**Requirements and Constraints Document for Inventory Management System**

**Introduction**

This document outlines the requirements and constraints for the development of an Android-based inventory management system. The system operates in both offline and online modes, leveraging SQLite for offline functionality and MySQL with a PHP-based REST web service for online operations.

**Functional Requirements**

**1. User Authentication**

* Users must log in to access the application.
* Online authentication generates a login token valid for 10 minutes.
* The login token is required for all subsequent API requests.
* The token's validity is refreshed with each successful operation.

**2. Inventory Management**

* Users can create, edit, and delete multiple inventories.
* Each inventory can contain multiple items.

**3. Item Management**

* Users can add, edit, and delete items within an inventory.
* Each item includes details such as:
  1. Name
  2. Location
  3. Stock quantity
  4. Notification threshold (for low stock alerts)

**4. Notifications**

* Alerts are triggered when an item's stock falls below the specified threshold.
* Notifications are supported in both offline and online modes.

**5. Offline Functionality**

* SQLite is used to manage local data.
* All operations (inventory and item management) are available offline.
* Synchronization with the online server is performed when connectivity is restored.

**6. Online Functionality**

* Uses a PHP-based REST web service to communicate with a MySQL database.
* All operations mirror the offline functionality but also include:
  1. Token-based authentication.
  2. Data synchronization between local and server databases.

**7. Data Synchronization**

* Changes made offline are synchronized with the server once the device reconnects.
* Conflicts (e.g., simultaneous updates) are resolved by applying the most recent change.

**Non-Functional Requirements**

**1. Performance**

* Offline operations must execute within 200ms.
* Online operations, including API calls, must execute within 1 second.

**2. Security**

* All online communications must use HTTPS.
* Tokens must be securely stored on the device and transmitted only when required.

**3. Scalability**

* The server must handle up to 10,000 concurrent users.
* The system must scale to support large inventories (up to 10,000 items per user).

**4. Usability**

* The UI must be intuitive, with clear navigation for inventory and item management.
* Notifications must be visually distinct and actionable.

**5. Reliability**

* The app must function offline without requiring an internet connection for basic operations.
* Data integrity must be maintained during synchronization.

**Constraints**

**1. Platform**

* The mobile application will run on Android devices with API level 23 (Android 6.0) or higher.

**2. Database**

* Offline database: SQLite
* Online database: MySQL

**3. Programming Languages**

* Mobile application: Java or Kotlin
* Web service: PHP

**4. Token Expiry**

* Login tokens have a fixed validity of 10 minutes.
* Tokens are renewed upon successful user activity.

**5. Synchronization Frequency**

* Synchronization is triggered:
  1. When the app detects an active internet connection.
  2. Upon user request (manual sync option).

**6. Resource Constraints**

* Limited server resources; optimization is required to minimize server load.
* Offline storage is limited by the device's available memory.

**Database Schema Overview**

**Offline Schema (SQLite)**

* Tables:
  1. users
  2. inventory
  3. items
  4. notification

**Online Schema (MySQL)**

* Tables:
  1. users
  2. session
  3. inventory
  4. item
  5. notification

**Key Challenges**

* Ensuring seamless synchronization between offline and online databases.
* Maintaining token security and renewal logic.
* Handling conflicts during data synchronization.
* Providing a consistent user experience across offline and online modes.