

# Movie Booking Management System - Technical Report

## Table of Contents

1. [Introduction](#)
2. [System Architecture Overview](#)
3. [Design Decisions and Rationale](#)
4. [User Flows](#)
5. [Administrative Operations](#)
6. [Database Architecture and Design](#)
7. [API Design and Implementation](#)
8. [Appendices](#)

## 1. Introduction

### 1.1 Project Overview

The Movie Booking Management System is a full-stack web application designed to streamline cinema operations and enhance customer experience. Built with modern technologies including Next.js frontend, Flask backend, and MySQL database, the system provides comprehensive functionality for both customers and administrators.

### 1.2 Key Features

#### Customer Features:

- Intuitive movie browsing and showtime selection
- Real-time seat availability with visual seat map
- Secure booking with temporary seat locking
- Personal booking history and ticket management

#### Administrative Features:

- Comprehensive dashboard with revenue analytics

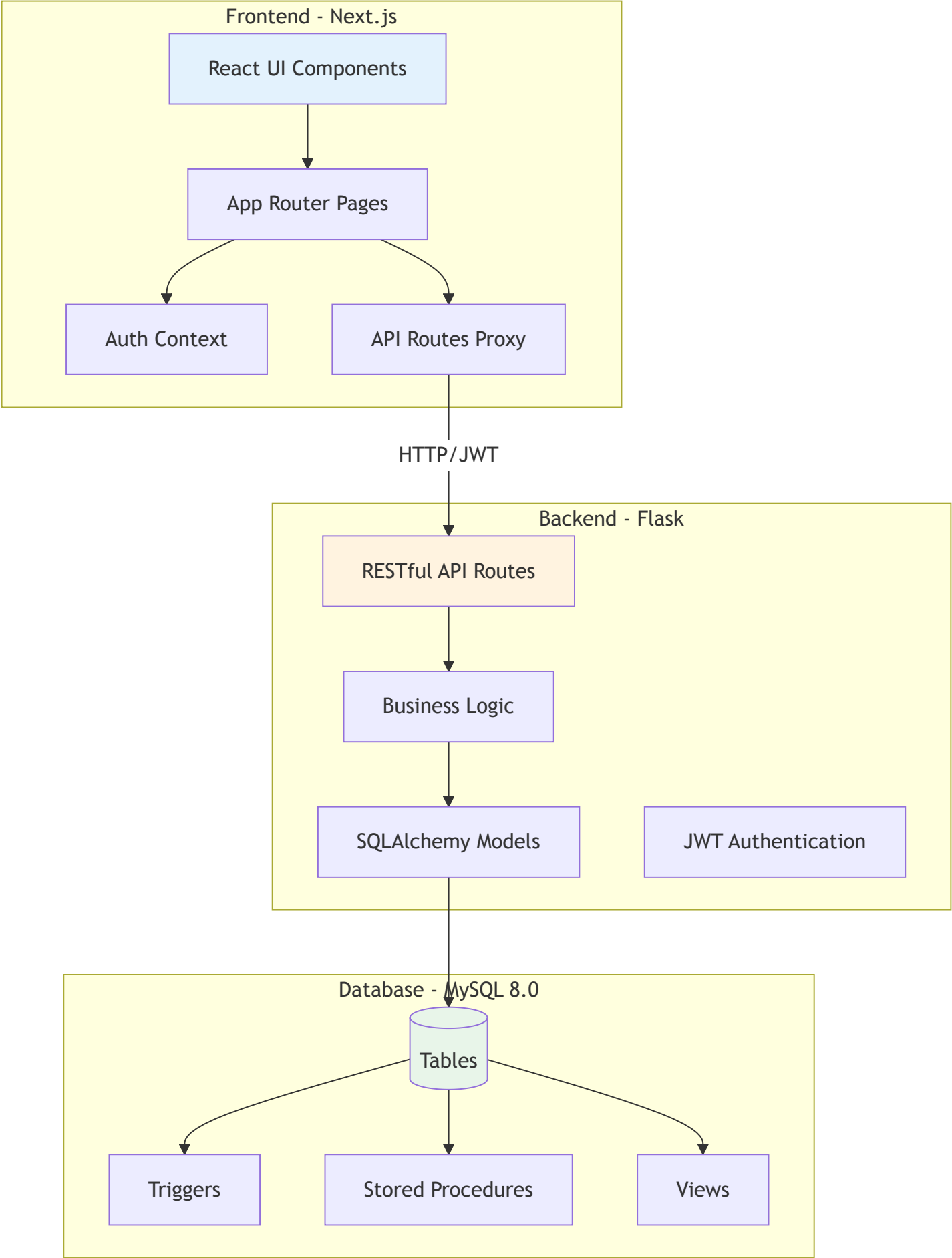
- Movie and showtime management with conflict detection
- Performance metrics and reporting
- Timeslot management across multiple screens

## 1.3 Technical Highlights

- **Database-First Architecture:** Schema-driven development with auto-generated models
- **Real-Time Seat Management:** Temporary locking mechanism preventing double-booking
- **Atomic Operations:** Stored procedures ensuring data consistency
- **Performance Optimization:** Database views for complex queries
- **Scalable Design:** Modular architecture supporting future expansion

# 2. System Architecture Overview

## 2.1 High-Level Architecture



## 2.2 Technology Stack

### Frontend (Next.js 14)

- React 18 with Server Components
- App Router for navigation
- Tailwind CSS for styling
- Radix UI for accessible components
- Context API for state management

### Backend (Flask)

- RESTful API architecture
- SQLAlchemy ORM with auto-generated models
- Structured error handling

### Database (MySQL 8.0)

- Relational schema with referential integrity and indexes
- Stored procedures for complex operations
- Triggers for business rule enforcement
- Views

## 2.3 Communication Flow

The system implements a clear request-response pattern:

1. Frontend components make requests through Next.js API routes
2. API routes proxy requests to Flask backend with authentication
3. Flask processes requests using business logic in serializers
4. Database operations executed through ORM with stored procedures for complex transactions

## 3. Design Decisions and Rationale

### 3.1 Database-First Approach (Instead of Code-first)

**Decision:** Implement schema-first development using `sqlacodegen` for model generation.

**Rationale:**

- **Data Integrity:** Database constraints ensure data consistency at the source

- **Performance:** Native SQL optimizations and indexing strategies
- **Maintainability:** Single source of truth for schema changes

## 3.2 Unified Seat Layout Design

**Decision:** Standardize all cinema screens to 8×12 seat grid (96 seats total).

**Rationale:**

- **Consistency:** Uniform user experience across all screens
- **Simplicity:** Single codebase for seat rendering
- **Scalability:** Easy to add new screens without UI changes
- **Pricing Logic:** Clear tier separation (Standard: A-E, Premium: F-H)

**Seat Configuration:**

Rows A-E: Standard class (60 seats) – \$8 base price

Rows F-H: Premium class (36 seats) – \$16 base price

Format multipliers: 2D (1.0x), 3D (1.25x), IMAX (1.5x)

## 3.3 Temporary Seat Locking Strategy

**Decision:** Implement 15-minute seat locks during selection process.

**Rationale:**

- **Prevents Double-Booking:** Ensures seats remain available during purchase
- **User Experience:** Provides reasonable time for decision-making
- **Resource Management:** Automatic cleanup prevents permanent locks
- **Scalability:** Database-managed expiration reduces server overhead

## 3.4 Atomic Transaction Design

**Decision:** Use stored procedures for complex multi-table operations.

**Rationale:**

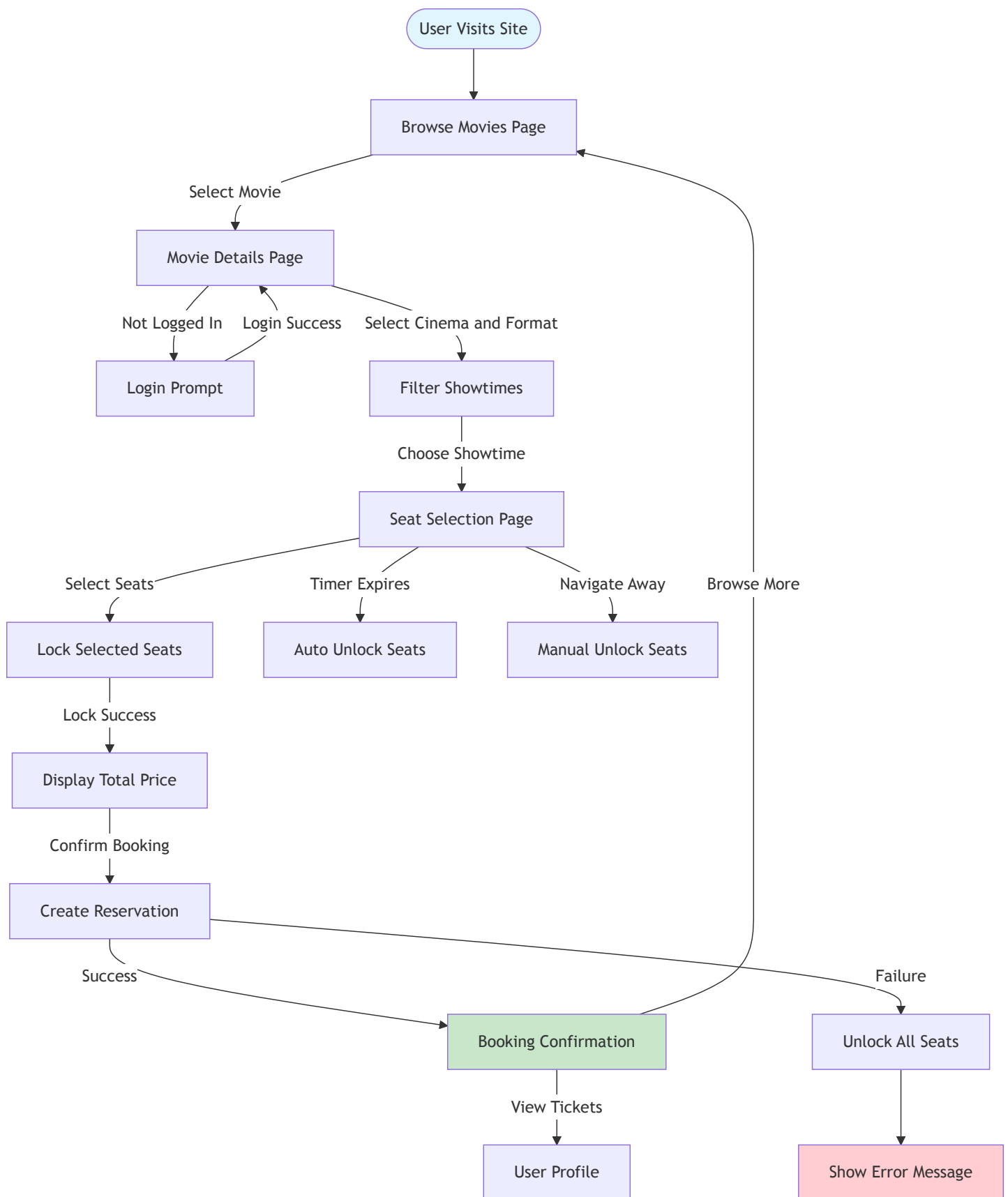
- **Data Consistency:** ACID compliance for critical operations
- **Performance:** Reduced network round-trips
- **Business Logic:** Database-enforced business rules

