Software Requirements Specification (SRS) Document

Document Version: 1.0
Date: 13 January 2026
Prepared for: Client
Prepared by: REKI Development Team

IEEE 830-1998 Standard Template for Software Requirements Specification

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document describes the functional and non-functional requirements for the REKI platform - a real-time nightlife discovery application. The intended audience includes stakeholders, developers, testers, and project managers involved in the implementation of the REKI system.

1.2 Document Conventions

Must/Shall: Indicates mandatory requirements

Should: Indicates desirable but not mandatory requirements

May: Indicates optional features

RFC: Reference to external documents

code: Indicates technical specifications

1.3 Scope

REKI is a mobile application platform that connects nightlife venues with users through real-time crowd data, venue atmosphere ("vibe") information, and exclusive offers. The system comprises:

Mobile Application (iOS/Android) for end-users

Business Dashboard for venue management

Admin Panel for system administration

Backend API for data processing and real-time updates

1.4 References

IEEE 830-1998: Recommended Practice for Software Requirements Specifications

ISO/IEC/IEEE 29148:2011 - Systems and software engineering

Google Maps Platform Documentation

Apple App Store Guidelines

GDPR Compliance Requirements

2. Overall Description

2.1 Product Perspective

REKI operates as an independent platform connecting three primary user groups:

text

```
┌─────────────────┐   ┌─────────────────┐
┌─────────────────┐
│  End Users    │──────►│  REKI Platform │◄──────│  Venues      │
│  (Consumers)  │    │          │    │  (Businesses) │
└─────────────────┘   └─────────────────┘
└─────────────────┘
            │
            ▼
        ┌─────────────────┐
        │  Admin       │
        │  (System)     │
        └─────────────────┘
```

System Interfaces:

Google Maps API for geolocation and mapping

Apple/Google authentication services

Payment gateways (future implementation)

Push notification services (APNs, FCM)

2.2 Product Functions

Core Capabilities:

Real-time venue status monitoring and display

Personalized venue discovery based on user preferences

Offer creation, distribution, and redemption

Business analytics and insights

Social interaction features (check-ins, sharing)

Location-based services and navigation

2.3 User Characteristics

2.3.1 End Users:

Age: 21-35 years

Tech-savvy urban residents

Frequent nightlife participants

Mobile-first users

2.3.2 Business Users:

Venue owners/managers

Marketing personnel

Age: 25-55 years

Requires business intelligence

2.3.3 Administrators:

Technical staff

System monitoring capabilities

Content moderation responsibilities

2.4 Constraints

Regulatory: GDPR compliance for EU users

Technical: Must support iOS 14+ and Android 10+

Performance: Response time < 3 seconds for 95% of requests

Data: Must handle Manchester-specific venue data initially

Time: 6-week MVP development timeline

2.5 Assumptions and Dependencies

Assumptions:

Users have smartphones with location services enabled

Venues have internet connectivity for status updates

Manchester has sufficient venue density for MVP testing

Dependencies:

Google Maps Platform API availability

Apple App Store approval process

PostgreSQL database stability

Third-party authentication services

3. Specific Requirements

3.1 Functional Requirements

3.1.1 User Registration and Authentication

ID	Requirement	Priority	Status
FR-001	The system shall allow user registration via email and password	High	To Implement
FR-002	The system shall support social login (Apple, Google)	Medium	To Implement
FR-003	The system shall provide guest access without registration	Low	To Implement
FR-004	The system shall implement JWT-based authentication	High	To Implement
FR-005	The system shall allow password recovery via email	Medium	To Implement

3.1.2 User Profile Management

ID	Requirement	Priority	Status
FR-006	Users shall be able to set preferences (vibes, neighborhoods)	High	To Implement
FR-007	Users shall be able to save favorite venues	Medium	To Implement
FR-008	Users shall be able to view activity history	Low	To Implement
FR-009	Users shall be able to update profile information	Medium	To Implement

3.1.3 Venue Discovery and Search

ID	Requirement	Priority	Status
FR-010	The system shall display venues on an interactive map	High	To Implement
FR-011	The system shall provide list view of venues	High	To Implement
FR-012	Users shall be able to filter venues by busyness level	High	To Implement
FR-013	Users shall be able to filter venues by vibe tags	High	To Implement
FR-014	The system shall sort venues by distance, rating, or trending	Medium	To Implement
FR-015	Users shall be able to search venues by name or location	Medium	To Implement

