

<div style="page-break-after: always;">

License Management System

Comprehensive Knowledge Transfer Document

Document Version: 2.0
Last Updated: June 5, 2025
Prepared by: [Your Name]

Table of Contents

Quick Navigation
[System Overview](#) • [Architecture](#) • [Admin Guide](#) • [User Guide](#) • [Troubleshooting](#)

- [1. System Overview](#)
- [2. System Architecture](#)
- [3. Technical Stack](#)
- [4. User Workflows](#)
 - [◦ Admin Workflow](#)
 - [◦ End User Workflow](#)
- [5. Screenshots and Functionality](#)
- [6. Integration Points](#)
- [7. Troubleshooting Guide](#)
- [8. FAQs](#)

System Overview








Introduction

The **License Management System (LMS)** is an enterprise-grade solution designed to streamline software license management, user access control, and product activation processes. This system provides organizations with complete control over their software distribution, ensuring compliance with licensing agreements while offering a seamless experience for both administrators and end-users.

Business Value

- **Revenue Protection:** Prevent unauthorized software usage and license sharing
- **Operational Efficiency:** Automate license provisioning and management
- **Compliance:** Maintain audit trails and ensure software license compliance
- **User Experience:** Provide self-service options for users while maintaining security
- **Business Intelligence:** Gain insights into software usage patterns and license utilization

Key Features

Feature	Description	Business Benefit
 User Authentication	Secure login with role-based access control	Ensures only authorized personnel can manage licenses
 License Generation	Automated key generation with customizable templates	Reduces manual work and errors
 Dashboard	Real-time insights and analytics	Enables data-driven decision making
 Activation Tracking	Monitor license activations and deactivations	Prevents unauthorized usage
 User Management	Centralized user and role administration	Simplifies access control
 Self-Service Portal	Users can manage their own licenses	Reduces IT support load
 Reporting	Comprehensive reporting and audit trails	Ensures compliance and provides insights

System Requirements

- **Server:** Node.js 14+ with MongoDB 4.4+
 - **Client:** Modern web browser (Chrome, Firefox, Edge, Safari)
 - **Network:** Internet connection for license validation
 - **Storage:** Minimum 1GB free disk space for database
-

System Architecture

High-Level Architecture

System Architecture

Figure 1: System Architecture Diagram

Diagram Description:

1. **Client Applications:** End-user applications that interact with the system
2. **Frontend (React.js):** Web-based admin and user interfaces
3. **Backend API (Node.js/Express):** Handles business logic and data processing
4. **MongoDB Database:** Persistent data storage
5. **License Validation:** Service for validating and managing licenses
6. **Client Environment:** Local environment where the License Manager Library runs

Note: The diagram above is a conceptual representation. To generate or modify the diagram:

1. Use a tool like [Mermaid Live Editor \(https://mermaid.live/\)](https://mermaid.live/) or [draw.io \(https://app.diagrams.net/\)](https://app.diagrams.net/)
2. Save the diagram as an image file in the `Screenshots` folder
3. Update the image reference if needed

Component Architecture

1. Frontend Layer (React.js)

Key Components:

- **Dashboard Module:** Real-time metrics and quick actions
- **User Management:** CRUD operations for user accounts
- **License Administration:** Generate, view, and manage licenses
- **Product Catalog:** Manage software products and versions
- **Reporting Engine:** Generate usage and compliance reports

Key Features:

- Responsive design for all devices
- Real-time updates using WebSockets
- Role-based access control (RBAC)
- Audit logging for all actions

2. Backend Layer (Node.js/Express)

Core Services:

Service	Description	Key Endpoints
Auth Service	Handles authentication & authorization	/api/auth/*
User Service	Manages user accounts and roles	/api/users/*
License Service	License generation and validation	/api/licenses/*
Product Service	Product catalog management	/api/products/*
Activation Service	Handles license activations	/api/activations/*

Key Features:

- RESTful API design
- JWT-based authentication
- Request validation and sanitization
- Rate limiting and security headers
- Comprehensive error handling

3. Data Layer (MongoDB)

Collections Schema:

```
// Users Collection
{
  _id: ObjectId,
  email: String,
  password: String,
  role: { type: String, enum: ['admin', 'user'] },
  createdAt: Date,
  lastLogin: Date
}

// Products Collection
{
  _id: ObjectId,
  name: String,
  version: String,
  description: String,
  features: [String],
  pricing: {
    type: { type: String, enum: ['one-time', 'subscription'] },
    amount: Number,
    currency: String
  },
  activationLimit: Number
}

// Licenses Collection
{
  _id: ObjectId,
  key: String,
  productId: ObjectId,
  userId: ObjectId,
  status: { type: String, enum: ['active', 'suspended', 'revoked'] },
  activations: [{
    deviceId: String,
    activatedAt: Date,
    lastValidated: Date,
    ipAddress: String
  }],
  validFrom: Date,
  validUntil: Date
}
```

4. Client Library

Key Features:

- Machine fingerprinting

- Offline validation
- Tamper detection
- Graceful degradation
- Automatic reactivation

Integration Example:

```
const license = new LicenseManager({
  apiUrl: 'https://api.yourdomain.com',
  productId: 'your-product-id',
  storage: 'localStorage', // or 'file' for Node.js
  checkInterval: 3600 // Check every hour
});

// Check license status
const { valid, message } = await license.validate();
if (!valid) {
  console.error('License validation failed:', message);
  // Handle invalid license
}
```

Data Flow

1. License Activation

```
sequenceDiagram
    participant C as Client App
    participant F as Frontend
    participant B as Backend
    participant D as Database

    C->>F: Request Activation
    F->>B: POST /api/activations
    B->>D: Validate License
    D-->>B: License Data
    B->>D: Record Activation
    B-->>F: Activation Response
    F-->>C: Success/Failure
```

