Software Requirements Specification (SRS)

Bakery Bliss - Artisan Bakery Management System

Document Information

- **Document Title**: Software Requirements Specification for Bakery Bliss
- **Version**: 1.0
- **Date**: July 14, 2025
- **Team**: Team Anonymous
- **Project**: Bakery Bliss - Artisan Bakery Management System

Table of Contents

1. Introduction
2. Overall Description
3. System Features
4. External Interface Requirements
5. Non-Functional Requirements
6. Other Requirements
7. Appendices

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document provides a comprehensive description of the Bakery Bliss system - an artisan bakery management platform that facilitates custom cake design, order management, team collaboration, and financial tracking. This document is intended for developers, project managers, stakeholders, and quality assurance teams involved in the development and maintenance of the system.

1.2 Document Conventions

- **Functional Requirements**: Denoted as FR-XX
- **Non-Functional Requirements**: Denoted as NFR-XX
- **Business Rules**: Denoted as BR-XX
- **User Stories**: Denoted as US-XX

1.3 Intended Audience and Reading Suggestions

This document is intended for:

- **Development Team**: Complete document review for implementation guidance
- **Project Managers**: Focus on Sections 2, 3, and 5 for project planning
- **Stakeholders**: Focus on Sections 1, 2, and 3 for business understanding
- **QA Team**: Focus on Sections 3, 4, and 5 for testing requirements
- **System Administrators**: Focus on Sections 4, 5, and 6 for deployment

1.4 Product Scope

Bakery Bliss is a comprehensive web-based platform designed to streamline bakery operations through:

- **Custom Cake Design Studio**: Interactive cake builder with real-time preview

- **Multi-Role Management**: Customer, Junior Baker, Main Baker, and Administrator workflows
- **Order Management**: Complete order lifecycle from creation to delivery
- **Team Collaboration**: Real-time chat and task assignment system
- **Financial Tracking**: Baker earnings, commission calculation, and payment distribution
- **Quality Control**: Multi-stage approval process for order completion

1.5 References

- IEEE Standard 830-1998 for Software Requirements Specifications
- W3C Web Content Accessibility Guidelines (WCAG) 2.1
- OWASP Security Guidelines for Web Applications
- React 18.3.1 Documentation
- Node.js and Express.js Best Practices
- PostgreSQL Database Documentation

2. Overall Description

2.1 Product Perspective

Bakery Bliss operates as a standalone web application with the following system context:

[SPACE FOR SYSTEM CONTEXT DIAGRAM]

Figure 2.1: System Context Diagram showing external entities and system boundaries

The system integrates with:

- **Database System**: PostgreSQL for data persistence
- **File Storage**: Local file system for design assets and images
- **Authentication System**: Session-based authentication with bcrypt encryption

- **Real-time Communication**: WebSocket connections for chat functionality

2.2 Product Functions

The primary functions of Bakery Bliss include:

2.2.1 User Management

- Multi-role user registration and authentication
- Role-based access control (Customer, Junior Baker, Main Baker, Admin)
- Profile management and career progression

2.2.2 Custom Cake Design

- Interactive visual cake builder
- Layer, shape, color, and decoration selection
- Real-time price calculation
- Design preview and customization

2.2.3 Order Management

- Order creation and processing
- Status tracking and updates
- Assignment to bakers
- Quality control workflow

2.2.4 Team Collaboration

- Real-time messaging system
- Task assignment and tracking
- Team performance monitoring
- Baker-customer communication

2.2.5 Financial Management

- Automatic commission calculation
- Earnings tracking and distribution

- Payment processing
- Financial reporting

[SPACE FOR FUNCTIONAL OVERVIEW DIAGRAM]

Figure 2.2: High-level functional overview of Bakery Bliss system

2.3 User Classes and Characteristics

2.3.1 Customer

- **Description**: End users who place orders for bakery products
- **Technical Expertise**: Basic web browsing skills
- **Primary Activities**: Browse products, design custom cakes, place orders, track progress
- **System Usage**: Occasional to regular use

2.3.2 Junior Baker

- **Description**: Entry-level bakers who execute assigned orders
- **Technical Expertise**: Basic computer skills, bakery knowledge
- **Primary Activities**: View assigned tasks, update order status, communicate with customers
- **System Usage**: Daily use during work hours

2.3.3 Main Baker

- **Description**: Experienced bakers who manage teams and oversee quality
- **Technical Expertise**: Intermediate computer skills, advanced bakery expertise
- **Primary Activities**: Assign orders, manage team, quality control, customer interaction
- **System Usage**: Daily use throughout business operations

2.3.4 Administrator

- **Description**: System administrators managing platform operations
- **Technical Expertise**: Advanced technical and business knowledge

