SRS

# SOFTWARE REQUIREMENTS SPECIFICATION

Kolam Design Pattern Recognition and Recreation System

Smart India Hackathon 2025

| DOCUMENT VERSION | DATE |
| --- | --- |
| 1.0 | September 2025 |
| **PROBLEM STATEMENT ID** | **ORGANIZATION** |
| 25107 | AICTE - IKS |

## 1. INTRODUCTION

### 1.1 Purpose

This Software Requirements Specification (SRS) document provides a comprehensive description of the functional and non-functional requirements for the Kolam Design Pattern Recognition and Recreation System. It serves as a contract between stakeholders and the development team, ensuring a clear understanding of system expectations and deliverables.

### 1.2 Scope

The system will provide comprehensive tools for analyzing, understanding, and recreating traditional Indian Kolam designs using advanced computer vision, machine learning algorithms, and mathematical pattern analysis. The system encompasses web-based and mobile applications with capabilities for pattern recognition, design recreation, cultural documentation, and educational content delivery.

### 1.3 Document Conventions

- **FR-X.Y:** Functional Requirement with module identifier (X) and requirement number (Y)
- **NFR-X:** Non-Functional Requirement with sequential numbering
- **UC-X:** Use Case identifier

- **Shall/Must:** Mandatory requirements
- **Should:** Highly desirable requirements
- **May/Could:** Optional requirements

### 1.4 Intended Audience

- Software development team and architects
- Quality assurance and testing teams
- Project managers and stakeholders
- Cultural experts and domain specialists
- System administrators and deployment teams

## 2. OVERALL SYSTEM DESCRIPTION

### 2.1 Product Perspective

The Kolam Design System is a standalone software solution that integrates multiple technologies including computer vision, machine learning, web development, and mobile applications. It operates as a comprehensive platform for cultural heritage preservation, mathematical education, and artistic creation.

### 2.2 Product Functions

The system provides the following major functions:

- Automated analysis and classification of traditional Kolam designs
- Mathematical pattern extraction and geometric property identification
- Algorithmic recreation of authentic traditional patterns
- Interactive design creation and editing tools
- Cultural documentation and educational content management
- Multi-platform accessibility (web, mobile, tablet)
- Community collaboration and content contribution features

### 2.3 User Characteristics

| User Type | Technical Expertise | Domain Knowledge | Primary Goals |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Cultural Researchers | Medium | High | Analysis, Documentation, Research |
| Educators | Medium | Medium | Teaching, Learning Resources |
| Artists | Low-Medium | Medium-High | Creation, Inspiration, Practice |
| Students | Low-Medium | Low-Medium | Learning, Exploration, Practice |
| General Public | Low | Low | Entertainment, Cultural Appreciation |

### 2.4 Operating Environment

- **Web Application:** Modern browsers (Chrome 90+, Firefox 88+, Safari 14+, Edge 90+)
- **Mobile Applications:** Android 8.0+ (API level 26+), iOS 12.0+
- **Server Environment:** Linux-based cloud infrastructure (AWS/Azure/GCP)
- **Database Systems:** PostgreSQL 13+, MongoDB 5.0+, Redis 6.2+
- **Processing Requirements:** GPU acceleration for computer vision tasks

## 3. FUNCTIONAL REQUIREMENTS

### 3.1 Pattern Recognition Module (FR-1)

**FR-1.1**

**Image Input Processing**

**Description:** The system shall accept and process various image formats containing Kolam designs.

**Requirements:**

- Support JPEG, PNG, SVG, TIFF, and PDF file formats
- Handle image sizes from 100x100 to 4096x4096 pixels
- Process batch uploads of up to 50 images simultaneously
- Automatic image preprocessing (noise reduction, contrast enhancement, rotation correction)
- Image quality validation and user feedback for low-quality inputs

**Priority:** High | **Verification:** Automated testing with diverse image datasets

FR-1.2

**Design Classification and Recognition**

**Description:** The system shall automatically classify and identify different types of traditional designs.

**Requirements:**

- Classify designs into primary categories: Kolam, Muggu, Rangoli, Rangavalli
- Identify regional variations with minimum 80% accuracy
- Detect seasonal and ceremonial design types
- Categorize complexity levels (Beginner, Intermediate, Advanced, Expert)
- Provide confidence scores for all classifications
- Handle incomplete or partially visible patterns

**Priority:** High | **Verification:** Machine learning model validation with expert-labeled datasets

FR-1.3

**Mathematical Pattern Analysis**

**Description:** The system shall extract and analyze mathematical properties embedded in designs.

**Requirements:**

- Detect symmetry types: rotational, reflectional, translational, and glide reflection
- Identify geometric shapes and their relationships
- Calculate mathematical properties: angles, ratios, proportions, and scaling factors
- Extract underlying grid structures and dot patterns
- Analyze curve properties and path continuity
- Generate mathematical descriptors for pattern similarity comparison