2. License Validation

```
sequenceDiagram
    participant C as Client App
    participant B as Backend
    participant D as Database

    C->>B: Validate License (with machine ID)
    B->>D: Check Activation
    alt Valid License
        D-->>B: Activation Record
        B-->>C: { valid: true, expires: date }
    else Invalid/Expired
        D-->>B: No/Invalid Record
        B-->>C: { valid: false, reason: '...' }
    end
end
```

Security Considerations

- **Data Encryption:** All sensitive data encrypted at rest and in transit
- **Rate Limiting:** Protection against brute force attacks
- **Input Validation:** Protection against injection attacks
- **CORS:** Properly configured CORS policies
- **Audit Logging:** All sensitive operations are logged
- **Secure Storage:** License data stored securely on client devices

Technical Stack

Frontend

- **Framework:** React.js
- **State Management:** Redux/Context API
- **UI Components:** Material-UI
- **Routing:** React Router
- **HTTP Client:** Axios

Backend

- **Runtime:** Node.js
- **Framework:** Express.js

- **Authentication:** JWT (JSON Web Tokens)
- **Database:** MongoDB with Mongoose ODM
- **Validation:** Joi
- **Logging:** Winston

Client Library

- **Language:** JavaScript/Node.js
- **Cryptography:** crypto-js
- **HTTP Client:** Axios
- **Persistence:** File system for local storage

User Workflows

Admin Workflow

1. Authentication

Login Process

```
graph TD
    A[Access Login Page] --> B[Enter Credentials]
    B --> C[Submit]
    C --> D{Validation}
    D -->|Success| E[Redirect to Dashboard]
    D -->|Failure| F[Show Error]
    F --> B
```

Key Features:

- Secure login with JWT authentication
- Role-based access control (RBAC)
- Session management
- Failed login attempt tracking

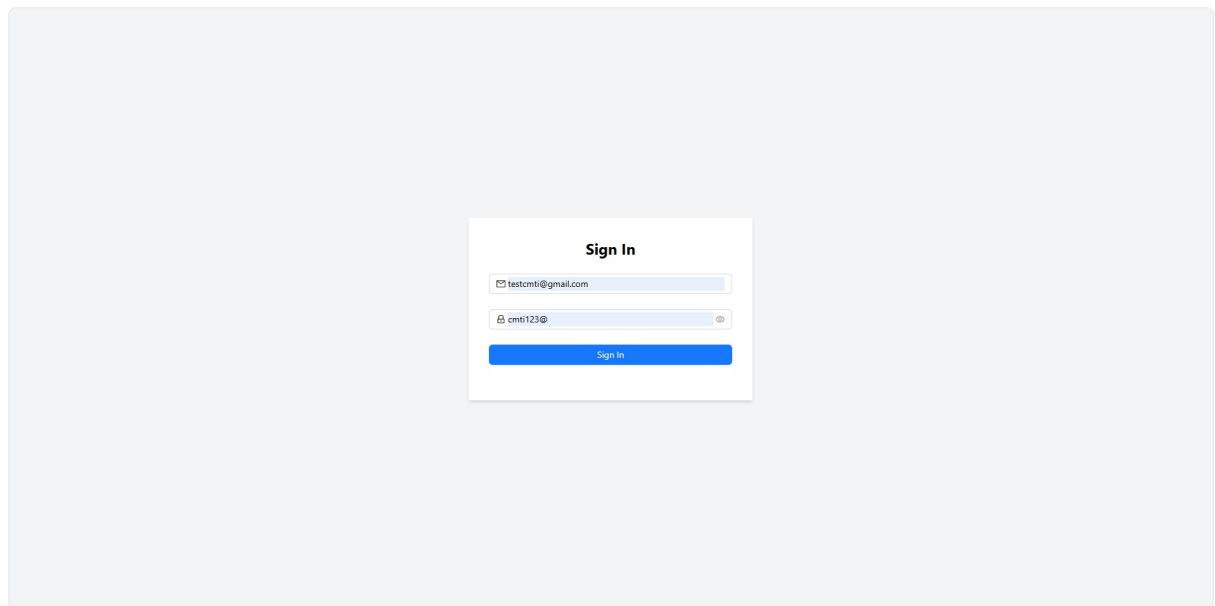


Figure 1: Admin Login Interface

2. Dashboard

Dashboard Components:

- System health metrics
- License usage statistics
- Recent activities
- Quick action buttons

