KING SAUD UNIVERSITY
COLLEGE OF COMPUTER AND INFORMATION SCIENCES
DEPARTMENT OF SOFTWARE ENGINEERING

SWE 455 Software Maintenance and Evolution
Spring 2025

Project Description – Deliverable 1 & 2

I. Rawan AlAbdulrahman

Name	ID
Shouq Alqureshi	443200473
Miriam Almogren	443200568
Batool Alsugaih	443200073
Waad Almutairi	443200547
Riyam Alsuhaibani	443201116
Amal Alharbi	442202306

KING SAUD UNIVERSITY
COLLEGE OF COMPUTER AND INFORMATION SCIENCES
DEPARTMENT OF SOFTWARE ENGINEERING

Table of Contents

Introduction

Project Overview

In this Software Maintenance course, we will build upon a past project developed in the SWE 444 - Software Construction Laboratory course. The project, Achiva, is a goal-tracking and social productivity platform designed to help individuals set, track, and achieve their personal and professional goals while leveraging community support for motivation and accountability. This course focuses on the critical aspect of software maintenance, which ensures the long-term functionality, adaptability, and scalability of software systems.

The primary goal of this course is to develop and refine skills in software maintenance, a vital component of software engineering that extends the lifecycle of software systems. Maintenance activities include correcting defects, enhancing features, optimizing performance, and ensuring compatibility with evolving technologies and user needs. By working on Achiva, we will gain hands-on experience in maintaining a complex system, addressing real-world challenges, and implementing updates to improve its functionality and user experience.

The importance of software maintenance cannot be overstated. In today's fast-paced technological landscape, software systems must evolve to meet changing user demands, integrate with new technologies, and remain secure and efficient over time. Through this course, we will learn how to:

1. Identify and resolve defects to ensure system reliability.
2. Enhance existing features to improve user satisfaction and engagement.
3. Optimize performance to ensure the system remains efficient and scalable.
4. Adapt the system to new requirements, platforms, or technologies.

By applying these maintenance practices to Achiva, we will not only improve the application but also develop a deeper understanding of the challenges and strategies involved in maintaining software systems. This experience will prepare us to contribute effectively to the long-term success of software projects in our professional careers.

In summary, this course emphasizes the critical role of software maintenance in ensuring the sustainability and adaptability of software systems. Through hands-on work on Achiva, we will develop the skills and knowledge necessary to maintain and enhance software systems, ensuring they continue to meet user needs and remain relevant in a dynamic technological environment.

جامعة الملك سعود
King Saud University
كلية علوم الحاسب والمعلومات
قسم هندسة البرمجيات

KING SAUD UNIVERSITY
COLLEGE OF COMPUTER AND INFORMATION SCIENCES
DEPARTMENT OF SOFTWARE ENGINEERING

Software Overview

Staying productive and achieving long-term goals can be challenging, especially without proper structure, motivation, or accountability. Many people set ambitious goals but struggle to follow through because traditional productivity tools lack **personalized guidance, real-time collaboration, and AI-driven support**. Standard to-do lists and project management apps often fall short, leading to inefficiencies and disengagement The system not only simplifies goal tracking but also introduces social elements that foster motivation and accountability .Through features like progress sharing ,achievement comparisons ,and community support ,users stay engaged and motivated throughout their journey .Additionally ,Achiva's ranking dashboard and reminder system ensure users maintain momentum and celebrate their progress along the way..

Who is Achiva For?

Achiva is designed for individuals and teams who need a smarter, more intuitive approach to goal-setting and productivity.

• **Individuals & Professionals** – Whether you're a student, entrepreneur, or working professional, Achiva helps you define, track, and accomplish personal and career goals with AI-driven insights.

• **Teams & Organizations** – Small and medium-sized teams can collaborate seamlessly, track shared goals, and stay aligned with real-time updates.

• **Coaches & Mentors** – Ideal for professionals guiding clients or students, Achiva provides a structured system for setting, monitoring, and achieving milestones.

Key Objectives & Features of Achiva :

• **Boost Productivity & Goal Success** – AI-powered recommendations help users stay on track, breaking large objectives into manageable steps.

• **Enhance Collaboration** – Teams can work efficiently with shared workspaces, task synchronization, and real-time progress tracking.

• **Keep Users Engaged & Motivated** – Visual analytics, reminders, and motivational insights ensure consistent progress and commitment.

Screenshots of main screens:

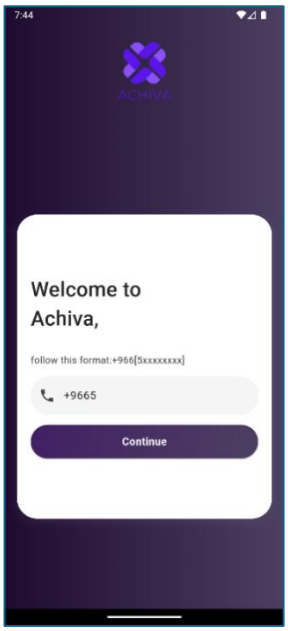


Figure 2 Sign Up Page

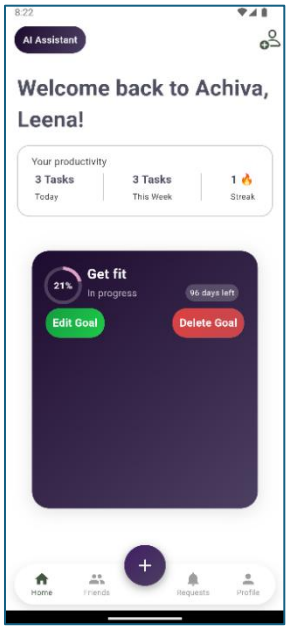


Figure 3 Home Page

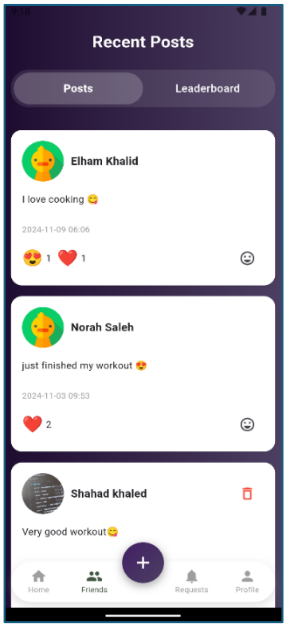


Figure 3 Friends Posts Page

Figure 1 Productivity Leaderboard

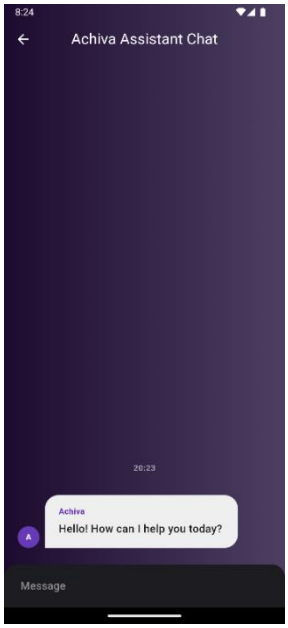


Figure 4 Chat with AI

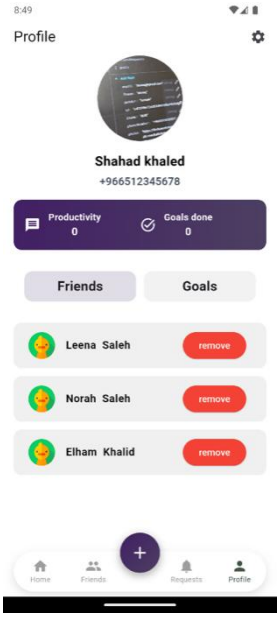


Figure 6 Profile Page

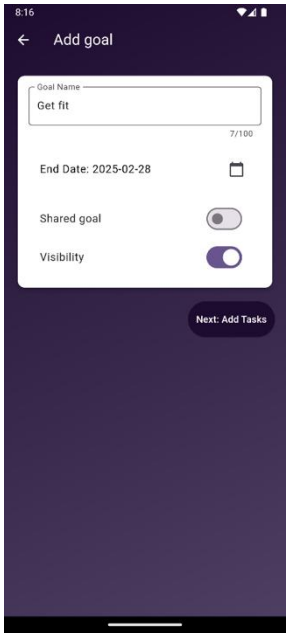


Figure 5 Add Goal Screen

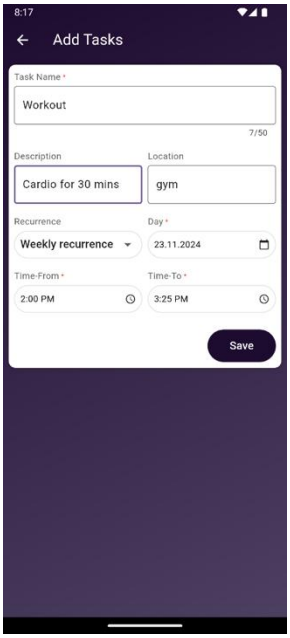


Figure 6 Add Task Screen

جامعة الملك سعود
King Saud University
كلية علوم الحاسب والمعلومات
قسم هندسة البرمجيات

KING SAUD UNIVERSITY
COLLEGE OF COMPUTER AND INFORMATION SCIENCES
DEPARTMENT OF SOFTWARE ENGINEERING

Program comprehension

#	Existing knowledge	Explanation	New knowledge
1	Product backlog (System functional features)	The system features and their relationship but having them only written takes longer time for the reader to comprehend the system than visualizing the relation in a use case diagram and will play a crucial role in the maintenance process.	System Use case Diagram in Figure 9.
2	System Architecture	Extracting the deployment diagram from the MVC architecture provides insights into component distribution, improving resource allocation, integration, scalability, security, and performance. This understanding helps optimize system maintenance and evolution.	System Deployment Diagram in Figure 11.
3	Programing language	Our application is built using Dart, a flexible programming language that works well with the MVC (Model-View-Controller) design. This structure helps organize the code better by separating the data, the user interface, and the control logic. This makes it easier to develop and maintain the app.	Learned how the MVC architecture separates the data model, user interface, and control logic, making the code easier to organize and maintain.
4	Firebase Database Structure	Existing data model stores basic goal and progress information	Required extensions to database schema for storing historical trends and analytics data
5	Goal and Task Data Model	The basic structure of goals and tasks was known, but the relationships between these entities and their serialization methods were not fully documented.	Discovered that goals and tasks use a composite pattern, where goals contain collections of tasks, allowing for nested hierarchical representation. This architecture supports the planned conversion from individual to shared goals but requires additional relationship mapping for assigned collaborators.

We extract the deployment from the component diagram so we can allocate each component to its specific device.

how is this a new knowledge? you already have MVC as an architecture style

what does this mean? what extentions?

where is the database scheema?

why was this identified as necessary? Is the current design inefficient? if yes, how?

Missing a sequence diagram or flowchart to illustrate how Achiva processes user actions (e.g., task completion, goal updates).

6	Non-functional aspects	By leveraging our current understanding of the system's non-functional aspects, we can clearly define the non-functional requirements, providing stakeholders with a comprehensive overview of the system.	Non-functional requirements : (Usability) :NFR1: The user shall be able to get familiar with the system's basic features in under 3 minutes.NFR2: The system shall provide clear and informative feedback when an invalid action is performed.NFR3: Buttons and interactive elements shall use distinguishable and intuitive colors.(Security) :NFR4: All sensitive user data shall be encrypted by default in the database to ensure confidentiality.(Reliability) :NFR5: The system shall be operational and accessible 99.9% of the time.NFR6: The system shall accurately fetch the user's location within 1 second, 100% of the time. (Performance) :NFR7: Task and goal creation, including their details, should be processed and displayed within 3 seconds.

Some (4 and 5) explanations are vague or too high-level
Some (3 and 4) new knowledges need specific insights on architecture, data processing, security, and performance... etc.

