



SRE
ASSIGNMENT -3
MENTAL
WELLNESS &
HABIT TRACKER
APP WITH AI
COACH



By

Zunaira Abdul Aziz BSSE23058

Faiqa Arshad BSSE23028

Hamna Fatima BSSE23080

Areeba Shahbaz BSSE23097



SOFTWARE REQUIREMENT ENGINEERING

5. Role Assignments

Group Members:

- Human Fatima (BSSE23080)
- Zunaira Abdul Aziz (BSSE23058)
- Faiqa Arshad (BSSE23028)
- Areeba Shahbaz (Bsse23097)

Mental Wellness & Habit Tracker App with AI Coach



Problem Definition:

Problem:

The increasing prevalence of mental health issues such as stress, anxiety, and depression in modern society is driving a need for more accessible and personalized wellness tools. Current mental health apps and trackers often fail to provide real-time, tailored support, and many people struggle to maintain consistent mental health routines. Wearable technology, while advanced in tracking physical health, is underutilized in the mental wellness space.

Many users experience mental health challenges without having a system that provides immediate, personalized feedback based on their real-time data, such as mood, stress levels, or daily habits.

Root Cause Analysis:

Issues and Causes:

1. Limited Personalization in Mental Health Apps:

- o Current mental health apps often provide generic advice, failing to account for individual user behavior, mood patterns, or lifestyle habits, which reduces their effectiveness.

2. Underutilization of Wearable Technology for Mental Health:

- o While wearable devices (e.g., smartwatches) have the capability to track physiological indicators like heart rate or sleep patterns, they are primarily used for physical fitness, leaving their potential for mental health management untapped.

3. Lack of Real-Time Feedback and Monitoring:

- o Users do not receive real-time insights or feedback from their mental health apps, which is crucial for managing stress as they occur.

4. Insufficient AI Integration for Personalized Recommendations:

- o Mental health tools lack robust AI systems that can analyze mood data, biometric information, and daily habits to provide personalized, proactive mental wellness advice.

5. Mental Health Stigma and User Reluctance:

- o Many individuals are reluctant to actively seek help for mental health issues due to stigma, making passive tools that integrate into daily life, such as wearables, a better option for many users.

6. Data Privacy Concerns:

- o Mental health data is highly sensitive, and users are often concerned about how their data is stored, processed, and shared. Without strong privacy protections, users may be hesitant to engage fully with mental health tracking apps.





Requirement Engineer 1:

R1: Personalized Mental Health Insights:

- The app should provide tailored mental health insights based on the user's daily habits, mood entries, and physiological data. These insights should help users recognize patterns in their mental wellness and offer relevant, actionable advice.

R2: Real-Time AI-Driven Feedback:

- An AI system should continuously analyze data from wearables and user inputs to provide real-time mental health advice. This ensures users receive timely suggestions or interventions to manage stress, anxiety, or low moods.

R3: Seamless Wearable Integration:

- The app must integrate with popular wearable devices (e.g., smartwatches) to track biometric data like heart rate, sleep patterns, and activity levels. These inputs are essential for providing a holistic view of the user's mental health.

R4: Mood and Habit Tracking:

- Users should be able to log their daily moods, activities, and habits in the app. The system should use this data, along with biometric inputs, to offer personalized recommendations aimed at improving mental wellness.

R5: Data Privacy and Security Compliance:

- The app should prioritize user data privacy, ensuring that all data is encrypted, stored securely, and compliant with data protection regulations like GDPR. Users should have control over their data and be able to delete it upon request.

R6: Low User Interaction Requirement:

- The app should operate with minimal manual input from the user. It should rely heavily on passive data collection through wearables and infrequent user check-ins, offering insights without burdening the user with frequent interaction.

R7: Offline Functionality for Mood and Habit Tracking:

- Users should be able to log their mood and habits even when they are offline. The data should sync with the cloud once the user regains connectivity, ensuring continuous tracking without interruptions.
- loud once the user regains connectivity, ensuring continuous tracking without interruptions.

INTERVIEW QUESTIONS:

1. Understanding the Customer Profile

- **Question:** How often do you experience stress or anxiety, and how do you currently manage it?
 - **Response:** "I experience stress almost daily, especially with work deadlines. I try to manage it through apps and exercise, but I find it hard to stay consistent and track my progress."
-

2. Identifying the Challenge

- **Question:** Can you describe a situation where you felt your current mental health tools were not helping you effectively?
 - **Response:** "There have been times when I felt overwhelmed, but the apps I use don't give me specific feedback or suggestions that are relevant to how I'm feeling at that moment. I feel like I need something more personalized."
-

3. Exploring the User Context

- **Question:** Do you use wearable devices like a smartwatch or fitness tracker? How do you feel about using them for mental health tracking?
 - **Response:** "I use a smartwatch to track my physical health, but I've never used it for mental wellness. I'd be open to trying if it could help with real-time feedback on my mood or stress."
-

4. Recap of Understanding:

- **Question:** So it sounds like you're looking for a solution that integrates your wearable device, provides real-time feedback on your mental wellness, and tailors advice based on your actual mood and habits. Does that sound correct?
 - **Response:** "Yes, exactly! I need something that understands me better and gives me actionable tips throughout the day."
-

5. Providing the Analyst's Perspective on the Problem

- **Question:** How do you feel about having an AI-driven app that continuously monitors your mental health, provides real-time insights, and adjusts its advice based on your data?
 - **Response:** "That would be amazing! If it could use my data to give personalized suggestions and remind me to take breaks or manage stress, it would be really helpful."
-

6. Evaluating the Solution's Impact

- **Question:** How do you think an app that integrates with your wearable device and provides mental health insights throughout the day would impact your routine?
 - **Response:** "It would definitely help me stay on track. If it could give me reminders or tips when I'm feeling stressed, I'd be more likely to follow through on wellness practices."
-

7. Assessing the opportunity:

- **Question:** What features would you consider important for a mental health app that works with wearables?
 - **Response:** "It should track my mood, stress levels, and sleep patterns. I'd also appreciate notifications to help me stay mindful and reminders for self-care activities."
-

8. Assessing the reliability, performance, and support needs:

Question: How important is data privacy to you when using mental health apps, and what level of security would you expect?

- **Response:** "Privacy is a huge concern for me. I'd want to ensure my data is stored securely, with strong encryption, and only shared with my consent. I would expect transparency about how my data is used."
-

9. Other requirements:

- **Question:** Are there any specific features or concerns you'd like addressed in the app, such as ease of use or customization options?
 - **Response:** "The app should be simple to use, with easy-to-understand features. I'd like it to be customizable to fit my specific needs, like adjusting how often I receive tips or reminders."
-

10.Wrap up:

Question: Is there anything else you would want to add or any other concerns you have about using mental wellness apps or wearables?

- **Response:** "I think having a daily check-in feature would be great. It would help me reflect on how I'm feeling and give me insights on how to manage my stress levels more effectively."
-

11.The Analyst's Summary:

- **Question:** Based on what you've shared, it seems your main challenges are finding personalized, real-time mental health support, integrating wearable technology, and ensuring privacy. Do you agree?
 - **Response:** "Yes, that's right. If an app could offer real-time, personalized support and work with my wearable device while keeping my data secure, it would be a game-changer for my mental wellness."
-



Requirement Engineer 2:

❖ Periodic Personalized Feedback:

The app should provide feedback on a periodic basis (e.g., weekly or monthly), summarizing users' mental health progress based on their logged data, rather than real-time insights.

❖ Basic Biometric Data Integration:

The app should integrate with wearables to track basic physical health data (e.g., steps, calories burned) but not include deeper insights into mental health-related biometric data such as heart rate variability or sleep quality.

❖ Static, Pre-Configured Recommendations:

The app should offer pre-configured wellness tips and advice based on generalized user data (like mood logs and basic activity levels) without using advanced AI to provide tailored recommendations.

❖ Manual Mood and Stress Input:

Users should be able to manually log their mood and stress levels at designated times during the day (e.g., once in the morning and once at night), rather than having the app track these metrics continuously in real time.

❖ Simplified User Interface:

The app should prioritize a simple and minimalistic user interface, focusing on ease of use and avoiding feature overload, which could overwhelm the user.

❖ Basic Data Privacy and Security:

The app should meet general data protection regulations but focus on basic data privacy standards. Encryption and security measures should be in place, but more complex features such as end-to-end encryption are not required at this stage.

❖ Fixed, Non-Customizable Notification Schedule:

The app should send notifications on a fixed schedule (e.g., daily reminders) without the option for users to customize the frequency, content, or type of notifications they receive.

INTERVIEW QUESTIONS:

1. Exploring Stress Management Strategies

- **Question:** How often do you experience stress or anxiety, and how do you currently manage it?
 - **Response:** "I experience stress a few times a week, mostly due to work and personal responsibilities. I manage it through a combination of taking breaks and exercising. I try to stay on top of my mood, but I don't always have the time to log it regularly."
-

2. Identifying Current Challenges

- **Question:** Can you describe a situation where you felt your current mental health tools were not helping you effectively?
 - **Response:** "I use a couple of mental wellness apps, but I don't feel like they offer enough personalized advice. I mostly get general tips, and sometimes I forget to input my data. I need something simpler and more manageable."
-

3. Exploring Device Use and Preferences

- **Question:** Do you use wearable devices like a smartwatch or fitness tracker? How do you feel about using them for mental health tracking?
 - **Response:** "I use a fitness tracker to monitor my physical activity, but I haven't considered using it for mental health. I would be open to it, but I'd prefer something that doesn't require constant input from me."
-

4. Recap of Understanding:

- **Question:** So it sounds like you're looking for a solution that tracks your activity, provides occasional feedback, and offers simple wellness tips without requiring too much input from you. Does that sound correct?
 - **Response:** "Yes, exactly! I want something straightforward that can give me advice when I need it, without being too overwhelming or needing constant attention."
-

5. The Analyst's Input on Customer's Problem:

- **Question:** How do you feel about having an app that checks in with you occasionally and provides you with tips based on your activity and mood logs, but without real-time AI-driven feedback?
 - **Response:** "I think that could work for me. I don't need constant feedback, but having something that can periodically check in and offer useful suggestions would be helpful without being too intrusive."
-

6. Assessing the Solution:

- **Question:** How do you think an app that provides basic wellness tips and checks in periodically, integrated with your wearable device, would impact your routine?
 - **Response:** "It would be useful to have something that reminds me to take care of myself at regular intervals. I'd probably follow through more if it didn't require too much effort, like logging my mood multiple times a day."
-

7. Assessing the Opportunity:

- **Question:** What features would you consider important for a mental health app that works with wearables?
 - **Response:** "I'd like the app to track my sleep and activity levels, provide reminders for self-care, and offer simple, pre-configured tips. I also want it to work without too much input on my part, like setting reminders for mental wellness activities."
-

8. Assessing Reliability, Performance, and Support Needs:

- **Question:** How important is data privacy to you when using mental health apps, and what level of security would you expect?
 - **Response:** "Data privacy is very important to me. I want to be sure my personal information is secure and not shared without my permission. Basic encryption and transparency about how my data is being used would be enough for me."
-

9. Other Requirements:

- **Question:** Are there any specific features or concerns you'd like addressed in the app, such as ease of use or customization options?
 - **Response:** "The app should be very easy to navigate. I'd like to have the ability to adjust notification settings, but I don't want to be bombarded with too many options. A simple, streamlined interface would be great."
-

10. Wrap Up:

- **Question:** Is there anything else you would want to add or any other concerns you have about using mental wellness apps or wearables?
 - **Response:** "I think having occasional check-ins would be really helpful. I don't want to feel like I'm constantly being monitored, but a gentle reminder to take a break or focus on my mental wellness would be good."
-

11. The Analyst's Summary:

- **Question:** Based on what you've shared, it seems your main challenges are finding an easy-to-use app that doesn't require constant input, but still provides useful tips and reminders to improve your mental wellness. Do you agree?
 - **Response:** "Yes, that's right. If the app could give me occasional reminders and simple tips without needing too much effort from me, I'd be much more likely to stick with it."
-

Interaction Matrix:

E1 / E2	R1	R2	R3	R4	R5	R6	R7
1. Personalized Mental Health Insights (R1)	0	1	1	1	0	0	0
2. Real-Time AI-Driven Feedback (R2)	1	0	1	1	0	1000	0
3. Seamless Wearable Integration (R3)	1	1	0	1	1000	0	0
4. Mood and Habit Tracking (R4)	1	1	1	0	1000	0	1000
5. Data Privacy and Security Compliance (R5)	0	0	1000	1000	0	0	0
6. Low User Interaction Requirement (R6)	0	1000	0	1000	0	0	0
7. Offline Functionality for Mood and Habit Tracking (R7)	0	0	0	1000	0	0	0

REQUIREMENTS: R1, R2...

Conflict: 1

Overlap: 1000

Non-conflicting: 0

Horizontal Reqs: Req.

Engineer 1

Vertical Reqs: Req.Engineer2

Conflicting Requirements:

R2 (Real-Time AI-Driven Feedback) vs. E2-1 (Periodic Personalized Feedback)

Engineer-01 advocates for real-time feedback driven by AI, while Engineer-02 prefers periodic feedback. The continuous nature of feedback in E1 conflicts with the more static approach in E2.

R3 (Seamless Wearable Integration) vs. E2-2 (Basic Biometric Data Integration)

E1 requires full integration with wearables to track detailed biometric data relevant to mental health (e.g., heart rate, sleep quality), while E2 only seeks basic data integration (steps, calories burned), without deeper mental health-related insights.

