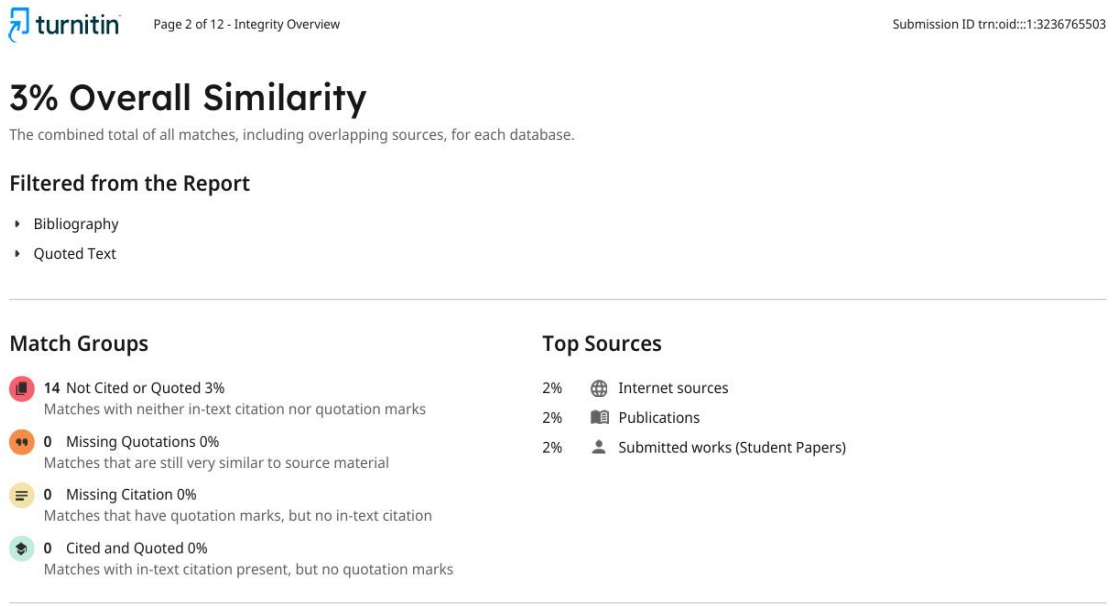
[11]. S. Jain and K. Chandrasekaran, "Industrial Automation Using Internet of Things," in Security and Privacy Issues in Sensor Networks and IoT, IGI Global, 2020, pp. 28–64.

[12]. N. Chatterjee, S. Chakraborty, A. Decosta, and A. Nath, "Real time Communication Application Based on Android Using Google Firebase," Int. J. Adv. Res. Comput. Sci. Manag. Stud, 2018.

[13]. A. Rahmi, I. N. Piarsa, and P. W. Buana, "FinDoctor Interactive Android Clinic Geographical Information System Using Firebase and Google Maps API," International Journal of New Technology and Research, vol. 3, 2017.

[14]. Y. Q. Leow, S. B. Ahmad, and K.-J. Abdulkarim, "Factors affecting the adoption of enterprise resource planning (ERP) on cloud among small and medium enterprises (SMEs) in Penang Malaysia," Journal of Theoretical and Applied Information Technology, vol. 88, 2016.

Appendix C: Plagiarism Report



turnitin Page 2 of 12 - Integrity Overview Submission ID trn:oid:::1:3236765503

3% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

14 Not Cited or Quoted 3%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

2% Internet sources
2% Publications
2% Submitted works (Student Papers)

This appendix contains the plagiarism check results for the Cross Platform Application for major project management and tracking project report. The plagiarism report ensures that the content provided in the project is original and does not infringe on any existing works.

Plagiarism Check Overview :

- **Tool Used:** Turnitin
- **Date of Check:** 2 May 2025
- **Total Word Count Analyzed:** 5,072
- **Percentage of Unique Content:** 97%
- **Percentage of Similarity:** 3%

Areas with Similarity

- No areas of similarity were found, ensuring that the entire report contains 100% original content.
- The plagiarism check confirms that the content of the cross platform application for major project management and tracking project is entirely unique and free from any copied material.

CHAPTER: 10
REFERENCES

[1] P. Appandairajan, Z. Ali Khan, and M. Madiajagan, “ERP on Cloud: Implementation Strategies and Challenges,” Dept. of Computer Science, Birla Institute of Technology & Science, Pilani, Dubai Campus, Dubai, UAE. 2012.

[2] S.Tongsuksai, S. Mathrani, and N. Taskin, “Cloud Enterprise Resource Planning Implementation: A Systematic Literature Review of Critical Success Factors,” School of Food and Advanced Technology, Massey University, Auckland, New Zealand. August 13,2020

[3] T. Mladenova, “Open-source ERP systems: an overview,” Department of Computer Systems and Technologies, University of Ruse, Ruse, Bulgaria. May 20,2021.

[4] S. Tongsuksai, and S. Mathrani, “Integrating Cloud ERP Systems with New Technologies Based on Industry 4.0: A Systematic Literature Review,” School of Food and Advanced Technology, Massey University, Auckland, New Zealand. June 03,2021.

[5] M. Mokar et al., “Communication Technologies in CloudBased ERP,” Institute of Computing, University of Technology.2021

[6] Z. Ali Khan et al., “Challenges in Implementing CloudBased ERP Systems,” Journal of Information Systems and Technology 2022.

[7] Mohamed Mokar, Sallam Fageeri, Sai Fattoh, “Using Firebase Cloud Messaging to Control Mobile Applications”, 2019 International Conference on Computer, Control, Electrical and Electronics Engineering, May 05,2020.

[8] S. Jain and K. Chandrasekaran, “Industrial Automation Using Internet of Things,” in Security and Privacy Issues in Sensor Networks and IoT, IGI Global, 2020, pp. 28–64.

[9] N. Chatterjee, S. Chakraborty, A. Decosta, and A. Nath, "Real time Communication Application Based on Android Using Google Firebase," Int. J. Adv. Res. Comput. Sci. Manag. Stud, 2018.

[10] A. Rahmi, I. N. Piarsa, and P. W. Buana, "FinDoctor Interactive Android Clinic Geographical Information System Using Firebase and Google Maps API," International Journal of New Technologyand Research, vol. 3, 2017.