Type text here

Use Case Diagram

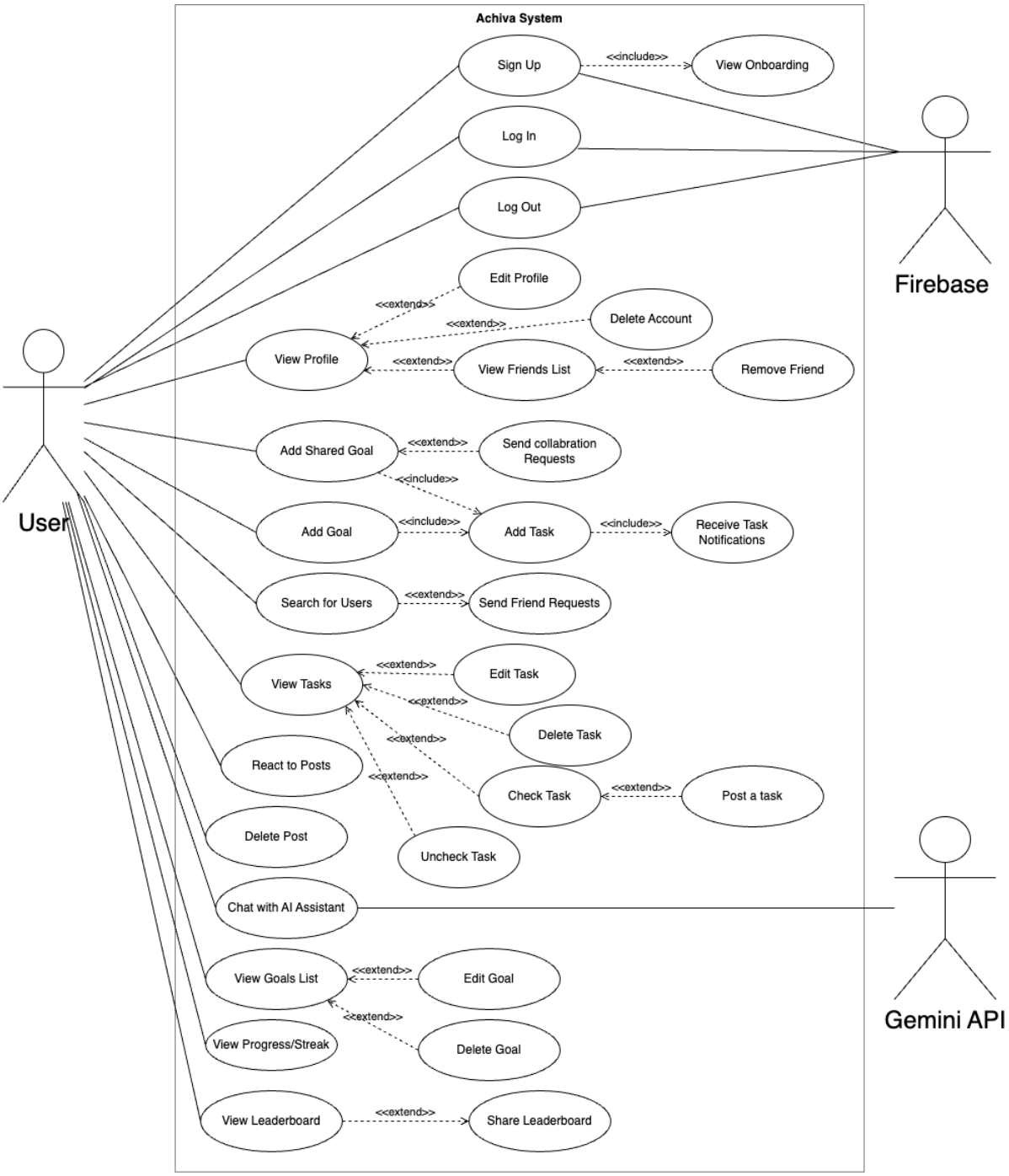
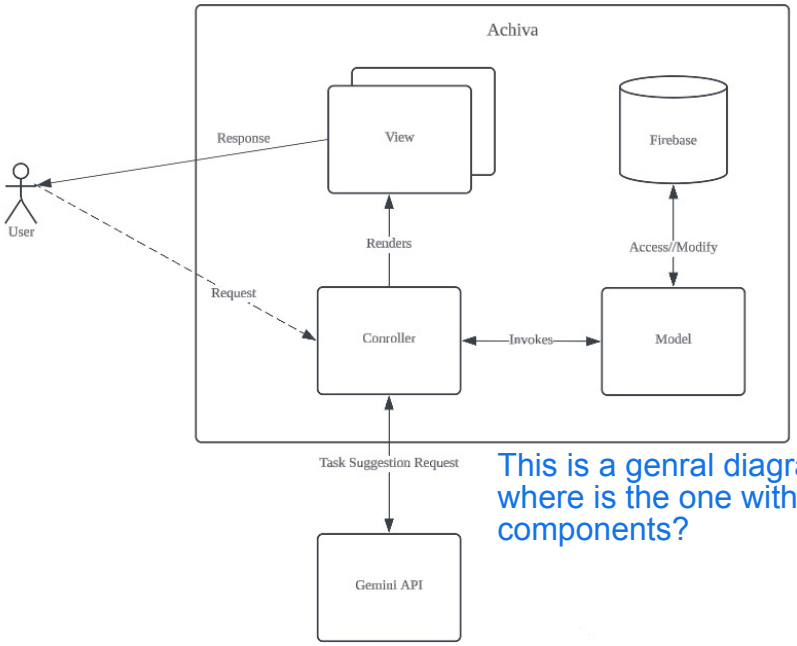


Figure 7 Achiva Use Case Diagram

Architecture & Deployment Diagram



This is a genral diagram for any MVC. where is the one with your system's components?

