**Software Design Specification (SDS)**

**Project Title:** Resourcify: Smart Community Resource Management System  
**Student Number:** 23/05373  
**Student Name:** Nicholas Kariuki Wambui

**1. Introduction**

**1.1 Goals and Objectives**  
The goal of Resourcify is to create an efficient and scalable resource management system that optimizes resource allocation, enhances transparency, and integrates automation and predictive analytics.

**1.2 Statement of Scope**  
The system will handle resource inventory management, request submission, approval workflows, predictive analytics, and geospatial mapping.

**1.3 Software Context**  
Resourcify will be used by communities, universities, and NGOs to manage and distribute resources efficiently.

**1.4 Major Constraints**

* The system must be developed using Java (Spring Boot) for the backend and HTML/CSS/JavaScript for the frontend.
* The system must integrate with Google Maps API.

**2. Data Design**

**2.1 Internal Software Data Structure**

* Resource data will be stored in a MySQL/SQLite database.
* User data will include username, password (encrypted), and role (Admin, Donor, User).

**2.2 Global Data Structure**

* Resource inventory will be accessible globally to Admins and Donors.

**2.3 Temporary Data Structure**

* Temporary data will be used for session management (e.g., user login sessions).

**2.4 Database Description**

* The database will store tables for users, resources, requests, and analytics.

**3. Architectural and Component-Level Design**

**3.1 System Structure**

* The system will follow a **Model-View-Controller (MVC)** architecture.

**3.2 Component Descriptions**

* **Component 1: User Authentication**
  + **Processing Narrative:** Validates user credentials and grants access.
  + **Interface Description:** Input: username and password. Output: Access granted or denied.

**4. User Interface Design**

**4.1 Description of the User Interface**

* The interface will include a login screen, dashboard, resource inventory, and request submission forms.

**4.2 Interface Design Rules**

* Follow the eight golden rules of interface design (e.g., consistency, feedback, simplicity).

**5. Restrictions, Limitations, and Constraints**

* The system must be compatible with modern web browsers.
* The system must support at least 100 concurrent users.

**6. Testing Issues**

**6.1 Classes of Tests**

* Unit tests, integration tests, and system tests will be conducted.

**6.2 Expected Software Response**

* The system should handle 100 concurrent users without performance degradation.

**7. References**

* Gao, P., & Thierer, A. (2018). "Smart resource allocation: The role of digital platforms in optimizing supply chains."
* Smith, J. (2020). "Data-driven decision-making in resource management."