Non-functional requirements

(Group-3)

Non –functional requirements addressing aspects of Performance, Scalability, Security, Reliability, Usability, and Maintainability.

1. Performance

- **Response Time**: The app should load core features (like assessments, resources, and check-ins) within 2 seconds for a smooth user experience.
- **Minimal Latency for Real-Time Alerts**: Emergency alerts (such as the panic button) should be triggered and received by emergency contacts within 5 seconds.
- **Efficient Data Retrieval**: Accessing the user's historical data (such as mental health trends and past assessments) should be optimized to avoid lag or delay.

2. Scalability

- **User Load Management**: The app should be able to scale to support a large number of simultaneous users without performance degradation.
- **Modular Resource Allocation**: As user numbers grow, the app should allocate resources dynamically for community support and group sessions, using load balancing where necessary.
- **Elastic Infrastructure**: The back-end infrastructure should be elastic, expanding or contracting based on real-time user demands, to support varying traffic levels (e.g., during peak times for group sessions).

3. Security

- **Data Encryption**: All user data should be encrypted both in transit and at rest to protect against unauthorized access.
- **Authentication and Authorization**: Implement secure authentication (e.g., two-factor authentication) and role-based access control to protect user data and restrict access based on user roles.
- **Regular Security Audits**: The system should undergo regular security checks and penetration testing to identify and resolve potential vulnerabilities.
- **Compliance with Health Data Regulations**: Ensure compliance with health information privacy laws and regulations, such as HIPAA or GDPR, as applicable.

4. Reliability

- **Uptime and Availability**: The app should maintain an uptime of at least 99.9% to ensure constant availability for users, especially for emergency alerts.
- **Redundancy and Backup**: Implement redundant data storage and regular data backups to prevent data loss and ensure quick recovery.

- **Error Recovery**: In case of system failures, ensure a robust recovery mechanism to minimize downtime and data loss.
- **Offline Access for Emergency Features**: Some critical features (e.g., the panic button) should be accessible offline if connectivity is poor, and the app should send the alert when connection is restored.

5. Usability

- **User-Friendly Interface**: The app should have an intuitive interface for seamless navigation, especially for users who may not be tech-savvy.
- **Accessibility**: Ensure compliance with accessibility standards (such as WCAG) so that users with disabilities can use the app easily.
- **Personalization Options**: Allow users to personalize their experience based on preferences, such as the frequency of mental health check-ins or types of notifications.
- **Language Support**: Include multi-language support to reach a diverse user base and improve usability for non-native speakers.

6. Maintainability

- **Modular Code Structure**: The app should have a modular codebase to facilitate easy updates, bug fixes, and feature enhancements.
- **Comprehensive Documentation**: Maintain up-to-date documentation on the app's architecture, APIs, and modules for developers and support teams.
- **Automated Testing**: Integrate automated testing for key functions (such as assessments, emergency alerts, and reminders) to ensure that updates do not disrupt existing features.
- **Scalable Database Design**: The database structure should support easy modifications and expansions, allowing future updates to the app's data model.