Figure 8 Achiva System Architecture Diagram

This is not how we draw the deployment diagram

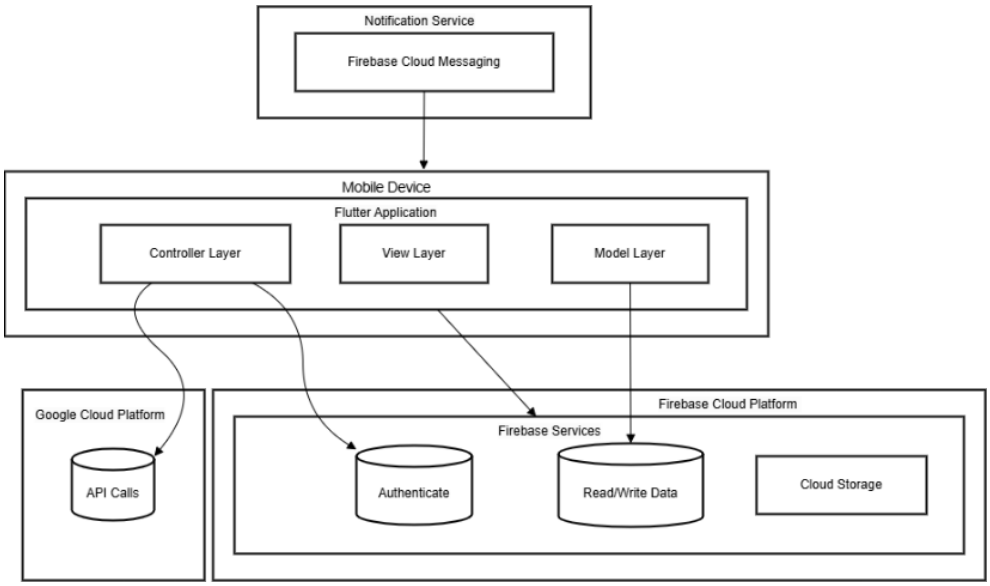


Figure 9 Achiva Deployment Diagram

جامعة الملك سعود
King Saud University
كلية علوم الحاسب والمعلومات
قسم هندسة البرمجيات

KING SAUD UNIVERSITY
COLLEGE OF COMPUTER AND INFORMATION SCIENCES
DEPARTMENT OF SOFTWARE ENGINEERING

Maintenance Request Management

Effective maintenance requires a systematic and organized approach that ensures all stakeholders understand the process and can easily follow necessary steps. The maintenance management process must be particularly robust given Achiva's interconnected features involving user collaboration, goal tracking, and analytics. This plan outlines our comprehensive approach to handling maintenance activities while maintaining system stability and user satisfaction throughout the development cycle. Maintenance Activity Management Plan:

1. **Centralized Request Management System** The foundation of our maintenance process is a centralized request management system that tracks all changes from submission through completion. This system includes standardized forms for maintenance requests, impact analysis documentation, and progress tracking. Using this systematic approach ensures no requests are overlooked and all stakeholders remain informed about the status of their requests.

2. **Review and Prioritization Process** The review process for maintenance requests follows a structured approach that ensures proper evaluation of each change's impact and importance. Our technical team conducts bi-weekly reviews to assess incoming requests. The review process includes detailed discussions with stakeholders, technical feasibility assessments, and resource availability analysis to ensure realistic implementation timelines.

3. **Cost and Resource Management** Resource allocation and cost estimation form a critical component of our maintenance planning. Each request undergoes thorough analysis to determine required development effort, testing resources, and infrastructure needs. We consider both immediate implementation costs and long-term maintenance implications, ensuring sustainable resource utilization across all maintenance activities.

4. **Impact Analysis** Our impact analysis process focuses on understanding the full scope of changes across Achiva's interconnected systems. This includes evaluating effects on user experience, system performance, data security, and feature integration. This thorough approach helps prevent unexpected issues and ensures smooth integration of new features with existing functionality.

5. **Testing and Quality Assurance** Quality assurance plays a central role in our maintenance process, with a comprehensive testing strategy that covers all aspects of feature implementation. Our testing approach includes unit testing of modified components, integration testing with existing features, and user acceptance testing for new functionality.

6. **Configuration Management** Version control and configuration management form the backbone of our maintenance implementation process. Using GitHub as our primary version control system, we maintain clear separation between development, staging, and production environments. Each maintenance request is developed in isolated feature branches, undergoes code review, and passes automated testing before integration.

7. **Documentation and Communication** Maintaining comprehensive documentation is essential for long-term system maintainability. Our documentation process covers technical specifications, user guides, API documentation, and support team training materials. Each maintenance change is thoroughly documented, including implementation details, configuration changes, and user impact

8. **Monitoring and Feedback** Post-implementation monitoring, and feedback collection form the final crucial component of our maintenance process. We maintain comprehensive monitoring of system performance metrics, error rates, and user engagement patterns following each change. This continuous feedback loop ensures our maintenance process remains effective and aligned with user needs.

Software Maintenance Requests Form Template

Request Information & Details

Request Information			
Request ID	**MR_#**	**Date**	**DD/MM/YYYY**
Requester Name	----------	**Email**	----------
Role/Department	----------	**System/Module Affected**	----------
Request Information			
Change Name	----------	**Justification**	
Description	Briefly describe the change and its expected outcome.	Why is this change necessary?	
Request Type	☐ Bug Fix ☐ Feature Enhancement ☐ Performance Optimization ☐ Security Update ☐ Compatibility Update ☐ Other: _____		
Maintenance Type	☐ Corrective ☐ Perfective ☐ Adaptive ☐ Preventive		
Severity	☐ High ☐ Medium ☐ Low	**Priority**	☐ High ☐ Medium ☐ Low

style -0.25

KING SAUD UNIVERSITY
COLLEGE OF COMPUTER AND INFORMATION SCIENCES
DEPARTMENT OF SOFTWARE ENGINEERING

Software Maintenance Requests & Cost Estimation

Seperate each MR in a heading

Request Information			
Request ID	**MR_1**	**Date**	**16/2 /2025**
Requester Name	Shouq alqureshi	**Email**	Shouq@gmail.com
Role/Department	IT department	**System/Module Affected**	Shared goal
Request Details			
Change Name	Task Assignment in Shared Goals	**Justification**	
Description	Tasks added in the shared goal will have the choice of assigning it to a collaborators restricting its un/check to them, reminders will be sent to assigned collaborators	Assigning specific tasks to collaborators will enhance accountability by clearly visualizing responsibility, preventing the current situation where tasks can be randomly checked/unchecked by any collaborator. This structured approach will enable more efficient goal management. Additionally, the system will optimize notifications by sending reminders only to assigned collaborators rather than alerting everyone unnecessarily.	
Request Type	☐Bug Fix ☒ Feature Enhancement ☐Performance Optimization ☐ Security Update ☐ Compatibility Update ☐ Other: _____		
Maintenance Type	☐ Corrective ☒ Perfective ☐ Adaptive ☐ Preventive		
Severity	☐ High ☒ Medium ☐ Low	**Priority**	☐ High ☒ Medium ☐ Low

where are the tables caption?