**Priority:** High | **Verification:** Mathematical validation by geometry experts

### 3.2 Design Recreation Module (FR-2)

**FR-2.1**

**Algorithmic Pattern Generation**

**Description:** The system shall generate authentic Kolam patterns using identified mathematical principles.

**Requirements:**

- Generate patterns based on extracted mathematical parameters
- Create variations of existing patterns with user-defined modifications
- Implement traditional construction rules and cultural constraints
- Support parameterized generation (size, complexity, style variations)
- Ensure generated patterns maintain cultural authenticity
- Provide real-time generation with response time under 3 seconds

**Priority:** High | **Verification:** Cultural expert validation and user testing

**FR-2.2**

**Interactive Design Tools**

**Description:** The system shall provide comprehensive interactive tools for design creation and modification.

**Requirements:**

- Real-time pattern editing with immediate visual feedback
- Parameter adjustment controls with live preview
- Grid-based and free-form drawing modes
- Symmetry tools for automatic pattern completion
- Undo/redo functionality with unlimited history
- Layer management and group operations
- Collaborative editing capabilities for multiple users

**Priority:** Medium | **Verification:** User interface testing and usability studies

FR-2.3

**Multi-Format Output Generation**

**Description:** The system shall export designs in multiple formats suitable for different use cases.

**Requirements:**

- Export designs in vector formats (SVG, PDF, EPS) and raster formats (PNG, JPEG)
- Generate step-by-step construction guides with visual instructions
- Create printable templates with customizable sizes
- Produce animated sequences showing construction process
- Generate 3D visualizations and models
- Support batch export for multiple designs

**Priority:** Medium | **Verification:** Format compatibility testing across platforms

### 3.3 Cultural Documentation Module (FR-3)

FR-3.1

**Comprehensive Design Database**

**Description:** The system shall maintain a comprehensive database of traditional designs with metadata.

**Requirements:**

- Store design metadata including origin, cultural context, and historical significance
- Maintain version history and design variations
- Link designs to specific geographic regions and communities
- Include ceremonial usage, seasonal relevance, and symbolic meanings
- Support tagging and categorization systems

- Implement search and filtering capabilities
- Provide data validation and quality control mechanisms

**Priority:** High | **Verification:** Cultural expert review and database integrity checks

FR-3.2

**Educational Content Management**

**Description:** The system shall provide rich educational content associated with designs and cultural practices.

**Requirements:**

- Provide detailed cultural background for each design category
- Explain mathematical concepts embedded in traditional patterns
- Include traditional stories, legends, and symbolic interpretations
- Offer historical context and evolution of design styles
- Support multi-language content (English, Hindi, Telugu, Tamil, Sanskrit)
- Include multimedia content (audio narrations, video demonstrations)

**Priority:** Medium | **Verification:** Educational effectiveness testing and cultural accuracy review

### 3.4 User Interface Module (FR-4)

FR-4.1

**Web Application Interface**

**Description:** The system shall provide a responsive and intuitive web-based interface.

**Requirements:**

- Responsive design supporting desktop, tablet, and mobile browsers
- Intuitive navigation with maximum 3-click access to core features
- Real-time feedback and progress indicators for long-running operations

- Accessibility compliance (WCAG 2.1 AA standards)
- Keyboard navigation and screen reader support
- Customizable user dashboard and preferences
- Integration with social media platforms for sharing

**Priority:** High | **Verification:** Cross-browser testing and accessibility audits

FR-4.2

**Mobile Application**

**Description:** The system shall provide native mobile applications for enhanced user experience.

**Requirements:**

- Native applications for Android and iOS platforms
- Camera integration for real-time pattern analysis and recognition
- Touch-optimized interface with gesture support
- Offline functionality for basic features and downloaded content
- Push notifications for community updates and new content
- AR visualization capabilities for pattern overlay
- Synchronization with web application accounts

**Priority:** Medium | **Verification:** Mobile device testing and app store compliance

# 4. NON-FUNCTIONAL REQUIREMENTS

## 4.1 Performance Requirements

**NFR-1: RESPONSE TIME REQUIREMENTS**

- Pattern recognition processing: Maximum 5 seconds per image
- Real-time design generation: Maximum 1 second response time
- Database query operations: Maximum 2 seconds response time
- Web page loading: Maximum 3 seconds for initial load
- Mobile application startup: Maximum 2 seconds

**NFR-2: THROUGHPUT REQUIREMENTS**

- Support minimum 100 concurrent users simultaneously
- Handle 1000 pattern analysis requests per hour
- Process batch operations of 50 images within 5 minutes
- Support 10,000 database queries per minute
- Handle 500 simultaneous mobile application users

**NFR-3: RESOURCE UTILIZATION**

- Maximum CPU utilization: 80% under normal load
- Maximum memory usage: 4GB per application instance
- Maximum disk I/O: 1000 IOPS for database operations
- Maximum network bandwidth: 100 Mbps per server instance
- Mobile application size: Maximum 50MB installation package