R4 (Mood and Habit Tracking) vs. E2-4 (Manual Mood and Stress Input)

E1 envisions continuous tracking of moods and habits, incorporating biometric data for personalized recommendations. E2 focuses on manual input at specific times, which means less continuous tracking and integration.

Overlapping Requirements:

R5 (Data Privacy and Security Compliance) vs. E2-6 (Basic Data Privacy and Security)

Both E1 and E2 emphasize data privacy, with E1 requiring GDPR compliance and secure data encryption. However, E2 advocates for basic security standards, which overlaps with E1's requirement but with less stringent measures.

R7 (Offline Functionality for Mood and Habit Tracking) vs. E2-5 (Simplified User Interface)

Both requirements align in terms of allowing basic functionality when offline, though the specific focus is slightly different. E1 requires mood and habit tracking to work offline, while E2 focuses on a simpler interface that may support offline usage.

Non-Conflicting Requirements:

R6 (Low User Interaction Requirement) vs. E2-3 (Static, Pre-Configured Recommendations)

While E1 seeks low user interaction and passive data collection, E2's approach of static recommendations doesn't directly conflict with this. Both are aimed at reducing the burden on users but with different methods for achieving it.

R6 (Low User Interaction Requirement) vs. E2-7 (Fixed Notification Schedule)

E1's goal of low user interaction aligns with E2's requirement for fixed, non-customizable notifications. Both aim to limit user engagement in a way that supports passive monitoring of mental wellness.

Workshop:

We conducted a workshop and provided our interaction matrix to our client and had a conversation.

Team Member	Critical Requirements	Important Requirements	Useful Requirements
Member 1	Real-Time AI-Driven Feedback (R2) (E1)	Seamless Wearable Integration (R3) (E1)	Offline Functionality for Mood and Habit Tracking (R7) (E1)
	Mood and Habit Tracking (R4) (E1)	Data Privacy and Security Compliance (R5) (E1)	Low User Interaction Requirement (R6) (E1)
Member 2	Seamless Wearable Integration (R3) (E1)	Data Privacy and Security Compliance (R5) (E1)	Real-Time AI-Driven Feedback (R2) (E1)
	Mood and Habit Tracking (R4) (E1)	Low User Interaction Requirement (R6) (E1)	Periodic Personalized Feedback (E2)
Member 3	Real-Time AI-Driven Feedback (R2) (E1)	Offline Functionality for Mood and Habit Tracking (R7) (E1)	Basic Biometric Data Integration (E2)
	Mood and Habit Tracking (R4) (E1)	Simplified User Interface (E2)	Periodic Personalized Feedback (E2)

Member 4	Real-Time AI-Driven Feedback (R2) (E1)	Periodic Personalized Feedback (E2)	Offline Functionality for Mood and Habit Tracking (R7) (E1)
	Seamless Wearable Integration (R3) (E1)	Basic Biometric Data Integration (E2)	Data Privacy and Security Compliance (R5) (E1)
Member 5	Mood and Habit Tracking (R4) (E1)	Data Privacy and Security Compliance (R5) (E1)	Real-Time AI-Driven Feedback (R2) (E1)
	Seamless Wearable Integration (R3) (E1)	Simplified User Interface (E2)	Basic Biometric Data Integration (E2)
Team Member	Critical Requirements	Important Requirements	Useful Requirements
Member 1	Real-Time AI-Driven Feedback (R2) (E1)	Seamless Wearable Integration (R3) (E1)	Offline Functionality for Mood and Habit Tracking (R7) (E1)
	Mood and Habit Tracking (R4) (E1)	Data Privacy and Security Compliance (R5) (E1)	Low User Interaction Requirement (R6) (E1)
Member 2	Seamless Wearable Integration (R3) (E1)	Data Privacy and Security Compliance (R5) (E1)	Real-Time AI-Driven Feedback (R2) (E1)
	Mood and Habit Tracking (R4) (E1)	Low User Interaction Requirement (R6) (E1)	Periodic Personalized Feedback (E2)
Member 3	Real-Time AI-Driven Feedback (R2) (E1)	Offline Functionality for Mood and Habit Tracking (R7) (E1)	Basic Biometric Data Integration (E2)

Voting Results

Critical Requirements:

1. **Real-Time AI-Driven Feedback (R2)** - Strongly supported by E1.
2. **Mood and Habit Tracking (R4)** - Consistently prioritized for continuous data collection.
3. **Seamless Wearable Integration (R3)** - Essential for biometric data and continuous monitoring.

Important Requirements:

1. **Data Privacy and Security Compliance (R5)** - Required by both E1 and E2, though E1 demands stricter measures.
2. **Low User Interaction Requirement (R6)** - Supported by E1's emphasis on minimal user engagement.
3. **Offline Functionality for Mood and Habit Tracking (R7)** - Needed for tracking data even without connectivity.

Useful Requirements:

1. **Basic Biometric Data Integration (E2-2)** - Somewhat important for tracking basic physical health metrics, though less emphasized by E1.
2. **Simplified User Interface (E2-5)** - Beneficial for ensuring usability, but not as critical as other requirements.
3. **Periodic Personalized Feedback (E2-1)** - Seen as useful for E2's approach but conflicts with E1's real-time feedback focus.

This table presents the outcome of the workshop, showing how the requirements were prioritized according to the feedback of each team member. The critical, important, and useful categories reflect the consensus formed during the discussion.

6. Section Distribution

1. Project Overview

Project Title: Mental Wellness & Habit Tracker App with AI Coach

Date: November 9, 2024

Document Version: 2.0

Authors:

Zunaira Abdul Aziz BSSE23058 (Section 1-3)

Faiqa Arshad BSSE23028 (Section 4-6)

Hamna Fatima BSSE23080 (Section 7-9)

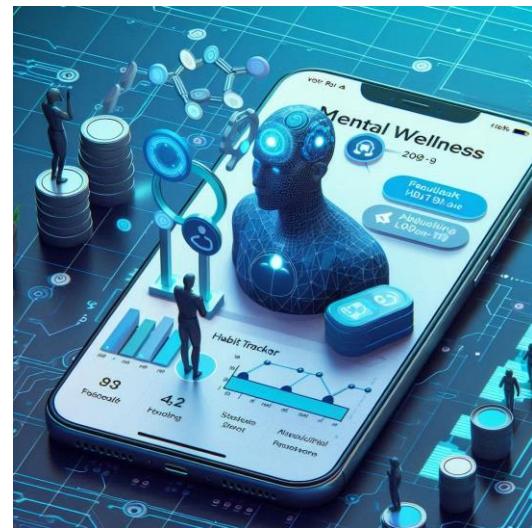
Areeba Shahbaz BSSE23097 (Section 10-11)

Purpose:

This project's goal is to build an app that helps people improve their mental well-being by tracking habits, recording moods, and getting personalized advice from an AI coach. The app aims to support users in developing healthy routines and understanding their emotions, with easy-to-use tools for better mental wellness.

Background:

With stress and mental health challenges becoming more common, people need ways to track and improve their mental wellness. Traditional habit and mood tracking can feel impersonal. The Mental Wellness & Habit Tracker App will solve this by adding an AI coach that offers helpful advice based on users' data, making the process more personalized and effective. By analysing patterns in users' habits and moods, the AI coach can suggest tailored routines and provide motivational support. This creates a more interactive and supportive experience, helping users stay consistent in their wellness journey.



Stakeholders:

- **Users:** The People who are using the app, and who wants to track personal habits and moods to enhance their mental health, as well as being helped by an AI coach.
- **Development Team:** The engineers, designers, and testers who build the app and make sure it works well and is easy to use.



Product Manager: It includes a person who is in charge of the project, ensuring the application fulfills the user requirement, and directing the development team for the similar purpose.

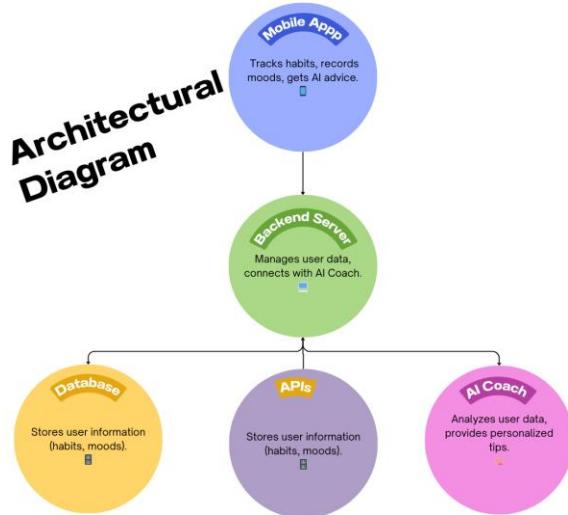
Mental Health Experts: Provide advice on how to design the app's features to be helpful and safe for users' mental health.

IT Support Team: Takes care of the app's servers, data storage, and security to make sure the app runs smoothly and safely.

Marketing Team: Promotes the app to users and gathers feedback to improve the app and attract more people.

2. System Architecture

Architecture Diagram:



Components:

- **Mobile App:** The main app where users track habits, record moods, and get AI advice.

- **Backend Server:** Stores and manages user data and connects with the AI coach.
- **AI Coach:** Analyzes data and provides personalized tips and reminders.
- **Database:** Saves all user information, like habits and mood entries.
- **APIs:** Connects the app to other services, like notifications or weather data.

Dependencies:

External Dependencies:

- **Third-party APIs:** The app may rely on external services like weather data or push notification services. If these services are down, it could affect app functionality.
- **App Stores:** The app's availability and updates depend on approval from app stores (Google Play, Apple App Store).

Internal Dependencies:

- **Backend Server:** The app depends on a reliable backend to store user data and process AI responses. If the server goes down, the app will not function properly.
- **AI Engine:** The app's ability to give personalized feedback depends on the performance of the AI system. If the AI engine has issues, it could affect the quality of the user experience.

3. Service Level Objectives (SLOs) & Indicators (SLIs)

Service Level Indicators (SLIs):

- **Response Time:** How quickly the app responds when the AI coach sends advice.
- **Error Rate:** How often there are issues with tracking or receiving AI tips.
- **Availability:** How often the app is up and running, so users can access it anytime.
- **Uptime:** The percentage of time the app is running without disruptions.
- **Data Sync Speed:** How quickly user data syncs across devices or between the app and server.
- **Request Processing Time:** Time taken to process user inputs or queries by the AI engine.

Service Level Objectives (SLOs):

- **Response Time:** The app should respond in under **1.5 seconds** for 90% of user actions.
- **Error Rate:** The error rate should be less than **0.5%** for all user actions.
- **Uptime:** The app should have at least **99.95% uptime**.
- **Data Sync Speed:** Syncing data should be completed in **less than 5 seconds**.
- **Request Processing Time:** AI responses should be processed in **under 3 seconds** for 80% of queries.

Service Level Agreements (SLAs):

- **App Availability:** The app will be available **99.95%** of the time, with a maximum downtime of **4 hours per year**.
- **Support Response:** Customer support will respond to reported issues within **12 hours**.
- **Issue Resolution:** Critical issues will be resolved within **24 hours**, while non-urgent issues will be addressed within **3 business days**.
- **Data Privacy:** The app will ensure user data is kept private and secure, following strict industry standards and legal requirements.
- **Updates & Maintenance:** Regular updates and maintenance will occur at least once every **three months**.

4. Monitoring & Alerting

Monitoring Tools

- Prometheus: Open-source systems monitoring and alerting toolkit, often paired with Grafana for visualization.
- Grafana: Visualization tool used to display time-series data from Prometheus and other sources.

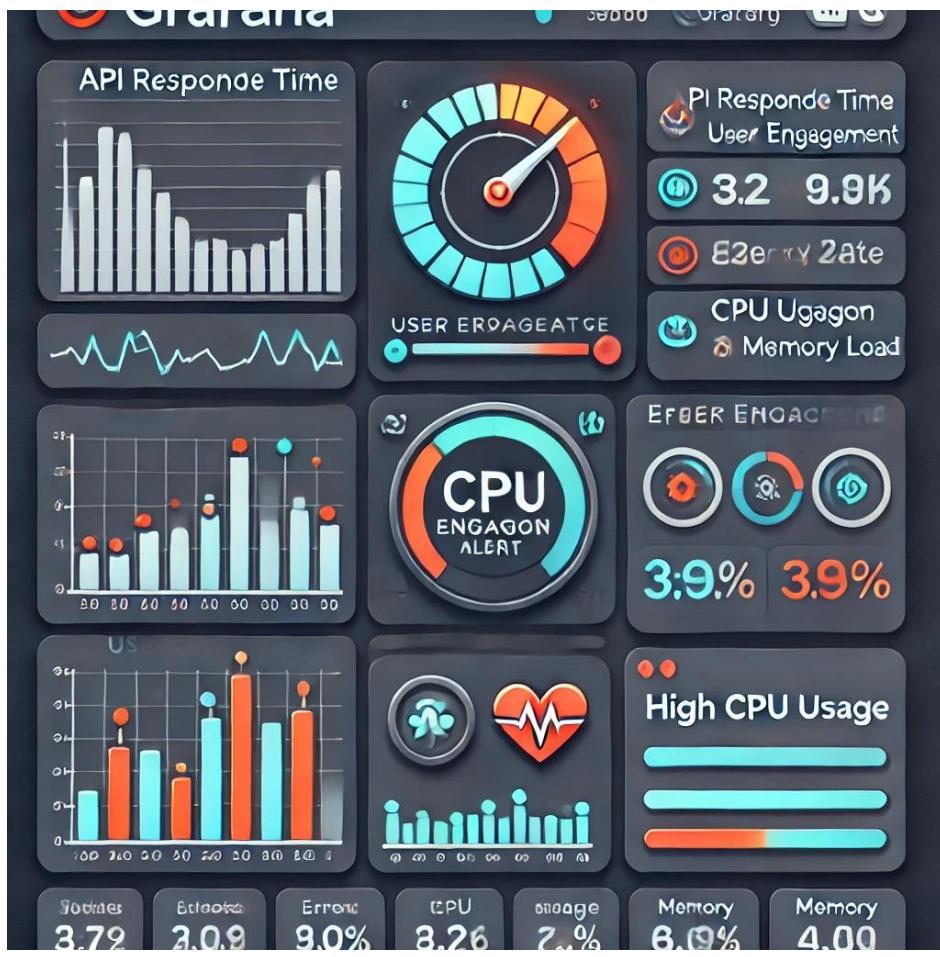


Figure 4: Example Grafana Dashboard for Monitoring and Alerting

- CloudWatch: Amazon's monitoring and observability service, providing detailed metrics on AWS resources.