Activity	**Estimate (cost)**	**Estimate (person-days)**
1. Understand the problem and identify required changes.	140$	(1-2)
2. Design the changes.	70$	(1-1)

3. Perform impact analysis.	70$	(1-1)
4. Implement changes in source code.	420$	(2-3)
5. Update SRS, SDD, STP, configuration documentation.	140$	(2-1)
6. Compile and integrate into baseline.	140$	(2-1)
7. Test functionality of changes.	280$	(2-2)
8. Perform regression testing.	140$	(2-1)
9. Release new baseline and report results.	140$	(2-1)
Total	**1540$**	**(15-13)**

Request Information			
Request ID	**MR_2**	**Date**	**16/2 /2025**
Requester Name	Waad Almutairi	**Email**	Waad@gmail.com
Role/Department	IT Department	**System/Module Affected**	User Notifications and Reminders
Request Details			
Change Name	Task Due Date Notifications Enhancement	**Justification**	
Description	The system will allow users to configure custom reminder times for each task. Users will be able to set reminders to be notified at different intervals (e.g., 24 hours before the due date, 1 hour before, etc.). These reminders will be sent via push notifications and emails, depending on user preferences	Currently, users are receiving notifications for tasks at a fixed time before the deadline. To improve task management and ensure timely completion, users should have the ability to set personalized reminder times for tasks. This will provide a more tailored user experience and enhance user productivity by allowing them to set reminders based on their individual preferences	

KING SAUD UNIVERSITY
COLLEGE OF COMPUTER AND INFORMATION SCIENCES
DEPARTMENT OF SOFTWARE ENGINEERING

Request Type	☐ Bug Fix ☒ Feature Enhancement ☐ Performance Optimization ☐ Security Update ☐ Compatibility Update ☐ Other: _____		
Maintenance Type	☐ Corrective ☒ Perfective ☐ Adaptive ☐ Preventive		
Severity	☐ High ☒ Medium ☐ Low	**Priority**	☐ High ☒ Medium ☐ Low

Activity	**Estimate (cost)**	**Estimate (person-days)**
1. Understand the problem and identify required changes.	120$	(1-2)
2. Design the changes.	90$	(1-1)
3. Perform impact analysis.	90$	(1-1)
4. Implement changes in source code.	500$	(2-3)
5. Update SRS, SDD, STP, configuration documentation.	120$	(2-1)
6. Compile and integrate into baseline.	120$	(2-1)
7. Test functionality of changes.	240$	(2-2)
8. Perform regression testing.	120$	(2-1)
9. Release new baseline and report results.	120$	(2-1)
Total	**1410$**	**(15-13)**

KING SAUD UNIVERSITY
COLLEGE OF COMPUTER AND INFORMATION SCIENCES
DEPARTMENT OF SOFTWARE ENGINEERING

Request Information			
Request ID	**MR_3**	**Date**	**19/2/2025**
Requester Name	Batool Alsugaih	**Email**	Batool@gmail.com
Role/Department	IT Department	**System/Module Affected**	User goals
Request Details			
Change Name	Convert Individual Goal to Shared Goal	**Justification**	
Description	Users will have the option to convert an existing personal goal into a shared goal with collaborators. This will: - Retain all progress, history, and attachments. - Allow users to add collaborators without duplicating the goal or recreating it.	Currently, users must recreate an existing goal if they want to make it a shared goal. This leads to redundancy, loss of progress tracking, and extra effort. Allowing users to directly convert an individual goal into a shared goal will improve user experience and efficiency. The update will also enhance collaboration by making goal-sharing more seamless.	
Request Type	☐ Bug Fix ☒ Feature Enhancement ☐ Performance Optimization ☐ Security Update ☐ Compatibility Update ☐ Other: _____		
Maintenance Type	☐ Corrective ☐ Perfective ☒ Adaptive ☐ Preventive		
Severity	☐ High ☒ Medium ☐ Low	**Priority**	☐ High ☒ Medium ☐ Low

Activity	**Estimate (cost)**	**Estimate (person-days)**
1. Understand the problem and identify required changes.	120$	(1-2)
2. Design the changes.	90$	(1-1)
3. Perform impact analysis.	60$	(1-1)
4. Implement changes in source code.	450$	(2-3)

5. Update SRS, SDD, STP, configuration documentation.	150$	(2-1)
6. Compile and integrate into baseline.	120$	(2-1)
7. Test functionality of changes.	240$	(2-1)
8. Perform regression testing.	120$	(2-2)
9. Release new baseline and report results.	140$	(2-1)
Total	**1490$**	**(15-13)**

Request Information			
Request ID	**MR_4**	**Date**	**20/2/2025**
Requester Name	Riyam Alsuhaibani	**Email**	Riyam@gmail.com
Request Details			
Change Name	Advanced Goal Progress Analytics and Insights	**Justification**	
Description	Implement an advanced analytics dashboard that provides: • Visual representations of goal completion trends over time • Pattern analysis of most productive days/times • Success rate comparisons across different goal categories • Predictive completion dates based on current progress • Personalized recommendations for goal achievement based on historical data	Currently, users can only see basic progress tracking for their goals. To improve user engagement and goal completion rates, users need deeper insights into their productivity patterns and goal achievement trends. This enhancement will provide valuable analytics that can help users understand their performance better and make data-driven adjustments to their goal-setting strategies.	

