Walansi Kontonbile

Software Requirements Specification

1.1 Executive Summary

Walansi Kontonbile addresses the critical need for culturally relevant digital safety education for girls and young people in Ghana, particularly in underserved communities where technology-facilitated gender-based violence, misinformation, and cyber threats are prevalent.

1.2 Business Objectives

- **Primary Goal**: Reduce digital exclusion of girls and young people from online spaces
- **Secondary Goals**:
    - Decrease incidents of TGBV through prevention education
    - Combat misinformation in local communities
    - Increase civic engagement among youth
    - Build digital literacy using culturally appropriate methods

1.3 Success Metrics

- User engagement: 70% monthly active users within 6 months
- Safety impact: 40% reduction in reported harassment cases among users
- Fact-checking usage: 1000+ daily fact-check requests
- Language adoption: 60% of interactions in local languages
- Community reach: Deployment in 50+ schools/communities

1.4 Target Stakeholders

- **Primary Users**: Girls and young people (13-25 years) in Ghana
- **Secondary Users**: Educators, community leaders, parents
- **Partners**: NGOs, schools, government agencies, fact-checking organizations

2. Functional Requirements

2.1 User Management

- **FR-001**: System shall support user registration via WhatsApp, SMS, and Telegram
- **FR-002**: System shall maintain anonymous user profiles (no personal data collection)
- **FR-003**: System shall allow users to select preferred language (Waali, Dagbani, Twi, Hausa, or English)
- **FR-004**: System shall remember user language preference across sessions

2.2 Multilingual & Multimodal Interface

- **FR-010**: System shall process text messages in different languages (Waali, Dagbani, Twi, Hausa or English)
- **FR-011**: System shall accept and process voice messages
- **FR-012**: System shall generate audio responses in user's preferred language
- **FR-013**: System shall support SMS for low-connectivity environments
- **FR-014**: System shall integrate with WhatsApp Business API
- **FR-015**: System shall integrate with Telegram Bot API

2.3 AI Story-Teacher Module

- **FR-020**: System shall embed cultural elements (proverbs, folktales, idioms) in responses
- **FR-021**: System shall maintain database of culturally appropriate teaching content
- **FR-022**: System shall adapt explanation complexity based on user literacy level
- **FR-023**: System shall provide contextual digital safety education

2.4 Fact-Checking Engine

- **FR-030**: System shall accept text, links, and image inputs for verification
- **FR-031**: System shall integrate with fact-checking APIs (Dubawa, Africa Check)
- **FR-032**: System shall return verification status (True/False/Misleading)
- **FR-033**: System shall generate shareable fact cards
- **FR-034**: System shall provide explanations in user's preferred language
- **FR-035**: System shall maintain local misinformation database

2.5 SafeSpace SOS Module

- **FR-040**: System shall recognize trigger words ("Help", "Harassment", "Walansi")
- **FR-041**: System shall provide immediate safety guidance
- **FR-042**: System shall offer escalation options to trusted partners
- **FR-043**: System shall maintain directory of support organizations
- **FR-044**: System shall ensure confidential handling of SOS requests

2.6 Campaign Generator

- **FR-050**: System shall generate advocacy content templates
- **FR-051**: System shall create culturally relevant memes and graphics
- **FR-052**: System shall produce audio content with local language voiceovers
- **FR-053**: System shall provide TikTok-style script templates
- **FR-054**: System shall incorporate Ghanaian cultural elements in designs

2.7 Data Analytics & Reporting

- **FR-060**: System shall log anonymous interaction data

- **FR-061**: System shall generate monthly community dashboards
- **FR-062**: System shall track misinformation trends
- **FR-063**: System shall provide usage analytics for stakeholders
- **FR-064**: System shall ensure full data anonymization

3. Non-Functional Requirements

3.1 Performance Requirements

- **NFR-001**: Response time ≤ 3 seconds for text messages
- **NFR-002**: Voice message processing ≤ 10 seconds
- **NFR-003**: System availability ≥ 99.5%
- **NFR-004**: Support for 10,000+ concurrent users
- **NFR-005**: Fact-checking response ≤ 15 seconds

3.2 Security Requirements

- **NFR-010**: End-to-end encryption for all communications
- **NFR-011**: No storage of personal identifiable information
- **NFR-012**: Secure API integrations with third-party services
- **NFR-013**: Regular security audits and penetration testing
- **NFR-014**: Compliance with data protection laws

3.3 Scalability Requirements

- **NFR-020**: Horizontal scaling capability
- **NFR-021**: Multi-region deployment support
- **NFR-022**: Database partitioning for performance
- **NFR-023**: CDN integration for media content

3.4 Usability Requirements

- **NFR-030**: Interface accessible to users with basic literacy
- **NFR-031**: Support for low-bandwidth environments
- **NFR-032**: Intuitive command structure
- **NFR-033**: Cultural appropriateness validation by local experts

3.5 Reliability Requirements

- **NFR-040**: Automated failover mechanisms
- **NFR-041**: Data backup and recovery procedures
- **NFR-042**: Error handling with user-friendly messages
- **NFR-043**: Monitoring and alerting systems

4. Technical Requirements

4.1 Platform Requirements

- **Cloud-based deployment** (AWS/Azure/GCP)
- **Microservices architecture**
- **API-first design approach**
- **Mobile-responsive web interface**

4.2 Integration Requirements

- WhatsApp Business API
- Telegram Bot API
- SMS gateway providers (e.g., Twilio)
- Fact-checking APIs (Dubawa, Africa Check)
- Speech-to-text and text-to-speech services
- Local language processing tools

4.3 Data Storage Requirements

- **NoSQL database** for conversation logs
- **Relational database** for structured data
- **File storage** for media content
- **Cache layer** for performance optimization

4.4 Architecture Patterns

Microservices: Enables independent scaling of AI-intensive services vs. simple messaging services

Layered Architecture: Provides clear separation of concerns and maintainability

Event-Driven: Essential for real-time messaging and analytics requirements

API Gateway: Necessary for managing multiple client types (WhatsApp, Telegram, SMS)

CQRS: Optimizes for both real-time interactions and analytical reporting

Caching: Critical for performance in low-bandwidth environments

5. Constraints and Assumptions

5.1 Technical Constraints

- Limited internet connectivity in rural areas
- Varying smartphone capabilities among users
- Local language processing limitations
- Third-party API dependencies

5.2 Budget Constraints

- Development budget: To be determined based on funding
- Operational costs including API usage fees
- Content creation and localization costs

5.3 Timeline Constraints

- MVP development: 6 months
- Beta testing: 2 months
- Full deployment: 3 months

5.4 Assumptions

- Users have access to basic mobile phones
- Community partners will support user acquisition
- Fact-checking partners will provide API access
- Local language experts will be available for content validation

6. Risk Assessment

6.1 Technical Risks

- **API dependencies**: Mitigation through multiple provider contracts
- **Language processing accuracy**: Extensive testing with native speakers
- **Scalability challenges**: Phased rollout approach

6.2 User Adoption Risks

- **Cultural resistance**: Community leader engagement strategy
- **Digital literacy barriers**: Simplified interface design
- **Trust building**: Transparency in AI decision-making

6.3 Operational Risks

- **Content moderation**: Human oversight protocols
- **False information spread**: Robust fact-checking workflows
- **Privacy concerns**: Clear data handling policies