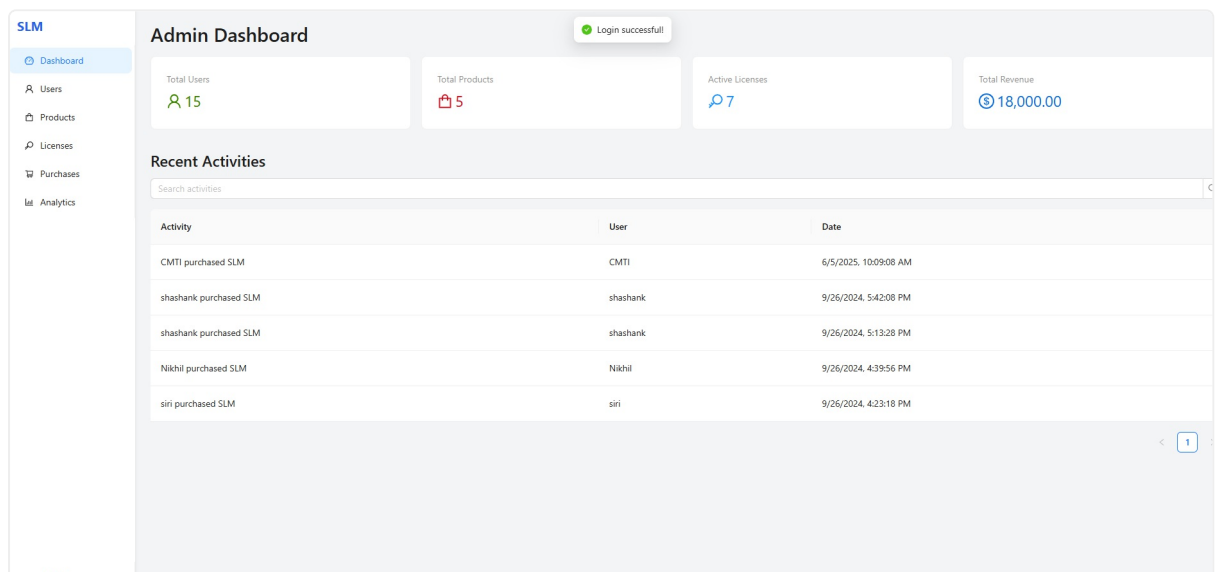


Figure 2: Admin Dashboard Overview

3. User Management

User Management Workflow



Features:

- User CRUD operations
- Role assignment
- Bulk operations
- Activity logging

SLM

Dashboard

Users

Products

Licenses

Purchases

Analytics

Users Management

Add User

Name	Email	Role	Actions
Nihal	admin@gmail.com	user	<div><div></div><div></div></div>
Nihal	admin2@gmail.com	user	<div><div></div><div></div></div>
haju	admin3@gmail.com	user	<div><div></div><div></div></div>
haju2	admin4@gmail.com	admin	<div><div></div><div></div></div>
DMN	omn@gmail.com	admin	<div><div></div><div></div></div>
Nihal	Nihalme5@gmail.com	user	<div><div></div><div></div></div>
Nikhil	Nachappa057@gmail.com	user	<div><div></div><div></div></div>
Naanu	Naanu@gmail.com	user	<div><div></div><div></div></div>
Shashank	shashank@gmail.com	user	<div><div></div><div></div></div>
yedu	yedu@gmail.com	user	<div><div></div><div></div></div>

<

1

2

>

Figure 3: User Management Interface

4. Product Management

Product Lifecycle

1. **Create Product**
 - Basic information
 - Version control
 - Feature sets
 - Pricing models