- **Error Handling:** Atomic rollback on failures

## 4. User Flows

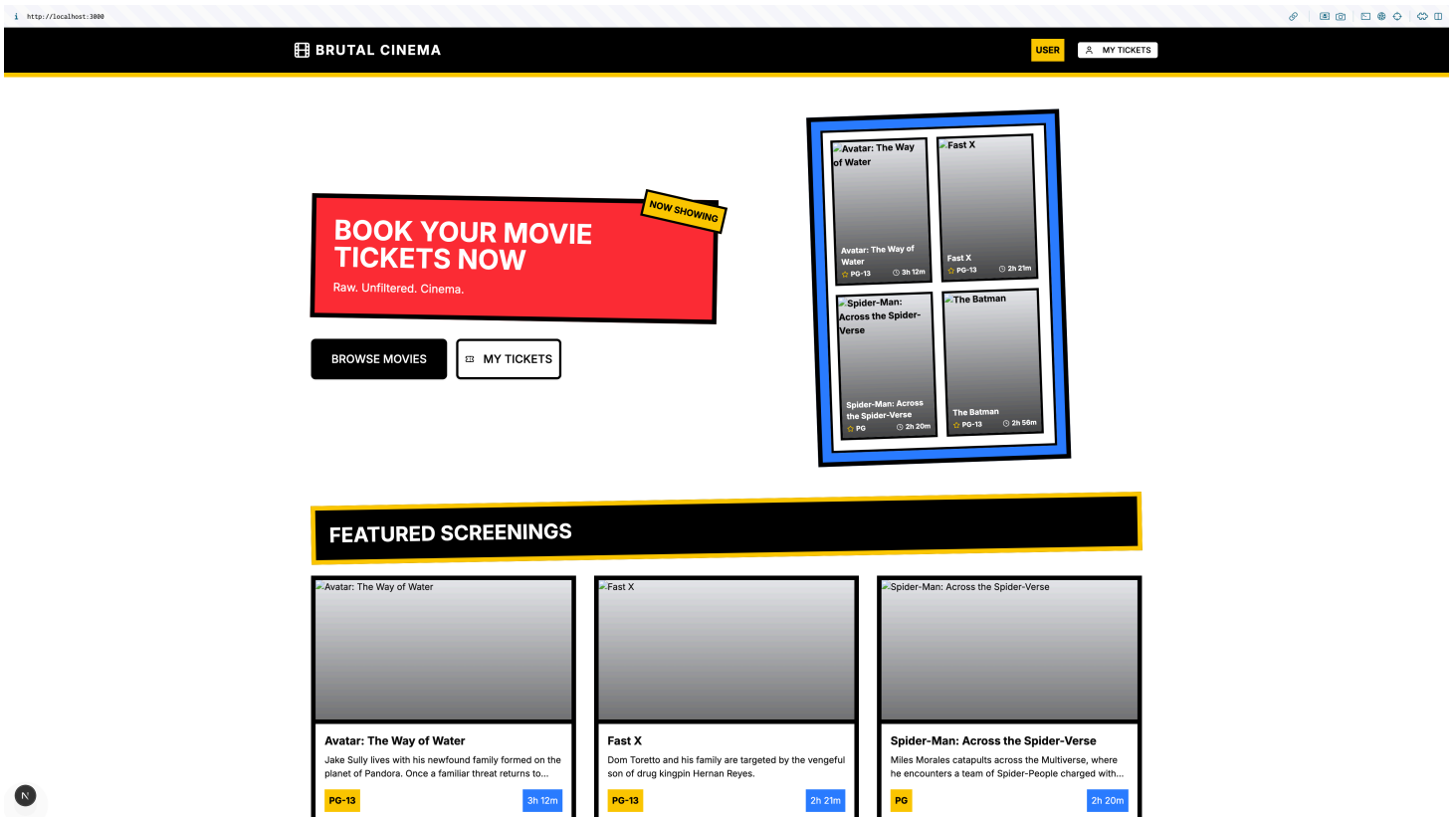
### 4.1 Customer Booking

The customer booking flow prioritizes simplicity while ensuring data integrity and preventing conflicts.