3.1.4 Real-time Venue Status

ID	Requirement	Priority	Status
FR-016	The system shall display real-time busyness indicators	High	To Implement
FR-017	The system shall show current vibe tags for each venue	High	To Implement
FR-018	The system shall display estimated wait times	Medium	To Implement
FR-019	The system shall show occupancy percentages	Medium	To Implement
FR-020	The system shall indicate special conditions (live music, line at door)	Medium	To Implement

3.1.5 Offer Management

ID	Requirement	Priority	Status
FR-021	Users shall be able to view active offers for venues	High	To Implement
FR-022	Users shall be able to redeem offers via QR code	High	To Implement
FR-023	Business users shall be able to create new offers	High	To Implement
FR-024	Business users shall be able to set offer time windows	Medium	To Implement
FR-025	Business users shall be able to track offer redemptions	Medium	To Implement

3.1.6 Business Dashboard

ID	Requirement	Priority	Status
FR-026	Business users shall be able to update venue status manually	High	To Implement
FR-027	Business users shall be able to view analytics dashboard	High	To Implement
FR-028	The system shall provide footfall tracking	Medium	To Implement
FR-029	Business users shall be able to boost venue visibility	Low	To Implement
FR-030	The system shall forecast venue activity patterns	Low	To Implement

3.1.7 Notification System

ID	Requirement	Priority	Status
FR-031	The system shall send push notifications for venue updates	Medium	To Implement

FR-032	The system shall notify users of new offers	Medium	To Implement
FR-033	Users shall be able to manage notification preferences	Low	To Implement
FR-034	The system shall provide in-app notifications	Medium	To Implement

3.1.8 Social Features

ID	Requirement	Priority	Status
FR-035	Users shall be able to check in at venues	Medium	To Implement
FR-036	Users shall see friend check-ins (opt-in)	Low	To Implement
FR-037	Users shall be able to share venues via external apps	Low	To Implement

3.2 Non-Functional Requirements

3.2.1 Performance Requirements

ID	Requirement	Metric	Priority
NFR-001	Application startup time	< 3 seconds	High
NFR-002	API response time (95th percentile)	< 2 seconds	High
NFR-003	Real-time update latency	< 5 seconds	High
NFR-004	Map loading time	< 4 seconds	Medium
NFR-005	Image loading time	< 2 seconds	Medium

3.2.2 Reliability Requirements

ID	Requirement	Metric	Priority
NFR-006	System availability	99.5% uptime	High
NFR-007	Data persistence reliability	99.9% successful saves	High
NFR-008	Error recovery time	< 30 seconds	Medium
NFR-009	Data backup frequency	Daily automatic backups	High

3.2.3 Security Requirements

ID	Requirement	Priority
NFR-010	All user data must be encrypted in transit (TLS 1.2+)	High
NFR-011	Passwords must be hashed using bcrypt	High
NFR-012	API endpoints must implement rate limiting	Medium
NFR-013	Business data must be accessible only to authorized users	High
NFR-014	GDPR compliance for EU users	High

3.2.4 Usability Requirements

ID	Requirement	Priority
NFR-015	Mobile interface must follow platform design guidelines	High
NFR-016	App must be navigable with one hand on mobile	Medium
NFR-017	Color coding must be accessible (WCAG 2.1 compliant)	Medium
NFR-018	Text must be readable without zooming	High

3.2.5 Compatibility Requirements

ID	Requirement	Priority
NFR-019	iOS compatibility: iOS 14.0+	High
NFR-020	Android compatibility: Android 10.0+	High
NFR-021	Screen size support: 4.7" to 6.9"	High
NFR-022	Orientation support: Portrait and Landscape	Medium

3.3 External Interface Requirements

3.3.1 User Interfaces

Home Screen: Displays personalized venue feed with status indicators

Map Screen: Interactive Google Maps interface with venue markers

Venue Detail Screen: Comprehensive venue information with offers

Business Dashboard: Analytics and management interface

Profile Screen: User preferences and activity history

3.3.2 Hardware Interfaces

GPS receiver for location services

Camera for QR code scanning

Accelerometer for UI interactions

Network interfaces (Wi-Fi, Cellular)

3.3.3 Software Interfaces

text

```
┌─────────────────────────────────────────────────────────────────────
──┐
│              External Dependencies              │
├────────────────┬────────────────┬────────────────┬────────────────
──┤
│ Google Maps │   Apple ID  │ Google OAuth│   Firebase    │
│    API      │             │             │ (Notifications)│
├────────────────┼────────────────┼────────────────┼────────────────
──┤
│  PostgreSQL │    Redis    │  WebSocket │  Payment      │
│  Database   │ (Caching)  │  Server    │ Gateway     │
```

```
└────────────────────┴────────────────────┴────────────────────┴────────────────────
───┘
```