2. Update Product

- Version updates
- Feature modifications
- Deprecation handling

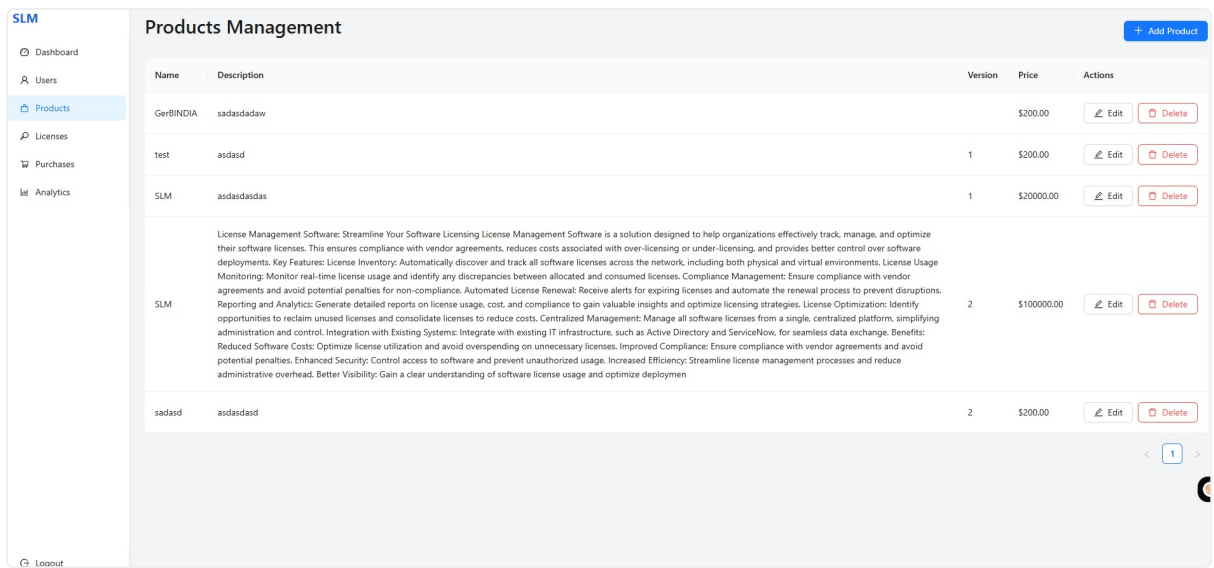


Figure 4: Product Catalog View

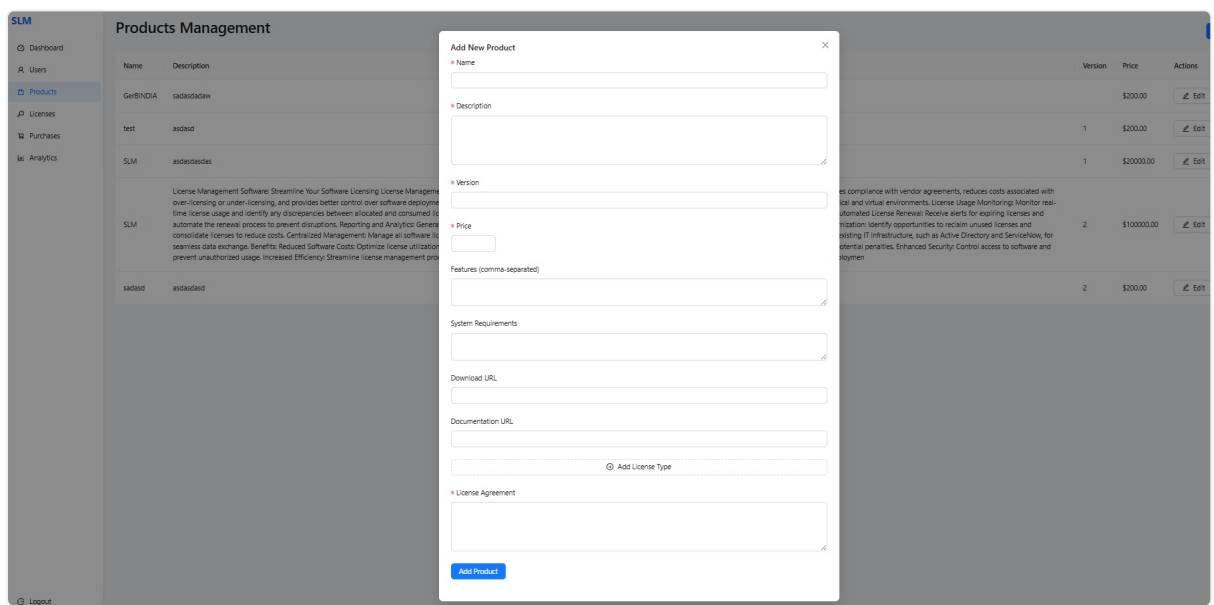


Figure 5: Adding a New Product

5. License Management

License Generation Process

sequenceDiagram

Admin->>+Backend: Generate License

Backend->>+Database: Create License Record

Database-->>-Backend: Confirm Creation

Backend-->>-Admin: License Key

Admin->>+User: Distribute Key

User->>+System: Activate License

Key Features:

- Batch generation
- Custom validation rules
- Expiration settings
- Activation limits

SLM	License Management					
	User	Product	License Key	Type	Status	Actions
Dashboard	shashank	SLM	6w3ogpF3n9r6c32yh4h	Subscription	active	Edit Delete
Users	shashank	SLM	rth0t8jcaqpod5u48o	Subscription	active	Edit Delete
Products	shashank	SLM	2njt62q1fkgp75c7enf	Subscription	active	Edit Delete
Licenses	siri	SLM	qkx6x1tdn8c2eoc98	Subscription	active	Edit Delete
Purchases	Nikhil	SLM	8476cmkgwqyrbhvcwin	Subscription	active	Edit Delete
Analytics	shashank	SLM	0pbn0auyew7mukmsdia19	Subscription	active	Edit Delete
	shashank	SLM	hncqpn4w3nhqz320g63o	Subscription	active	Edit Delete
	OMT	SLM	949g11qjh8cw65fagq	Perpetual	active	Edit Delete

Figure 6: License Management Console

SLM	License Management					
	User	Product	License Key	Type	Status	Actions
Dashboard	shashank	SLM	6w3ogpF3n9r6c32yh4h	Subscription	active	Edit Delete
Users	shashank	SLM	rth0t8jcaqpod5u48o	Subscription	active	Edit Delete
Products	shashank	SLM	2njt62q1fkgp75c7enf	Subscription	active	Edit Delete
Licenses	siri	SLM	qkx6x1tdn8c2eoc98	Subscription	active	Edit Delete
Purchases	Nikhil	SLM	8476cmkgwqyrbhvcwin	Subscription	active	Edit Delete
Analytics	shashank	SLM	0pbn0auyew7mukmsdia19	Subscription	active	Edit Delete
	shashank	SLM	hncqpn4w3nhqz320g63o	Subscription	active	Edit Delete
	OMT	SLM	949g11qjh8cw65fagq	Perpetual	active	Edit Delete