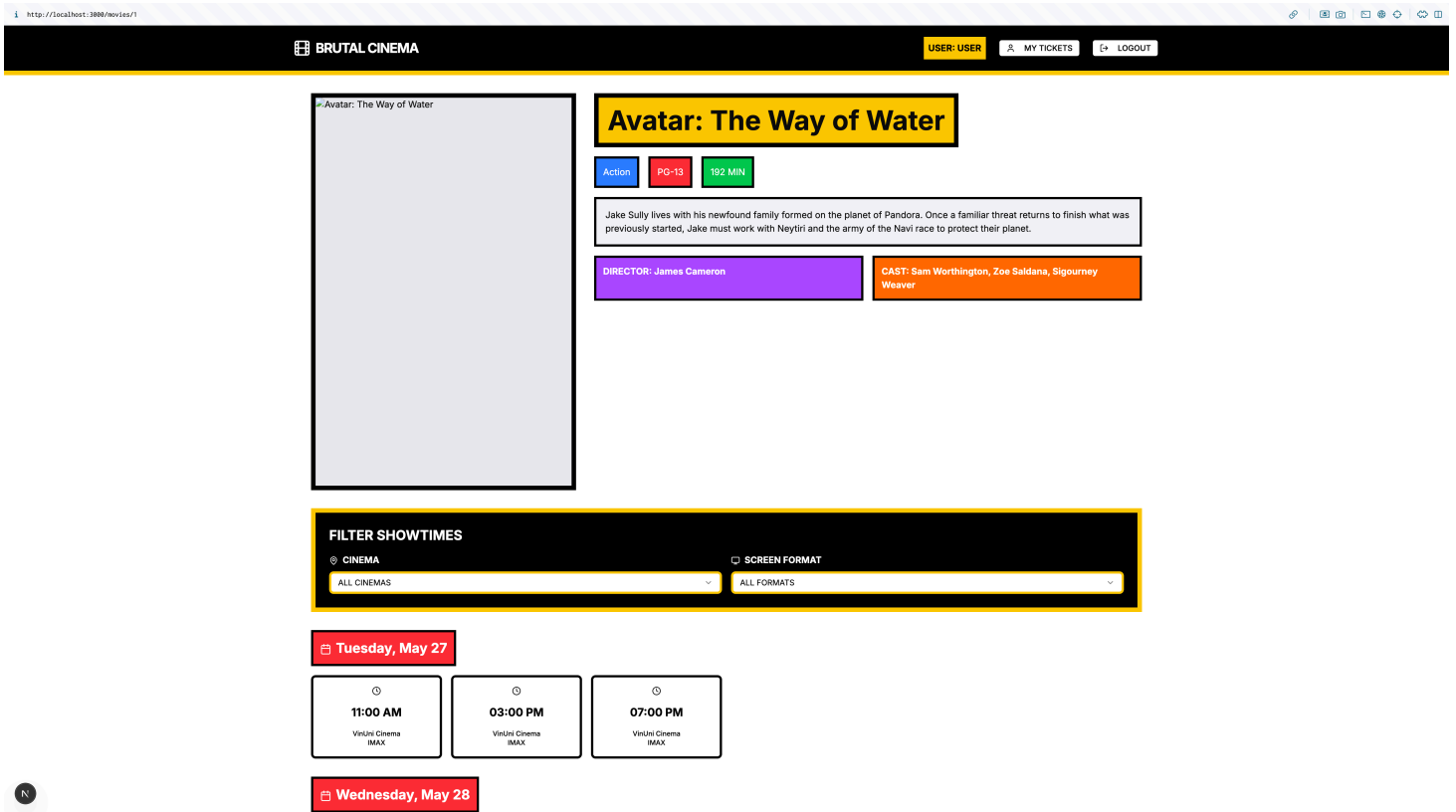


1. Landing page showing featured movies

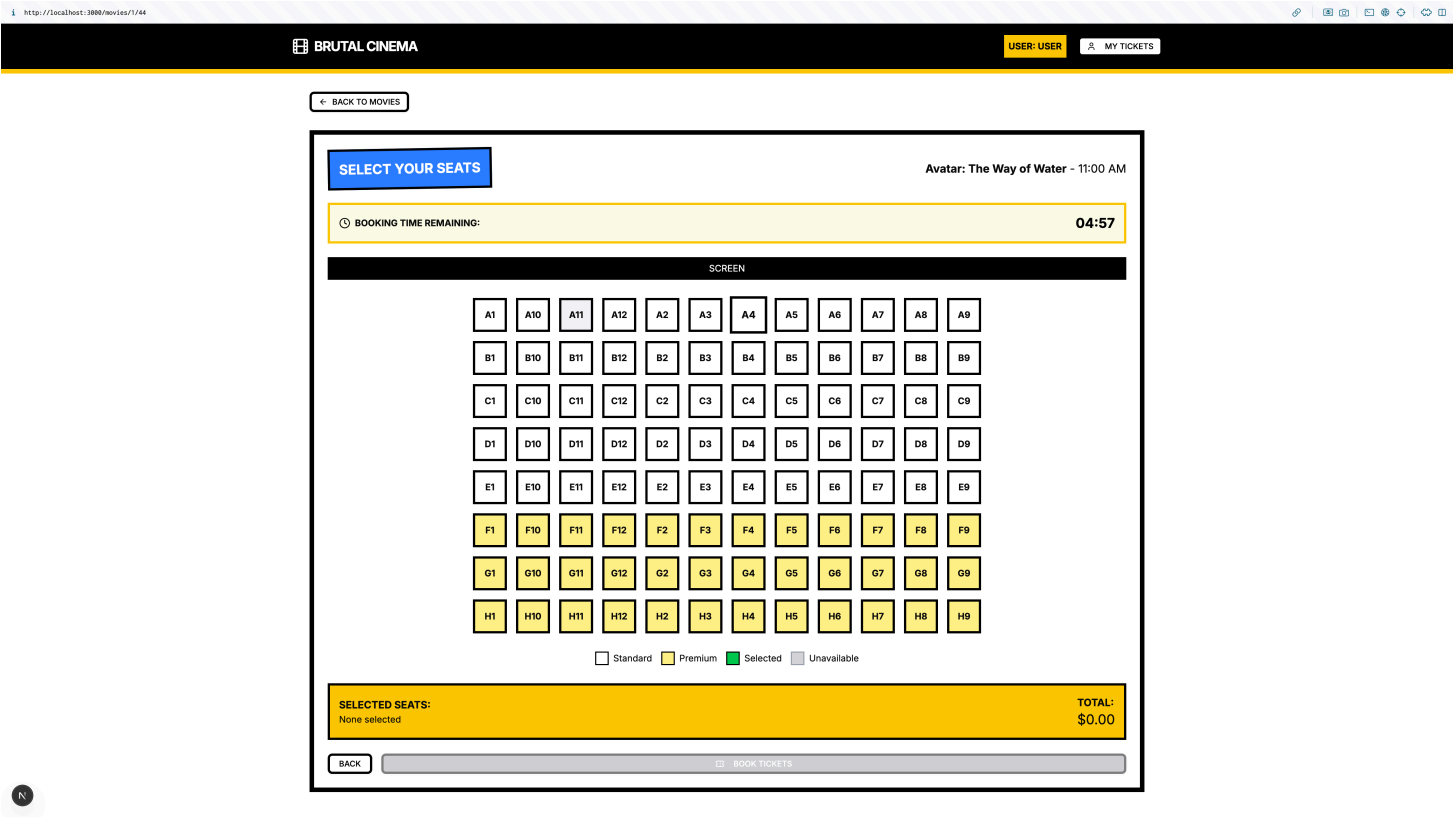




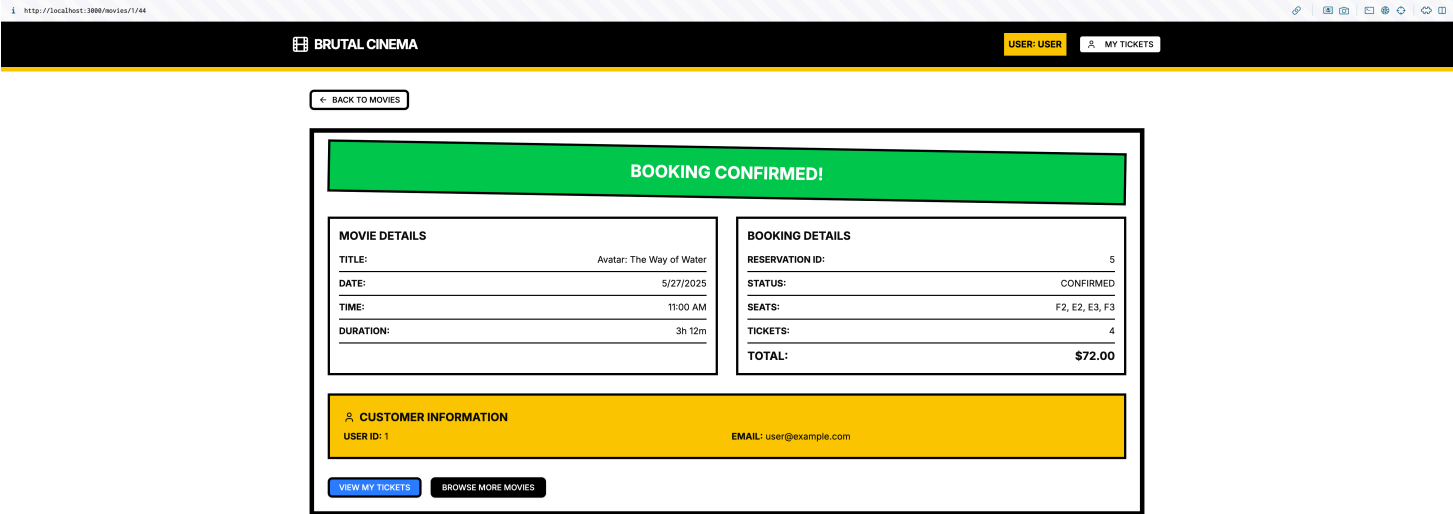
## 2. Movie details page with showtime filtering options