Metrics Collected

- CPU Usage: Monitors CPU load across servers to track resource availability.
- Memory Usage: Tracks RAM usage, alerting if the system is near capacity.
- Response Time: Measures the time taken for the system to respond to user requests.
- Error Rate: Tracks the frequency of errors over time, helping identify service health.
- Disk I/O: Monitors the read and write speed on the disk, crucial for I/O-intensive applications.

Alerting Strategy

- Alert Thresholds: Set thresholds based on business needs. For example, CPU usage above 85% for more than 5 minutes may trigger an alert.
- Notification Channels:
 - Slack: Integrate with monitoring tools to alert teams immediately.

- PagerDuty: Escalate critical alerts and notify relevant stakeholders for quick action.
- Escalation Paths:
 - Tier 1: Notify on-call engineer within 5 minutes of an alert.
 - Tier 2: If unresolved within 15 minutes, escalate to the engineering lead.
 - Tier 3: For prolonged issues, notify executives after 1 hour.

5. Incident Management

Incident Response Process

- Detection: Incident detected via alert from monitoring tools.
- Analysis: Identify the root cause using logs and metrics.
- Resolution: Implement a fix, such as restarting services or rolling back updates.
- Communication: Update stakeholders with incident status and estimated resolution time.

Runbooks

- Server Outage: Guide includes steps to check server health, network connectivity, and recovery options.
- Latency Issues: Analyse logs, identify slow queries, and adjust resource allocations if necessary.
- Database Failures: Procedure for restoring from backups and minimizing data loss.

Post-Incident Review

- Timeline: Document key timestamps from detection to resolution.
- Root Cause Analysis: Identify underlying causes to prevent future incidents.
- Action Items: List improvements, such as system upgrades or additional alerts.

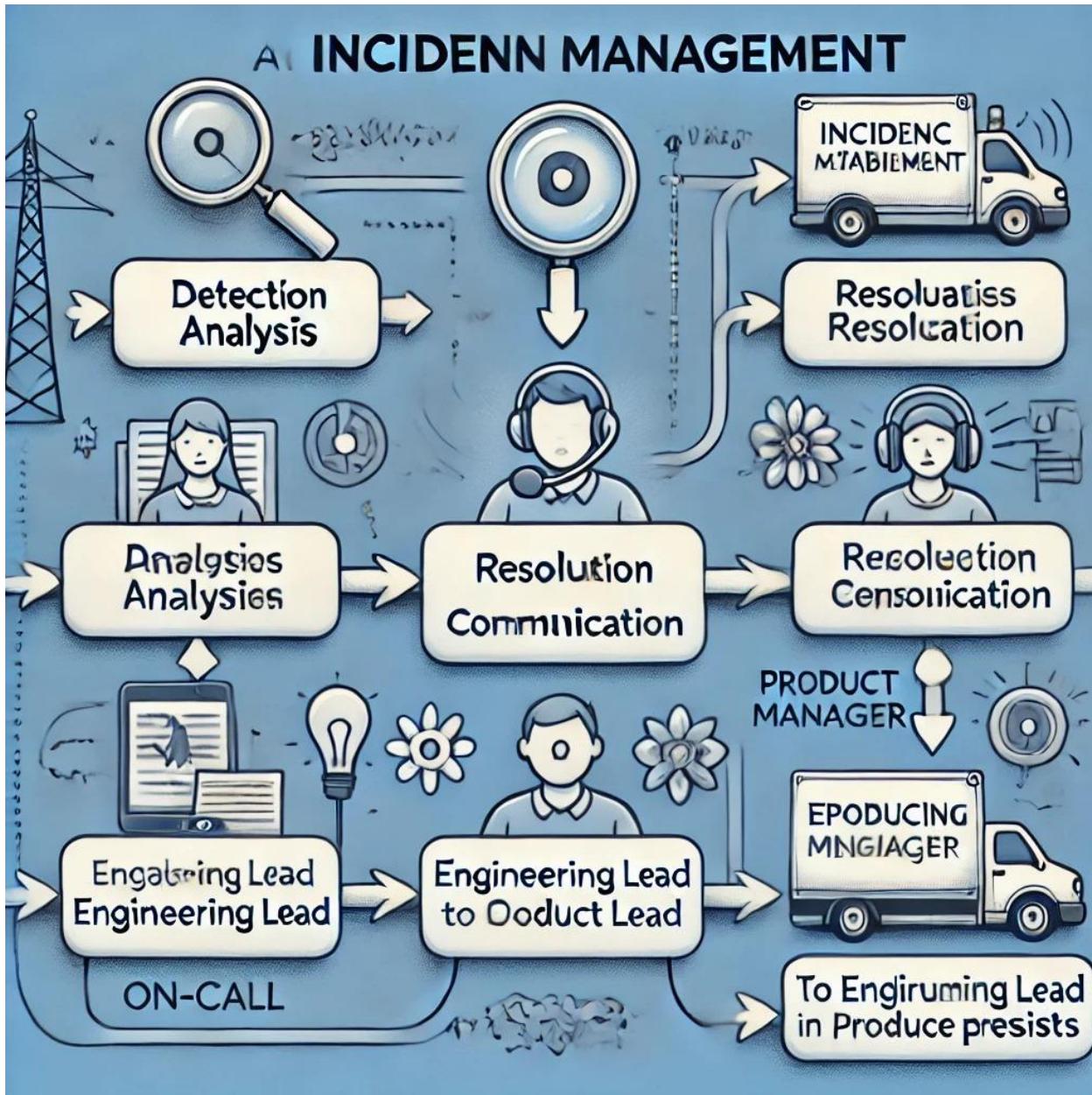


Figure 5: Incident Management Flowchart

6. Capacity Planning & Scaling

Resource Forecasting

- Usage Trends: Analyse historical traffic patterns to project future needs.
- Growth Factors: Account for business initiatives, seasonal trends, and user growth.
- Capacity Calculation: Determine CPU, memory, and storage needs based on projected growth.

Scaling Procedures

- Horizontal Scaling: Add more servers to distribute load, often using a load balancer.
 - Vertical Scaling: Increase resource capacity (e.g., CPU, memory) on an existing server.
 - Database Sharding: Split databases into smaller, more manageable pieces.

Auto-Scaling Configurations

- Auto-Scaling Groups: Define rules for automatically adding/removing instances based on demand.
 - Scaling Triggers: Set triggers like CPU usage ≥ 75% for 10 minutes to initiate scaling.
 - Load Balancer Configuration: Distribute traffic evenly across instances in the auto-scaling group

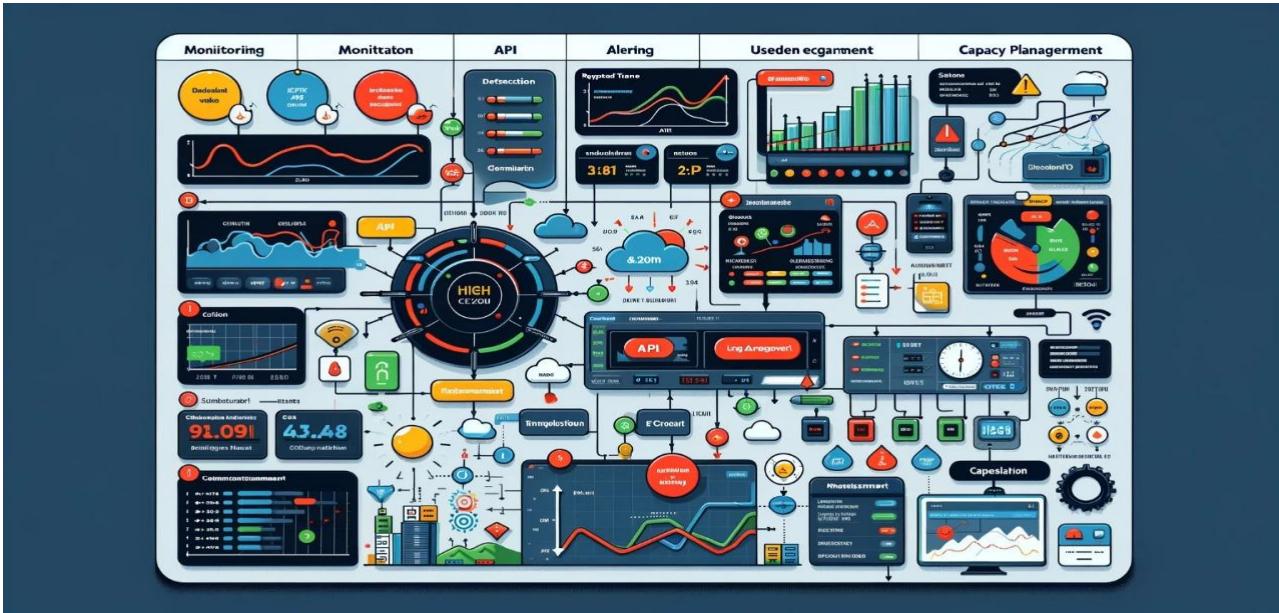


Figure 6: Capacity Planning and Scaling Flowchart

7. Change Management

Change management plays a crucial role in the lifecycle of a project, ensuring that updates and modifications are handled systematically to maintain stability and reduce risks. To achieve this, the **Mental Wellness & Habit Tracker App** follows a well-defined process for proposing, reviewing, and implementing changes while providing fallback options if issues arise.

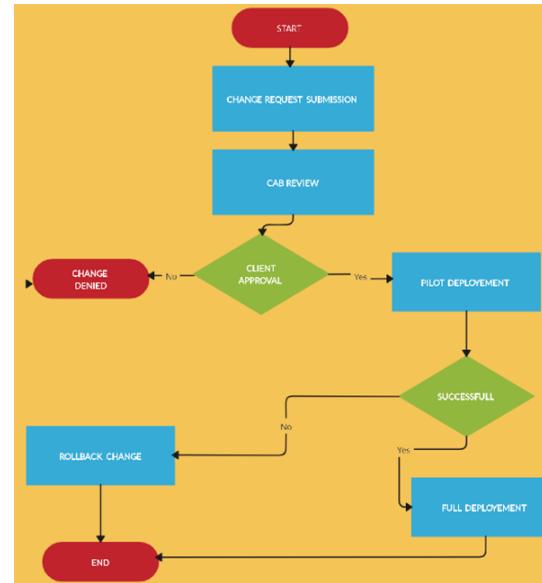
Change Policies:

Change policies outline the steps for proposing and approving modifications to the app's functionality, ensuring that all changes are carefully assessed and documented.

Proposing Changes: Changes can be proposed by any member of the development or operations team, or based on user feedback. Each change request must include a description of the modification, the rationale, potential risks, and the expected impact on system performance.

Review Process: Change proposals are reviewed by a change advisory board (CAB), which includes members from the development, product management, and IT teams. They assess the technical feasibility, potential impact on users, and alignment with business goals.

Approval: Only changes that pass the review process are approved. For critical updates, a pilot implementation may be conducted before full deployment.



Change Proposal Process Flowchart:

- This diagram outlines the flow of proposing, reviewing, and approving a change request.
- Diagram Elements:** Change Request Submission → CAB Review → Approval/Denial → Pilot Deployment → Full Deployment

Rollout Plans:



Rollout plans ensure that updates to the app are introduced in a controlled and predictable manner, minimizing disruption to users.

- Preparation:** Before rolling out any change, detailed testing is conducted in a staging environment that mirrors the production setup. This includes functional testing, performance testing, and user acceptance testing (UAT).
- Step-by-Step Deployment:** Rollouts are typically staged to reduce risk. For example:

- **Pilot Group:** Initial deployment to a small group of users.
- **Gradual Scaling:** If no issues are detected, the update is gradually deployed to the remaining users over a period of time.
- **Monitoring:** Post-deployment monitoring ensures that performance and user experience metrics remain within acceptable thresholds.
- **Communication:** Users are informed of upcoming changes via in-app notifications and emails. Downtime, if any, is scheduled during non-peak hours to minimize impact.