Figure 7: Editing License Details

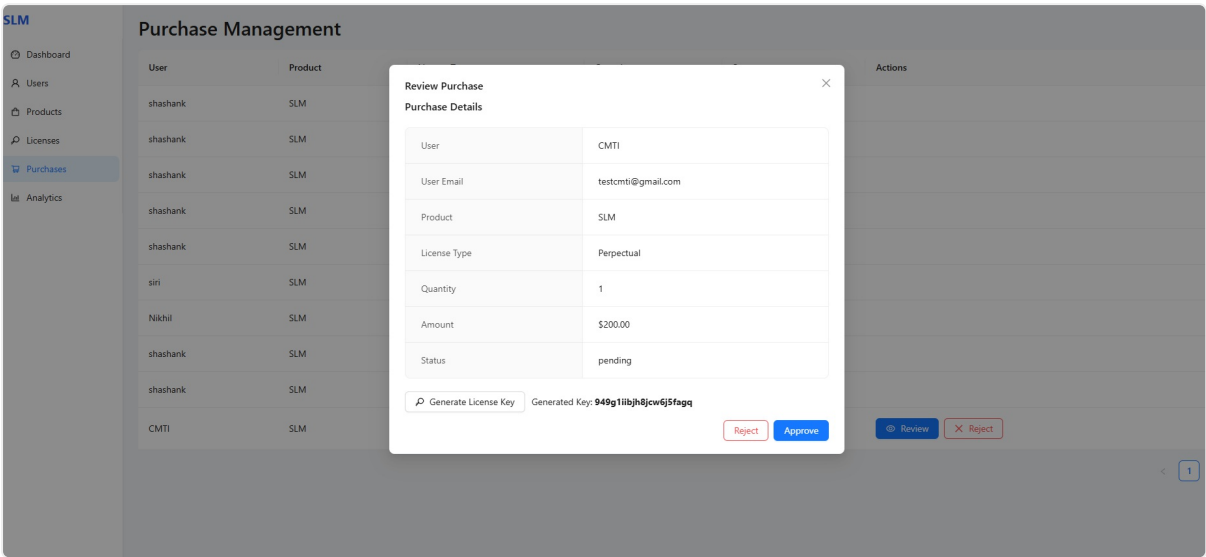


Figure 8: License Key Generation

6. Purchase Management

Purchase Workflow

1. Order Processing
2. Payment Verification
3. License Assignment
4. Notification

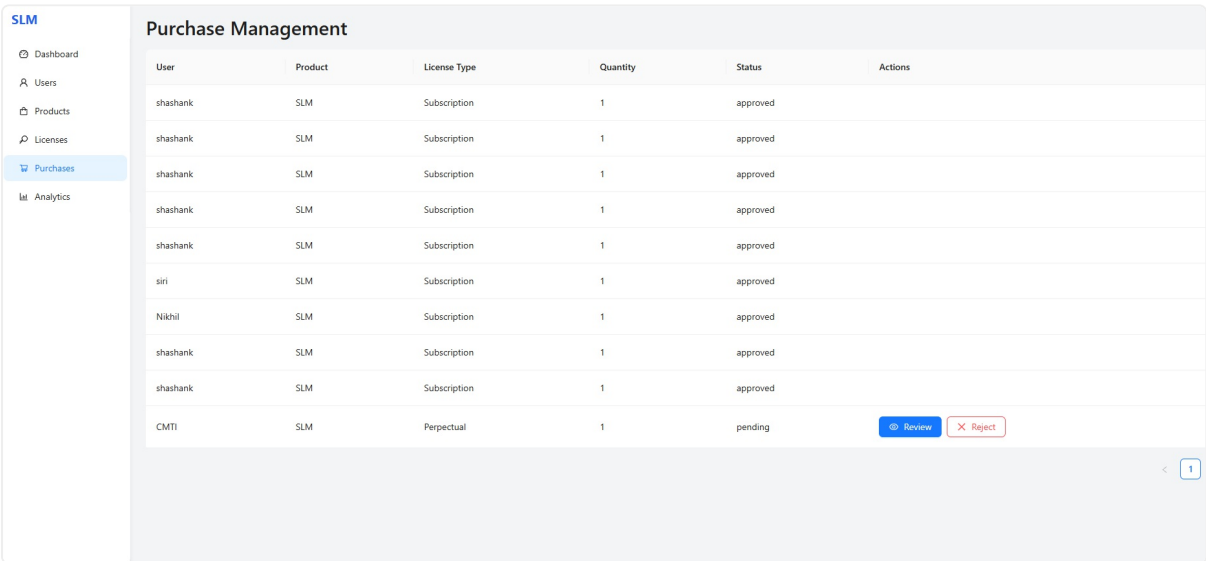


Figure 9: Purchase Management View

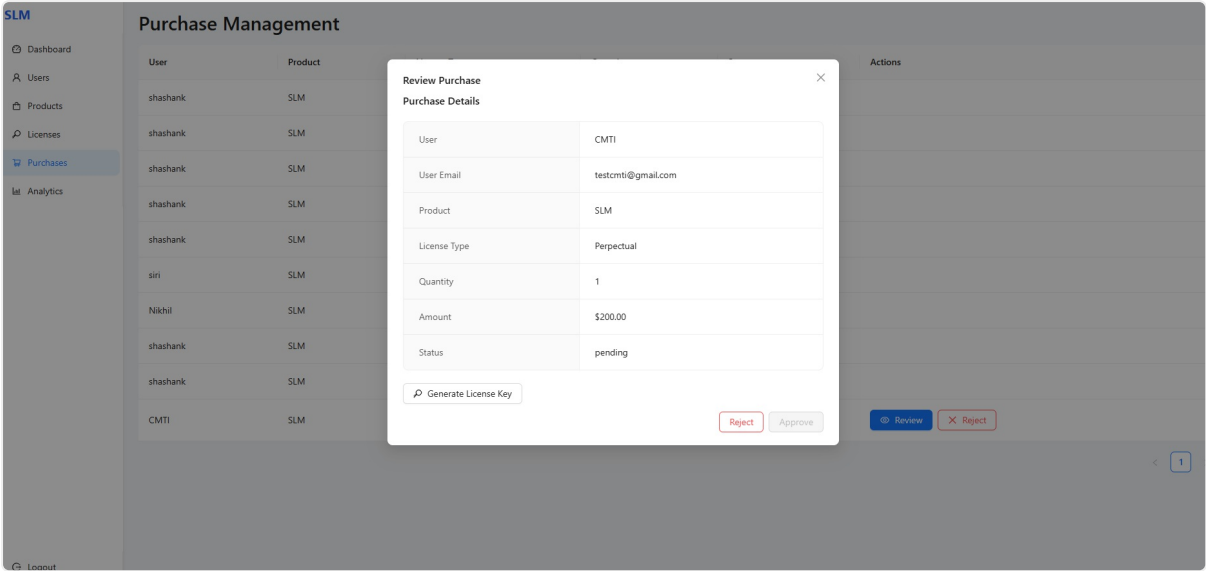
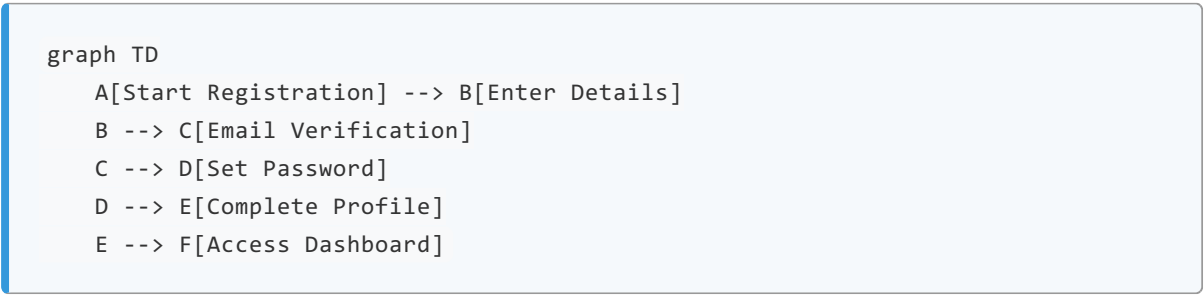