جامعة الملك سعود
King Saud University
كلية علوم الحاسب والمعلومات
قسم هندسة البرمجيات

KING SAUD UNIVERSITY
COLLEGE OF COMPUTER AND INFORMATION SCIENCES
DEPARTMENT OF SOFTWARE ENGINEERING

Request Type	☐ Bug Fix ☒ Feature Enhancement ☐ Performance Optimization ☐ Security Update ☐ Compatibility Update ☐ Other: _____		
Maintenance Type	☐ Corrective ☒ Perfective ☐ Adaptive ☐ Preventive		
Severity	☐ High ☒ Medium ☐ Low	**Priority**	☒ High ☐ Medium ☐ Low

Activity	**Estimate (cost)**	**Estimate (person-days)**
1. Understand the problem and identify required changes.	$150	(1-2)
2. Design the changes.	$100	(1-1)
3. Perform impact analysis.	$80	(1-1)
4. Implement changes in source code.	$600	(3-3)
5. Update SRS, SDD, STP, configuration documentation.	150$	(2-1)
6. Compile and integrate into baseline.	$150	(2-1)
7. Test functionality of changes.	$300	(2-2)
8. Perform regression testing.	$150	(2-1)
9. Release new baseline and report results.	$150	(2-1)
Total	**$1830**	**(16-13)**

Request Information			
Request ID	**MR_5**	**Date**	**20/2/2025**
Requester Name	Amal Alharbi	**Email**	Amal@gmail.com
Role/Department	IT Department.	**System/Module Affected**	AI Chat Module
Request Details			
Change Name	Multi-Input Support for AI chat.	**Justification**	
Description	The system's current AI chat functionality is limited to text input, restricting users from utilizing more diverse communication methods when	Expanding the AI chat's input capabilities will allow users to communicate more effectively by enabling voice, image, and file uploads.	

	interacting with the AI for task management assistance. To enhance user experience and improve the AI's ability to understand and organize tasks effectively, it's essential to expand this capability. Therefore, enabling users to send voice recordings, upload images, and share files within the AI chat will provide a more flexible and interactive way to communicate, allowing for more accurate task breakdowns and suggestions.	This will enhance the AI's ability to analyze and provide better task management suggestions, leading to improved user experience and efficiency.	
Request Type	☐ Bug Fix ☒ Feature Enhancement ☐ Performance Optimization ☐ Security Update ☐ Compatibility Update ☐ Other: _____		
Maintenance Type	☐ Corrective ☒ Perfective ☐ Adaptive ☐ Preventive		
Severity	☐ High ☒ Medium ☐ Low	**Priority**	☐ High ☒ Medium ☐ Low

Activity	**Estimate (cost)**	**Estimate(person-days)**
1. Understand the problem and identify required changes.	150$	(2-3)
2. Design the changes.	120$	(2-2)
3. Perform impact analysis.	80$	(2-2)
4. Implement changes in source code.	500$	(3-4)
5. Update SRS, SDD, STP, configuration documentation.	180$	(2-2)
6. Compile and integrate into baseline.	150$	(2-2)
7. Test functionality of changes.	300$	(3-2)
8. Perform regression testing.	150$	(3-2)
9. Release new baseline and report results.	160$	(2-2)
Total	**1790$**	**(21-21)**

جامعة الملك سعود
King Saud University
كلية علوم الحاسب والمعلومات
قسم هندسة البرمجيات

KING SAUD UNIVERSITY
COLLEGE OF COMPUTER AND INFORMATION SCIENCES
DEPARTMENT OF SOFTWARE ENGINEERING

Request Information			
Request ID	**MR_6**	**Date**	**20/2/2025**
Requester Name	Miriam Almogren	**Email**	miriamoffic@gmail.com
Role/Department	IT Department.	**System/Module Affected**	Productivity Leaderboard
Request Details			
Change Name	Improved Productivity Ranking System	**Justification**	
Description	The updated productivity ranking system will: Implement a dynamic scoring algorithm that considers task difficulty and consistency. Introduce ranking filters allowing users to view rankings based on different time periods. Optimize database queries to handle increased complexity efficiently.	The current productivity ranking system in Achiva lacks a dynamic ranking algorithm that adjusts based on user engagement, task difficulty, and goal completion consistency. The enhancement will introduce a more accurate and competitive ranking system that rewards users fairly and motivates engagement. Additionally, new leaderboard filters will allow users to compare productivity rankings based on custom time frames (e.g., weekly, monthly, yearly).	
Request Type	☐ Bug Fix ☒ Feature Enhancement ☐ Performance Optimization ☐ Security Update ☐ Compatibility Update ☐ Other: _____		
Maintenance Type	☐ Corrective ☒ Perfective ☐ Adaptive ☐ Preventive		
Severity	☒ High ☐ Medium ☐ Low	**Priority**	☐ High ☒ Medium ☐ Low

Activity	**Estimated Cost ($)**	**Estimated Person-Days**
Requirement Analysis & Planning	150$	(1-2)
UI/UX Design Enhancements	100$	(1)
Algorithm Design & Implementation	500$	(3)

Database Query Optimization	200$	(2)
Update SRS, SDD, STP, Documentation	100$	(1)
System Integration & Testing	300$	(2-3)
Regression Testing	150$	(2)
Deployment & Report Generation	150$	(1-2)
Total	**1650$**	**(13-15)**

very long texts inside the tables..

Impact analysis

MR_1		
Current State	**Future State**	**Gap**
Currently, tasks in shared goals can be checked/unchecked by any collaborator without specific assignment, leading to confusion about responsibility and ownership. Notifications and reminders are sent to all collaborators regardless of actual task ownership, causing unnecessary notifications and potentially missed responsibilities.	users will be able to explicitly assign tasks to specific collaborators, Notifications will be targeted only to assigned collaborators for specific tasks, un/check functionality will be restricted to assigned collaborators	The system needs new UI components for task assignment ,backend logic for tracking assignments, and modifications to the notification system to target specific users based on assignments.
Type of Impact	**Timing of Impact**	**Level of Impact**
☒ Functional	☐ Immediate	☐ Low
☒ UI/UX	☒ Short-term	☒ Medium
☐ Performance	☐ Long-term	☐ High
☐ ____		
Summary of Impact	**Scale of Change**	**Key Risks**
This change will affect all users who participate in shared goals, improving task management and accountability while reducing confusion and notification noise.	☐ Organization ☐ Program ☒ Project ☐ ____	- Potential user confusion during transition. - Risk of overcomplicating the UI. - Data handling of existing shared goals
Roles Affected	**Number Affected**	**Communication Requirements**
End users, Project managers, Team leads, developers, testers	All users of the shared goals feature (approximately 80%)	• In-app notifications about the new feature

		• Meeting with all stakeholder to be informed of the impact of the change
Training Requirements	**Leadership Oversight**	**Alignment & Collaboration Needed**
Light training will be needed, through a walkthrough via app.	Product Manager, UX Lead they will be approving the MR	1. UX/UI team 2. Backend development team 3. QA team
Dependencies & Affected Componnts		
• Database schema. • Task management backend. • Notification service. • UI components.		