Rollout Plan Stages:

- This diagram shows the step-by-step stages involved in rolling out updates to the user base.
- **Diagram Elements:** Testing → Pilot Deployment → Full Rollout → Monitoring & Feedback

Rollback Procedures:

Despite careful planning, unforeseen issues can occur. Therefore, rollback procedures are essential to quickly revert changes and restore the previous stable state of the system.

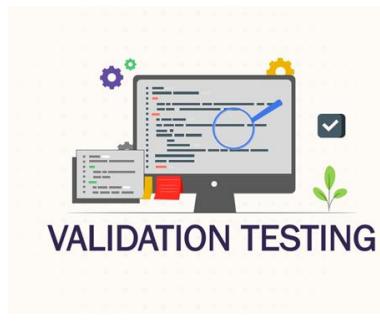
- **Rollback Triggers:** A rollback may be initiated if:
 - Significant performance degradation occurs.
 - Key functionality is impacted.
 - User complaints spike due to the new changes.
- **Reverting the Change:** A rollback process is triggered by the development team, where the last stable version of the app is redeployed. This includes reverting any database schema changes, clearing caches, and restoring previous configurations.
- **Post-Rollback Analysis:** Once the rollback is complete, the development team conducts a thorough investigation to identify the root cause of the issue and make necessary corrections before attempting to reintroduce the update.

Rollback Procedure Diagram:

- A flowchart illustrating the steps taken when rolling back a failed update.
- **Diagram Elements:** Rollout Failure → Revert Deployment → Database Restore → Monitoring Post-Rollback



8. Testing & Validation



Testing and validation are critical phases in the development lifecycle of the **Mental Wellness & Habit Tracker App** to ensure that the system meets its performance, security, and functional requirements. This section outlines the test scenarios, testing environments, and validation process used to maintain a stable and reliable system.

Test Scenarios:

The testing process covers a wide range of test cases to ensure that the app is thoroughly evaluated in different areas. Each test scenario is designed to identify potential issues early, ensuring smooth functionality when deployed in production.

- **Load Testing:** Evaluates how the app performs under heavy user traffic and high data loads. The goal is to identify the system's limits and ensure it can handle peak loads without performance degradation.
 - Test Case Example: Simulating 10,000 concurrent users logging mood data and receiving AI-generated advice.
- **Failover Testing:** Ensures the system can recover gracefully in case of server or network failures. This involves simulating server crashes or network disconnections to validate the app's ability to automatically switch to backup systems.
 - Test Case Example: Triggering a server failure to ensure automatic failover to secondary servers without disrupting user sessions.
- **Regression Testing:** Ensures that new features or updates do not negatively impact existing functionality. Automated regression tests are run after every update to verify that previous functionalities continue to work as expected.

- Test Case Example: Verifying that habit tracking and mood analysis features remain functional after a new AI module is introduced.
- **Security Testing:** Assesses vulnerabilities in the app's security, focusing on protecting user data and preventing unauthorized access.
 - Test Case Example: Testing for SQL injection, cross-site scripting (XSS), and encryption integrity during user data transfers.
- **User Acceptance Testing (UAT):** Involves real users testing the app to ensure that it meets their expectations in terms of usability and functionality.
 - Test Case Example: Gathering feedback from a group of beta users on the ease of use of mood tracking and AI advice features.

Testing Environments:

To ensure that all test scenarios are conducted in controlled environments, several testing stages are used. Each environment replicates key aspects of the app's production system, allowing for realistic testing conditions.

- **Development Environment:** This is where developers initially test their code using unit and integration tests. It is less stable than other environments and used primarily by the development team.
- **Staging Environment:** A replica of the production environment, staging is used for comprehensive testing of new features and bug fixes before they are released. This environment ensures that any issues are caught before deployment to the live system.
 - Key Features: Mirrors production infrastructure, allowing for load and failover testing.
- **Production Environment:** The final environment where the app is live and accessible to users. Testing here is limited to critical updates, and thorough validation is required before any changes are made.
 - Key Features: Used for limited smoke testing of minor updates and patches.



Validation Process:

The validation process ensures that the app is stable and reliable before any deployment to the production environment. It focuses on critical performance, security, and functionality criteria.

- **Performance Validation:** Ensures that the app meets its defined performance objectives, such as maintaining response times under 1.5 seconds for 90% of user actions and handling concurrent users without significant slowdowns.
 - Criteria: Successful load tests with no more than 5% performance degradation under peak loads.
- **Security Validation:** Validates that user data is encrypted and secure, and that the app complies with privacy regulations. Security tests confirm that vulnerabilities such as SQL injection or unauthorized access are not present.
 - Criteria: Zero critical security vulnerabilities in final testing.
- **Functional Validation:** Confirms that all user-facing features work as intended, including mood tracking, habit tracking, and AI-generated advice. Functional testing ensures that the system delivers a smooth and intuitive user experience.
 - Criteria: 100% pass rate on critical functionality tests, including regression testing for new features.
- **User Feedback Validation:** Ensures that feedback from beta users is incorporated into the final product, refining any features or addressing usability concerns.
 - Criteria: Positive feedback from UAT participants with no major usability issues reported.

The validation process concludes with a final review from the product manager and development team, after which the app is approved for production deployment.

Validation Process Flowchart:

- A detailed flowchart showcasing the steps involved in validating performance, security, functionality, and user feedback before deployment.
- **Diagram Elements:** Performance Testing → Security Validation → Functional Testing → User Feedback → Final Approval

9. Security & Compliance



In any app dealing with user data, especially a **Mental Wellness & Habit Tracker** that processes personal habits and mental health information, security and compliance are paramount. This section ensures that the

app is built and maintained with strong security policies, adheres to relevant compliance standards, and is regularly audited to ensure continued safety and legality.

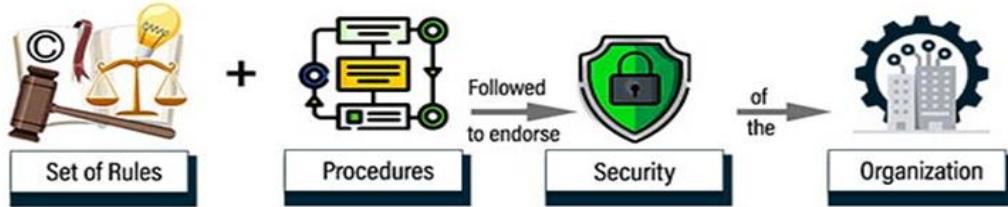
Security Policies:

Security policies define how user data is protected, how access to the system is controlled, and how the app stays compliant with industry standards for data security.

- **Data Protection Measures:**
 - **Encryption:** All user data, including habits, mood entries, and personal information, is encrypted both in transit and at rest. Encryption ensures that even if data is intercepted, it cannot be read without the decryption keys.
 - **Regular Backups:** Data is backed up hourly, and full backups are performed daily to protect against data loss.
 - **Firewall Protection:** Firewalls are set up to monitor and control incoming and outgoing network traffic based on predetermined security rules, preventing unauthorized access.
- **Access Control:**
 - **Role-Based Access Control (RBAC):** Access to system resources and data is restricted based on user roles. For example, users can access their personal data, but only the system administrators have access to the full database.

- **Multi-Factor Authentication (MFA):** System access, especially for admin and development teams, requires multi-factor authentication, adding an extra layer of protection beyond passwords.
- **Incident Response Plan:**
 - In case of a security breach, an incident response plan is in place, outlining the steps for identifying, containing, and resolving the issue while notifying affected users and authorities if needed.

Security Policies



- A flowchart detailing key security measures like data encryption, access control, regular backups, and firewall protection.
- **Diagram Elements:**
 - Central node: **Security Policies**
 - Branches: **Data Protection** (encryption, backups, firewalls), **Access Control** (RBAC, MFA), and **Incident Response**.

Compliance Standards:

Compliance ensures that the app adheres to industry standards and legal regulations to protect



user privacy and ensure data security.

- **General Data Protection Regulation (GDPR):**
 - The app complies with the GDPR, ensuring that users' personal data is collected, stored, and processed in a way that respects their privacy rights. This includes

providing clear consent options for data collection and allowing users to request deletion of their data.

- **SOC 2 (Service Organization Control 2):**
 - SOC 2 compliance demonstrates that the app is secure, available, and processes data confidentially and with integrity. This is crucial for the handling of sensitive mental health data, where trust and confidentiality are paramount.
- **HIPAA (Health Insurance Portability and Accountability Act) (if applicable):**



If the app processes health-related data, HIPAA compliance ensures that all personal health information (PHI) is handled in accordance with strict privacy and security standards, preventing unauthorized access to medical data.

Auditing Procedures:

Regular audits are critical to maintaining security and compliance. These procedures

ensure that the app's security policies are enforced and that the app remains compliant with industry standards.

- **Audit Schedule:**
 - **Quarterly Security Audits:** Every three months, a full security audit is conducted, reviewing access logs, encryption methods, firewall settings, and backups to ensure compliance with internal policies and external standards like GDPR and SOC 2.
 - **Annual Compliance Review:** A full compliance review is conducted annually to ensure continued adherence to GDPR, SOC 2, and other relevant standards.
- **Steps for Security and Compliance Audits:**
 - **Preparation:** Identify areas to be audited, gather all relevant documentation, and assign responsibilities to the auditing team.
 - **Execution:** Auditors review system configurations, access logs, and data storage practices, and run penetration tests to identify vulnerabilities.
 - **Reporting:** The results of the audit are compiled into a report, highlighting any vulnerabilities, non-compliance issues, or security gaps, along with recommendations for remediation.
 - **Remediation:** Any identified issues are addressed promptly, with follow-up audits to ensure that corrective actions are effective.

Auditing Procedures Timeline:



- A timeline or cycle diagram illustrating the schedule and steps for quarterly security audits and annual compliance reviews.
- **Diagram Elements:**
 - Key points: **Preparation, Execution, Reporting, and Remediation**, with arrows showing the recurring nature of audits.

Audit Process



techqualitypedia.com

10. Disaster Recovery & Backup

Recovery Plan:

In preparation for potential disruptions, the Mental Wellness & Habit Tracker App is equipped with a disaster recovery plan that prioritizes the continuity of service and data security. The plan addresses multiple types of possible events, from minor service disruptions to significant data loss, with clear protocols and response measures:

- **Defined Recovery Time Objective (RTO):**
For critical services—such as user data access and AI-driven recommendations—the RTO is set at 1 hour to minimize downtime. Supporting systems, such as analytics and non-core functions, have an RTO of up to 4 hours.
- **Defined Recovery Point Objective (RPO):**

To ensure data integrity, the maximum allowable data loss (RPO) is limited to 5 minutes. This objective is achieved through a combination of continuous data replication to a backup environment and frequent, incremental data snapshots.

The recovery plan is regularly tested through simulated outage scenarios to validate response times, ensuring the team and systems are fully prepared for rapid recovery in an actual emergency.



Backup Strategy:

The backup strategy aims to provide comprehensive protection against data loss while ensuring rapid recovery:

- **Scheduled Backups:**

Data is backed up every hour, with a full system backup conducted daily. This regular schedule ensures that users' progress, mood logs, and habit data are always protected.

- **Backup Redundancy & Storage Locations:**

Backups are securely stored in geographically diverse data centers, ensuring availability even in the event of regional disruptions. Backup data is encrypted in transit and at rest, adhering to stringent security standards.

- **Backup Retrieval Process:**

Should data recovery be needed, restoration processes are streamlined for efficiency. Automated scripts handle the re-importing of data, reducing human error and ensuring data accuracy. The goal is to restore backups with minimal user impact, allowing for seamless continuity in user experience.



Failover Mechanisms:

Failover mechanisms are implemented to ensure consistent application availability, even during server issues or outages:

- **Primary-Secondary Server Architecture:**

The system utilizes a primary-secondary server setup, with active-passive failover to automatically reroute traffic to the secondary server if the primary fails. This switch-over is nearly instantaneous, thanks to a constant health check system.

- **Health Monitoring & Activation Triggers:**

Server health is continuously monitored to detect potential issues early. Predefined activation triggers, such as latency spikes or resource exhaustion, prompt automatic

failover to backup servers. This redundancy ensures users can access the app without interruption, even during infrastructure issues.

11. Continuous Improvement & Optimization

Performance Tuning:

To provide an efficient and enjoyable user experience, the app's performance is continually optimized. This involves proactive tuning to reduce latency, improve response times, and enhance the app's reliability:

- **Database Indexing & Query Optimization:**

To ensure efficient access to frequently used data, the database is regularly optimized. Indexed fields and optimized query structures improve retrieval speeds for high-traffic features, like mood logging and habit analysis.

- **Load Balancing & Adaptive Scaling:**

A load-balancing system distributes user requests evenly across multiple servers, preventing any single server from becoming overloaded. This system works in tandem with auto-scaling configurations, which adapt server capacity to handle fluctuations in user demand. During peak times, additional servers are automatically brought online, while in quieter periods, resources are scaled down to optimize costs.

- **Memory and CPU Optimization:**

Routine analysis of memory and CPU usage helps identify resource-intensive functions, such as AI-driven recommendations. These functions are optimized through code refinements and memory management improvements to keep the app responsive.

Automation Opportunities:

Automation is key to improving operational reliability and efficiency across multiple aspects of app management:

- **Automated Backups & Failover Checks:**

Backup creation and verification processes are fully automated, ensuring that data protection occurs without needing manual intervention. Automated failover checks and tests are regularly conducted to ensure redundancy systems function seamlessly.