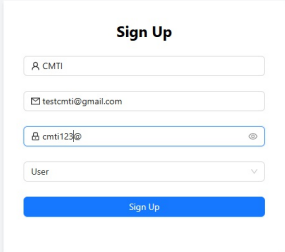
Figure 10: Purchase Review Process

End User Workflow

1. Account Management


Registration Process





A user registration form titled "Sign Up" is centered on a light gray background. The form contains four input fields: a text field for "CM11", an email field for "testcmti@gmail.com", a password field for "cmti123@" with a toggle icon, and a dropdown menu labeled "User". A blue "Sign Up" button is positioned at the bottom of the form.

Figure 11: User Registration Form



A user login screen titled "Sign In" is centered on a light gray background. The screen features two input fields: an email field for "testcmti@gmail.com" and a password field for "cmti123@" with a toggle icon. A blue "Sign In" button is located at the bottom of the form.

Figure 12: User Login Screen

2. User Dashboard

Dashboard Features:

- License overview
- Quick actions
- Notifications
- Recent activities

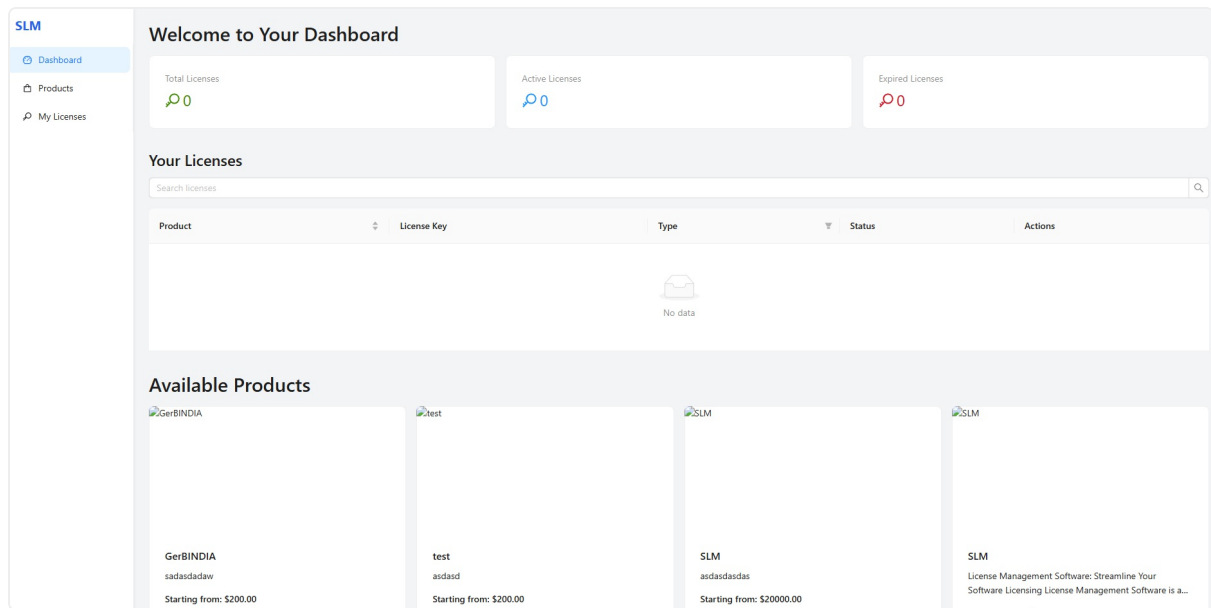


Figure 13: User Dashboard Overview

3. License Management

License Operations:

- View active licenses
- Check activation status
- Request additional licenses
- Download license files

