Software Requirements Specification (SRS)

Warzone Ticketing System (WTS) for 7,000-Seat Arena

| **Version:** | **1.7 (Tech Stack & Detail Finalized)** |
| --- | --- |
| **Purpose:** | To define the complete functional and non-functional requirements for the 7,000-seat arena **Warzone Ticketing System (WTS)**. |
| **Scope:** | Customer Web Portal, Staff Mobile App (Gate/Counter), Administration Portal, and Support Center Interface. |

1. Introduction

The **Warzone Ticketing System (WTS)** is a mission-critical platform designed to handle high-volume ticket sales and rapid, secure access control for a 7,000-seat reserved seating arena. The primary goal is stability and fraud prevention during peak demand. This document serves as the final blueprint for development.

2. General Requirements

| **Requirement** | **Description** | **Rationale** |
| --- | --- | --- |
| **System Name** | The official system name is the **Warzone Ticketing System (WTS)**. | Standardizes documentation and communication. |
| **Users** | Customer (Public), Gate Staff (Scanner), Counter Staff (Box Office), Support Staff (Customer Service), and Administrator. | Defines all necessary system interfaces and Role-Based Access Control (RBAC). |

3. Functional Requirements (FR)

3.1 Inventory and Seat Assignment

| **ID** | **Requirement** | **Detail** |
| --- | --- | --- |
| **FR 3.1.1** | Inventory Management | The system must manage a 7,000-seat, reserved inventory, allowing administrators to define sections, rows, and specific price points. The inventory status (Available, Held, Sold) must be updated in real-time. |
| **FR 3.1.2** | Administrator Configuration | Administrators must be able to define price zones, dynamic pricing rules, and event-specific holds/complimentary ticket allocations via the Admin Portal. |
| **FR 3.1.3** | **Best Available Assignment** | The system **must** employ a "Best Available" algorithm (prioritizing contiguous blocks, proximity to the stage/center) to automatically assign seats **immediately upon payment confirmation**, based only on the customer's selected Zone and Quantity. |
| **FR 3.1.4** | Time-limited Seat Hold | Upon assignment, seats must be placed on an irreversible **8-10 minute time-limited hold**. If the transaction fails or times out, the seats are immediately released back to the general inventory to maximize sales opportunities. |

3.2 Ticket Generation and Access Control

| **ID** | **Requirement** | **Detail** |
| --- | --- | --- |
| **FR 3.2.1** | Unique Barcode Generation | The system **must** generate a **unique, non-reusable QR code or barcode** for **every individual seat** sold. This code serves as the cryptographic link to the specific seat coordinates and ticket status in the database. |
| **FR 3.2.2** | Real-time Validation | The dedicated mobile scanning app must validate tickets in real-time (NFR 4.1.1) against the database, checking for validity, duplication (fraud), and correct event/date/time. |
| **FR 3.2.3** | Fraud Feedback | The scanner app must provide immediate, full-screen, high-contrast feedback: **GREEN** (Access Granted), **RED** (Already Scanned/Invalid), and **YELLOW** (Wrong Gate/Zone) to ensure quick decision-making by Gate Staff. |

3.3 Financial Transaction and Sales

| **ID** | **Requirement** | **Detail** |
| --- | --- | --- |
| **FR 3.3.1** | Pricing Rules | The system must support dynamic pricing (time-based or quantity-based), tiered fees (service fees, taxes), and promotional discount codes. |
| **FR 3.3.2** | Payment Gateways | The system **must integrate with Stripe** for all secure online credit card processing, leveraging their tokenization features. |
| **FR 3.3.3** | Box Office Sales | The Counter Staff interface must allow for quick sales, including the ability to process payment via Cash, Card (via Stripe terminal integration), and manual Complimentary (Comp) overrides. |

3.4 Scalability and Virtual Queue (CRITICAL)

| **ID** | **Requirement** | **Detail** |
| --- | --- | --- |
| **FR 3.4.1** | Scalable Architecture | The WTS must be built on an architecture designed for horizontal scaling, specifically utilizing **Laravel's Queue system** for background tasks (e.g., emailing, report generation). |
| **FR 3.4.2** | Database Integrity | Database transactions involving seat assignment must utilize ACID principles and appropriate locking mechanisms to prevent data race conditions and the double-selling of seats. |
| **FR 3.4.3** | **Virtual Queue System** | A mandatory virtual queuing system must be implemented before the ticket sales page. This system will dynamically throttle traffic, holding users and releasing them in controlled batches to protect the core inventory and database during peak load (NFR 4.1.2). |