- **Continuous Integration & Continuous Deployment (CI/CD):**

An automated CI/CD pipeline ensures rapid and safe deployment of updates. This pipeline incorporates unit and integration testing, code reviews, and automated rollbacks for failed deployments, reducing the risk of downtime from new code.

- **Routine Health Checks & Self-Healing Scripts:**

Automated scripts regularly monitor the app's components, such as the AI engine, data sync functions, and server response times. If a service fails or underperforms, self-healing scripts attempt recovery actions, such as restarting services or clearing caches. These measures improve system resilience and reduce the burden on the IT team.



Feedback Loops:

A robust feedback system enables continuous improvement by capturing user and operational feedback to guide future updates:

- **User Feedback Mechanisms:**

Feedback is gathered via in-app surveys, direct feedback forms, and app store reviews. Using AI-driven sentiment analysis, this feedback is categorized, prioritized, and flagged for attention by the product and development teams, ensuring that user-reported issues are addressed promptly.

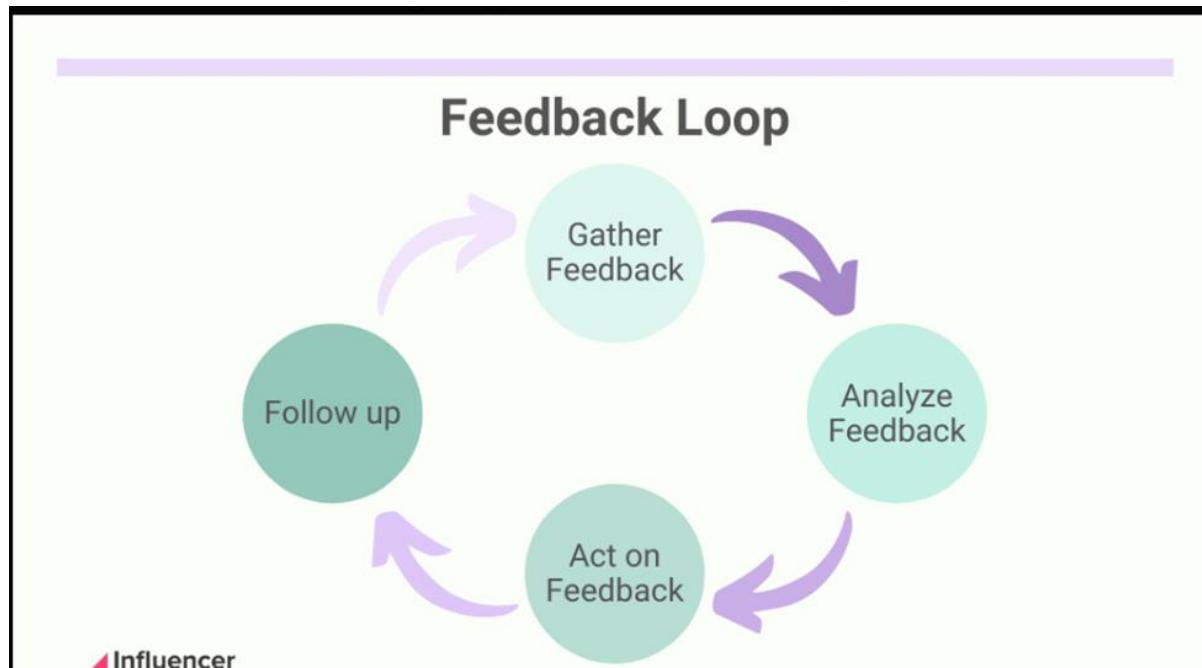
- **Operations & Development Feedback:**

Internal feedback from operations and development teams is collected during regular retrospectives and incident reviews. This feedback informs adjustments to system

monitoring, performance tuning, and operational workflows, helping to identify areas for enhancement in technical stability and user experience.

- **Iterative Improvement Cycles:**

Using agile methodology, the app undergoes continuous improvement cycles where the team incorporates feedback, releases feature updates, and addresses identified issues. Each cycle includes testing for quality assurance and stakeholder feedback sessions to ensure changes align with both user needs and business goals.



7. Vision Document

Software Requirement Engineering

Project Title: Mental Wellness & Habit Tracker App with AI Coach

Group Members:

Zunaira Abdul Aziz BSSE23058

Faiqa Arshad BSSE23028

Hamna Fatima BSSE23080

Areeba Shahbaz BSSE23097

VISION DOCUMENT

- 1. Introduction**
- 2. User Description**
- 3. Product Overview**
- 4. Feature Attributes**
- 5. Product Features**
- 6. Exemplary Use Cases**
- 7. Other Product Requirements**
- 8. Documentation Requirements**
- 9. Glossary**



01

Introduction

The Mental Wellness & Habit Tracker app is designed to help users monitor and improve their mental well-being and daily habits. By offering a secure, intuitive platform, this app encourages users to record mood entries, track habits, and receive personalized recommendations. With strong security policies and compliance with industry standards, the app aims to protect sensitive user data and build trust.

2. User Description

The primary users are individuals looking to enhance their mental wellness and develop positive daily routines. These users may include:

General Users: People seeking tools to track moods, habits, and receive insights into their mental health.

Mental Health Enthusiasts: Users focused on self-improvement through daily reflections and habit tracking.

Health and Wellness Professionals: Professionals may use the app as a supplementary tool for clients to monitor progress.

3. Product Overview

The Mental Wellness & Habit Tracker app serves as a supportive resource for mental health by enabling users to record personal data, track progress, and gain insights. It prioritizes data security, ease of use, and reliability, integrating advanced encryption, secure backups, and compliance with GDPR, SOC 2, and HIPAA standards to safeguard sensitive data.

4. Feature Attributes

The app's core attributes focus on privacy, reliability, and user engagement:

Security-Centric Design: Encryption, backups, and access control ensure robust data protection.

Compliance & Regular Audits: The app meets strict standards like GDPR, SOC 2, and HIPAA to protect user data privacy.

Usability & Accessibility: Designed for an intuitive user experience across multiple devices, making mental wellness tracking accessible for all users.

5. Product Features

Data Encryption: Ensures user data is encrypted both in transit and at rest.

Role-Based Access Control (RBAC): Limits access based on user roles, restricting sensitive data to authorized personnel only.

Multi-Factor Authentication (MFA): Adds an extra security layer for administrators and developers.

Incident Response Plan: An established plan for breach handling to protect and notify users promptly.

GDPR, SOC 2, and HIPAA Compliance: Adheres to these standards to ensure data privacy and security.

Automated Backups and Redundancy: Data backups are taken every hour and stored in diverse locations for recovery in case of data loss.

Performance Optimization: Features like load balancing, database indexing, and adaptive scaling to enhance app responsiveness.

Feedback Loops and CI/CD Pipelines: Enables ongoing improvement with user feedback, automated testing, and regular updates.

6. Exemplary Use Cases

Mood Tracking and Insights Generation: Users can log their moods and receive AI-driven insights into their emotional patterns.

Habit Tracking: Users create and monitor habits to achieve personal wellness goals.

Data Recovery: In case of a system failure, the app's disaster recovery plan ensures rapid data restoration.

Audit Preparation and Reporting: Scheduled audits help maintain regulatory compliance and reassure users about data security.

7. Other Product Requirements

Security Measures: The app implements firewalls, RBAC, encryption, and multi-factor authentication.

Compliance Standards: Regular reviews for GDPR, SOC 2, and HIPAA adherence.

Disaster Recovery: Backup redundancy and failover mechanisms to handle disruptions without data loss.

Continuous Improvement: Optimization of database, server load balancing, and self-healing scripts to ensure uptime and performance.

8. Documentation Requirements

The documentation includes user manuals, developer guides, security policy outlines, and compliance checklists. Key documents:

User Guide: Step-by-step instructions for app usage.

Security & Compliance Guidelines: Details on data security policies, incident response, and compliance.

System Audit Reports: Records of quarterly security audits and annual compliance reviews.

9. Glossary

Encryption: A method to secure data by converting it into a code to prevent unauthorized access.

GDPR (General Data Protection Regulation): European Union regulation for data privacy.

SOC 2 (Service Organization Control 2): A standard for managing customer data based on five “trust service principles” (security, availability, processing integrity, confidentiality, privacy).

HIPAA (Health Insurance Portability and Accountability Act): U.S. regulation for data privacy and security of medical information.

RBAC (Role-Based Access Control): Security protocol that restricts access based on user roles.

MFA (Multi-Factor Authentication): Security process requiring multiple forms of identification before granting access.

RTO (Recovery Time Objective): The target time set for the recovery of IT and business activities after a disaster.

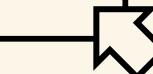
RPO (Recovery Point Objective): The maximum tolerable period in which data might be lost in case of a disruption.

8. StoryBoard Creation

Mental Wellness & Habit Tracker

App - User Storyboard

User Interaction Flow





>>>

Table of contents

.....

01

Scene 1:
Login/Sign-
up Screen

02

Scene 2:
Home
Dashboard

03

Scene 3:
Track Mood

04

Scene 4:
Track Habit

>>>

Table of contents

.....

05 **Scene 5:**
AI Coach
Interaction

06 **Scene 6:**
Progress
Analytics

07 **Scene 7:**
Notifications/
Reminders

08 **Scene 8:** Final
Thoughts
and Closing

01

.....**Scene 1:**

Login/Sign-up

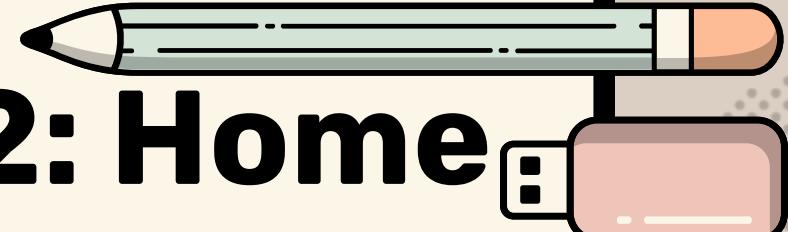
Screen



- User opens the app.
- Logs in or signs up.
- Option to reset password.

02

Scene 2: Home Dashboard



- Displays user's mood and habit progress.
- Access buttons for mood, habits, and AI coach.

03

.....

>>>



Scene 3: Track Mood



- User logs mood.
- Select from mood options (happy, sad, neutral).
- Option to add comments.

04

.....

>>>>



Scene 4: Track Habit



- User checks off completed habits.
- Option to add new habits.
- Displays habit progress.

05

Scene 5: AI Coach

Interaction

- User interacts with AI Coach for personalized advice.
- AI gives suggestions based on habits and mood.



06

.....Scene 6:
|||||

Progress Analytics



- User views mood and habit progress over time.
 - Displays graphs and trends.

07

.....Scene 7:

Notifications/R
eminders

eminders



- User receives push notifications for habit reminders.
- Option to open app from notification.

Mental Wellness & Habit Tracker App with AI Coach

by Zunaira Abdul Aziz

Submission date: 11-Nov-2024 02:51PM (UTC+0500)

Submission ID: 2515730674

File name: SRE_Assignment_3_1_1.pdf (1.52M)

Word count: 4022

Character count: 23934



**SRE
ASSIGNMENT -3
MENTAL
WELLNESS &
HABIT TRACKER
APP WITH AI
COACH**



GROUP 9

By

Zunaira Abdul Aziz BSSE23058

Faiqa Arshad BSSE23028

Hamna Fatima BSSE23080

Areeba Shahbaz BSSE23097



1. Project Overview

Project Title: Mental Wellness & Habit Tracker App with AI Coach

Date: November 9, 2024

Document Version: 2.0

Authors:

Zunaira Abdul Aziz BSSE23058 (Section 1-3)

Faiqa Arshad BSSE23028 (Section 4-6)

Hamna Fatima BSSE23080 (Section 7-9)

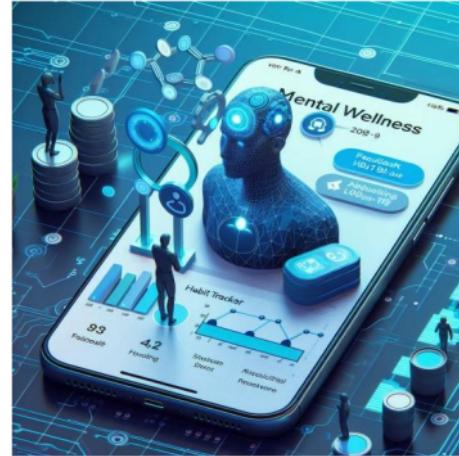
Areeba Shahbaz BSSE23097 (Section 10-11)

Purpose:

This project's goal is to build an app that helps people improve their mental well-being by tracking habits, recording moods, and getting personalized advice from an AI coach. The app aims to support users in developing healthy routines and understanding their emotions, with easy-to-use tools for better mental wellness.

Background:

With stress and mental health challenges becoming more common, people need ways to track and improve their mental wellness. Traditional habit and mood tracking can feel impersonal. The Mental Wellness & Habit Tracker App will solve this by adding an AI coach that offers helpful advice based on users' data, making the process more personalized and effective. By analysing patterns in users' habits and moods, the AI coach can suggest tailored routines and provide motivational support. This creates a more interactive and supportive experience, helping users stay consistent in their wellness journey.



Stakeholders:

- **Users:** The People who are using the app, and who wants to track personal habits and moods to enhance their mental health, as well as being helped by an AI coach.

- **Development Team:** The engineers, designers, and testers who build the app and make sure it works well and is easy to use.



