NAME: OLUYEMI OLUWAFOLASEKEMI

MATRIC NUMBER: 23/0188

DEPARTMENT: SOFTWARE ENGINEERING GROUP E

Requirement Analysis for the Campus Safety App

1. Functional and Non-functional Requirements

Functional Requirements

-User Registration and Login

- Secure login for students and staff.
- Password recovery functionality.

Emergency Contacts

- Quick access to predefined emergency contacts (campus security, local police, fire department).
- Ability to add custom emergency contacts.

Panic Button

- Single-tap activation to alert campus security.
- Sends the user's real-time location and profile information to responders.

Live Chat

- Instant messaging functionality with campus security.
- Support for multimedia attachments (e.g., photos, videos).

Safety Map

- Real-time map displaying safety zones, incident reports, and location of security personnel.
- GPS-enabled geolocation tracking for users and emergency responders.

Notifications

- Real-time alerts about campus safety updates and incidents.
- Push notifications for emergency broadcasts.

Geolocation Sharing

- Allows users to share their real-time location with security personnel during emergencies.
- Option to stop location sharing after the situation is resolved.

Non-functional Requirements

Usability

- Intuitive and user-friendly interface.
- Accessible for all types of users, including those with disabilities.

Performance

- Must handle up to 100,000 concurrent users.
- Real-time notifications and geolocation updates within 2 seconds.

Security

- Data encryption for all communications.
- Compliance with data privacy laws (e.g., GDPR, CCPA).

Availability

- Minimum 99.9% uptime.
- Redundant systems to ensure functionality during emergencies.

Compatibility

- Native support for Android and iOS.
- Compatible with GPS services and third-party APIs (e.g., Google Maps).

Scalability

- Designed to accommodate increasing user numbers and data loads.

2. Development Process Model: Agile Development

Justification

- **Iterative Development**: Agile allows continuous feedback and iteration, ensuring the app meets evolving user needs.

- Flexibility: Agile handles changes effectively, ideal for integrating user feedback from paper prototyping and scenario analysis.

- Faster Delivery: With sprints, features like the panic button or safety map can be developed and tested incrementally.

- Collaboration: Involves stakeholders (e.g., students, campus security) in the development cycle for better alignment.

3. Use Case Diagram and Scenario Analysis

Use Case Diagram

The use case diagram includes the following actors:

- User (Student/Staff)

- Campus Security Personnel

- Administrator

Key use cases:

- Activate Panic Button

- Chat with Campus Security

- View Safety Map

- Receive Notifications

- Share Geolocation

Scenario Analysis

Use Case: Activate Panic Button

i. Actor: User

ii. Trigger: User feels threatened and activates the panic button.

iii. Preconditions: User is logged in, and location services are enabled.

iv. Normal Flow:

- User taps the panic button.
- The app sends an alert to campus security with the user's real-time location and profile details.
- Security acknowledges the alert and responds.

v. Exceptions:

- If GPS is unavailable, fallback to last known location.
- Alert delivery failure due to server downtime or network issues.

---

4. Validating Requirements through Paper Prototyping

- Create low-fidelity prototypes of the app's interface (e.g., login screen, panic button, safety map).

- Test with students and campus security to gather feedback on usability and functionality.

- Iterate on designs based on stakeholder feedback.

5. Risk Analysis

Risk	Likelihood	Impact	Mitigation

Server Downtime	Medium	High	Implement redundant servers and failover systems.
GPS Inaccuracy	High	Medium	Use advanced geolocation algorithms.
Data breach	Low	High	Implement encryption and regular audits.
User misuse of panic button	Medium	Medium	Add confirmation prompt and log all activations.
Scalability issues	Medium	High	Use cloud based infrastructure.

6. Project Planning and Management

- Project Phases:

1. Requirement Gathering
2. Design
3. Development
4. Testing
5. Deployment
6. Support and Maintenance

Gantt Chart:

A Gantt chart showing phases, tasks, resources, and timelines will be developed using MS Project.

7. Software Proposal

Title: Campus Safety Mobile App

Objective: To enhance student and staff safety by providing real-time access to emergency services and campus security resources.

Deliverables:

- Fully functional mobile app with all specified features.
- User training manual and onboarding tutorials.
- Backend system for security personnel to manage alerts and communications.
- Post-deployment maintenance for one year.

Estimated Timeline: 6 months