3.3.4 Communication Interfaces

HTTPS/SSL for API communication

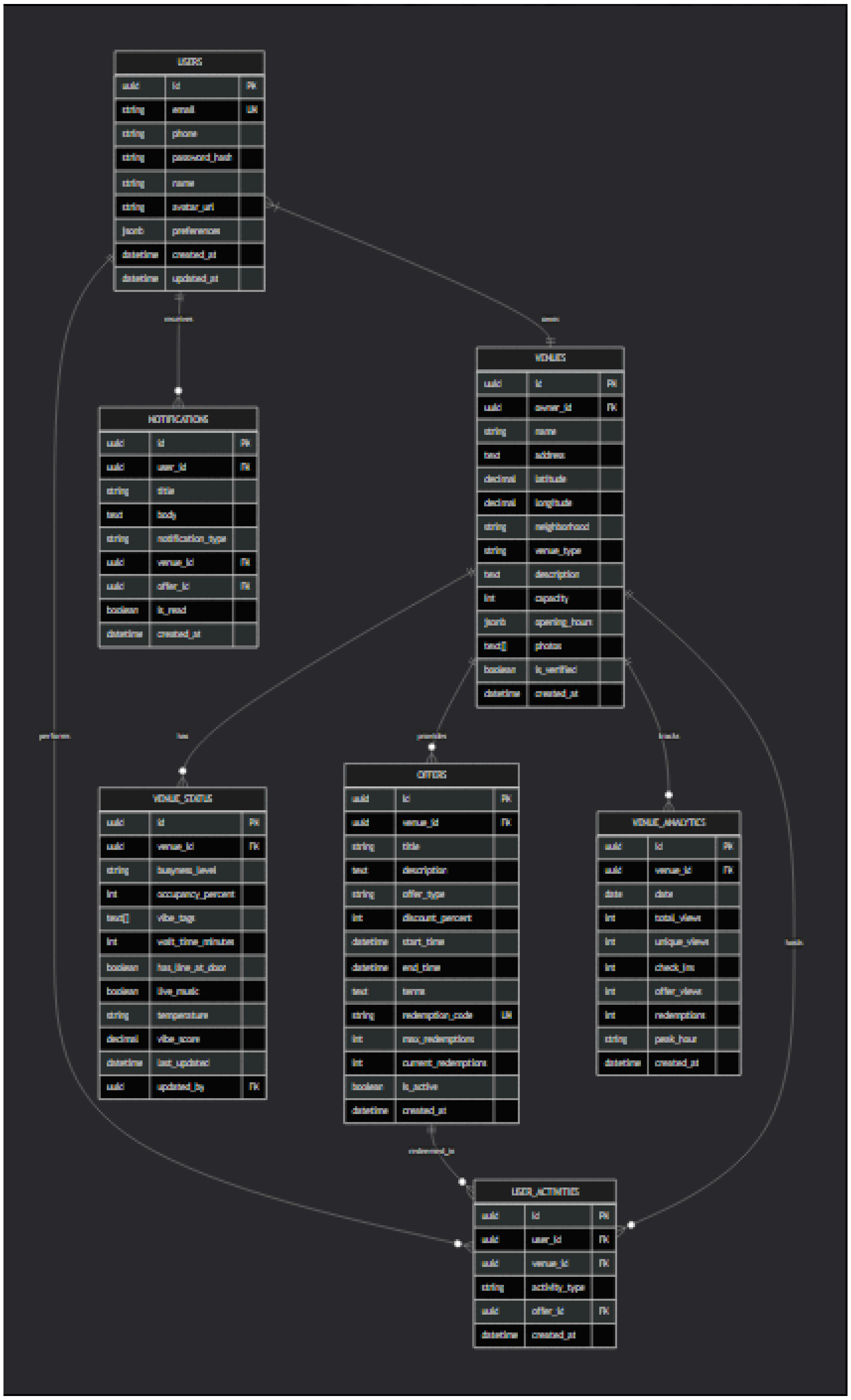
WebSocket for real-time updates

APNs/FCM for push notifications

SMTP for email communications

3.4 Data Requirements

3.4.1 Data Entities and Relationships



3.4.2 Data Dictionary

Entity	Attribute	Type	Description	Constraints
USERS	id	UUID	Unique user identifier	Primary Key
USERS	email	VARCHAR(255)	User email address	Unique, Not Null

