

# SOFTWARE REQUIREMENTS

## SPECIFICATION

DOXEN

AI-Powered Business Requirements Document Generator

Version	1.0
Date	February 21, 2026
Status	Final
Document Type	Software Requirements Specification

## **TABLE OF CONTENTS**

- 1. Introduction
  - 1.1 Purpose
  - 1.2 Scope
  - 1.3 Definitions and Acronyms
  - 1.4 References
- 2. Overall Description
  - 2.1 Product Perspective
  - 2.2 Product Functions
  - 2.3 User Classes and Characteristics
  - 2.4 Operating Environment
  - 2.5 Design and Implementation Constraints
- 3. System Features
  - 3.1 User Authentication
  - 3.2 Multi-Source Data Integration
  - 3.3 AI-Powered Requirement Extraction
  - 3.4 BRD Generation
  - 3.5 Natural Language Refinement
  - 3.6 Traceability Management
  - 3.7 Document Export
- 4. External Interface Requirements
  - 4.1 User Interfaces
  - 4.2 Hardware Interfaces
  - 4.3 Software Interfaces
  - 4.4 Communication Interfaces
- 5. Non-Functional Requirements
  - 5.1 Performance Requirements

5.2 Safety Requirements

5.3 Security Requirements

5.4 Software Quality Attributes

6. Other Requirements

7. Appendix

# 1. INTRODUCTION

## 1.1 Purpose

This Software Requirements Specification (SRS) document provides a comprehensive description of the Doxen system - an AI-powered Business Requirements Document (BRD) generator. This document is intended for project stakeholders, development teams, quality assurance personnel, and system architects. It defines functional and non-functional requirements, system constraints, and interface specifications necessary for the successful development and deployment of Doxen.

## 1.2 Scope

Doxen is a web-based application that automates the creation of Business Requirements Documents by intelligently processing scattered business communications from multiple sources including emails, Slack messages, meeting transcripts, and uploaded documents. The system leverages artificial intelligence to extract, categorize, and structure requirements while maintaining complete traceability to source materials.

### Key Capabilities:

- Multi-source data integration (Slack, documents, email threads, transcripts)
- AI-powered requirement extraction and categorization
- Automated generation of structured BRD documents
- Natural language document refinement
- Complete requirement traceability
- Export to PDF and Markdown formats
- User authentication and project management

## 1.3 Definitions, Acronyms, and Abbreviations

<b>BRD</b>	Business Requirements Document - A formal document describing business solution requirements
<b>AI</b>	Artificial Intelligence
<b>LLM</b>	Large Language Model
<b>RLS</b>	Row Level Security - Database security model
<b>API</b>	Application Programming Interface
<b>SRS</b>	Software Requirements Specification
<b>UI/UX</b>	User Interface / User Experience
<b>PDF</b>	Portable Document Format
<b>OAuth</b>	Open Authorization - Authentication protocol

## 1.4 References

- IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications
- Supabase Documentation: <https://supabase.com/docs>
- React 18 Documentation: <https://react.dev>
- OpenAI API Documentation: <https://platform.openai.com/docs>
- Slack API Documentation: <https://api.slack.com>

## 2. OVERALL DESCRIPTION

### 2.1 Product Perspective

Doxen is a standalone web application that operates within the following system context:

#### System Dependencies:

- Frontend Layer: React 18 application served via modern web browsers
- Backend Layer: Supabase (PostgreSQL database, Authentication, Storage, Edge Functions)
- AI Layer: Lovable AI Gateway (Gemini Flash) or alternative LLM providers
- Integration Layer: Slack Web API, Email systems

### 2.2 Product Functions

Doxen provides the following high-level functions:

<b>User Management</b>	Account creation, authentication (email/password, Google OAuth), profile management
<b>Project Management</b>	Create, read, update, delete projects with status tracking
<b>Data Source Integration</b>	Upload documents, import Slack channels, paste transcripts and text
<b>Requirement Extraction</b>	AI-powered analysis to identify and categorize requirements from sources
<b>BRD Generation</b>	Automated creation of structured, professional BRD documents
<b>Document Refinement</b>	Natural language commands to modify and enhance generated documents
<b>Traceability</b>	Complete mapping of requirements back to source excerpts with confidence scores
<b>Export &amp; Sharing</b>	PDF and Markdown export with professional formatting