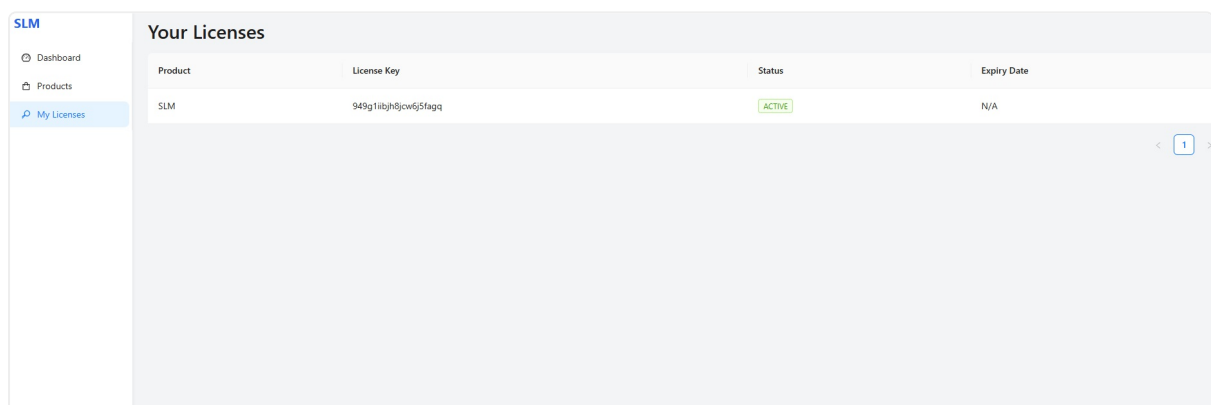


Figure 14: User License Management

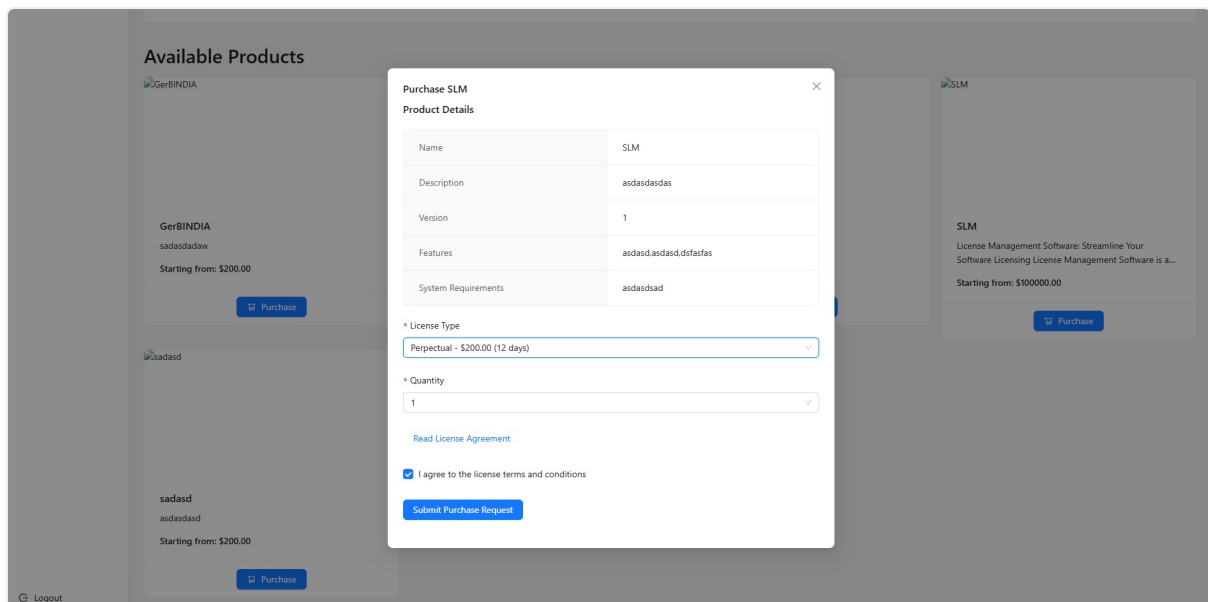


Figure 15: Purchase History

4. License Activation

Activation Flow

1. Enter License Key
2. System Validation
3. Machine Binding
4. Confirmation

Offline Activation

1. Generate Activation Request
2. Submit Request File
3. Receive Activation File
4. Complete Activation

```

PS D:\Naanu\Codes\CMTI\License_managment\LMS_Simulation> node activation.js
🚀 Starting license activation process...
Checking for existing license...
No existing license file found, attempting to activate...
Attempting to activate license: an10ypfuzg8zhki92twrm
Entering activateLicense function
Request body: {
  licenseKey: 'an10ypfuzg8zhki92twrm',
  machineId: '151366df81d31c39409b87692204905c3cdf766af53b74612ec0a7c0aba70838'
}
License activation response: {
  message: 'License activated successfully',
  license: {
    _id: '68412508d60b36487ad6757a',
    userId: '68411f11d60b36487ad67468',
    productId: '66e96eeb7a7149dee0d6f4fd',
    licenseKey: 'an10ypfuzg8zhki92twrm',
    type: 'perpetual',
    startDate: '2025-06-05T05:03:04.826Z',
    status: 'active',
    activations: [ [Object] ],
    createdAt: '2025-06-05T05:03:04.827Z',
    updatedAt: '2025-06-05T05:03:58.031Z',
    __v: 1
  }
}
License data saved successfully
✅ License activated successfully. Access granted to the application.
🚀 Application started successfully!

```

Figure 16: License Activation Process

Support Workflow

Ticket Lifecycle

```

graph LR
    A[Submit Ticket] --> B[Triage]
    B --> C{Complexity}
    C -->|Simple| D[Resolve]
    C -->|Complex| E[Escalate]
    E --> F[Resolution]
    D --> G[Close]
    F --> G

```

Support Channels:

- In-app messaging
- Email support

- Knowledge base
- Community forums

Integration Points

API Reference

Base URL

```
https://api.yourdomain.com/v1
```

Authentication

All API requests require an API key or JWT token in the header:

```
Authorization: Bearer YOUR_API_KEY
```

Core Endpoints

1. Authentication

Login

```
POST /api/auth/login
```

Request:

```
{
  "email": "user@example.com",
  "password": "your_password"
}
```

Response:

```
{
  "token": "jwt_token_here",
  "user": {
    "id": "user_id",
    "email": "user@example.com",
    "role": "admin"
  }
}
```

2. License Management

Generate License

```
POST /api/licenses
```

Validate License

```
GET /api/licenses/validate/{key}
```

Get License Details

```
GET /api/licenses/{id}
```

Webhooks

Webhooks allow you to receive real-time updates about system events.