- **Primary Activities**: User management, system configuration, analytics review
- **System Usage**: Regular monitoring and maintenance

[SPACE FOR USER ROLE HIERARCHY DIAGRAM]
Figure 2.3: User role hierarchy and relationships

2.4 Operating Environment

2.4.1 Client-Side Environment

- **Supported Browsers**: Chrome 90+, Firefox 88+, Safari 14+, Edge 90+
- **Mobile Compatibility**: Responsive design for tablets and smartphones
- **Screen Resolutions**: 1024x768 minimum, optimized for 1920x1080
- **Internet Connection**: Broadband connection recommended

2.4.2 Server-Side Environment

- **Operating System**: Cross-platform (Windows, macOS, Linux)
- **Runtime**: Node.js 18+ LTS
- **Database**: PostgreSQL 14+
- **Web Server**: Express.js 4.21+
- **Deployment**: Docker containerization support

2.5 Design and Implementation Constraints

2.5.1 Technical Constraints

- **Programming Language**: TypeScript/JavaScript
- **Frontend Framework**: React 18.3+ with Vite build system
- **Backend Framework**: Express.js with RESTful API architecture
- **Database**: PostgreSQL with Drizzle ORM
- **Authentication**: Session-based with bcrypt encryption

2.5.2 Business Constraints

- **Budget Limitations**: Development within allocated budget constraints

- **Timeline**: Phased delivery approach with MVP and incremental features
- **Compliance**: Food safety regulations and data protection requirements
- **Scalability**: Support for 1000+ concurrent users

2.6 User Documentation

- **User Manual**: Comprehensive guide for all user roles
- **API Documentation**: RESTful API reference for integration
- **Administrator Guide**: System configuration and maintenance procedures
- **Developer Documentation**: Technical implementation details
- **Quick Start Guide**: Onboarding instructions for new users

2.7 Assumptions and Dependencies

2.7.1 Assumptions

- Users have stable internet connections
- Modern web browsers with JavaScript enabled
- Basic computer literacy among all user types
- Availability of product images and design assets

2.7.2 Dependencies

- **External Libraries**: React, Express.js, PostgreSQL drivers
- **Third-party Services**: None (self-contained system)
- **Hardware Dependencies**: Standard web server infrastructure
- **Network Dependencies**: HTTP/HTTPS communication protocols

3. System Features

3.1 User Authentication and Authorization

3.1.1 Feature Description

Priority: High
Risk: High

The system shall provide secure user authentication and role-based authorization for all platform access.

3.1.2 Stimulus/Response Sequences

[SPACE FOR AUTHENTICATION SEQUENCE DIAGRAM]
Figure 3.1: User authentication sequence diagram

3.1.3 Functional Requirements

FR-01: The system shall allow user registration with email, password, and role selection
FR-02: The system shall authenticate users using email and password combination
FR-03: The system shall implement session-based authentication with 24-hour expiration
FR-04: The system shall enforce role-based access control for all system functions
FR-05: The system shall encrypt passwords using bcrypt with salt rounds

3.2 Custom Cake Builder

3.2.1 Feature Description

Priority: High
Risk: Medium

Interactive visual cake design studio allowing customers to create custom cakes with real-time preview and dynamic pricing.

3.2.2 Stimulus/Response Sequences

[SPACE FOR CAKE BUILDER FLOW DIAGRAM]
Figure 3.2: Custom cake builder interaction flow

3.2.3 Functional Requirements

FR-06: The system shall provide layer selection (2-4 layers)
FR-07: The system shall offer shape options (circle, heart, square, oval, hexagon)
FR-08: The system shall provide color scheme selection
FR-09: The system shall allow side design customization
FR-10: The system shall enable upper decoration selection
FR-11: The system shall calculate dynamic pricing based on selections
FR-12: The system shall provide real-time visual preview

3.3 Order Management System

3.3.1 Feature Description

Priority: High
Risk: Medium

Comprehensive order lifecycle management from creation through delivery with status tracking and role-based workflows.

3.3.2 Stimulus/Response Sequences

[SPACE FOR ORDER WORKFLOW DIAGRAM]
Figure 3.3: Order management workflow diagram

3.3.3 Functional Requirements

FR-13: The system shall allow order creation with products and custom cakes
FR-14: The system shall implement order status progression (pending → processing → quality_check → ready → delivered)
FR-15: The system shall enable order assignment to junior bakers by main bakers
FR-16: The system shall provide order tracking for customers
FR-17: The system shall implement quality control workflow

FR-18: The system shall support order cancellation with proper state management

3.4 Real-Time Communication System

3.4.1 Feature Description

Priority: Medium
Risk: Medium

Order-specific chat system enabling real-time communication between customers, junior bakers, and main bakers.