MR_2		
Current State	**Future State**	**Gap**
Currently, the system sends fixed reminder notifications for tasks, without providing users with the ability to customize when they are reminded. This can lead to users receiving notifications that are not timely or relevant, potentially affecting their task completion efficiency	The system will allow users to customize reminder times for each task, enabling them to choose the intervals that best suit their schedule. These personalized reminders will be sent via push notifications and emails, depending on the user's preferences, improving user engagement and task management	The system currently does not support custom reminder intervals for tasks. To enable this feature, significant changes will be needed to the user interface for configuring reminder times, backend logic to store and process the custom settings, and the notification system to trigger reminders according to the user-defined intervals.
Type of Impact	**Timing of Impact**	**Level of Impact**
☒ Functional	☐ Immediate	☐ Low
☒ UI/UX	☒ Short-term	☒ Medium
☐ Performance	☐ Long-term	☐ High
☐ ____		
Summary of Impact	**Scale of Change**	**Key Risks**
This change will impact all users who interact with the task management and notification	☐ Organization ☐ Program ☒ Project	- Some users may find it difficult to set custom

features, which comprise about 80% of the user base.	☐ ____	reminders if the UI is not intuitive. - Adding more options to configure reminders may clutter the UI if not designed carefully.
Roles Affected	**Number Affected**	**Communication Requirements**
• End users. • Developers. • Testers.	All users of the task management system (approximately 80%)	• In-app notifications will be sent to users informing them about the new feature and how to use it. • Stakeholder meetings with product managers, developers, and UX/UI teams will be necessary to ensure alignment on the feature's design and implementation.
Training Requirements	**Leadership Oversight**	**Alignment & Collaboration Needed**
A simple walkthrough within the app will be provided to guide users on how to set up custom reminders.	Product Manager and UX/UI Lead needs to approve the MR	• UX/UI team • Backend development team • QA team
Dependencies & Affected Components		
• Database schema. • Task management system. • Notification system. • UI components.		

MR_3		
Current State	**Future State**	**Gap**
Currently, users cannot convert an existing goal into a shared goal. Instead, they must recreate it, losing progress, history, and attachments. This results in redundancy, inefficiency, and a poor collaboration experience.	Users will be able to **convert an existing goal into a shared goal** while keeping all progress, attachments, and tracking history. They can add collaborators seamlessly,	The system needs **new UI components** for goal sharing, **backend modifications** to update goal ownership and permissions dynamically, **database updates** to support shared goal states, and **notification system adjustments**

	and notifications will be sent to inform them.	to inform collaborators when a goal is shared.
Type of Impact	**Timing of Impact**	**Level of Impact**
☒ Functional	☒ Immediate	☐ Low
☒ UI/UX	☒ Short-term	☒ Medium
☐ Performance	☐ Long-term	☐ High
☐ ____		
Summary of Impact	**Scale of Change**	**Key Risks**
This change will impact all users who always wants to convert their individual goals into shared goals.	☐ Organization ☐ Program ☒ Project ☐ ____	- Users may be confused about how to convert goals into shared goals. - Increased **UI complexity** due to the addition of a new sharing option.
Roles Affected	**Number Affected**	**Communication Requirements**
End users, developers, testers.	All users utilizing goal-sharing (80%)	• A meeting with key stakeholders to explain changes and gather feedback.
Training Requirements	**Leadership Oversight**	**Alignment & Collaboration Needed**
Minimal training needed. A short tutorial or in-app guide will be provided.	Product Manager and UX Lead will review and approve the change.	• Backend development team • Testers team • UX/UI team
Dependencies & Affected Components		
• UI components for goal sharing. • Database schema updates. • Goal management backend.		

KING SAUD UNIVERSITY
COLLEGE OF COMPUTER AND INFORMATION SCIENCES
DEPARTMENT OF SOFTWARE ENGINEERING

MR_4		
Current State	**Future State**	**Gap**
Users can only see basic progress tracking for their goals with limited insight into patterns or trends. The current system lacks detailed analytics and visualization tools for understanding goal achievement patterns.	Users will have access to an **advanced analytics dashboard** with visual representations of goal trends, pattern analysis, success rate comparisons, predictive completion dates, and personalized recommendations based on historical data.	Users will have access to an **advanced analytics dashboard** with visual representations of goal trends, pattern analysis, success rate comparisons, predictive completion dates, and personalized recommendations based on historical data.
Type of Impact	**Timing of Impact**	**Level of Impact**
☒ Functional	☐ Immediate	☐ Low
☒ UI/UX	☒ Short-term	☒ Medium
☒ Performance	☒ Long-term	☐ High
☐ ____		
Summary of Impact	**Scale of Change**	**Key Risks**
This change will impact all users who track goals and want deeper insights into their progress patterns and achievement trends.	☐ Organization ☐ Program ☒ Project ☐ ____	- System **performance degradation** due to increased data processing requirements - **Data accuracy concerns** with predictive algorithms - **User overwhelm** from too many analytics options. - **Mobile responsiveness issues** with complex visualization components.
Roles Affected	**Number Affected**	**Communication Requirements**
End users, developers, data analysts, testers.	All users utilizing goal tracking (80%)	- Email announcement to all users about the new analytics features - In-app notifications highlighting new capabilities - Documentation updates in the knowledge base.
Training Requirements	**Leadership Oversight**	**Alignment & Collaboration Needed**
Moderate training needed. Tutorial videos and tooltips will be provided to help users interpret analytics data.	Product Manager, Analytics Lead, and UX Lead will review and approve the change.	• Backend development team. • Data science team. • UX/UI team.

		• QA testing team. • Performance testing team.