3.5 Administration and Reporting

| **ID** | **Requirement** | **Detail** |
| --- | --- | --- |
| **FR 3.5.1** | RBAC | Role-Based Access Control must be enforced for all system interfaces, restricting access based on staff role: Gate Staff (Scanner only), Counter Staff (Sales/Search), Support Staff (Order Lookup/Re-issue), and Administrator (Full reporting/Configuration). |
| **FR 3.5.2** | Admittance Report | The system must generate a real-time "Admittance Report" detailing every successful scan, including Ticket ID, Scan Time, Gate location, and the specific Staff User ID who performed the scan. |
| **FR 3.5.3** | Financial Reporting | The Administrator must be able to generate financial reports filtered by event, date range, payment type (Cash/Stripe), and revenue by Zone/Ticket Type. |

3.6 Email and Notification System

| **ID** | **Requirement** | **Detail** |
| --- | --- | --- |
| **FR 3.6.1** | **Customer Email:** Automatic email must be sent immediately upon successful purchase, containing a detailed confirmation and the attached unique QR code ticket(s). |  |
| **FR 3.6.2** | **Staff Email:** Staff must receive automated notifications regarding operational issues, gate assignment changes, or system maintenance schedules. |  |
| **FR 3.6.3** | **Administrator Email:** System must send real-time email alerts to technical administrators upon critical system failures (e.g., database connection issues, Stripe API failures, Virtual Queue failures). |  |

3.7 Customer Purchase Workflow (WTS)

1. **Enter Virtual Queue:** Customer accesses the event sale page and is placed in the queue (FR 3.4.3).
2. **Access Sales Page:** Customer exits the queue and is granted access to the sales interface.
3. **Select Zone & Quantity:** Customer selects the desired Price Zone (e.g., Lower Bowl) and the number of seats (e.g., 4).
4. **Time Hold & Review:** The system calculates the total price.
5. **Payment Processing:** Customer enters information and proceeds via the **Stripe gateway** (FR 3.4.1).
6. **Automatic Assignment:** Successful payment triggers the **Best Available Assignment (FR 3.1.3)** job in the background (using the Laravel Queue system).
7. **Confirmation & Delivery:** The sale finalizes, and tickets with unique QR codes are delivered via email (FR 3.2.1, FR 3.6.1).

4. Non-Functional Requirements (NFR)

NFR 4.1 Performance and Scalability

| **ID** | **Requirement** | **Metric/Details** |
| --- | --- | --- |
| **NFR 4.1.1 (Scan Time)** | Mobile scanning app must validate and process a ticket in under **1.0 second**. | Sub-1.0s validation time is critical for smooth entry. |
| **NFR 4.1.2 (Peak Load)** | System must be stable and maintain inventory integrity during a sudden surge of **2,500 concurrent users** entering the Virtual Queue/Sale process. | inventory accuracy at 2,500 users. |
| **NFR 4.1.3 (Database)** | Database queries for inventory and seat assignment must execute in less than **200 milliseconds** under standard load. | query time. |

NFR 4.2 Security

| **ID** | **Requirement** | **Metric/Details** |
| --- | --- | --- |
| **NFR 4.2.1 (Compliance)** | System must be **PCI DSS Compliant** for payment handling via Stripe integration. | Annual PCI compliance certification. |
| **NFR 4.2.2 (Encryption)** | All data communication (client to server, scanner app to server) must use HTTPS/TLS 1.2 or higher. All sensitive data must be encrypted at rest. | All traffic encrypted. |

NFR 4.3 Reliability

| **ID** | **Requirement** | **Metric/Details** |
| --- | --- | --- |
| **NFR 4.3.1 (Uptime)** | Minimum guaranteed application uptime. | uptime. |

NFR 4.4 Technology Stack

| **ID** | **Requirement** | **Metric/Details** |
| --- | --- | --- |
| **NFR 4.4.1 (Framework)** | Server-side development must use the **Laravel Framework** (latest LTS), leveraging its Queue and Eloquent ORM features. | Mandated development framework. |
| **NFR 4.4.2 (Database)** | Database must use **MySQL** (configured for high concurrency/replication) for inventory and transactional data. | Mandated database system. |