USERS	preferences	JSONB	User preferences for discovery	Default: {}
VENUES	neighborhood	VARCHAR(100)	Manchester neighborhood	Northern Quarter, Spinningfields, etc.
VENUES	capacity	INTEGER	Maximum venue capacity	> 0
VENUE_STATUS	busyness_level	VARCHAR(20)	Current crowd level	['quiet','moderate','busy','very_busy']
VENUE_STATUS	occupancy_percent	INTEGER	Percentage of capacity filled	0-100
VENUE_STATUS	vibe_tags	TEXT[]	Atmosphere descriptors	Array of predefined tags
OFFERS	offer_type	VARCHAR(50)	Type of promotion	['2-for-1','discount','free_entry','flash_sale']
OFFERS	redemption_code	VARCHAR(50)	Unique code for redemption	Unique, Not Null

3.5 System Features

3.5.1 Feature 1: Real-time Venue Status Updates

Description:

The system shall provide real-time updates of venue status including busyness levels, atmosphere, and special conditions.

Stimulus/Response Sequences:

Stimulus: Venue staff updates status via Business Dashboard

Response: System broadcasts update to all subscribed users

Stimulus: User opens venue detail screen

Response: System displays current status with timestamp

Stimulus: Time-based automatic status simulation triggers

Response: System updates status based on historical patterns

Functional Requirements:

FR-016, FR-017, FR-018, FR-019, FR-020

FR-026 (Business-side updates)

3.5.2 Feature 2: Personalized Venue Discovery

Description:
The system shall provide personalized venue recommendations based on user preferences, location, and historical behavior.

Stimulus/Response Sequences:

Stimulus: User sets preferences during onboarding

Response: System creates personalized feed

Stimulus: User applies filters (busyness, vibe, offers)

Response: System returns filtered venue list

Stimulus: User interacts with venues (views, saves)

Response: System refines recommendations

Functional Requirements:

FR-006, FR-010, FR-011, FR-012, FR-013, FR-014, FR-015

3.5.3 Feature 3: Offer Management System

Description:
The system shall manage creation, distribution, and redemption of venue offers.

Stimulus/Response Sequences:

Stimulus: Business user creates new offer

Response: System generates unique redemption code

Stimulus: User views and redeems offer

Response: System validates and tracks redemption

Stimulus: Offer expiration time reached

Response: System deactivates offer automatically

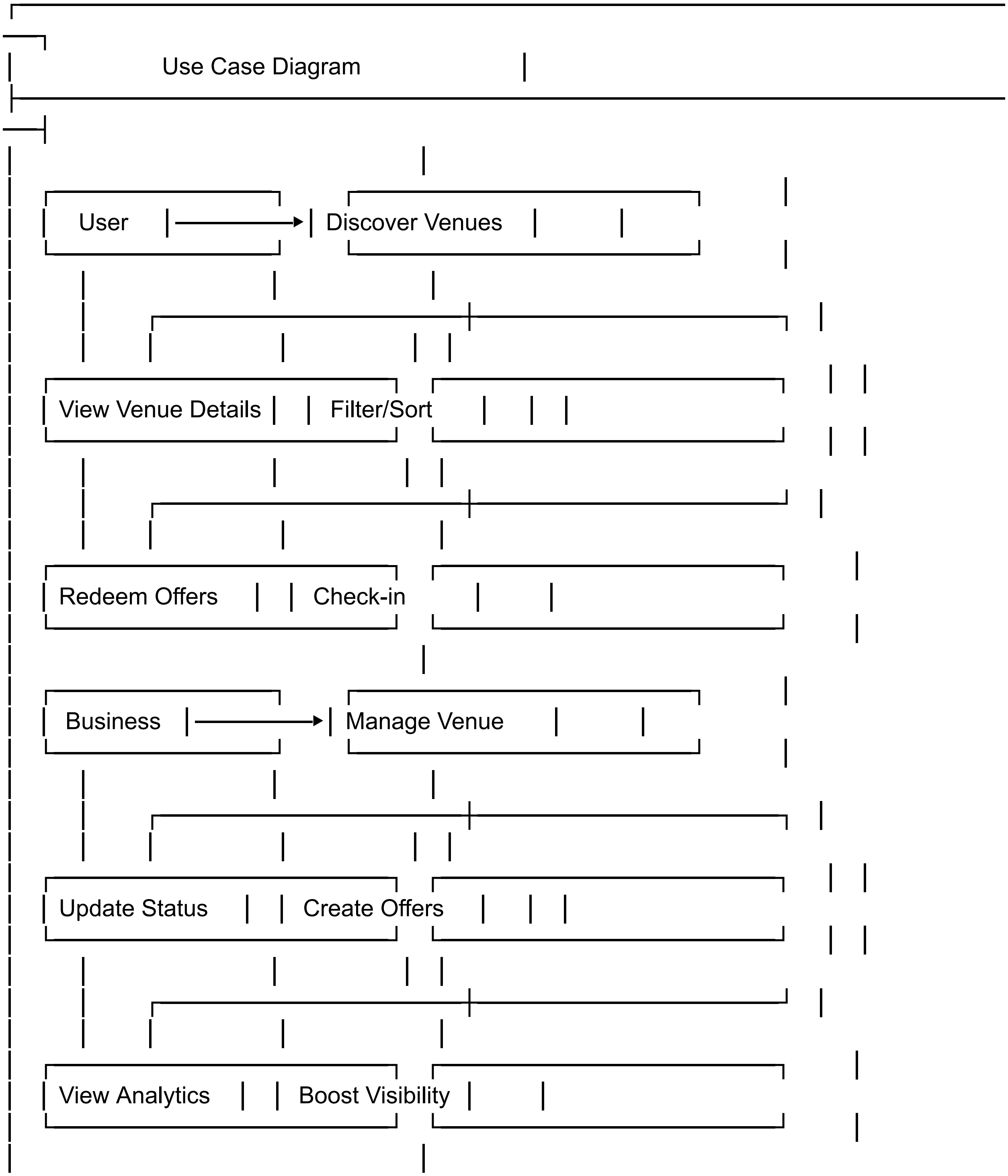
Functional Requirements:

FR-021, FR-022, FR-023, FR-024, FR-025

4. Appendices

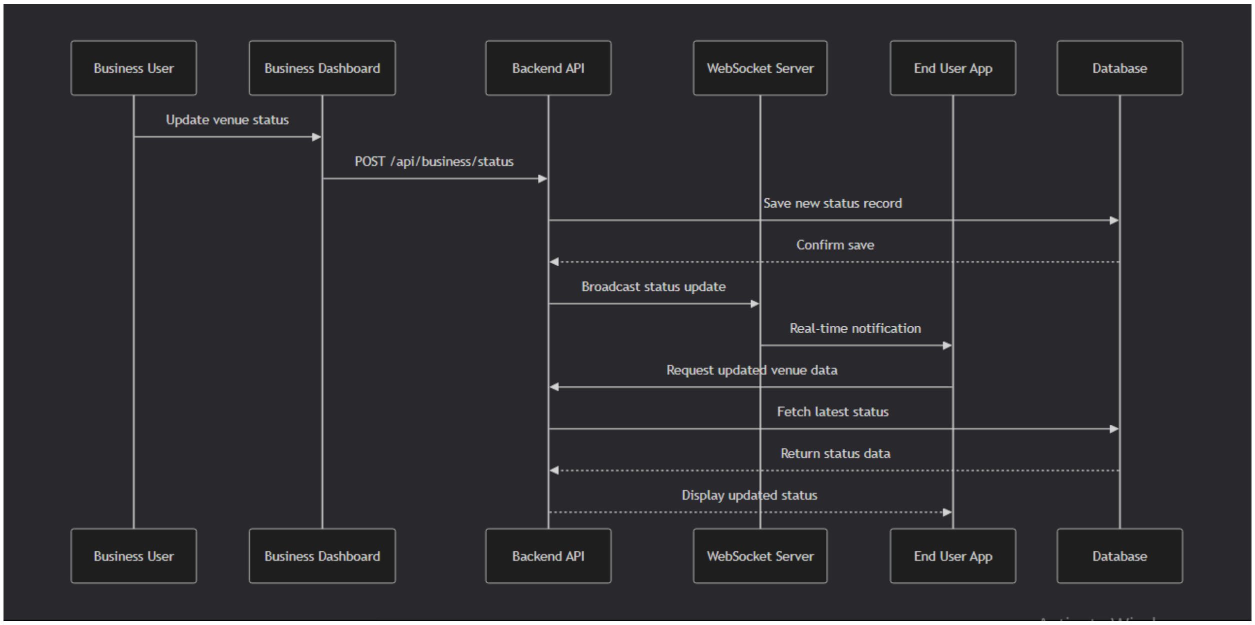
Appendix A: Use Case Diagrams

text

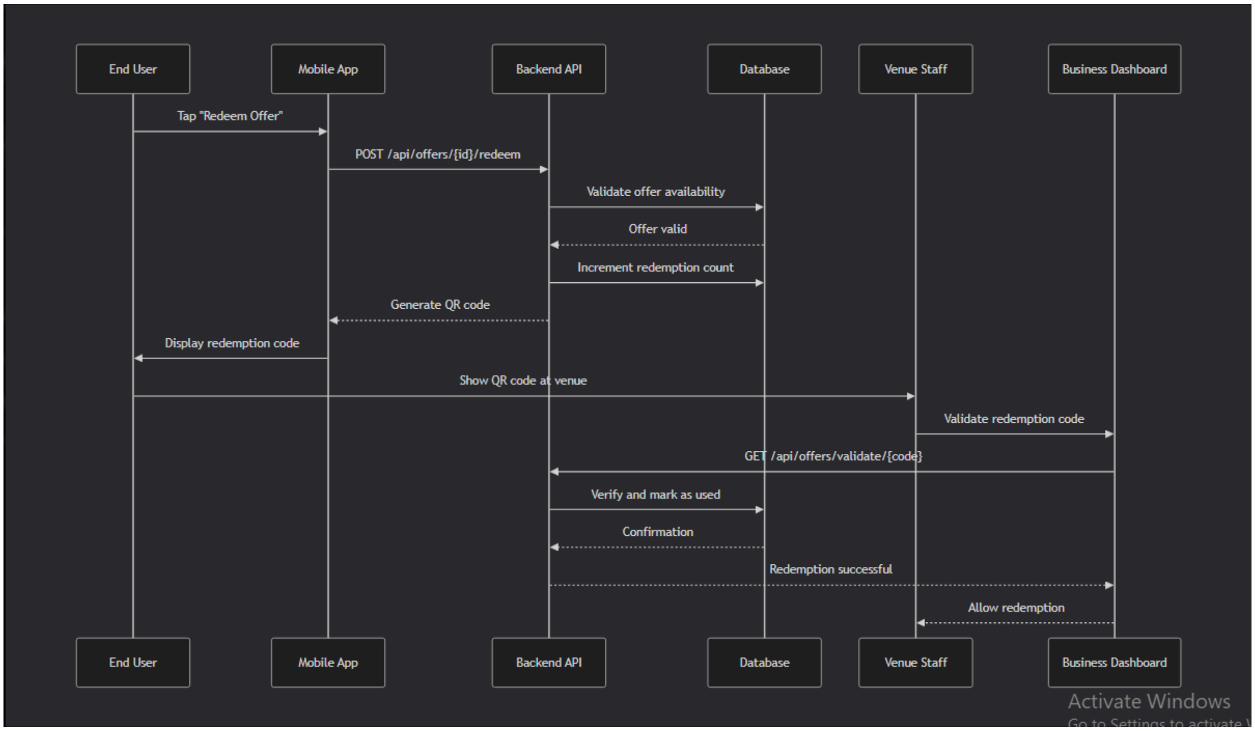


Appendix B: Sequence Diagrams

B.1 Venue Status Update Sequence



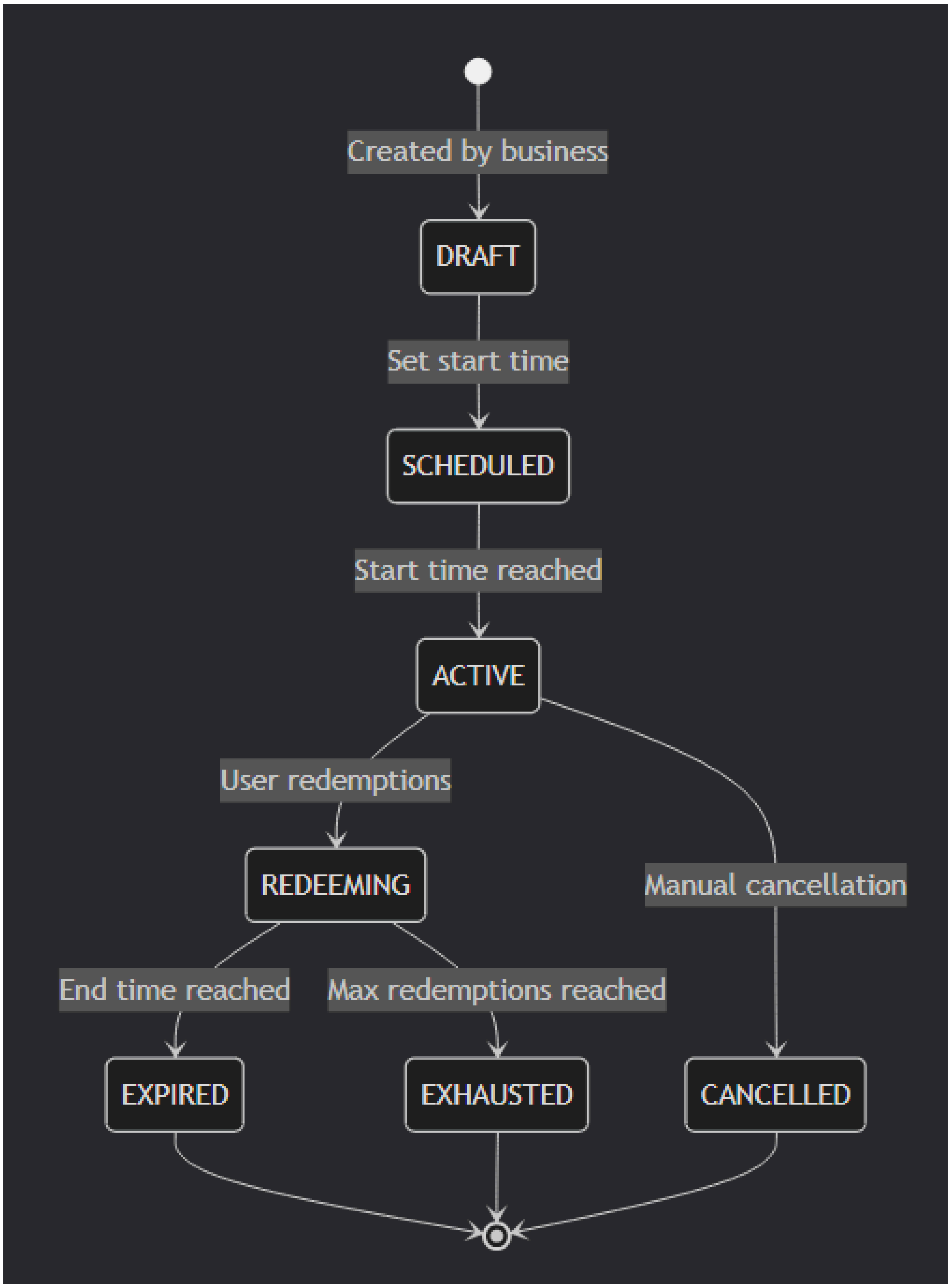
B.2 Offer Redemption Sequence



Appendix C: State Diagrams

C.1 Venue Status State Machine

C.2 Offer Lifecycle State Machine



Appendix D: Data Flow Diagrams

D.1 Level 0: Context Diagram

text

```
┌─────────────────┐     ┌──────────────────────────────────────┐
│  End User       │────▶│                    │
│  (Mobile App)   │◀────│    REKI SYSTEM              │
└─────────────────┘     │                         │
         │                  │