Product Manager: It includes a person who is in charge of the project, ensuring the application fulfills the user requirement, and directing the development team for the similar purpose.

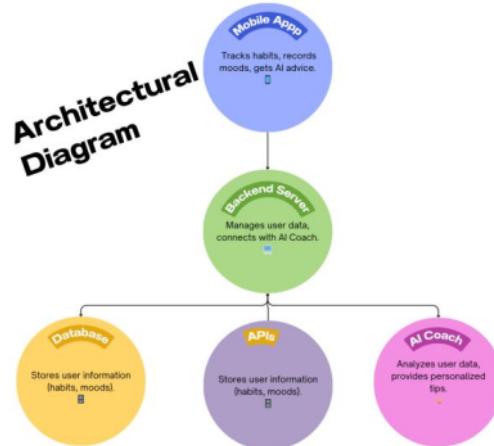
Mental Health Experts: Provide advice on how to design the app's features to be helpful and safe for users' mental health.

IT Support Team: Takes care of the app's servers, data storage, and security to make sure the app runs smoothly and safely.

Marketing Team: Promotes the app to users and gathers feedback to improve the app and attract more people.

2. System Architecture

Architecture Diagram:



Components:

- **Mobile App:** The main app where users track habits, record moods, and get AI advice.
- **Backend Server:** Stores and manages user data and connects with the AI coach.
- **AI Coach:** Analyzes data and provides personalized tips and reminders.
- **Database:** Saves all user information, like habits and mood entries.

- **APIs:** Connects the app to other services, like notifications or weather data.

Dependencies:

External Dependencies:

- **Third-party APIs:** The app may rely on external services like weather data or push notification services. If these services are down, it could affect app functionality.
- **App Stores:** The app's availability and updates depend on approval from app stores (Google Play, Apple App Store).

Internal Dependencies:

- **Backend Server:** The app depends on a reliable backend to store user data and process AI responses. If the server goes down, the app will not function properly.
- **AI Engine:** The app's ability to give personalized feedback depends on the performance of the AI system. If the AI engine has issues, it could affect the quality of the user experience.

19

3. Service Level Objectives (SLOs) & Indicators (SLIs)

Service Level Indicators (SLIs):

- **Response Time:** How quickly the app responds when the AI coach sends advice.
- **Error Rate:** How often there are issues with tracking or receiving AI tips.
- **Availability:** How often the app is up and running, so users can access it anytime.
- **Uptime:** The percentage of time the app is running without disruptions.
- **Data Sync Speed:** How quickly user data syncs across devices or between the app and server.
- **Request Processing Time:** Time taken to process user inputs or queries by the AI engine.

Service Level Objectives (SLOs):

- **Response Time:** The app should respond in under **1.5 seconds** for 90% of user actions.
- **Error Rate:** The error rate should be less than **0.5%** for all user actions.
- **Uptime:** The app should have at least **99.95% uptime**.
- **Data Sync Speed:** Syncing data should be completed in **less than 5 seconds**.
- **Request Processing Time:** AI responses should be processed in **under 3 seconds** for 80% of queries.

Service Level Agreements (SLAs):

- **App Availability:** The app will be available **99.95%** of the time, with a maximum downtime of **4 hours per year**.
- **Support Response:** Customer support will respond to reported issues within **12 hours**.

10

- **Issue Resolution:** Critical issues will be resolved within **24 hours**, while non-urgent issues will be addressed within **3 business days**.
- **Data Privacy:** The app will ensure user data is kept private and secure, following strict industry standards and legal requirements.
- **Updates & Maintenance:** Regular updates and maintenance will occur at least once every **three months**.

4. Monitoring & Alerting

14

Monitoring Tools

- Prometheus: Open-source systems monitoring and alerting toolkit, often paired with Grafana for visualization.
- Grafana: Visualization tool used to display time-series data from Prometheus and other sources.

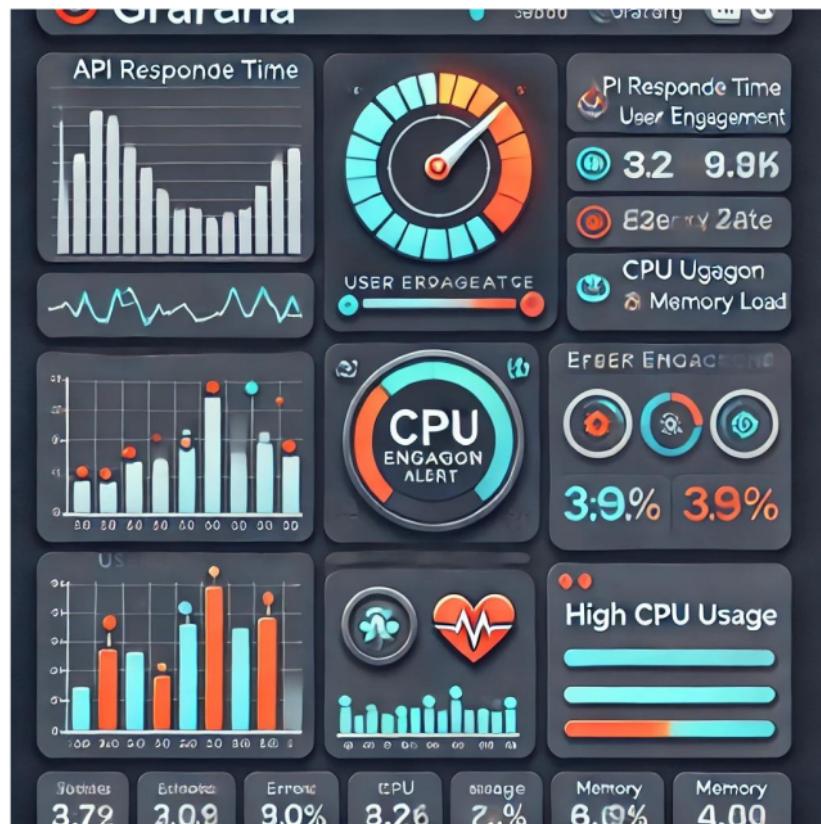


Figure 4: Example Grafana Dashboard for Monitoring and Alerting

- CloudWatch: Amazon's monitoring and observability service, providing detailed metrics on AWS resources.

Metrics Collected

- CPU Usage: Monitors CPU load across servers to track resource availability.
- Memory Usage: Tracks RAM usage, alerting if the system is near capacity.
- Response Time: Measures the time taken for the system to respond to user requests.¹³
- Error Rate: Tracks the frequency of errors over time, helping identify service health.
- Disk I/O: Monitors the read and write speed on the disk, crucial for I/O-intensive applications.

Alerting Strategy

- Alert Thresholds: Set thresholds based on business needs. For example, CPU usage above 85% for more than 5 minutes may trigger an alert.
- Notification Channels:
 - Slack: Integrate with monitoring tools to alert teams immediately.
 - PagerDuty: Escalate critical alerts and notify relevant stakeholders for quick action.
- Escalation Paths:
 - Tier 1: Notify on-call engineer within 5 minutes of an alert.
 - Tier 2: If unresolved within 15 minutes, escalate to the engineering lead.
 - Tier 3: For prolonged issues, notify executives after 1 hour.

5. Incident Management

Incident Response Process

- Detection: Incident detected via alert from monitoring tools.
- Analysis: Identify the root cause using logs and metrics.
- Resolution: Implement a fix, such as restarting services or rolling back updates.
- Communication: Update stakeholders with incident status and estimated resolution time.

Runbooks

- Server Outage: Guide includes steps to check server health, network connectivity, and recovery options.
- Latency Issues: Analyse logs, identify slow queries, and adjust resource allocations if necessary.
- Database Failures: Procedure for restoring from backups and minimizing data loss.

Post-Incident Review

- Timeline: Document key timestamps from detection to resolution.
- Root Cause Analysis: Identify underlying causes to prevent future incidents.
- Action Items: List improvements, such as system upgrades or additional alerts.

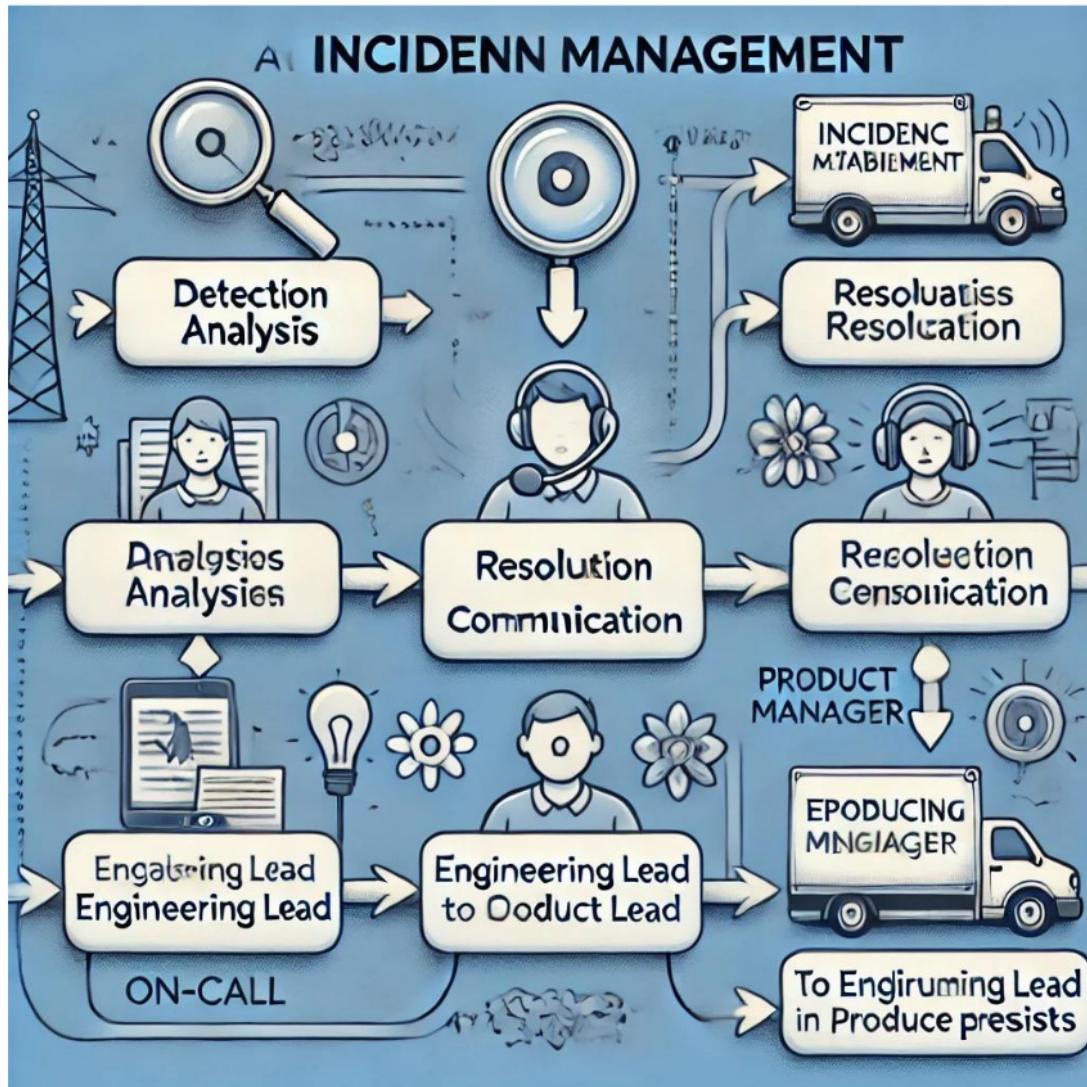


Figure 5: Incident Management Flowchart

6. Capacity Planning & Scaling

Resource Forecasting

- Usage Trends: Analyse historical traffic patterns to project future needs.
- Growth Factors: Account for business initiatives, seasonal trends, and user growth.
- Capacity Calculation: Determine CPU, memory, and storage needs based on projected growth.

Scaling Procedures

- Horizontal Scaling: Add more servers to distribute load, often using a load balancer.
- Vertical Scaling: Increase resource capacity (e.g., CPU, memory) on an existing server.
- Database Sharding: Split databases into smaller, more manageable pieces.

Auto-Scaling Configurations

- Auto-Scaling Groups: Define rules for automatically adding/removing instances based on demand.
- Scaling Triggers: Set triggers like CPU usage $\geq 75\%$ for 10 minutes to initiate scaling.
- Load Balancer Configuration: Distribute traffic evenly across instances in the auto-scaling group

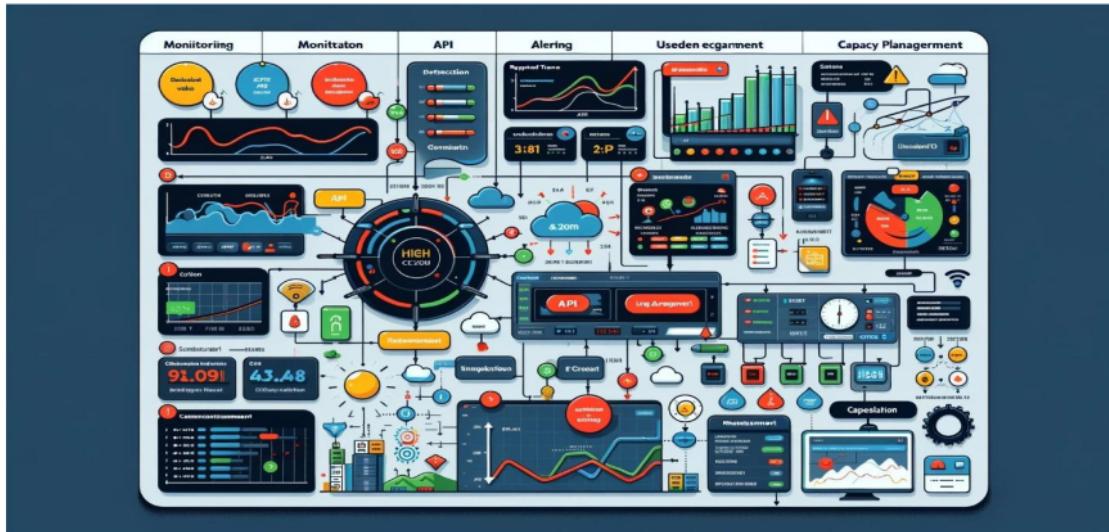


Figure 6: Capacity Planning and Scaling Flowchart

7. Change Management

18

Change management plays a crucial role in the lifecycle of a project, ensuring that updates and modifications are handled systematically to maintain stability and reduce risks. To achieve this, the **Mental Wellness & Habit Tracker App** follows a well-defined process for proposing, reviewing, and implementing changes while providing fallback options if issues arise.