## 4.2 Scalability Requirements

NFR-4: HORIZONTAL SCALING

- Support automatic scaling based on user load
- Load balancing across multiple server instances
- Database sharding and replication capabilities
- CDN integration for global content delivery
- Microservices architecture for independent scaling

NFR-5: DATA GROWTH MANAGEMENT

- Support database growth up to 10TB of design data
- Handle 100,000+ registered users
- Manage 50,000+ traditional design records
- Archive and compress historical data automatically
- Implement data lifecycle management policies

### 4.3 Security Requirements

NFR-6: AUTHENTICATION AND AUTHORIZATION

- Multi-factor authentication for administrative accounts
- Role-based access control (RBAC) with granular permissions
- OAuth 2.0 integration with third-party providers
- Session management with secure token handling
- Password policy enforcement and account lockout mechanisms

NFR-7: DATA PROTECTION

- End-to-end encryption for data transmission (TLS 1.3)

- Encryption at rest for sensitive data (AES-256)
- Input validation and sanitization for all user inputs
- SQL injection and XSS attack prevention
- Regular security audits and penetration testing
- GDPR compliance for European users

**NFR-8: SYSTEM SECURITY**

- Regular security updates and patch management
- Intrusion detection and monitoring systems
- Secure API endpoints with rate limiting
- Container security for deployment environments
- Backup encryption and secure storage

### 4.4 Reliability Requirements

**NFR-9: AVAILABILITY REQUIREMENTS**

- System uptime: 99.5% availability (maximum 43.8 hours downtime per year)
- Planned maintenance windows: Maximum 4 hours monthly
- Disaster recovery time: Maximum 4 hours RTO (Recovery Time Objective)
- Data recovery point: Maximum 1 hour RPO (Recovery Point Objective)
- Health monitoring and alerting systems

**NFR-10: ERROR HANDLING AND RECOVERY**

- Graceful degradation during partial system failures
- Automatic retry mechanisms for transient failures
- Comprehensive error logging and monitoring

- User-friendly error messages and recovery guidance
- Automated backup and recovery procedures

### 4.5 Usability Requirements

**NFR-11: USER EXPERIENCE**

- Intuitive interface requiring minimal training for basic operations
- Maximum 3-click navigation to access core features
- Contextual help and guided tutorials for new users
- Consistent design patterns across all interfaces
- Support for multiple languages with accurate translations

**NFR-12: ACCESSIBILITY**

- WCAG 2.1 AA compliance for web accessibility
- Screen reader compatibility and keyboard navigation
- High contrast mode and scalable fonts
- Voice navigation support for mobile applications
- Alternative text for all images and visual elements

## 5. TECHNICAL CONSTRAINTS

**TC-1: DEVELOPMENT PLATFORM CONSTRAINTS**

- Primary development language: Python 3.9 or higher
- Web framework: FastAPI or Django for backend services
- Frontend framework: React.js with TypeScript
- Mobile development: React Native or native development

- Database systems: PostgreSQL for structured data, MongoDB for documents

#### TC-2: DEPLOYMENT AND INFRASTRUCTURE CONSTRAINTS

- Cloud deployment on AWS, Azure, or Google Cloud Platform
- Containerized deployment using Docker and Kubernetes
- CI/CD pipeline integration for automated deployments
- Load balancing and auto-scaling capabilities
- Monitoring and logging infrastructure setup

#### TC-3: INTEGRATION CONSTRAINTS

- RESTful API design for system integration
- Open-source library preferences for cost-effectiveness
- Standard web technologies for maximum compatibility
- Mobile platform compliance (App Store and Google Play policies)
- Third-party service integration limitations

## 6. EXTERNAL INTERFACE REQUIREMENTS

### 6.1 User Interfaces

- Web-based dashboard with responsive design
- Mobile applications for iOS and Android
- Administrative interface for content management
- API documentation and developer portal

### 6.2 Hardware Interfaces

- Camera integration for mobile devices
- Touch screen support for tablets and mobile devices

- GPU acceleration for computer vision processing
- Network connectivity for cloud services

### 6.3 Software Interfaces

- Operating system compatibility (Windows, macOS, Linux, iOS, Android)
- Browser compatibility (Chrome, Firefox, Safari, Edge)
- Database management system interfaces
- Cloud service provider APIs
- Social media platform integration

### 6.4 Communication Interfaces

- HTTPS/TLS for secure web communication
- WebSocket connections for real-time features
- RESTful APIs for system integration
- Email notifications and communications
- Push notifications for mobile applications

---

**Document Version:** 1.0 | **Date:** September 2025 | **Smart India Hackathon 2025**

This document serves as the official Software Requirements Specification for the Kolam Design Pattern Recognition and Recreation System