### 2.3 User Classes and Characteristics

<b>Business Analyst</b>	Primary user who creates BRDs regularly. Needs efficient requirement gathering.	Medium
<b>Product Manager</b>	Uses system to document product requirements from stakeholder communications.	Medium
<b>Project Manager</b>	Converts project communications into	Low to Medium

	formal requirements documents.	
<b>Technical Lead</b>	Reviews generated BRDs for technical feasibility and completeness.	High
<b>Stakeholder</b>	May view and provide feedback on generated requirements.	Low

## 2.4 Operating Environment

Doxen operates in the following environment:

- Client Side: Modern web browsers (Chrome 90+, Firefox 88+, Safari 14+, Edge 90+)
- Server Side: Cloud-hosted Supabase infrastructure
- Database: PostgreSQL 14+ with Row Level Security enabled
- Edge Functions: Deno runtime for serverless function execution
- AI Services: Lovable AI Gateway or compatible LLM API endpoints
- Storage: Supabase Storage for document uploads
- Network: HTTPS required for all communications

## 2.5 Design and Implementation Constraints

- Technology Stack: Must use React 18, TypeScript, Supabase, and compatible LLM services
- Browser Compatibility: Must support modern browsers with ES6+ JavaScript
- Security: Must implement Row Level Security (RLS) for all database tables
- Performance: Must maintain 60fps on mobile devices for UI interactions
- Authentication: Must use Supabase Auth with support for email and OAuth providers
- API Rate Limits: Must respect LLM provider rate limits and implement appropriate throttling
- Data Privacy: User data must be isolated; users can only access their own projects
- Accessibility: Must meet WCAG 2.1 Level AA standards

## 3. SYSTEM FEATURES

### 3.1 User Authentication

#### 3.1.1 Description and Priority

**Priority:** High

User authentication provides secure access control to the Doxen platform. Users must authenticate before accessing any features. The system supports multiple authentication methods to accommodate different user preferences and organizational requirements.

#### 3.1.2 Functional Requirements

- FR-AUTH-001: System SHALL support email and password authentication
- FR-AUTH-002: System SHALL support Google OAuth authentication
- FR-AUTH-003: System SHALL enforce password complexity requirements (minimum 8 characters)
- FR-AUTH-004: System SHALL send email confirmation for new account registration
- FR-AUTH-005: System SHALL require unique username selection on first login
- FR-AUTH-006: System SHALL maintain user session for 7 days with option to extend
- FR-AUTH-007: System SHALL provide password reset functionality via email
- FR-AUTH-008: System SHALL display appropriate error messages for failed authentication attempts

### 3.2 Multi-Source Data Integration

#### 3.2.1 Description and Priority

**Priority:** High

The system must accept business communications from multiple sources to consolidate scattered requirements information. This feature is fundamental to the product's value proposition.

#### 3.2.2 Functional Requirements

- FR-INT-001: System SHALL support document upload (PDF, DOCX, TXT formats)
- FR-INT-002: System SHALL support Slack channel message import via OAuth bot token
- FR-INT-003: System SHALL support direct text paste for meeting transcripts
- FR-INT-004: System SHALL store uploaded documents in secure cloud storage
- FR-INT-005: System SHALL maintain metadata for each data source (name, type, upload date)
- FR-INT-006: System SHALL support multiple data sources per project
- FR-INT-007: System SHALL display list of all sources for a project with preview capability
- FR-INT-008: System SHALL allow users to delete data sources



- FR-INT-009: System SHALL extract text content from uploaded documents automatically

### **3.3 AI-Powered Requirement Extraction**

#### **3.3.1 Description and Priority**

**Priority:** Critical

The core AI functionality that analyzes data sources and extracts structured requirements. This feature differentiates Doxen from manual documentation methods.