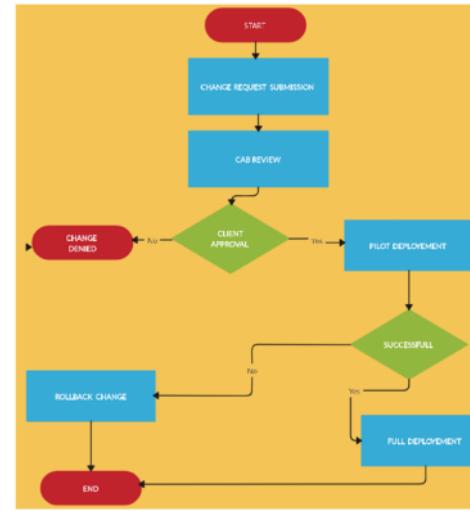
Change Policies:

Change policies outline the steps for proposing and approving modifications to the app's functionality, ensuring that all changes are carefully assessed and documented.

Proposing Changes: Changes can be proposed by any member of the development or operations team, or based on user feedback. Each change request must include a description of the modification, the rationale, potential risks, and the expected impact on system performance. **Review**

Process: Change proposals are reviewed by a change advisory board (CAB), which includes members from the development, product management, and IT teams. They assess the technical feasibility, potential impact on users, and alignment with business goals.

Approval: Only changes that pass the review process are approved. For critical updates, a pilot implementation may be conducted before full deployment.



Change Proposal Process Flowchart:

- This diagram outlines the flow of proposing, reviewing, and approving a change request.
- **Diagram Elements:** Change Request Submission → CAB Review → Approval/Denial → Pilot Deployment → Full Deployment

Rollout Plans:



Rollout plans ensure that updates to the app are introduced in a controlled and predictable manner, minimizing disruption to users.

- **Preparation:** Before rolling out any change, detailed testing is conducted in a staging environment that mirrors the production setup. This includes functional testing, performance testing, and user acceptance testing (UAT).
- **Step-by-Step Deployment:** Rollouts are typically staged to reduce risk. For example:
 - **Pilot Group:** Initial deployment to a small group of users.
 - **Gradual Scaling:** If no issues are detected, the update is gradually deployed to the remaining users over a period of time.
 - **Monitoring:** Post-deployment monitoring ensures that performance and user experience metrics remain within acceptable thresholds.
- **Communication:** Users are informed of upcoming changes via in-app notifications and emails. Downtime, if any, is scheduled during non-peak hours to minimize impact.

Rollout Plan Stages:

- This diagram shows the step-by-step stages involved in rolling out updates to the user base.
- Diagram Elements: Testing → Pilot Deployment → Full Rollout → Monitoring & Feedback

Rollback Procedures:

Despite careful planning, unforeseen issues can occur. Therefore, rollback procedures are essential to quickly revert changes and restore the previous stable state of the system.

- **Rollback Triggers:** A rollback may be initiated if:
 - Significant performance degradation occurs.
 - Key functionality is impacted.
 - User complaints spike due to the new changes.

- **Reverting the Change:** A rollback process is triggered by the development team, where the last stable version of the app is redeployed. This includes reverting any database schema changes, clearing caches, and restoring previous configurations.
- **Post-Rollback Analysis:** Once the rollback is complete, the development team conducts a thorough investigation to identify the root cause of the issue and make necessary corrections before attempting to reintroduce the update.

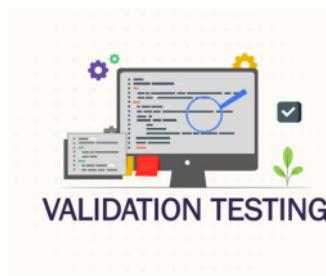
9

Rollback Procedure Diagram:

- A flowchart illustrating the steps taken when rolling back a failed update.
- **Diagram Elements:** Rollout Failure → Revert Deployment → Database Restore → Monitoring Post-Rollback



8. Testing & Validation



Testing and validation are critical phases in the development lifecycle of the **Mental Wellness & Habit Tracker App** to ensure that the system meets its performance, security, and functional requirements. This section outlines the test scenarios, testing environments, and validation process used to maintain a stable and reliable system.

Test Scenarios:

The testing process covers a wide range of test cases to ensure that the app is thoroughly evaluated in different areas. Each test scenario is designed to identify potential issues early, ensuring smooth functionality when deployed in production.

- **Load Testing:** Evaluates how the app performs under heavy user traffic and high data loads. The goal is to identify the system's limits and ensure it can handle peak loads without performance degradation.

1

- Test Case Example: Simulating 10,000 concurrent users logging mood data and receiving AI-generated advice.
- **Failover Testing:** Ensures the system can recover gracefully in case of server or network failures. This involves simulating server crashes or network disconnections to validate the app's ability to automatically switch to backup systems.
 - Test Case Example: Triggering a server failure to ensure automatic failover to secondary servers without disrupting user sessions.
- **Regression Testing:** Ensures that new features or updates do not negatively impact existing functionality. Automated regression tests are run after every update to verify that previous functionalities continue to work as expected.
 - Test Case Example: Verifying that habit tracking and mood analysis features remain functional after a new AI module is introduced.
- **Security Testing:** Assesses vulnerabilities in the app's security, focusing on protecting user data and preventing unauthorized access.
 - Test Case Example: Testing for SQL injection, cross-site scripting (XSS), and encryption integrity during user data transfers.
- **User Acceptance Testing (UAT):** Involves real users testing the app to ensure that it meets their expectations in terms of usability and functionality.
 - Test Case Example: Gathering feedback from a group of beta users on the ease of use of mood tracking and AI advice features.

Testing Environments:

To ensure that all test scenarios are conducted in controlled environments, several testing stages are used. Each environment replicates key aspects of the app's production system, allowing for realistic testing conditions.

- **Development Environment:** This is where developers initially test their code using unit and integration tests. It is less stable than other environments and used primarily by the development team.
- **Staging Environment:** A replica of the production environment, staging is used for comprehensive testing of new features and bug fixes before they are released. This environment ensures that any issues are caught before deployment to the live system.
 - Key Features: Mirrors production infrastructure, allowing for load and failover testing.
- **Production Environment:** The final environment where the app is live and accessible to users. Testing here is limited to critical updates, and thorough validation is required before any changes are made.
 - Key Features: Used for limited smoke testing of minor updates and patches.



Validation Process:

The validation process ensures that the app is stable and reliable before any deployment to the production environment. It focuses on critical performance, security, and functionality criteria.

- **Performance Validation:** Ensures that the app meets its defined performance objectives, such as maintaining response times under 1.5 seconds for 90% of user actions and handling concurrent users without significant slowdowns.
 - Criteria: Successful load tests with no more than 5% performance degradation under peak loads.
- **Security Validation:** Validates that user data is encrypted and secure, and that the app complies with privacy regulations. Security tests confirm that vulnerabilities such as SQL injection or unauthorized access are not present.
 - Criteria: Zero critical security vulnerabilities in final testing.
- **Functional Validation:** Confirms that all user-facing features work as intended, including mood tracking, habit tracking, and AI-generated advice. Functional testing ensures that the system delivers a smooth and intuitive user experience.
 - Criteria: 100% pass rate on critical functionality tests, including regression testing for new features.
- **User Feedback Validation:** Ensures that feedback from beta users is incorporated into the final product, refining any features or addressing usability concerns.
 - Criteria: Positive feedback from UAT participants with no major usability issues reported.

The validation process concludes with a final review from the product manager and development team, after which the app is approved for production deployment.

Validation Process Flowchart:

- A detailed flowchart showcasing the steps involved in validating performance, security, functionality, and user feedback before deployment.
- **Diagram Elements:** Performance Testing → Security Validation → Functional Testing → User Feedback → Final Approval

9. Security & Compliance



In any app dealing with user data, especially a **Mental Wellness & Habit Tracker** that processes personal habits and mental health information, security and compliance are paramount.

This section ensures that the

app is built and maintained with strong security policies, adheres to relevant compliance standards, and is regularly audited to ensure continued safety and legality.

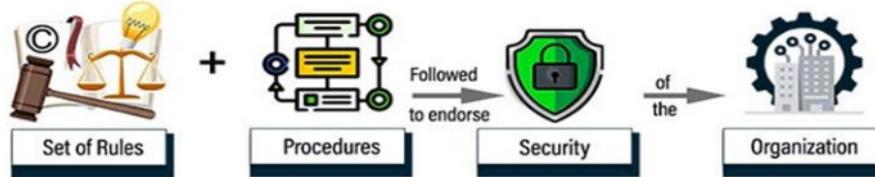
Security Policies:

Security policies define how user data is protected, how access to the system is controlled, and how the app stays compliant with industry standards for data security.

- **Data Protection Measures:**
 - **Encryption:** All user data, including habits, mood entries, and personal information, is encrypted both in transit and at rest. Encryption ensures that even if data is intercepted, it cannot be read without the decryption keys.
 - **Regular Backups:** Data is backed up hourly, and full backups are performed daily to protect against data loss.
 - **Firewall Protection:** Firewalls are set up to monitor and control incoming and outgoing network traffic based on predetermined security rules, preventing unauthorized access.
- **Access Control:**
 - **Role-Based Access Control (RBAC):** Access to system resources and data is restricted based on user roles. For example, users can access their personal data, but only the system administrators have access to the full database.

- **Multi-Factor Authentication (MFA)**: System access, especially for admin and development teams, requires multi-factor authentication, adding an extra layer of protection beyond passwords.
- **Incident Response Plan**:
 - In case of a security breach, an incident response plan is in place, outlining the steps for identifying, containing, and resolving the issue while notifying affected users and authorities if needed.

Security Policies



- A flowchart detailing key security measures like data encryption, access control, regular backups, and firewall protection.
- **Diagram Elements**:
 - Central node: **Security Policies**
 - Branches: **Data Protection** (encryption, backups, firewalls), **Access Control** (RBAC, MFA), and **Incident Response**.

Compliance Standards:

3 Compliance ensures that the app adheres to industry standards and legal regulations to protect 20

user privacy and ensure data security.



- **General Data Protection Regulation (GDPR)**:
 - The app complies with the GDPR, ensuring that users' personal data is collected, stored, and processed in a way that respects their privacy rights. This includes

providing clear consent options for data collection and allowing users to request deletion of their data.

- **SOC 2 (Service Organization Control 2):**

- SOC 2 compliance demonstrates that the app is secure, available, and processes data confidentially and with integrity. This is crucial for the handling of sensitive mental health data, where trust and confidentiality are paramount.

- **HIPAA (Health Insurance Portability and Accountability Act) (if applicable):**



15

If the app processes health-related data, HIPAA compliance ensures that all personal health information (PHI) is handled in accordance with strict privacy and security standards, preventing unauthorized access to medical data.

Auditing Procedures:

Regular audits are critical to maintaining security and compliance. These procedures

ensure that the app's security policies are enforced and that the app remains compliant with industry standards.

- **Audit Schedule:**

- **Quarterly Security Audits:** Every three months, a full security audit is conducted, reviewing access logs, encryption methods, firewall settings, and backups to ensure compliance with internal policies and external standards like GDPR and SOC 2.
- **Annual Compliance Review:** A full compliance review is conducted annually to ensure continued adherence to GDPR, SOC 2, and other relevant standards.

- **Steps for Security and Compliance Audits:**

- **Preparation:** Identify areas to be audited, gather all relevant documentation, and assign responsibilities to the auditing team.
- **Execution:** Auditors review system configurations, access logs, and data storage practices, and run penetration tests to identify vulnerabilities.
- **Reporting:** The results of the audit are compiled into a report, highlighting any vulnerabilities, non-compliance issues, or security gaps, along with recommendations for remediation.
- **Remediation:** Any identified issues are addressed promptly, with follow-up audits to ensure that corrective actions are effective.

Auditing Procedures Timeline:



- A timeline or cycle diagram illustrating the schedule and steps for quarterly security audits and annual compliance reviews.

- **Diagram Elements:**

- Key points: **Preparation, Execution, Reporting, and Remediation**, with arrows showing the recurring nature of audits.