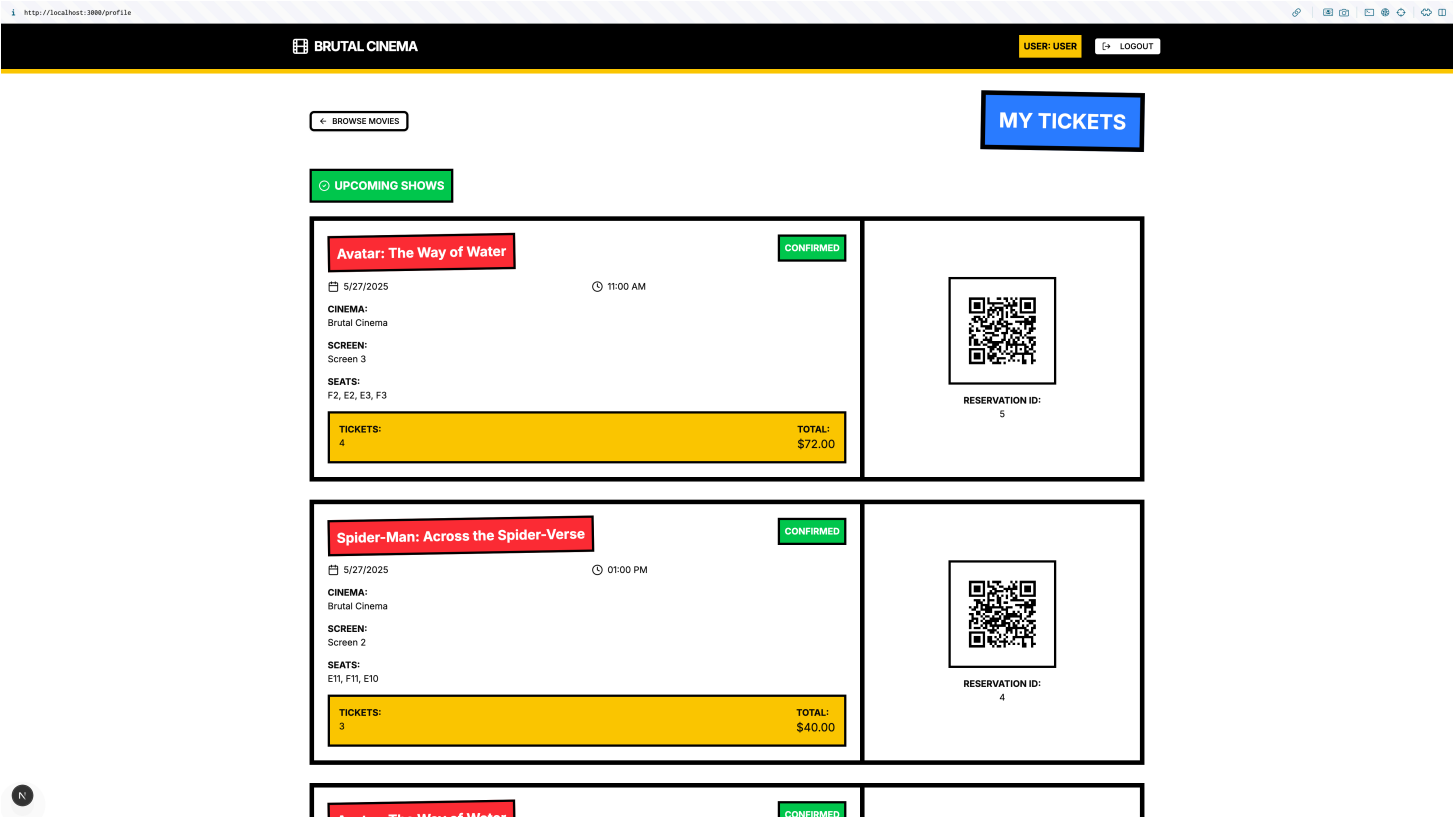
## 3. Seats options



4. Booking confirmation page with reservation details

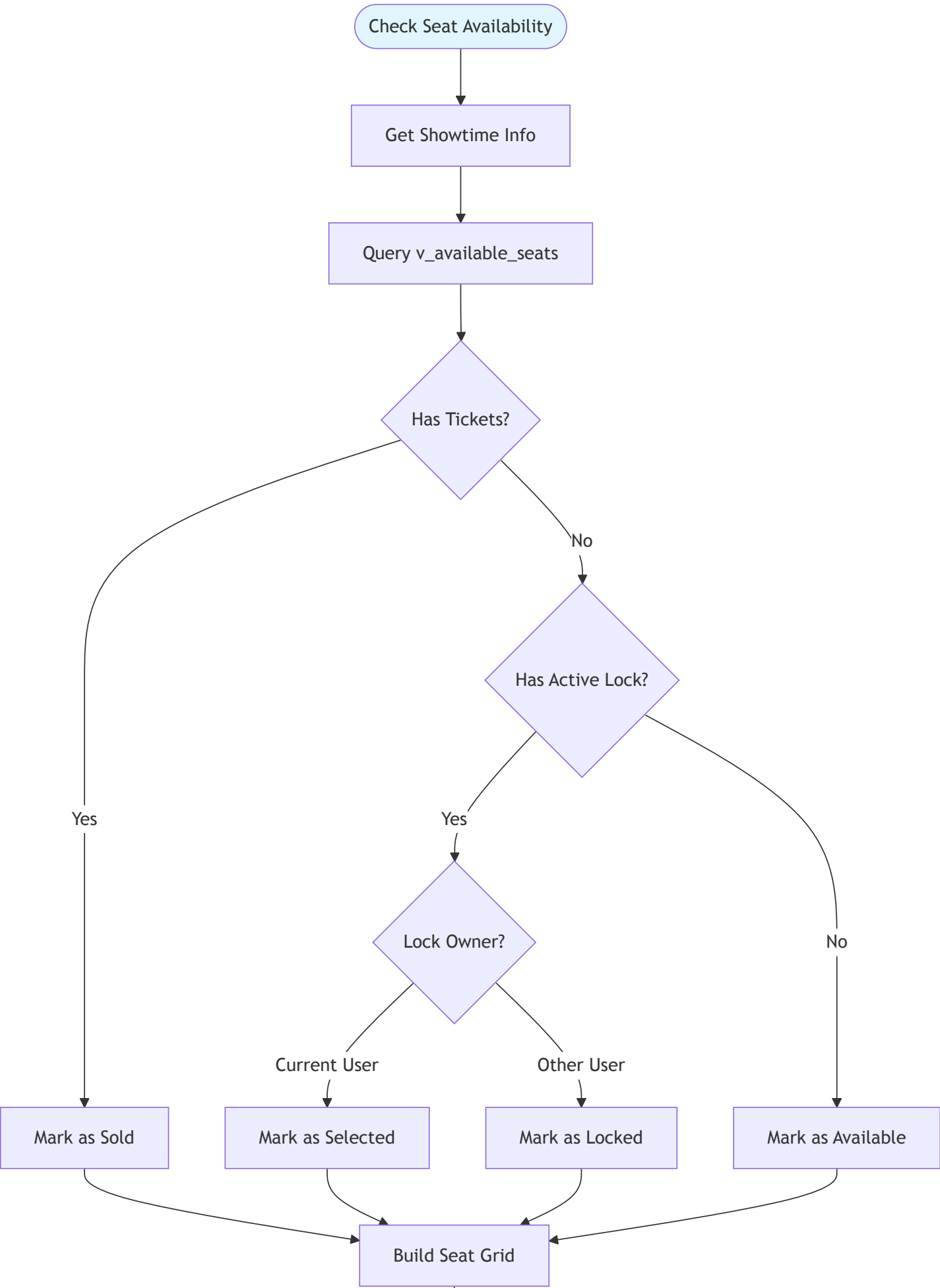


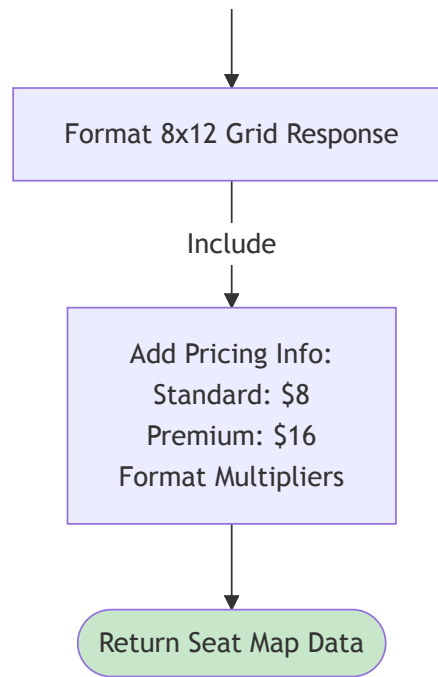
5. User profile showing booking history and tickets



## 4.2 Seat Availability Management

Real-time seat availability ensures accurate booking information and prevents conflicts.



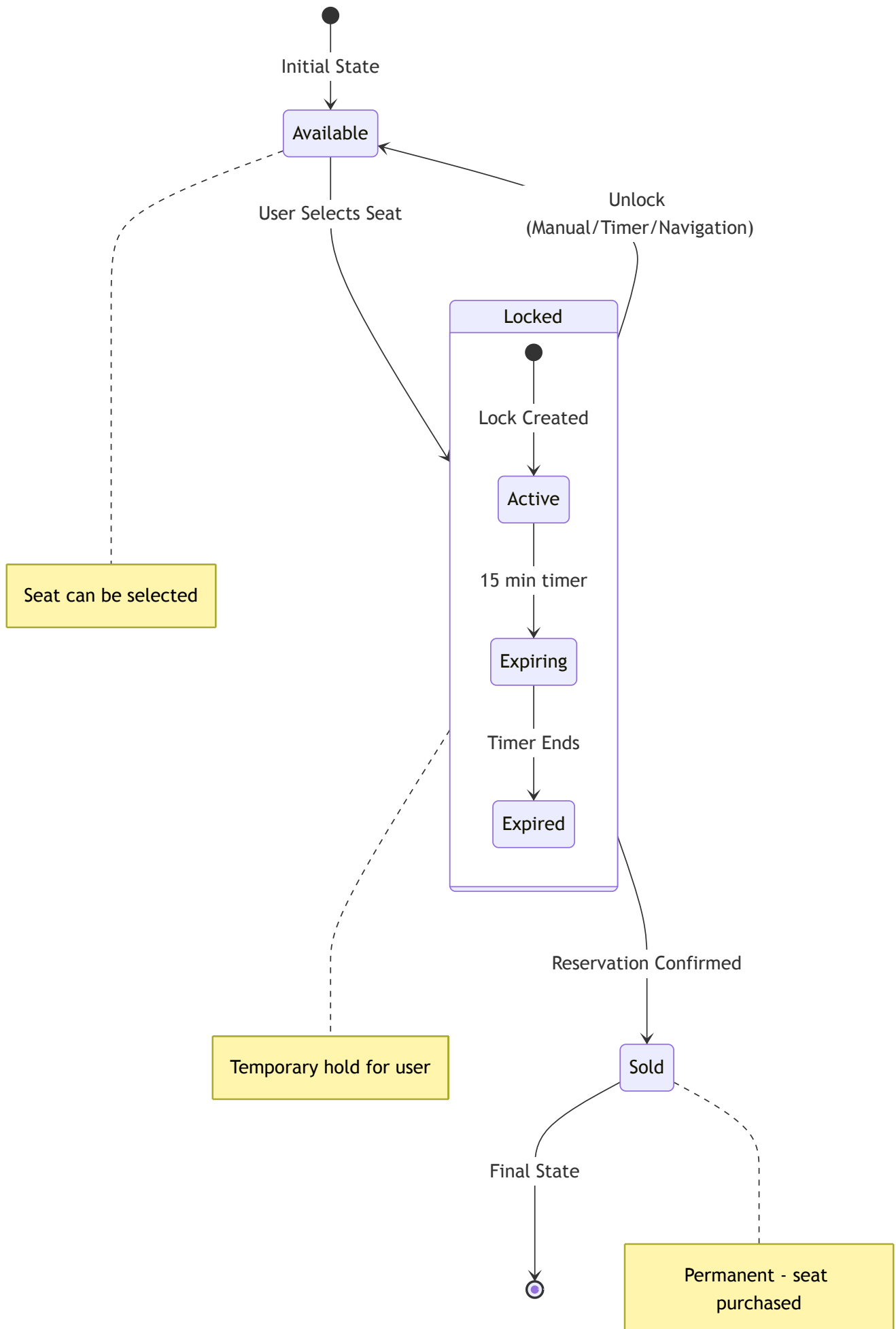


#### Key Features:

- **Real-Time Updates:** Immediate reflection of seat status changes
- **User-Specific Locking:** Users can modify their own selections
- **Automatic Cleanup:** Expired locks automatically released
- **Visual Feedback:** Clear indication of seat states and pricing

## 4.3 Seat State Machine

The seat state machine ensures consistent behavior across all user interactions.