3.4.2 Stimulus/Response Sequences

[SPACE FOR CHAT SYSTEM DIAGRAM]
Figure 3.4: Real-time chat system architecture

3.4.3 Functional Requirements

FR-19: The system shall provide order-specific chat rooms
FR-20: The system shall enable real-time message delivery
FR-21: The system shall support message history and persistence
FR-22: The system shall implement participant management for chat rooms
FR-23: The system shall provide unread message indicators

3.5 Baker Earnings and Payment System

3.5.1 Feature Description

Priority: High
Risk: High

Automated commission calculation and payment distribution system for baker compensation.

3.5.2 Stimulus/Response Sequences

[SPACE FOR PAYMENT FLOW DIAGRAM]
Figure 3.5: Baker payment processing flow

3.5.3 Functional Requirements

FR-24: The system shall calculate baker commissions automatically upon order completion
FR-25: The system shall distribute payments based on predefined commission rates
FR-26: The system shall track total earnings for each baker
FR-27: The system shall provide earnings breakdown and history
FR-28: The system shall implement payment audit trails

3.6 Career Progression System

3.6.1 Feature Description

Priority: Medium
Risk: Low

Application and approval system for role advancement from Customer to Junior Baker to Main Baker.

3.6.2 Functional Requirements

FR-29: The system shall allow customers to apply for junior baker roles
FR-30: The system shall enable junior bakers to apply for main baker promotion
FR-31: The system shall implement application review and approval workflow
FR-32: The system shall automatically update user roles upon approval

3.7 Analytics and Reporting

3.7.1 Feature Description

Priority: Medium
Risk: Low

Comprehensive analytics dashboard for performance monitoring and business intelligence.

3.7.2 Functional Requirements

FR-33: The system shall provide role-specific dashboard analytics
FR-34: The system shall track order completion rates and performance metrics
FR-35: The system shall generate financial reports and earnings summaries
FR-36: The system shall monitor user engagement and platform usage

4. External Interface Requirements

4.1 User Interfaces

4.1.1 General UI Requirements

[SPACE FOR UI MOCKUP DIAGRAMS]
Figure 4.1: Main user interface layouts for different user roles

UI-01: The system shall provide responsive web interface compatible with desktop and mobile devices
UI-02: The system shall implement consistent design system using Tailwind CSS and Radix UI components
UI-03: The system shall support dark/light theme switching
UI-04: The system shall provide accessible interface following WCAG 2.1 guidelines

4.1.2 Role-Specific Interfaces

Customer Interface:

- Product catalog with search and filtering
- Interactive cake builder with real-time preview
- Order tracking and history
- Profile management and application submission

Junior Baker Interface:

- Task dashboard with assigned orders
- Order status update controls
- Chat interface for customer communication

- Earnings tracking display

Main Baker Interface:

- Team management dashboard
- Order assignment interface
- Quality control review system
- Performance analytics overview

Administrator Interface:

- User management console
- System configuration panel
- Analytics and reporting dashboard
- Application approval interface

4.2 Hardware Interfaces

HW-01: The system shall operate on standard web server hardware
HW-02: The system shall support deployment on cloud infrastructure
HW-03: The system shall be compatible with standard database server configurations

4.3 Software Interfaces

4.3.1 Database Interface

SI-01: PostgreSQL 14+ database connection using Drizzle ORM
SI-02: Connection pooling for optimal database performance
SI-03: Automated database migrations using Drizzle Kit

4.3.2 External Libraries

SI-04: React 18.3+ for user interface components
SI-05: Express.js 4.21+ for server-side API implementation
SI-06: TanStack Query for client-side data management
SI-07: Zod for data validation and type safety

4.4 Communication Interfaces

CI-01: RESTful API communication using JSON over HTTP/HTTPS

CI-02: WebSocket connections for real-time chat functionality

CI-03: Session-based authentication using HTTP cookies

CI-04: File upload support for images and design assets

5. Non-Functional Requirements

5.1 Performance Requirements

[SPACE FOR PERFORMANCE METRICS DIAGRAM]

Figure 5.1: System performance benchmarks and targets

NFR-01: The system shall respond to user requests within 200ms for 95% of operations

NFR-02: The system shall support 1000+ concurrent users without performance degradation

NFR-03: The system shall load initial page content within 2 seconds

NFR-04: The system shall handle database queries within 50ms average response time

NFR-05: The system shall maintain 99.9% uptime availability

5.2 Safety Requirements

NFR-06: The system shall implement automatic data backup every 24 hours

NFR-07: The system shall maintain transaction integrity during system failures

NFR-08: The system shall provide graceful degradation during partial service outages