┌─────────────────┐     │  ┌──────────────────────────┐    │
│  Business       │────▶│  │ Database (PostgreSQL)  │    │
│  (Dashboard)    │◀────│  └──────────────────────────┘    │
└─────────────────┘     │                         │
         │                  │
┌─────────────────┐     │  ┌──────────────────────────┐    │
│  Admin          │────▶│  │ External Services      │    │
│  (System)       │◀────│  │ (Maps, Auth, Notif)    │    │
└─────────────────┘     │  └──────────────────────────┘    │
         └─────────────────────────────────────┘
```

D.2 Level 1: Major Processes

text

```
┌─────────────────────────────────────────────────────────────────┐
│           REKI System - Level 1 DFD               │
├──────────────┬───────────────┬───────────────┬──────────────────┤
│         │         │         │          │
│ User Mgmt │ Venue      │ Offer      │ Analytics     │
│ Process   │ Discovery  │ Management │ Process       │
│  (1.0)    │ Process    │ Process    │  (4.0)        │
│         │  (2.0)    │  (3.0)    │          │
├──────────────┼───────────────┼───────────────┼──────────────────┤
│ User Data │ Venue      │ Offer Data │ Analytics     │
│         │ Data       │         │ Data       │
└──────────────┴───────────────┴───────────────┴──────────────────┘
    │         │         │         │
    ▼         ▼         ▼         ▼
┌─────────────────────────────────────────────────────────────────┐
│              Central Database                   │
└─────────────────────────────────────────────────────────────────┘
```