Available Events:

- license.created
- license.activated
- license.expired
- purchase.completed
- user.registered

Webhook Payload Example:

```
{
  "event": "license.activated",
  "data": {
    "licenseKey": "LIC-123-456-789",
    "productId": "prod_123",
    "activatedAt": "2025-06-05T11:30:00Z",
    "machineId": "machine_fingerprint"
  },
  "timestamp": "2025-06-05T11:30:01Z"
}
```

Client Integration

Installation

```
npm install @your-org/license-manager
# or
yarn add @your-org/license-manager
```

Basic Usage

```
const { LicenseManager } = require('@your-org/license-manager');

// Initialize with your API key
const license = new LicenseManager({
  apiKey: 'your_api_key',
  productId: 'your_product_id',
  storage: 'localStorage' // or 'file' for Node.js
});

// Activate license
async function activateLicense(key) {
  try {
    const result = await license.activate(key);
    console.log('Activation successful:', result);
  } catch (error) {
    console.error('Activation failed:', error.message);
  }
}

// Check license status
async function checkLicense() {
  const status = await license.validate();
  if (status.valid) {
    console.log('License is valid until:', status.expiresAt);
  } else {
    console.error('License is invalid:', status.reason);
  }
}
```

Troubleshooting Guide

Common Issues and Solutions

1. Activation Issues

Problem: License activation fails

Solution:

1. Check internet connection
2. Verify license key format
3. Ensure machine ID hasn't changed
4. Check server logs for errors

Error Reference:

Error: Invalid license key

- Verify the key was entered correctly
- Check if the key has been revoked
- Contact support if issue persists

2. Validation Failures

Problem: License validation fails

Solution:

```
graph TD
    A[Validation Failed] --> B{Check Error}
    B -->|Expired| C[Renew License]
    B -->|Invalid| D[Verify Key]
    B -->|Max Activations| E[Deactivate Old Devices]
    C --> F[Contact Support]
    D --> G[Retry Activation]
    E --> G
```

3. API Connectivity

Problem: Cannot connect to license server

Troubleshooting Steps:

1. Check network connectivity
2. Verify API endpoint URL
3. Check CORS configuration
4. Test with Postman/cURL

Test Command:

```
curl -X GET "https://api.yourdomain.com/health" \
-H "Authorization: Bearer YOUR_API_KEY"
```

4. Performance Issues

Problem: Slow response times

Optimization Tips:

- Enable caching where possible

- Use batch operations for multiple licenses
- Implement retry logic with exponential backoff

5. Security Concerns

Best Practices:

- Always use HTTPS
- Rotate API keys regularly
- Implement rate limiting
- Use secure storage for sensitive data

Getting Help

If you're still experiencing issues, please provide:

1. Error message
2. Steps to reproduce
3. Environment details
4. Relevant logs (with sensitive information redacted)

Contact support at: support@yourdomain.com

FAQs

Technical

Q: How do I integrate the license check in my application?

A: Include the client library and follow the integration guide. Detailed integration examples are available in the [Integration Guide](#).

Q: What data is collected during activation?

A: Only the following data is collected:

- Machine ID (hardware fingerprint)
- License key
- Timestamp
- IP address (for security purposes)
- Product/version information

Conclusion

This document has provided a comprehensive overview of the License Management System, covering its architecture, features, and usage guidelines. The system is designed to be robust, secure, and scalable, meeting the needs of both administrators and end-users.

Key Takeaways

1. For Administrators

- Centralized license management
- Real-time monitoring and reporting
- Granular access control
- Automated workflows

2. For Developers

- Well-documented RESTful API
- Client libraries for easy integration
- Webhook support for real-time updates
- Comprehensive error handling

3. For End Users

- Simple license activation
- Self-service portal
- Clear license status
- Easy device management

Appendix

A. Glossary

Term	Definition
Activation	Process of binding a license to a specific device
API Key	Unique identifier for API authentication
JWT	JSON Web Token for secure authentication
Machine ID	Unique identifier generated from hardware components
Webhook	Method for real-time notifications

B. Error Codes

Code	Description	Resolution
400	Bad Request	Verify request parameters
401	Unauthorized	Check API key/token
403	Forbidden	Verify permissions
404	Not Found	Check endpoint URL
429	Too Many Requests	Implement rate limiting
500	Server Error	Contact support

C. Version History

Version	Date	Changes
1.0	2024-01-15	Initial Release
1.1	2024-03-22	Added webhook support
2.0	2025-06-05	Major UI overhaul, New APIs

D. Additional Resources

- API Reference Documentation (<https://docs.yourdomain.com/api>)

- Client SDK GitHub Repository (<https://github.com/your-org/license-manager-sdk>)
- Support Portal (<https://support.yourdomain.com>)
- Knowledge Base (<https://help.yourdomain.com>)

Document Version: 2.0
Last Updated: June 5, 2025
Prepared by: [Your Name/Team]
Confidential & Proprietary

Document Control

Version	Date	Author	Changes
1.0	2024-01-15	[Author]	Initial version
1.1	2024-03-22	[Author]	Added webhook section
2.0	2025-06-05	[Author]	Major update, new features

Reviewers

Name	Role	Approval Date
[Name]	[Role]	[Date]
[Name]	[Role]	[Date]

Distribution List

- [Stakeholder 1]
- [Stakeholder 2]
- [Stakeholder 3]

This document and the information contained herein is the confidential and proprietary property of [Your Company Name]. Unauthorized copying or distribution is strictly prohibited.