NFR-09: The system shall implement proper error handling to prevent data corruption

5.3 Security Requirements

[SPACE FOR SECURITY ARCHITECTURE DIAGRAM]

Figure 5.2: Security layers and protection mechanisms

NFR-10: The system shall encrypt all user passwords using bcrypt with minimum 10 salt rounds

NFR-11: The system shall implement HTTPS for all client-server communication

NFR-12: The system shall validate and sanitize all user inputs to prevent injection attacks

NFR-13: The system shall implement session timeout after 24 hours of inactivity

NFR-14: The system shall log all security-relevant events for audit purposes

NFR-15: The system shall implement role-based access control for all sensitive operations

5.4 Software Quality Attributes

5.4.1 Reliability

NFR-16: The system shall have a Mean Time Between Failures (MTBF) of 720 hours

NFR-17: The system shall recover from failures within 5 minutes

NFR-18: The system shall maintain data consistency across all operations

5.4.2 Availability

NFR-19: The system shall be available 24/7 with maximum 0.1% downtime

NFR-20: The system shall support planned maintenance windows with advance notification

5.4.3 Maintainability

NFR-21: The system shall be developed with modular architecture for easy maintenance

NFR-22: The system shall include comprehensive documentation for all components

NFR-23: The system shall implement automated testing with minimum 80% code coverage

5.4.4 Portability

NFR-24: The system shall be deployable on multiple operating systems (Windows, macOS, Linux)
NFR-25: The system shall support containerized deployment using Docker

5.5 Business Rules

BR-01: Only authenticated users may access system functionality
BR-02: Junior bakers can only view and update their assigned orders
BR-03: Main bakers can assign orders only to their team members
BR-04: Commission rates are fixed at 15% for junior bakers and 5% for main bakers
BR-05: Orders must progress through quality check before being marked as ready
BR-06: Customer applications for junior baker roles require main baker approval
BR-07: Payment distribution occurs automatically when orders are marked as delivered

6. Other Requirements

6.1 Legal Requirements

LR-01: The system shall comply with data protection regulations (GDPR, CCPA where applicable)
LR-02: The system shall maintain user privacy and consent management
LR-03: The system shall comply with food safety regulations for order processing
LR-04: The system shall maintain audit trails for financial transactions

6.2 Standards Compliance

SC-01: The system shall follow W3C web standards for HTML, CSS, and JavaScript
SC-02: The system shall implement WCAG 2.1 accessibility guidelines

SC-03: The system shall follow RESTful API design principles
SC-04: The system shall adhere to OWASP security best practices

6.3 Cultural and Political Requirements

CR-01: The system shall support multiple currency formats for international expansion
CR-02: The system shall be designed for future localization and internationalization
CR-03: The system shall respect cultural differences in design preferences

6.4 Environmental Requirements

ER-01: The system shall be optimized for energy-efficient operation
ER-02: The system shall minimize resource usage through efficient code and caching

---

7. Appendices

7.1 Glossary

API: Application Programming Interface - methods for communication between software components
Baker Commission: Percentage-based payment for completed orders
Custom Cake: User-designed cake with specific layers, shapes, colors, and decorations
Junior Baker: Entry-level baker role responsible for order execution
Main Baker: Senior baker role responsible for team management and quality control
Order Lifecycle: Complete process from order creation to delivery
Quality Check: Review process ensuring order meets standards before delivery
Role-Based Access: Permission system based on user roles and responsibilities

Session Management: System for maintaining user authentication state
WebSocket: Protocol enabling real-time bidirectional communication

7.2 Analysis Models

[SPACE FOR SYSTEM ARCHITECTURE DIAGRAM]
Figure 7.1: Complete system architecture overview

[SPACE FOR DATABASE SCHEMA DIAGRAM]
Figure 7.2: Database entity relationship diagram

[SPACE FOR API ARCHITECTURE DIAGRAM]
Figure 7.3: RESTful API structure and endpoints

7.3 To Be Determined List

TBD-01: Integration with external payment gateways (Stripe, PayPal)
TBD-02: Mobile application development using React Native
TBD-03: Advanced analytics and business intelligence features
TBD-04: Integration with inventory management systems
TBD-05: Multi-language support and localization
TBD-06: Advanced notification system (email, SMS)
TBD-07: Integration with delivery tracking services
TBD-08: AI-powered design suggestions for cake builder

Document Control

- **Version**: 1.0
- **Last Modified**: July 14, 2025
- **Review Status**: Draft
- **Approval Status**: Pending
- **Next Review Date**: August 14, 2025

Authors: Team Anonymous
Reviewers: [To be assigned]
Approvers: [To be assigned]

This document contains confidential and proprietary information. Distribution is restricted to authorized personnel only.