Dependencies & Affected Components

- Goal tracking database and historical data storage
- Data processing pipeline
- UI dashboard components
- Reporting system
- Mobile app interface

MR_5		
Current State	**Future State**	**Gap**
The AI chat functionality currently supports only text input, limiting users' ability to interact with the system using more diverse communication methods. This restriction hinders effective task management assistance, as users cannot provide voice recordings, upload images, or share files, which are necessary for more accurate task breakdowns and recommendations.	The system will allow users to interact with the AI chat by sending voice recordings, uploading images, and sharing files. This will enable a more flexible and interactive communication method, leading to better task breakdowns, more accurate suggestions, and a more intuitive user experience.	The lack of support for multiple input types (voice, images, files) restricts users from engaging with the AI in a more dynamic and interactive way. This affects the AI's ability to fully understand and analyze tasks, reducing the accuracy of task management suggestions.
Type of Impact	**Timing of Impact**	**Level of Impact**
☒ Functional	☐ Immediate	☐ Low
☒ UI/UX	☒ Short-term	☒ Medium
☐ Performance	☐ Long-term	☐ High
☐ ____		
Summary of Impact	**Scale of Change**	**Key Risks**
This change will enhance the user experience by enabling multiple input	☐ Organization ☐ Program ☒ Project	User Adoption: The introduction of multiple input types may require users to adjust to new features, and there could be

types, improving the AI's ability to process and analyze tasks. Users will be able to communicate with the AI in a more interactive manner, leading to more effective task management and improved task suggestions.	☐ ____	resistance or confusion without proper training or user-friendly design. Data Privacy Concerns: Handling voice recordings and images raises concerns about user data privacy, requiring clear consent and secure storage methods.
Roles Affected	**Number Affected**	**Communication Requirements**
End users Developers Testers UX/UI Team Number Affected: All users of the AI chat feature (approximately 80%).	All users of the AI chat feature(approximately 100% of active AI chat).	A meeting with key stakeholders to explain changes and gather feedback. Collaboration with the backend development team, UX/UI team, and testing team.
Training Requirements	**Leadership Oversight**	**Alignment & Collaboration Needed**
Light training will be required. A walkthrough of the new features will be provided through an in-app tutorial to ensure users understand the new input methods.	The Product Manager and UX Lead will review and approve the MR	Backend team to modify the AI's input handling system. UX/UI team to update the user interface for new input methods. Testing teams for functionality and integration validation.
Dependencies & Affected Components		
Task management backend to support file handling and AI processing. UI components for displaying new input options (voice, image, file upload).		

KING SAUD UNIVERSITY
COLLEGE OF COMPUTER AND INFORMATION SCIENCES
DEPARTMENT OF SOFTWARE ENGINEERING

MR_6		
Current State	**Future State**	**Gap**
The existing productivity ranking system provides static rankings that do not account for varying levels of task difficulty or engagement patterns. Users may feel that the rankings are not reflective of their actual productivity, leading to reduced motivation and engagement.	With the new ranking system, users will have a more dynamic and fair assessment of their productivity. The filtering options will provide flexibility in viewing rankings over different time periods, making the system more interactive and engaging.	The system currently lacks a mechanism for dynamically adjusting rankings based on various performance metrics. This update requires changes in the UI, backend logic, and database queries to support the new ranking methodology.
Type of Impact	**Timing of Impact**	**Level of Impact**
☒ Functional	☐ Immediate	☐ Low
☒ UI/UX	☒ Short-term	☒ Medium
☐ Performance	☐ Long-term	☐ High
☐ ____		
Summary of Impact	**Scale of Change**	**Key Risks**
- Users will experience a more competitive and engaging ranking system. - Database performance must be optimized to handle additional queries.	☐ Organization ☐ Program ☒ Project ☐ ____	- Users may need time to understand the new ranking methodology. - Potential performance issues if database queries are not optimized efficiently.
Roles Affected	**Number Affected**	**Communication Requirements**
End users, Project managers, UX/UI designers, Backend developers, Testers	**All users interacting with the productivity dashboard (estimated 80% of active users)**	• **In-app notifications explaining the new ranking system.** • **Documentation and FAQs updated to guide users.** • **Stakeholder meetings to ensure alignment with expectations.**
Training Requirements	**Leadership Oversight**	**Alignment & Collaboration Needed**

A simple in-app walkthrough for users to familiarize themselves with the new ranking system	**Product Manager and UX Lead will review and approve the changes**	• **UX/UI team** • **Backend development team** • **QA team**
Dependencies & Affected Components		
• **Database schema updates** • **Productivity dashboard UI** • **Ranking algorithm backend logic**		

جامعة الملك سعود
King Saud University
كلية علوم الحاسب والمعلومات
قسم هندسة البرمجيات

KING SAUD UNIVERSITY
COLLEGE OF COMPUTER AND INFORMATION SCIENCES
DEPARTMENT OF SOFTWARE ENGINEERING

MR prioritization

Prioritization Criteria

Our prioritization criteria are user-centered focusing on the MRs that gain user engagement and satisfaction, providing ***business values***. Also, ***strategic alignment with core features*** of the app plays a big role in prioritizing MRs. To ensure efficiency we will consider ***implementation complexity*** such as resource requirements and technical challenges is taken into consideration

Ranked MRS

1. **Task Assignment in Shared Goals**
   Our Highest Priority MR : It addresses a fundamental usability issue with one of the app core functionalities which is collaborative goals that feed the app ultimate goal that is user productivity. Also this MRs impact goes beyond task assignment which enhance users experience by reducing the amount of notification received of unrelated tasks
2. **Task Due Date Notifications Enhancement**
   It impacts core task management functionality and directly improve user experience in a repetitive used function and will increase user satisfaction and it will be Built on existing notification infrastructure with low implementation complexity
3. **Convert Individual Goal to Shared Goal**
   It provides the user flexibility that will enhance it's satisfaction in handling core functions and avoid user's frustration.
4. **Advanced Goal Progress Analytics and Insights**
   It Enhances user understanding of their productivity and improve user experince. Also, related to a core functionality
5. **Improved Productivity Ranking System**
   Since Adding a New algorithm and custom time frames would improve user engagement in the same time it has high implementation complexity requiring algorithm development and it impact is limited to competitive users.
6. **Multi-Input Support for AI Chat**
   It's a valuable MR, but has a high implementation complexity to have the resources in handling multi-input for a side functionality