AURORE
HOLDINGS

# Software Development Life Cycle (SDLC)

## Policy Statement

Aurore Holdings LLC is dedicated to delivering superior software products and services while upholding the highest standards of security and best practices throughout the software development lifecycle (SDLC). Our commitment extends to ensuring that all software solutions meet stringent quality benchmarks and comply with rigorous data protection and security requirements. This Software Development Lifecycle (SDLC) Policy delineates the comprehensive processes and procedures that govern the development, testing, deployment, and maintenance of our software offerings, with a particular focus on safeguarding customer Personally Identifiable Information (PII) and adhering to industry regulations.

## Scope

This policy applies to all software development activities undertaken by Aurore Holdings LLC, including but not limited to:

E-commerce Solutions: Development and maintenance of platforms and systems that facilitate online transactions, product management, and customer interactions.

Logistics Systems: Creation and upkeep of software solutions for managing and optimizing supply chain operations, including our 3PL shipping systems.

Enterprise Development Consulting: Delivery of customized software solutions and consulting services to enhance enterprise operations and efficiency.

IT Services: Development and support of IT systems and infrastructure, including enterprise applications and IT management tools.

Products Across Sectors: Software development related to various product sectors such as food, beauty, home decor, consumables, construction materials, and warehouse management systems.

Website and App Development: Design, development, and maintenance of websites and mobile applications tailored to our diverse business needs.

End-of-Life Software Libraries: Management of software libraries that are no longer in active development but continue to be distributed under commercial licenses.

The scope of this policy encompasses every phase of the software development lifecycle, from initial requirements gathering and design to rigorous testing, deployment, and ongoing maintenance. It also covers the handling and protection of customer PII data, ensuring compliance with applicable data protection regulations and internal security protocols.

Key aspects of this policy include:

Requirements Gathering: Comprehensive documentation and analysis of stakeholder needs, including data protection and security requirements.

Design and Development: Implementation of secure coding practices, with attention to data encryption, access controls, and the use of dummy data in development environments.

Testing: Rigorous testing protocols to ensure functionality, security, and compliance, including specialized tests for data protection.

Deployment: Secure deployment practices, including the use of encrypted communication channels and data handling procedures to protect PII.

Maintenance: Ongoing support and updates, with a focus on addressing security vulnerabilities, optimizing performance, and ensuring compliance with evolving regulations.

Data Handling: Strict policies for the encryption, access control, and deletion of PII data in production environments, as well as the use of dummy data in non-production environments to simulate real data flows without compromising security.

## 1. Planning and Requirement Analysis

Objectives: Define the project goals, scope, deliverables, and constraints.

Stakeholders: Identify and engage stakeholders including management, users, and development teams.

Requirement Gathering: Collect and document detailed requirements from stakeholders for e-commerce, logistics, consulting, IT services, and product offerings.

Feasibility Study: Assess technical, operational, and economic feasibility.

Risk Analysis: Identify potential risks and create mitigation plans.

## 2. System Design

High-Level Design (HLD): Define system architecture, modules, data flow, and interfaces. Ensure the design includes security measures for PII data handling.

Low-Level Design (LLD): Detail the specific functions of each module, including data structures, algorithms, and interface design.

UI/UX Design: Create wireframes and mockups for user interfaces, focusing on user experience and accessibility.

## 3. Implementation

Environment Setup: Configure development, testing, and production environments.

Development Environment: Use dummy data to simulate real customer PII data.

Production Environment: Implement strict access controls, encryption, and deletion policies for real customer PII data.

Coding: Develop the application in iterations using agile methodologies. Regularly review code for quality and security compliance.

Integration: Integrate different modules and ensure they work together seamlessly.

## 4. Testing

Unit Testing: Test individual components for functionality.

Integration Testing: Verify that integrated components work together correctly.

System Testing: Conduct end-to-end testing of the entire system.

User Acceptance Testing (UAT): Validate the system with stakeholders to ensure it meets requirements.
Security Testing: Focus on PII data protection, ensuring encryption, restricted access, and data deletion processes are functioning correctly.

## 5. Deployment

Staging Environment: Test the deployment process and perform final checks using a staging environment.
Production Deployment: Deploy the application to the production environment with real customer data.
Monitoring: Continuously monitor system performance, security, and user feedback.

## 6. Maintenance

Bug Fixes: Identify and fix any issues that arise post-deployment.

Updates and Enhancements: Regularly update the system with new features and improvements based on user feedback and market trends.
Performance Tuning: Optimize system performance and scalability.

Security Audits: Conduct regular security audits to ensure ongoing compliance with data protection regulations.

## 7. Data Handling and Security

Data Encryption: Implement robust encryption mechanisms for PII data in storage and transit.

Access Controls: Use role-based access controls to restrict access to PII data.

Data Deletion: Ensure PII data is deleted from the production environment once it is no longer needed.

Dummy Data: Utilize dummy data sets in the development environment to simulate real data flows without compromising security.

## **8.** Documentation

User Documentation: Provide detailed user manuals and help guides.

Technical Documentation: Maintain comprehensive technical documentation including system design, codebase, APIs, and data handling procedures.

Compliance Documentation: Ensure documentation complies with relevant data protection regulations such as GDPR and CCPA.

## **9.** Review and Retrospective

Post-Implementation Review: Assess the project's success against initial objectives and requirements.

Lessons Learned: Document lessons learned and best practices for future projects.

Continuous Improvement: Implement feedback loops to continuously improve processes and products.