Appendix E: API Endpoint Specifications

E.1 Authentication Endpoints

yaml

```
/api/auth/login:
  post:
    summary: User login
    requestBody:
      required: true
      content:
        application/json:
          schema:
            type: object
            properties:
              email:
                type: string
                format: email
              password:
                type: string
                minLength: 8
    responses:
      200:
        description: Login successful
        content:
          application/json:
            schema:
              type: object
              properties:
                access_token:
                  type: string
                refresh_token:
                  type: string
                user:
                  $ref: '#/components/schemas/User'
      401:
        description: Invalid credentials
```

E.2 Venue Discovery Endpoints

yaml

```
/api/venues:
  get:
    summary: Get venues with filters
    parameters:
      - in: query
        name: busyness
        schema:
          type: string
          enum: [quiet, moderate, busy, very_busy]
      - in: query
        name: vibeTags
        schema:
          type: array
          items:
            type: string
      - in: query
        name: hasOffers
        schema:
          type: boolean
      - in: query
        name: sortBy
        schema:
          type: string
          enum: [distance, trending, top_rated]
    responses:
      200:
        description: List of venues
        content:
          application/json:
            schema:
              type: object
              properties:
                venues:
                  type: array
                  items:
                    $ref: '#/components/schemas/Venue'
                total:
                  type: integer
                filters:
                  $ref: '#/components/schemas/Filters'
```

Appendix F: Error Handling Specifications

F.1 Error Codes and Messages

HTTP Code	Error Code	Message	Resolution
400	VALIDATION_ERROR	Invalid input parameters	Check request body format
401	UNAUTHORIZED	Authentication required	Login or provide valid token
403	FORBIDDEN	Insufficient permissions	Contact administrator
404	VENUE_NOT_FOUND	Venue does not exist	Check venue ID
409	OFFER_EXHAUSTED	Maximum redemptions reached	Offer no longer available
429	RATE_LIMITED	Too many requests	Wait and retry
500	SERVER_ERROR	Internal server error	Contact support

F.2 Error Response Format

json

```
{
  "error": {
    "code": "VALIDATION_ERROR",
    "message": "Invalid email format",
    "details": {
      "field": "email",
      "constraint": "Must be valid email address"
    },
    "timestamp": "2026-01-13T20:47:52Z",
    "request_id": "req_123456789"
  }
}
```

Appendix G: Deployment Specifications

G.1 Infrastructure Requirements

Component	Specification	Quantity	Notes
Application Server	4 vCPU, 8GB RAM	2	Load balanced
Database Server	8 vCPU, 16GB RAM, 100GB SSD	1	PostgreSQL 14+
Cache Server	2 vCPU, 4GB RAM	1	Redis 6+
File Storage	50GB	1	S3-compatible
CDN	Global	1	For static assets

G.2 Environment Variables

env

```
# Database
DB_HOST=localhost
DB_PORT=5432
```

```
DB_NAME=reki_db
DB_USER=reki_user
DB_PASSWORD=secure_password

# JWT
JWT_SECRET=your_jwt_secret_key
JWT_EXPIRATION=24h

# Google Maps
GOOGLE_MAPS_API_KEY=your_google_maps_key

# Redis
REDIS_HOST=localhost
REDIS_PORT=6379

# Email
SMTP_HOST=smtp.gmail.com
SMTP_PORT=587
SMTP_USER=your_email@gmail.com
SMTP_PASSWORD=your_app_password
```

5. Glossary

Term	Definition
Vibe	The atmosphere or mood of a venue at a given time
Busyness Level	Categorical representation of venue occupancy
Vibe Tags	Descriptive labels for venue atmosphere
Redemption	Process of claiming a venue offer
Footfall	Number of people entering a venue
Boost	Paid feature to increase venue visibility
Check-in	User action indicating presence at a venue
Occupancy Percent	Percentage of maximum capacity currently filled

6. Index

API Endpoints: 3.3.4, Appendix E

Data Models: 3.4.1, Appendix Diagrams

Error Handling: Appendix F

Functional Requirements: 3.1

Non-Functional Requirements: 3.2

Security Requirements: 3.2.3

Use Cases: Appendix A

User Interfaces: 3.3.1

Document Approval

Role	Name	Signature	Date
Project Manager			
Lead Developer			
Quality Assurance			
Client Representative			

Document Status: Approved

Next Review Date: 13 February 2026

Change History: Version 1.0 - Initial Release