10. Disaster Recovery & Backup

Recovery Plan:

In preparation for potential disruptions, the Mental Wellness & Habit Tracker App is equipped with a disaster recovery plan that prioritizes the continuity of service and data security. The plan addresses multiple types of possible events, from minor service disruptions to significant data loss, with clear protocols and response measures:

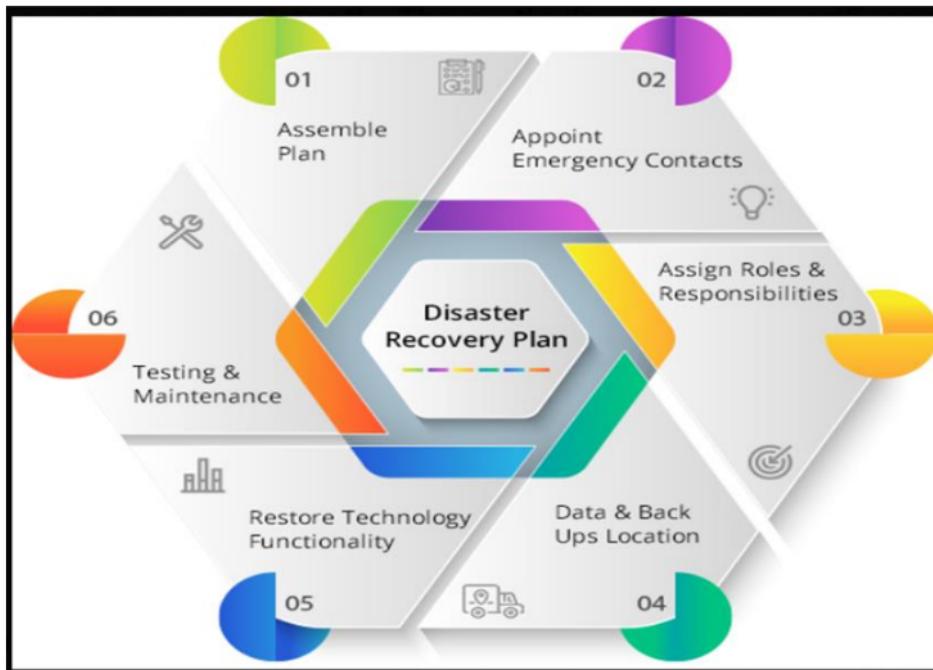
- **Defined Recovery Time Objective (RTO):**

For critical services—such as user data access and AI-driven recommendations—the RTO is set at 1 hour to minimize downtime. Supporting systems, such as analytics and non-core functions, have an RTO of up to 4 hours.

- **Defined Recovery Point Objective (RPO):**

To ensure data integrity, the maximum allowable data loss (RPO) is limited to 5 minutes. This objective is achieved through a combination of continuous data replication to a backup environment and frequent, incremental data snapshots.

The recovery plan is regularly tested through simulated outage scenarios to validate response times, ensuring the team and systems are fully prepared for rapid recovery in an actual emergency.



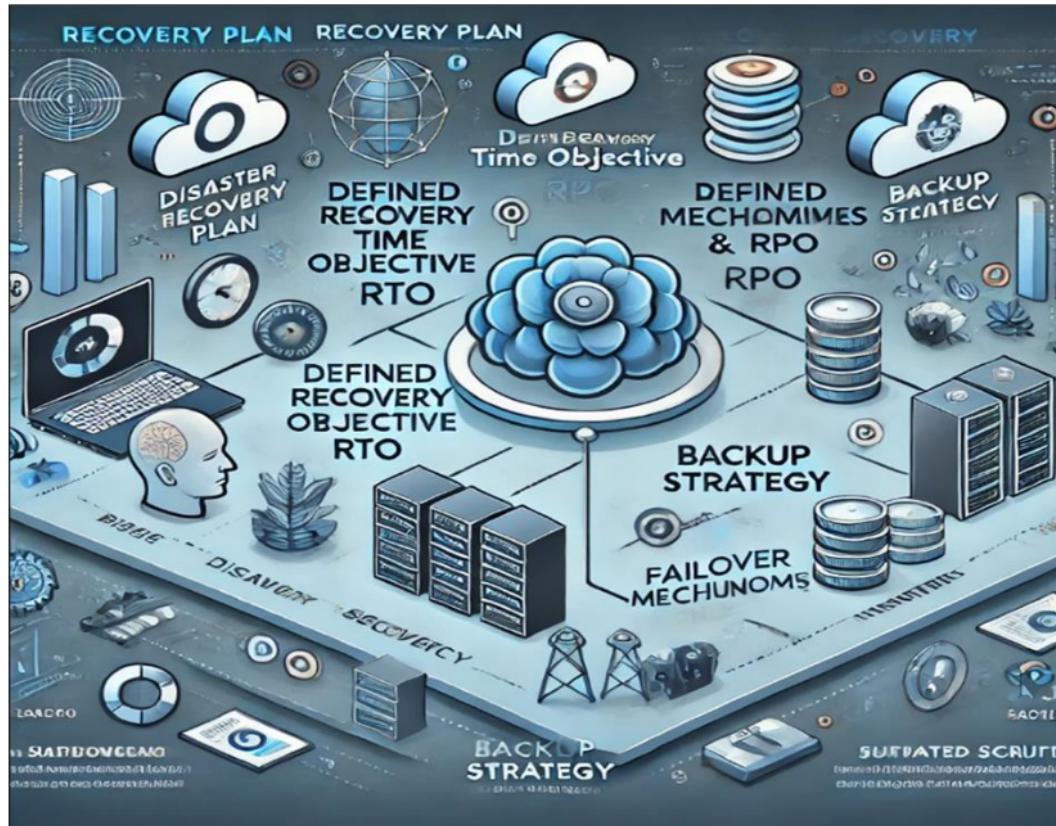
Backup Strategy:

The backup strategy aims to provide comprehensive protection against data loss while ensuring rapid recovery:

- **Scheduled Backups:**
Data is backed up every hour, with a full system backup conducted daily. This regular schedule ensures that users' progress, mood logs, and habit data are always protected.
- **Backup Redundancy & Storage Locations:**
Backups are securely stored in geographically diverse data centers, ensuring availability even in the event of regional disruptions. Backup data is encrypted in transit and at rest, adhering to stringent security standards.

- **Backup Retrieval Process:**

Should data recovery be needed, restoration processes are streamlined for efficiency. Automated scripts handle the re-importing of data, reducing human error and ensuring data accuracy. The goal is to restore backups with minimal user impact, allowing for seamless continuity in user experience.



Failover Mechanisms:

Failover mechanisms are implemented to ensure consistent application availability, even during server issues or outages:

- **Primary-Secondary Server Architecture:**

The system utilizes a primary-secondary server setup, with active-passive failover to automatically reroute traffic to the secondary server if the primary fails. This switch-over is nearly instantaneous, thanks to a constant health check system.

- **Health Monitoring & Activation Triggers:**

Server health is continuously monitored to detect potential issues early. Predefined activation triggers, such as latency spikes or resource exhaustion, prompt automatic

failover to backup servers. This redundancy ensures users can access the app without interruption, even during infrastructure issues.

11. Continuous Improvement & Optimization

Performance Tuning:

To provide an efficient and enjoyable user experience, the app's performance is continually optimized. This involves proactive tuning to reduce latency, improve response times, and enhance the app's reliability:

- **Database Indexing & Query Optimization:**

To ensure efficient access to frequently used data, the database is regularly optimized. Indexed fields and optimized query structures improve retrieval speeds for high-traffic features, like mood logging and habit analysis.

- **Load Balancing & Adaptive Scaling**¹¹

A load-balancing system distributes user requests evenly across multiple servers, preventing any single server from becoming overloaded. This system works in tandem with auto-scaling configurations, which adapt server capacity to handle fluctuations in user demand. During peak times, additional servers are automatically brought online, while in quieter periods, resources are scaled down to optimize costs.

- **Memory and CPU Optimization:**

Routine analysis of memory and CPU usage helps identify resource-intensive functions, such as AI-driven recommendations. These functions are optimized through code refinements and memory management improvements to keep the app responsive.

Automation Opportunities:

Automation is key to improving operational reliability and efficiency across multiple aspects of app management:

- **Automated Backups & Failover Checks:**

Backup creation and verification processes are fully automated, ensuring that data protection occurs without needing manual intervention. Automated failover checks and tests are regularly conducted to ensure redundancy systems function seamlessly.

- **Continuous Integration & Continuous Deployment (CI/CD):**

An automated CI/CD pipeline ensures rapid and safe deployment of updates. This pipeline incorporates unit and integration testing, code reviews, and automated rollbacks for failed deployments, reducing the risk of downtime from new code.

- **Routine Health Checks & Self-Healing Scripts:**

Automated scripts regularly monitor the app's components, such as the AI engine, data sync functions, and server response times. If a service fails or underperforms, self-healing scripts attempt recovery actions, such as restarting services or clearing caches. These measures improve system resilience and reduce the burden on the IT team.



Feedback Loops:

A robust feedback system enables continuous improvement by capturing user and operational feedback to guide future updates:

- **User Feedback Mechanisms:**

Feedback is gathered via in-app surveys, direct feedback forms, and app store reviews. Using AI-driven sentiment analysis, this feedback is categorized, prioritized, and flagged for attention by the product and development teams, ensuring that user-reported issues are addressed promptly.

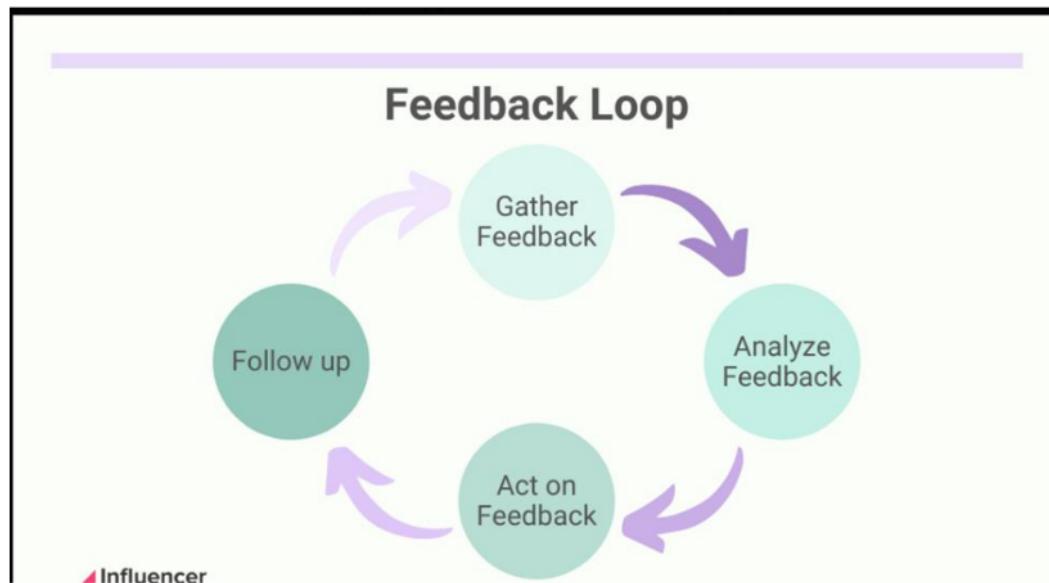
- **Operations & Development Feedback:**

Internal feedback from operations and development teams is collected during regular retrospectives and incident reviews. This feedback informs adjustments to system

monitoring, performance tuning, and operational workflows, helping to identify areas for enhancement in technical stability and user experience.

- **Iterative Improvement Cycles:**

Using agile methodology, the app undergoes continuous improvement cycles where the team incorporates feedback, releases feature updates, and addresses identified issues. Each cycle includes testing for quality assurance and stakeholder feedback sessions to ensure changes align with both user needs and business goals.



Mental Wellness & Habit Tracker App with AI Coach

ORIGINALITY REPORT



PRIMARY SOURCES

Rank	Source Details	Similarity (%)
1	Submitted to University of Maryland, Global Campus Student Paper	1%
2	Submitted to DeVry, Inc. Student Paper	1%
3	Shravan Pargaonkar. "A Guide to Software Quality Engineering", CRC Press, 2024 Publication	<1%
4	Submitted to University of West Florida Student Paper	<1%
5	Submitted to University of Wales Institute, Cardiff Student Paper	<1%
6	www.andolasoft.com Internet Source	<1%
7	Submitted to Applied Business Academy Student Paper	<1%
8	Submitted to University of Derby Student Paper	<1%

9	Submitted to American Public University System Student Paper	<1 %
10	Submitted to Manchester Metropolitan University Student Paper	<1 %
11	Submitted to NCC Education Student Paper	<1 %
12	www.appcial.com Internet Source	<1 %
13	www.coursehero.com Internet Source	<1 %
14	www.saashub.com Internet Source	<1 %
15	www.redzonetech.net Internet Source	<1 %
16	Submitted to University of Ulster Student Paper	<1 %
17	codinginterviewpro.com Internet Source	<1 %
18	inscricao.faculdadeitop.edu.br Internet Source	<1 %
19	www.leatherlink.net Internet Source	<1 %

20

Faheem Syeed Masoodi, Alwi M. Bamhdi,
Majid A. Charoo, Zubair Sayeed Masoodi, Arif
Mohammad, Gousiya Hussain. "Internet of
Things Vulnerabilities and Recovery
Strategies", Routledge, 2024

<1 %

Publication

Exclude quotes On

Exclude matches Off

Exclude bibliography On