Product Requirements Document

[Product Name]

Document Owner: [Your Name]

Last Updated: [Date]

Version: [1.0]

Status: [Draft / In Review / Approved]

Change Log

Track all changes to this PRD with date, author, and description of changes.

Date	Author	Changes

1. Problem Statement

Describe the customer problem in concrete terms. Use actual customer quotes and behavioral data. Avoid market analysis or competitive positioning.

Customer Problem

[Describe the specific task customers cannot complete or complete inefficiently. Example: Small business owners spend 3.5 hours weekly reconciling inventory across multiple sales channels.]

Customer Quotes

[Include direct quotes from customers describing their pain points]

Supporting Metrics

- Support ticket counts: [number]
- Time spent on workarounds: [hours per week]
- Revenue lost: [dollar amount]
- Other relevant metrics: [specify]

2. Success Metrics

Define measurable success criteria. Avoid vague goals. Each metric should connect to a business outcome.

Metric	Target	Business Impact
Example: Inventory reconciliation time	From 3.5 hours to 30 minutes weekly	$X saved hours × TAM

3. User Personas and Scenarios

Document who will use this product and how they'll use it in their actual work environment. Focus on workflow context, not demographics.

Primary User Persona

Name/Role: [e.g., Sarah Chen, Small Business Owner]
Current Workflow: [Describe current process]
Pain Points: [List main challenges]

User Scenario

[Create a narrative showing the complete user journey from problem to solution. Example: Sarah receives an order notification on her phone while at a craft fair. She opens our app, confirms inventory, and marks the item as sold. The system automatically updates stock levels across all channels in 15 seconds.]

4. Feature Requirements

Structure features in a table format that engineering and design teams can reference quickly.

Feature	User Story	Priority	Success Criteria	Dependencies
Example: Real-time sync	As a seller, I want stock updated across channels	P0	Updates within 2 seconds	API access

5. Technical Specifications

Document technical constraints without prescribing solutions. Include performance benchmarks, security requirements, and integration points.

Performance Requirements

- Concurrent users: [number]
- Transactions per second: [number]
- Uptime requirement: [percentage]
- Acceptable latency: [milliseconds]

Security Requirements

[List security standards, authentication methods, data encryption requirements]

Integration Points

- External APIs: [list with rate limits]
- Data storage: [requirements and volume]
- Third-party services: [specify]

6. Implementation Timeline

Break the project into measurable milestones that deliver incremental value. Each milestone should produce something customers can use.

Phase 1: [Weeks X-Y]

Goal: [What will be delivered]
User Base: [Number of beta users]
Key Features: [List main features]

Phase 2: [Weeks X-Y]

Goal: [What will be delivered]
User Base: [Number of users]
Key Features: [List main features]

Phase 3: [Weeks X-Y]

Goal: [General availability launch]
Key Features: [List main features]

7. Risk Assessment Matrix

Document potential risks and mitigation strategies.

Risk	Probability	Impact	Mitigation Strategy
Example: API changes	Medium	High	Build abstraction layer

8. Stakeholder Communication Plan

Define who needs what information and when. Create a RACI matrix for decision accountability.

Decision Type	Responsible	Accountable	Consulted/Informed
Feature prioritization			
Technical architecture			

9. Testing and Validation Framework

Specify how you'll validate that the product solves the original problem.

Quantitative Validation

[Define analytics tracking for success metrics from day one]

Qualitative Validation

[Plan user interviews at each milestone]

Failure Criteria

[Define what failure looks like and when to pivot. Example: If reconciliation time hasn't decreased by 50% after Phase 2, pause Phase 3 to investigate.]

10. AI Integration Guidelines (If Applicable)

Include if your product uses AI components.

- Accuracy thresholds: [percentage]
- Acceptable error rates: [percentage]
- Fallback mechanisms: [describe]
- Training data requirements: [specify]
- Bias testing protocols: [describe]

11. Migration and Rollback Plans

Document how existing users will transition to the new system.

Migration Strategy

[Include data migration scripts, training materials, and support documentation]

Rollback Criteria

[Define criteria for rolling back. If critical bugs appear, how quickly can you revert? What data must be preserved?]

Document created using the PRD Template