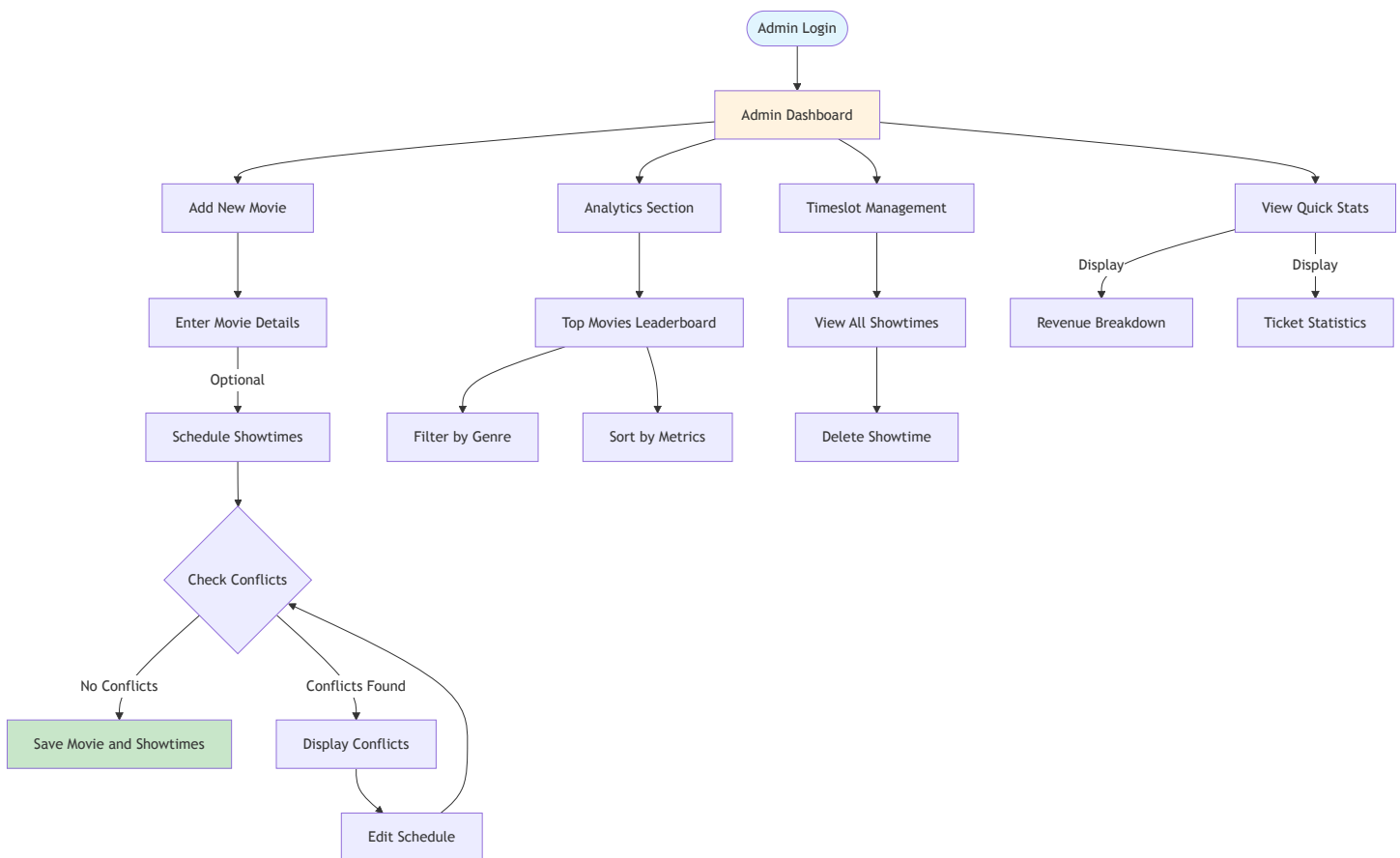
## State Transitions:

- **Available** → **Locked**: User clicks seat (immediate response)
- **Locked** → **Available**: Manual unlock, timer expiry, or navigation
- **Locked** → **Sold**: Successful reservation confirmation
- **Sold**: Terminal state (no further transitions)

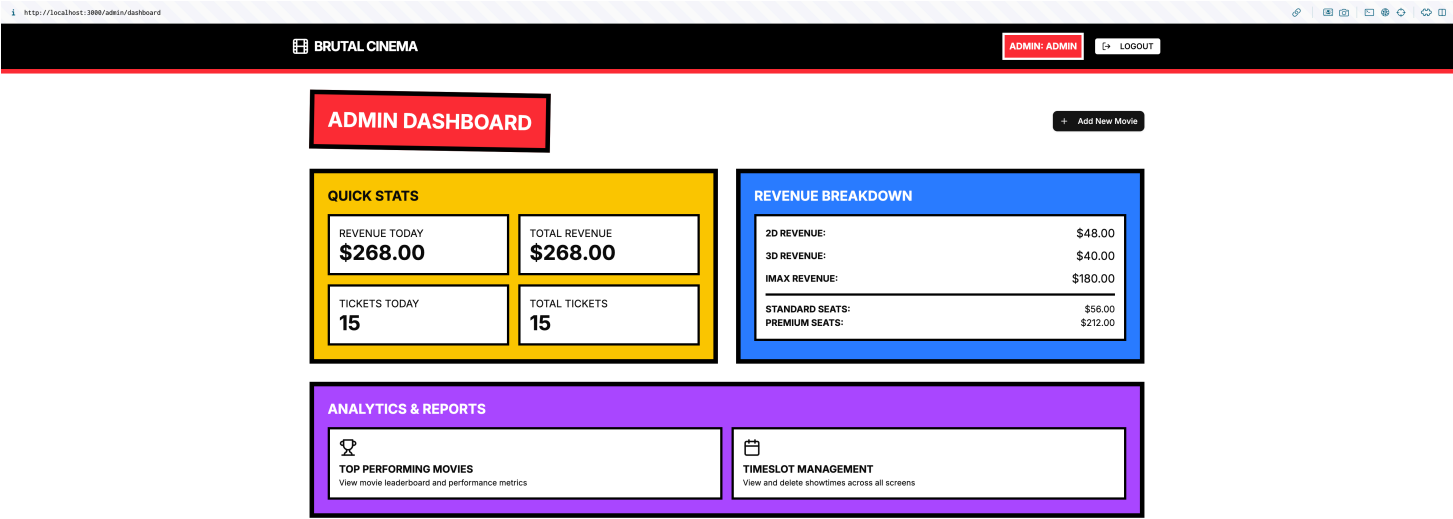
# 5. Administrative Operations

## 5.1 Admin Dashboard Workflow

The administrative interface provides comprehensive cinema management capabilities with intuitive workflows.



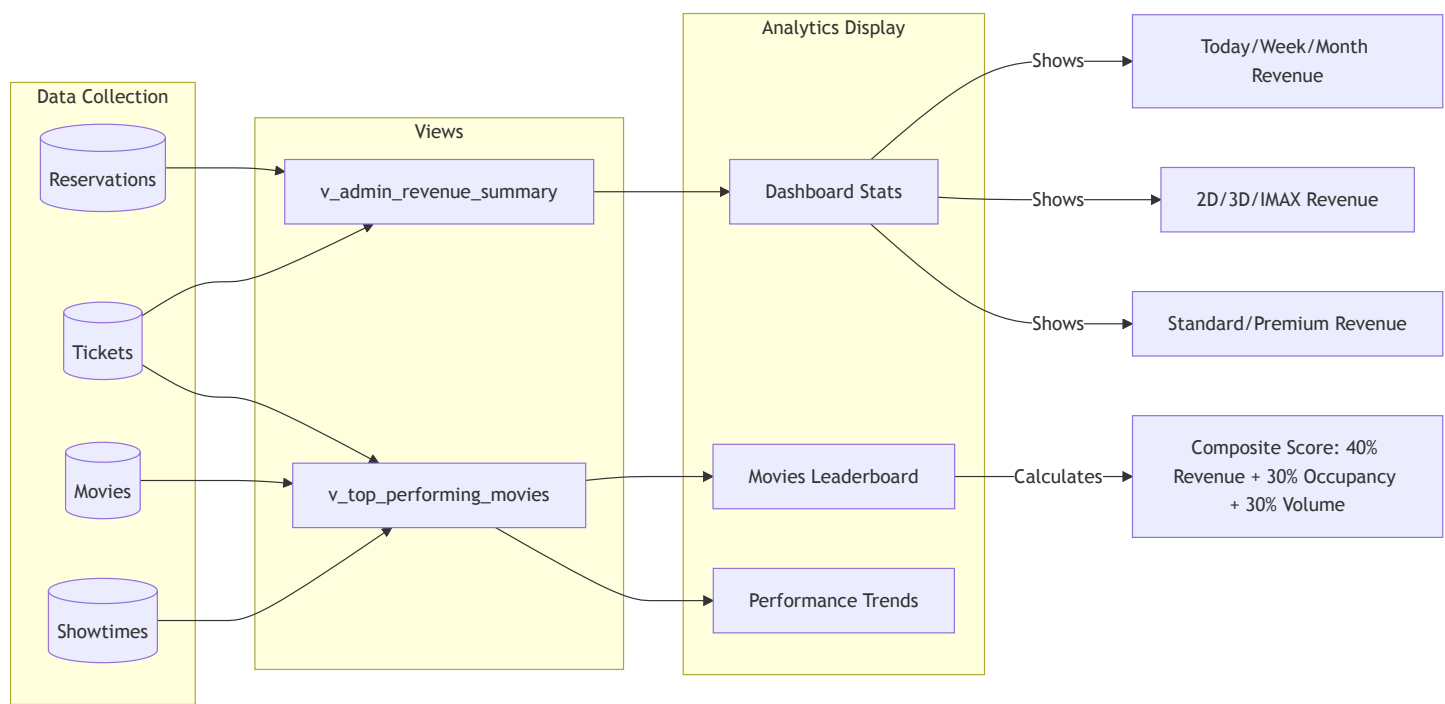
*Admin dashboard with quick stats and navigation options*



N

## 5.2 Analytics and Reporting System

Comprehensive analytics provide actionable insights for business decision-making.



