

SoleMate – E-Commerce Platform for Shoe Retail

Objective:

Design, develop, and deploy a fully functional e-commerce application for "SoleMate" – a shoe company aiming to expand its retail presence into the online marketplace. The application will provide an intuitive, secure, and responsive platform for customers to browse, purchase, and review products, with integrated inventory, order tracking, and analytics for business operations.

Business Context:

With increasing competition in the footwear industry and consumer preference for online shopping, SoleMate needs a robust e-commerce platform to cater to its customers across desktop and mobile devices. The platform must support scalability, secure transactions, seamless user experience, and integration with backend systems like inventory and CRM.

Scope of Work (Aligned to SDLC Phases)

1. Requirement Gathering & Analysis

Activities:

- Conduct stakeholder interviews (Sales, Marketing, Warehouse, IT teams) - Assume details
- Define business goals and target audience
- Capture functional requirements:
 - User registration & authentication
 - Product catalog with filtering and search
 - Product detail pages with reviews and ratings
 - Shopping cart and checkout
 - Payment gateway integration (Stripe/PayPal/UPI)
 - Order tracking & history
 - Customer profile management
 - Admin dashboard for product and order management
 - Inventory management & stock alerts
 - Promotional features (discount codes, offers)
- Capture non-functional requirements:
 - Performance: Load <2 seconds

- Scalability: Handle 50,000 concurrent users
- Security: PCI-DSS compliance, data encryption
- Compatibility: Mobile, tablet, and desktop
- Define acceptance criteria for each feature

Deliverables:

- SRS (Software Requirements Specification)
- Use case diagrams & user stories
- Requirements traceability matrix (RTM)

2. Planning & Estimation

Activities:

- Select tech stack:
 - **Frontend:** React.js, Next.js for SSR, TailwindCSS or any other technology
 - **Backend:** Node.js with Express or any other technology
 - **Database:** PostgreSQL for relational data, Redis for caching or any other technology
 - **Payment:** Stripe API or any other technology
 - **Cloud Deployment:** AWS (EC2, RDS, S3, CloudFront) or any other technology
- Create project plan with milestones
- Resource allocation (Frontend Devs, Backend Devs, QA, DevOps)
- Risk analysis and mitigation plan
- Budget estimation

Deliverables:

- Project plan with Gantt chart
- Resource allocation matrix
- Risk register
- Cost estimation report

3. System Design

Activities:

- **High-Level Architecture:**
 - MVC structure with API Gateway
 - Microservices for product, order, and payment services
 - CDN for static content delivery
- **Database Design:**
 - ER diagram for products, users, orders, payments, inventory
- **API Design:**
 - REST APIs for mobile and web
 - Swagger documentation
- **UI/UX Design:**
 - Wireframes for key screens (Figma)
 - Interactive prototypes
 - Accessibility compliance check (WCAG 2.1)

Deliverables:

- HLD & LLD documents
- ER diagram
- API documentation
- UI wireframes and prototypes

4. Development

Activities:

- Setup development environment & version control (GitHub/GitLab)
- Implement features in sprints (Agile methodology)
- Code reviews and merge approvals
- Integrate APIs with frontend components
- Unit testing for each module
- Build responsive layouts

Deliverables:

- Source code repository
- API endpoints with documentation
- Unit test reports

5. Testing

Activities:

- **Functional Testing:** Verify features against SRS
- **Integration Testing:** Ensure smooth interaction between modules
- **Performance Testing:** Load and stress tests
- **Security Testing:** SQL injection, XSS, CSRF checks
- **UAT (User Acceptance Testing):** Real-world usage simulation

Deliverables:

- Test plan & test cases
- Bug/defect reports in Jira
- UAT sign-off document

6. Deployment

Activities:

- Setup CI/CD pipelines with GitHub Actions/Jenkins
- Deploy backend to AWS EC2, frontend to AWS Amplify/CloudFront
- Configure domain, SSL, and CDN caching
- Data migration from legacy system (if any)
- Final production release

Deliverables:

- Deployment checklist
- Live application URL
- Rollback plan

7. Maintenance & Support

Activities:

- Monitor application using AWS CloudWatch & New Relic
- Apply security patches & framework updates

- Fix post-release defects
- Add new features based on customer feedback

Deliverables:

- Maintenance logs
- Performance reports
- Feature enhancement backlog

Project Evaluation Metrics

Area	Metric	Target
Requirement Quality	% of requirements covered in RTM	$\geq 100\%$
Design Quality	% of design changes after development start	$\leq 5\%$
Development Quality	Code coverage (unit tests)	$\geq 80\%$
Performance	Avg. page load time	$\leq 2s$
Security	Vulnerabilities detected post-launch	0 critical
Testing Efficiency	Defect detection ratio (pre vs. post-release)	$\geq 90\%$
Deployment	Downtime during deployment	≤ 30 min
User Experience	Customer satisfaction score (post-UAT)	$\geq 8/10$
Maintenance	Mean time to resolve critical issues	≤ 4 hrs
Business Impact	Increase in online sales in 3 months	$\geq 25\%$