#### **3.3.2 Functional Requirements**

- FR-EXT-001: System SHALL analyze all data sources using AI/LLM technology
- FR-EXT-002: System SHALL categorize extracted requirements as: Functional, Non-Functional, Business Rules, User Stories
- FR-EXT-003: System SHALL assign confidence score (0-1) to each extracted requirement
- FR-EXT-004: System SHALL preserve source excerpt for each requirement
- FR-EXT-005: System SHALL filter conversational noise and extract only project-relevant content
- FR-EXT-006: System SHALL generate title and description for each requirement
- FR-EXT-007: System SHALL display progress indicator during extraction process
- FR-EXT-008: System SHALL handle extraction failures gracefully with error messaging
- FR-EXT-009: System SHALL allow users to edit or remove extracted requirements
- FR-EXT-010: System SHALL support re-extraction if new sources are added

## 3.4 BRD Generation

### 3.4.1 Description and Priority

**Priority:** Critical

Automated generation of structured Business Requirements Documents from extracted requirements. This is the primary deliverable of the system.

### 3.4.2 Functional Requirements

- FR-BRD-001: System SHALL generate structured BRD with standard sections
- FR-BRD-002: BRD SHALL include: Executive Summary, Project Overview, Functional Requirements, Non-Functional Requirements, Assumptions & Constraints, Success Criteria
- FR-BRD-003: System SHALL maintain version history of generated BRDs
- FR-BRD-004: System SHALL display BRD in readable format within the application
- FR-BRD-005: System SHALL generate professional formatting and structure
- FR-BRD-006: System SHALL support multiple BRD versions per project
- FR-BRD-007: System SHALL indicate BRD generation progress to user
- FR-BRD-008: System SHALL handle generation failures with appropriate error messages

## 3.5 Natural Language Refinement

### 3.5.1 Description and Priority

**Priority:** Medium

Allows users to refine generated BRDs using simple natural language commands, making the system more flexible and user-friendly.

### 3.5.2 Functional Requirements

- FR-REF-001: System SHALL accept natural language commands to modify BRD
- FR-REF-002: System SHALL support commands like 'Add a security section', 'Expand technical requirements'
- FR-REF-003: System SHALL apply refinements to create new BRD version
- FR-REF-004: System SHALL maintain context of previous refinements
- FR-REF-005: System SHALL display refinement progress to user
- FR-REF-006: System SHALL preserve previous versions when creating refined versions

## 3.6 Traceability Management

### 3.6.1 Description and Priority

**Priority:** High

Provides complete traceability from requirements back to source materials, enabling validation and audit capabilities.

### 3.6.2 Functional Requirements

- FR-TRC-001: System SHALL maintain link between each requirement and its source
- FR-TRC-002: System SHALL display traceability table showing requirement-to-source mapping
- FR-TRC-003: System SHALL include source excerpt for each requirement
- FR-TRC-004: System SHALL display confidence score in traceability view
- FR-TRC-005: System SHALL allow navigation from requirement to full source document
- FR-TRC-006: System SHALL support filtering and searching in traceability table

## 3.7 Document Export

### 3.7.1 Description and Priority

**Priority:** High

Export functionality allows users to download and share generated BRDs in standard formats.

### 3.7.2 Functional Requirements

- FR-EXP-001: System SHALL support PDF export of generated BRDs
- FR-EXP-002: System SHALL support Markdown export of generated BRDs
- FR-EXP-003: System SHALL maintain professional formatting in exported documents
- FR-EXP-004: System SHALL include all BRD sections in exported document
- FR-EXP-005: System SHALL generate downloadable file with appropriate naming
- FR-EXP-006: System SHALL indicate export progress during generation

## 4. EXTERNAL INTERFACE REQUIREMENTS

### 4.1 User Interfaces

The user interface shall be web-based and responsive, supporting desktop, tablet, and mobile devices. The interface follows modern design principles with clean aesthetics and intuitive navigation.

**Key Interface Components:**

- Landing Page: Public-facing page with product information and call-to-action
- Authentication Pages: Sign in/Sign up forms with social OAuth options
- Dashboard: Grid view of all user projects with status indicators
- Project Workspace: Tabbed interface with sections for Sources, Requirements, BRD, and Traceability
- Data Sources Tab: Upload zone, Slack import controls, text paste area
- Requirements Tab: List view of extracted requirements with filtering
- BRD Tab: Document viewer with refinement input and export controls
- Traceability Tab: Table view showing requirement-to-source mappings

**UI Requirements:**

- UI-001: Interface SHALL be responsive and adapt to screen sizes from 320px to 4K displays
- UI-002: Interface SHALL maintain consistent design language using Tailwind CSS
- UI-003: Interface SHALL provide visual feedback for all user actions
- UI-004: Interface SHALL use loading indicators for asynchronous operations
- UI-005: Interface SHALL display appropriate error messages for failures
- UI-006: Interface SHALL achieve 60fps performance on mobile devices

### 4.2 Hardware Interfaces

As a web-based application, Doxen does not have direct hardware interfaces. The application relies on standard web browser capabilities to access device hardware (camera for uploads, storage for downloads, etc.).