Top movies leaderboard with performance metrics and filtering



## TOP PERFORMING MOVIES

LEADERBOARD BASED ON 90-DAY PERFORMANCE

SORT BY: ☒ OVERALL PERFORMANCE ☐ \$ REVENUE ☐ OCCUPANCY RATE ☐ TICKETS SOLDFILTER BY GENRE: ☒ ALL ☐ ACTION ☐ ANIMATION ☐ SCI-FI

RANK	MOVIE	GENRE	SHOWTIMES	TICKETS	REVENUE	OCCUPANCY	SCORE <small>REV + OCC + VOL</small>
1	<b>Avatar: The Way of Water</b> DIR: James Cameron RATING: PG-13	Action	25	9 AVG \$17.33	\$156.00 \$6.24/show	0.4%	95.0 REV: <div><div>100</div></div> 40% OCC: <div><div>83</div></div> 30% VOL: <div><div>100</div></div> 30%
2	<b>Test</b> DIR: RATING: PG-13		1	3 AVG \$24.00	\$72.00 \$72.00/show	3.1%	83.3 REV: <div><div>83</div></div> 40% OCC: <div><div>100</div></div> 30% VOL: <div><div>67</div></div> 30%
3	<b>Spider-Man: Across the Spider-Verse</b> DIR: Joaquim Dos Santos RATING: PG	Animation	25	3 AVG \$13.33	\$40.00 \$1.60/show	0.1%	66.7 REV: <div><div>67</div></div> 40% OCC: <div><div>67</div></div> 30% VOL: <div><div>67</div></div> 30%
4	<b>Fast X</b> DIR: Louis Leterrier RATING: PG-13	Action	27	0 AVG \$0.00	\$0.00 \$0.00/show	0.0%	0.0 REV: <div><div></div></div> 40% OCC: <div><div></div></div> 30% VOL: <div><div></div></div> 30%
5	<b>The Batman</b> DIR: Matt Reeves RATING: PG-13	Action	20	0 AVG \$0.00	\$0.00 \$0.00/show	0.0%	0.0 REV: <div><div></div></div> 40% OCC: <div><div></div></div> 30% VOL: <div><div></div></div> 30%
6	<b>Inception</b> DIR: Christopher Nolan RATING: PG-13	Sci-Fi	20	0 AVG \$0.00	\$0.00 \$0.00/show	0.0%	0.0 REV: <div><div></div></div> 40% OCC: <div><div></div></div> 30% VOL: <div><div></div></div> 30%
7	<b>Dune: Part Two</b> DIR: Denis Villeneuve RATING: PG-13	Sci-Fi	12	0 AVG \$0.00	\$0.00 \$0.00/show	0.0%	0.0 REV: <div><div></div></div> 40% OCC: <div><div></div></div> 30% VOL: <div><div></div></div> 30%
8	<b>Test</b> DIR: RATING: PG-13		0	0 AVG \$0.00	\$0.00 \$0.00/show	0.0%	0.0 REV: <div><div></div></div> 40% OCC: <div><div></div></div> 30% VOL: <div><div></div></div> 30%

## SCORING METHODOLOGY

COMPOSITE PERFORMANCE SCORE = 40% REVENUE + 30% OCCUPANCY + 30% VOLUME

Each metric is ranked using percentile scoring (0-100) based on the last 90 days of performance data. Higher scores indicate better relative performance compared to other movies.

## \$ REVENUE SCORE (40%)

## WHAT IT MEASURES:

- Total ticket revenue generated
- Financial performance
- Box office success

CALCULATION: Percentile rank of total revenue across all movies

## OCCUPANCY SCORE (30%)

## WHAT IT MEASURES:

- Average seat fill rate
- Operational efficiency
- Audience appeal per showing

CALCULATION: Percentile rank of (tickets sold + total seats available)

## VOLUME SCORE (30%)

## WHAT IT MEASURES:

- Total tickets sold
- Overall popularity
- Audience reach

CALCULATION: Percentile rank of total ticket count

## ADDITIONAL PERFORMANCE METRICS

## REVENUE EFFICIENCY:

Average revenue per seat. Indicates how well each movie performs financially.

## AVERAGE TICKET PRICE:

Average seat price. Calculated as total revenue divided by total seats sold.

**COLOR CODING:**

Occupancy rates: ≥75% GREEN, 50-74% YELLOW, <50% RED

**DATA WINDOW:**

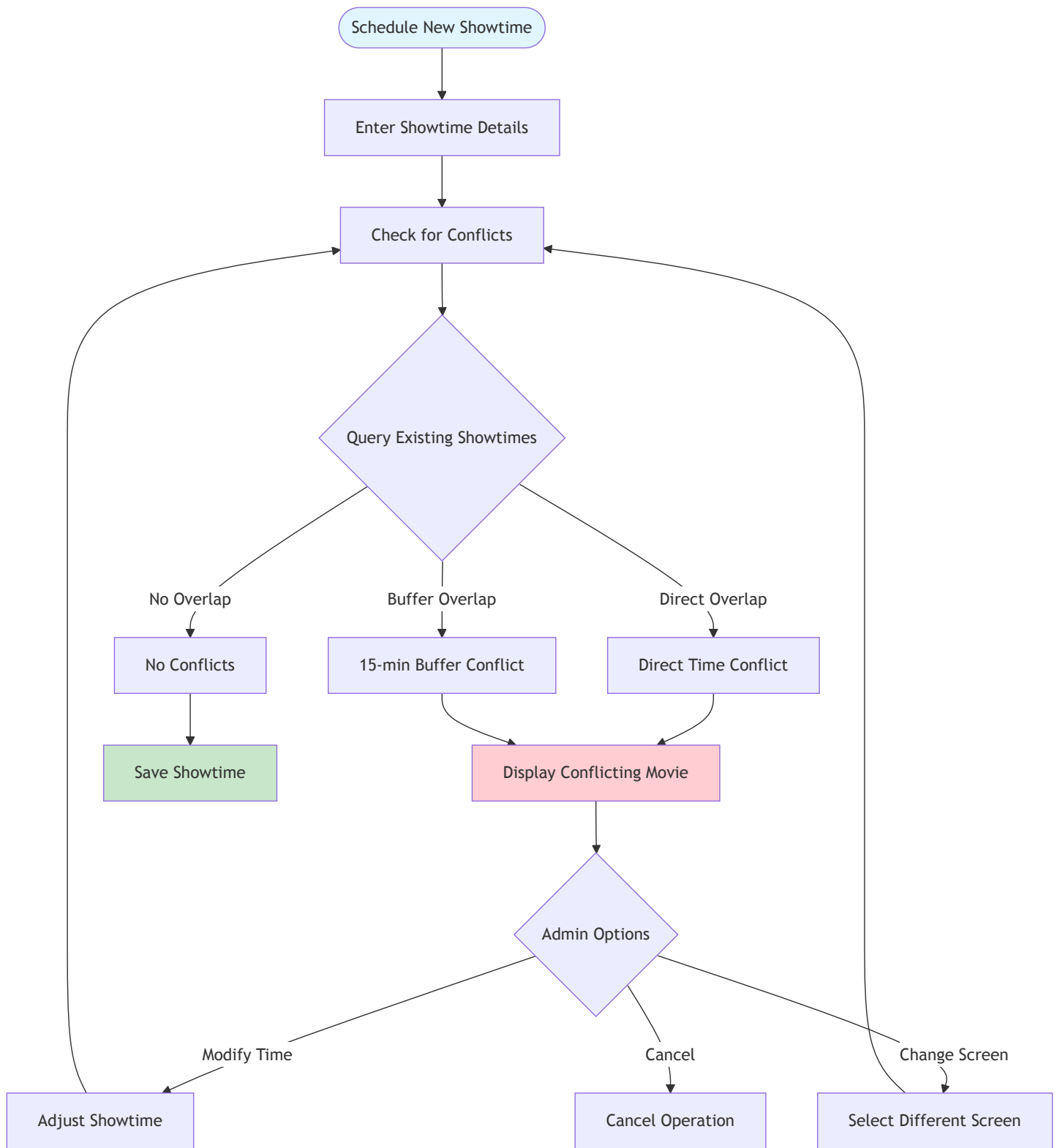
All calculations use the last 90 days of performance data for fair comparison