SoleMate – E-Commerce Application Project Evaluation Scoring Metrics

Total Score: 100 points (Weighting by SDLC phase)

1. Requirement Gathering & Analysis – 15 Points

Metric	Description	Target	Max Points	Scoring Guide
Requirements Coverage	% of requirements documented in RTM	100%	5	5 = Full coverage, 3–4 = Minor gaps, <3 = Major gaps

Clarity & Completeness	Clear, measurable, non-ambiguous requirements	$\geq 90\%$ clear	5	5 = Fully clear, 3–4 = Mostly clear, <3 = Unclear
Stakeholder Alignment	% of requirements approved by stakeholders	$\geq 95\%$	5	5 = Full sign-off, 3–4 = Minor disagreements, <3 = Major disagreements

2. Planning & Estimation – 10 Points

Metric	Description	Target	Max Points	Scoring Guide
Project Plan Accuracy	Milestones realistic and achievable	$\leq 5\%$ deviation	4	4 = On time, 2–3 = Slight delay, <2 = Major delay
Resource Allocation	Skills match project needs	$\geq 90\%$ alignment	3	3 = Fully matched, 2 = Mostly matched, 1 = Poor allocation
Risk Management	Risk register completeness	100% high risks covered	3	3 = All covered, 2 = Mostly covered, 1 = Gaps

3. System Design – 15 Points

Metric	Description	Target	Max Points	Scoring Guide
Architecture Completeness	HLD & LLD cover all modules	100%	5	5 = Complete, 3–4 = Minor gaps, <3 = Major gaps
Database Design Quality	Normalization, indexing, relationships	Fully optimized	5	5 = Optimized, 3–4 = Some optimization, <3 = Poor
UI/UX Quality	Accessibility, responsiveness, usability	WCAG compliant	5	5 = Excellent, 3–4 = Good, <3 = Poor

4. Development – 20 Points

Metric	Description	Target	Max Points	Scoring Guide
--------	-------------	--------	------------	---------------

Code Quality	Readability, modularity, adherence to standards	$\geq 90\%$ compliant	5	5 = Excellent, 3–4 = Good, <3 = Poor
Feature Implementation	% of planned features implemented	$\geq 95\%$	5	5 = Full, 3–4 = Minor missing, <3 = Major missing
Integration Quality	API & module connectivity	Zero major failures	5	5 = Smooth, 3–4 = Few minor issues, <3 = Frequent failures
Unit Test Coverage	% of code covered by unit tests	$\geq 80\%$	5	5 = Excellent, 3–4 = Adequate, <3 = Low coverage

5. Testing – 15 Points

Metric	Description	Target	Max Points	Scoring Guide
Test Case Coverage	% of requirements tested	100%	5	5 = Full coverage, 3–4 = Minor gaps, <3 = Major gaps
Defect Detection Efficiency	% of defects found pre-release	$\geq 90\%$	5	5 = Excellent, 3–4 = Good, <3 = Low
UAT Success Rate	% of UAT cases passed	$\geq 95\%$	5	5 = Excellent, 3–4 = Good, <3 = Poor

6. Deployment – 10 Points

Metric	Description	Target	Max Points	Scoring Guide
Deployment Success	First-attempt success rate	100%	4	4 = Perfect, 3 = Minor issues, <3 = Major issues
Downtime	Deployment downtime duration	≤ 30 min	3	3 = Within limit, 2 = Slightly over, 1 = Long outage
Rollback Preparedness	Tested rollback plan	Yes	3	3 = Fully tested, 2 = Partially tested, 1 = Not tested

7. Maintenance & Support – 10 Points

Metric	Description	Target	Max Points	Scoring Guide
Issue Resolution Time	Avg. time to fix critical bugs	≤ 4 hrs	4	4 = Fast, 3 = Acceptable, <3 = Slow
Feature Enhancement Delivery	Timely delivery of updates	$\geq 90\%$ on schedule	3	3 = On time, 2 = Slight delay, 1 = Late
Performance Monitoring	Regular monitoring reports	Weekly	3	3 = Consistent, 2 = Sporadic, 1 = None

8. Business Impact & User Experience – 5 Points

Metric	Description	Target	Max Points	Scoring Guide
Customer Satisfaction	Post-launch survey rating	$\geq 8/10$	3	3 = High, 2 = Medium, 1 = Low
Sales Growth	Increase in online sales in 3 months	$\geq 25\%$	2	2 = Target met, 1 = Partial, 0 = None

Scoring Summary Table

SDLC Phase	Max Points
Requirement Gathering	15
Planning & Estimation	10
System Design	15
Development	20
Testing	15
Deployment	10
Maintenance & Support	10
Business Impact & UX	5
Total	100

Important Note for Assignment Submission:

Learners should focus primarily on the **Deliverables** listed in each section for completion and submission. The **Activities** sections serve as reference material and process guidance to help learners understand the methodology and approach, enabling them to produce the highest quality and most comprehensive **Deliverables**.

Understanding the Activities will provide valuable context, purpose, and requirements for each deliverable, but the final submission should consist of the specific documents and artifacts outlined in the Deliverables sections of each phase.