### 4.3 Software Interfaces

<b>Supabase Auth</b>	User authentication and authorization	REST API / JWT
<b>Supabase Database</b>	Data persistence (PostgreSQL)	PostgREST API
<b>Supabase Storage</b>	Document file storage	REST API
<b>Supabase Edge Functions</b>	Serverless compute for AI operations	HTTP/HTTPS
<b>Lovable AI Gateway</b>	AI/LLM services for requirement extraction	REST API

<b>Slack Web API</b>	Channel and message retrieval	REST API / OAuth 2.0
<b>Browser Storage</b>	Client-side state management	LocalStorage / SessionStorage

#### 4.4 Communication Interfaces

- COM-001: All client-server communication SHALL use HTTPS/TLS 1.2 or higher
- COM-002: API requests SHALL include appropriate authentication headers
- COM-003: System SHALL implement CORS policies for cross-origin requests
- COM-004: WebSocket connections SHALL be used for real-time updates where applicable
- COM-005: System SHALL handle network failures gracefully with retry logic

## 5. NON-FUNCTIONAL REQUIREMENTS

### 5.1 Performance Requirements

- NFR-PERF-001: Page load time SHALL be under 3 seconds on 4G connection
- NFR-PERF-002: UI interactions SHALL maintain 60fps on mobile devices
- NFR-PERF-003: Requirement extraction SHALL process up to 50 sources within 5 minutes
- NFR-PERF-004: BRD generation SHALL complete within 2 minutes for typical projects
- NFR-PERF-005: System SHALL support up to 1000 concurrent users
- NFR-PERF-006: Database queries SHALL return results within 500ms for 95th percentile
- NFR-PERF-007: File uploads SHALL support documents up to 50MB

### 5.2 Safety Requirements

- NFR-SAF-001: System SHALL implement automatic data backup daily
- NFR-SAF-002: System SHALL maintain data redundancy across multiple availability zones
- NFR-SAF-003: System SHALL provide data recovery capability within 4 hours
- NFR-SAF-004: System SHALL log all critical operations for audit purposes

### 5.3 Security Requirements

- NFR-SEC-001: System SHALL encrypt all data in transit using TLS 1.2+
- NFR-SEC-002: System SHALL encrypt all data at rest in database and storage
- NFR-SEC-003: System SHALL implement Row Level Security (RLS) for all database tables
- NFR-SEC-004: System SHALL enforce password complexity (min 8 chars, mixed case, numbers)
- NFR-SEC-005: System SHALL implement rate limiting on API endpoints
- NFR-SEC-006: System SHALL sanitize all user inputs to prevent injection attacks
- NFR-SEC-007: System SHALL implement JWT-based authentication with token expiration
- NFR-SEC-008: System SHALL log and monitor suspicious activities
- NFR-SEC-009: System SHALL comply with GDPR and data protection regulations
- NFR-SEC-010: System SHALL provide secure API key management for integrations

### 5.4 Software Quality Attributes

#### 5.4.1 Reliability

- NFR-REL-001: System SHALL maintain 99.9% uptime
- NFR-REL-002: System SHALL handle graceful degradation if AI services are unavailable
- NFR-REL-003: System SHALL implement automatic retry for failed operations

#### 5.4.2 Availability

- NFR-AVL-001: System SHALL be accessible 24/7 with planned maintenance windows
- NFR-AVL-002: Planned maintenance SHALL not exceed 4 hours per month
- NFR-AVL-003: System SHALL provide status page for service availability

#### 5.4.3 Maintainability

- NFR-MAIN-001: Code SHALL follow TypeScript and React best practices
- NFR-MAIN-002: System SHALL provide comprehensive error logging
- NFR-MAIN-003: System SHALL use version control for all code and configuration
- NFR-MAIN-004: System SHALL maintain API documentation
- NFR-MAIN-005: System SHALL implement continuous integration/deployment pipelines