## Analytics Features:

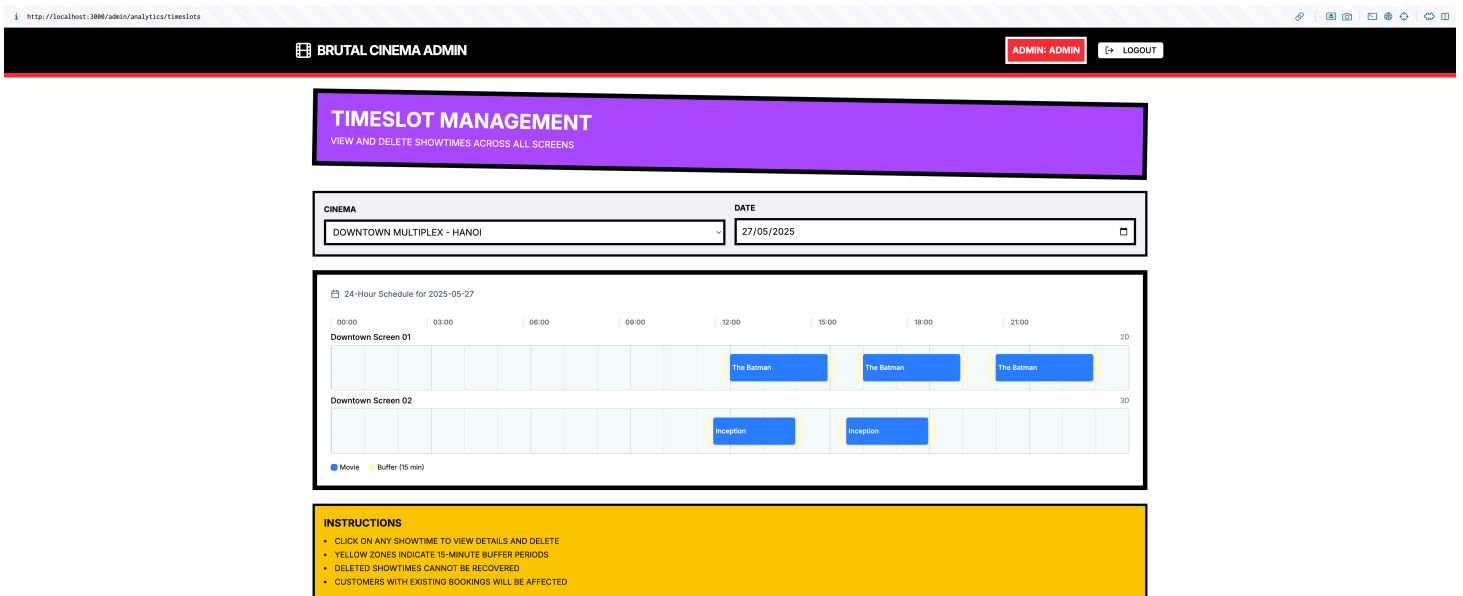
- **Revenue Tracking:** Time-based metrics (daily, weekly, monthly)
- **Performance Scoring:** Composite algorithm balancing multiple factors
- **Format Analysis:** Revenue breakdown by screen format (2D/3D/IMAX)
- **Occupancy Rates:** Real-time calculation of seat utilization

## 5.3 Conflict Resolution System

Automated conflict detection prevents scheduling errors and optimizes screen utilization.



*Timeslot management interface showing schedule conflicts*



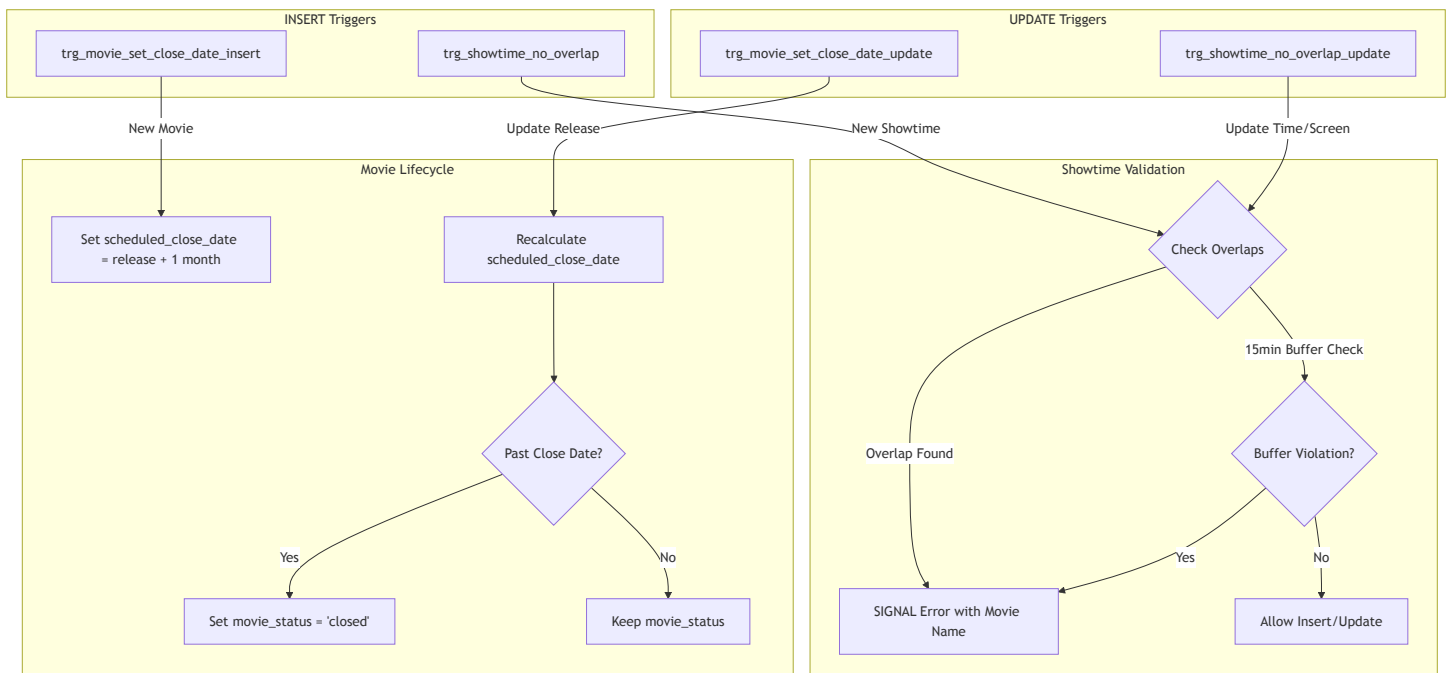
## Conflict Prevention Features:

- **15-Minute Buffer:** Automatic inclusion of cleaning/transition time
- **Real-Time Delete:** Immediate delete showtime on click (with confirmation)
- **Visual Indicators:** Clear display of showtimes
- **Flexible Resolution:** Multiple options for conflict resolution

# 6. Database Architecture and Design

## 6.1 Database Triggers

Implemented 4 triggers that enforce business rules and maintain data integrity at the database level.



## Trigger Functions:

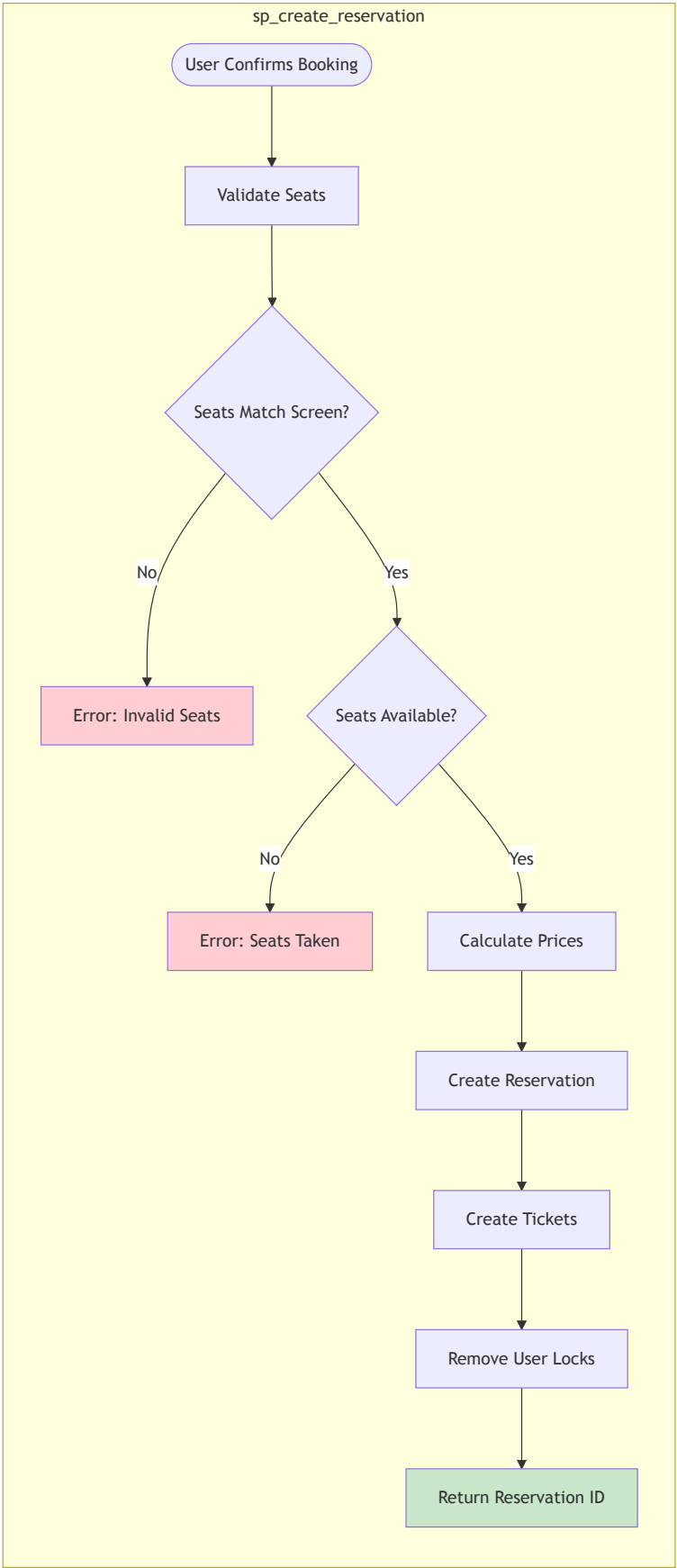
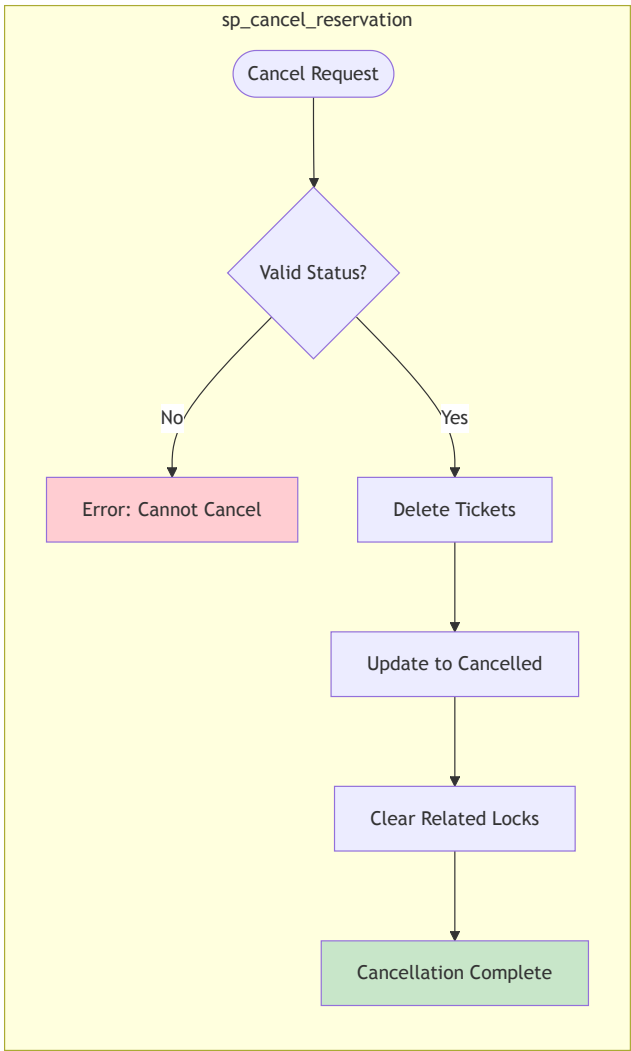
### 1. Showtime Overlap Prevention

- Validates new showtimes against existing schedules
- Enforces 15-minute buffer for cleaning and transitions
- Provides detailed error messages with conflicting movie names

### 2. Movie Lifecycle Management

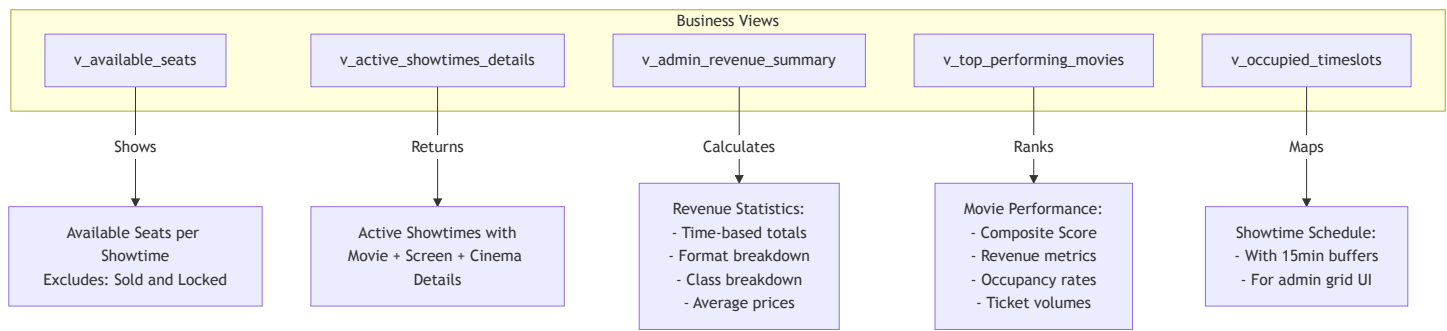
- Automatically sets close dates (release + 1 month)
- Updates close dates when release dates change
- Automatically closes expired movies

# 6.2 Stored Procedures



# 6.3 Views

Optimized views provide efficient access to complex analytical data. Including 5 views:



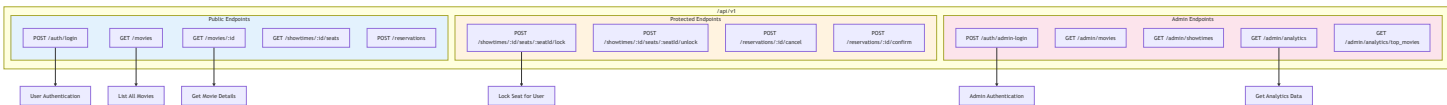
## View Optimization:

- **Performance:** Pre-calculated complex joins and aggregations
- **Consistency:** Single source of truth for analytical data

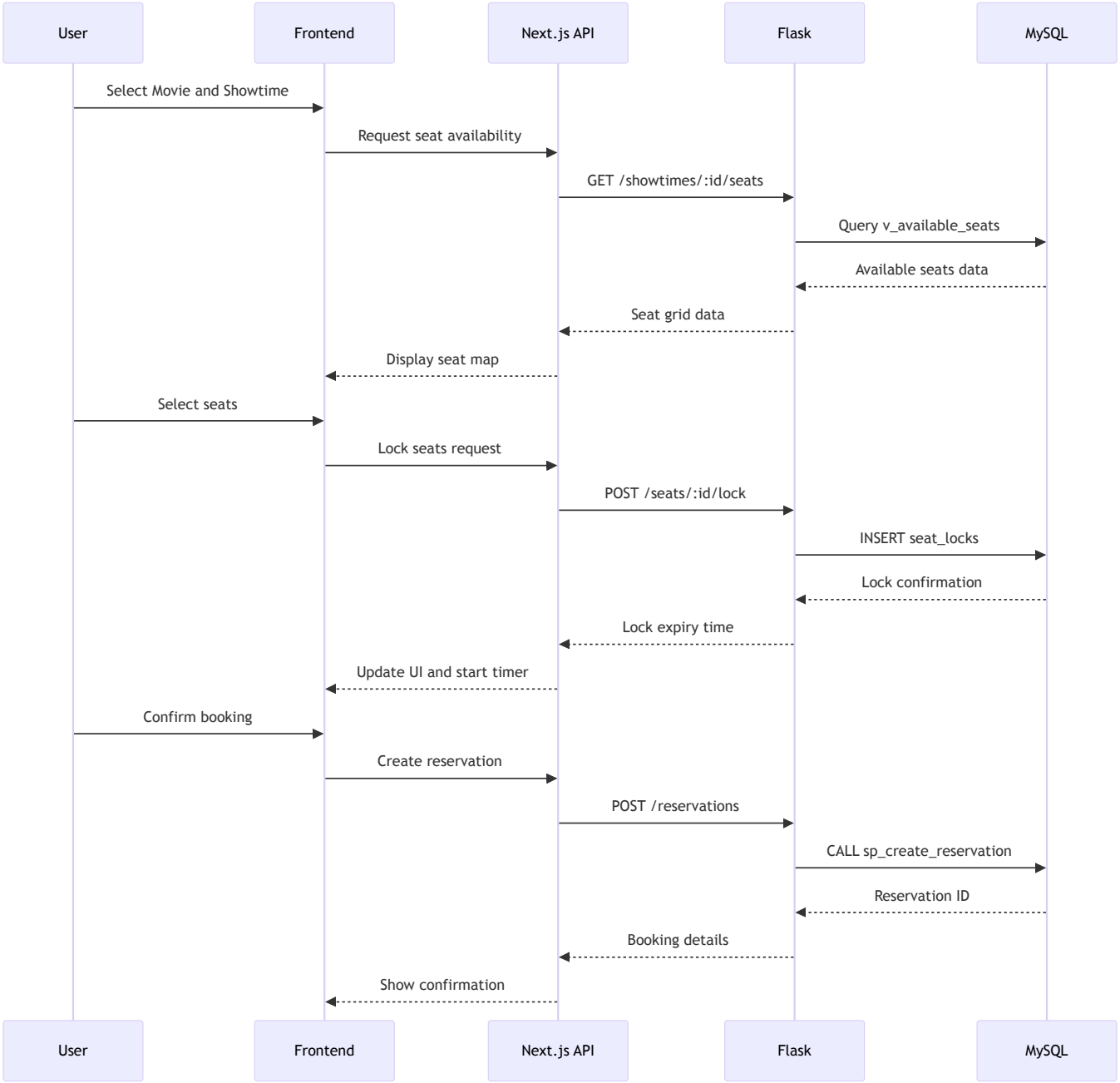
# 7. API Design and Implementation

## 7.1 API Structure

The RESTful API follows consistent patterns and provides comprehensive functionality for all system operations.



# 8. Component Interaction Sequence



## Conclusion

Our Movie Booking Management System is the combination between user-centric design and robust technical architecture. The database-first approach ensures data integrity while providing excellent performance, and the modular design supports future scalability and feature expansion.