#### 5.4.4 Portability

- NFR-PORT-001: System SHALL run on major web browsers (Chrome, Firefox, Safari, Edge)
- NFR-PORT-002: System SHALL be responsive across desktop, tablet, and mobile devices
- NFR-PORT-003: System SHALL support multiple LLM providers via configuration

#### 5.4.5 Usability

- NFR-USE-001: New users SHALL be able to create their first BRD within 10 minutes
- NFR-USE-002: System SHALL provide contextual help and tooltips
- NFR-USE-003: Error messages SHALL be clear and actionable
- NFR-USE-004: System SHALL meet WCAG 2.1 Level AA accessibility standards
- NFR-USE-005: System SHALL support keyboard navigation throughout

## 6. OTHER REQUIREMENTS

### 6.1 Data Requirements

- DATA-001: System SHALL store user profiles including username, display name, avatar
- DATA-002: System SHALL store projects with name, description, status, timestamps
- DATA-003: System SHALL store data sources with content, metadata, file paths
- DATA-004: System SHALL store extracted requirements with categories, confidence scores
- DATA-005: System SHALL store BRDs as versioned JSONB objects
- DATA-006: System SHALL implement soft deletes for critical data
- DATA-007: System SHALL maintain data integrity through foreign key constraints

### 6.2 Integration Requirements

- INT-001: Slack integration SHALL use OAuth 2.0 for authorization
- INT-002: Slack bot SHALL require invitation to channels before reading history
- INT-003: System SHALL handle Slack API rate limits appropriately
- INT-004: System SHALL support future integration with additional communication platforms

### 6.3 Scalability Requirements

- SCALE-001: System SHALL support horizontal scaling of edge functions
- SCALE-002: Database SHALL support read replicas for query optimization
- SCALE-003: Storage SHALL scale automatically with usage
- SCALE-004: System architecture SHALL support multi-region deployment

### 6.4 Documentation Requirements

- DOC-001: System SHALL provide user documentation including getting started guide
- DOC-002: System SHALL provide API documentation for developers
- DOC-003: System SHALL maintain technical architecture documentation
- DOC-004: System SHALL provide troubleshooting guides
- DOC-005: System SHALL include inline help within the application



## 7. APPENDIX

### 7.1 Database Schema

<b>profiles</b>	User profile information	id, user_id, username, display_name, avatar_url
<b>projects</b>	User projects	id, user_id, name, description, status
<b>data_sources</b>	Imported source materials	id, project_id, name, source_type, content, file_path
<b>extracted_requirements</b>	AI-extracted requirements	id, project_id, source_id, category, title, description, confidence_score
<b>generated_brds</b>	Generated BRD documents	id, project_id, content (JSONB), version

### 7.2 Technology Stack

<b>Frontend Framework</b>	React	18.x
<b>Build Tool</b>	Vite	Latest
<b>Language</b>	TypeScript	5.x
<b>Styling</b>	Tailwind CSS	3.x
<b>UI Components</b>	shadcn/ui, Radix UI	Latest
<b>Routing</b>	React Router	v6
<b>State Management</b>	TanStack Query	v5
<b>Forms</b>	React Hook Form + Zod	Latest
<b>Backend/Database</b>	Supabase (PostgreSQL)	14+
<b>Authentication</b>	Supabase Auth	Latest
<b>Storage</b>	Supabase Storage	Latest
<b>Edge Functions</b>	Deno Runtime	Latest
<b>AI/LLM</b>	Lovable AI Gateway / OpenAI	API v1
<b>PDF Generation</b>	jsPDF	Latest
<b>Markdown</b>	react-markdown	Latest

### 7.3 Glossary

#### Business Requirements Document (BRD):

A formal document that outlines business solution requirements for a project.

#### Confidence Score:

A numerical value (0-1) indicating the AI's certainty about an extracted requirement.

#### Edge Function:

Serverless function that runs close to the user for low latency.

**Natural Language Processing (NLP):**

AI technique for understanding and processing human language.

**Row Level Security (RLS):**

Database security model that restricts data access at the row level.

**Traceability:**

The ability to trace requirements back to their original source materials.

**Source Excerpt:**

A snippet of text from the original source that supports a requirement.