US5 – Screening/Theater Room Registration

User Story

As a cinema administrator
I want to register screenings by specifying date, time, theater room, and movie
So that I can schedule and organize movie projections in the system

Acceptance Criteria

1. The system must allow selection of a previously registered movie.
2. The administrator must be able to choose an available theater room from the catalog.
3. Date (DD/MM/YYYY) and time (HH:MM) must be enterable.
4. The system must validate that no scheduling conflicts exist in the same theater room.
5. A visual confirmation must be displayed when the screening is successfully registered.
6. Data must be persisted in the Cinema Management microservice database.
7. The system must prevent registration of screenings on past dates.

Agile Process – Product Discovery

Problem Discovery

- **Identified problem:** The administrator currently lacks an in-system tool to plan screenings, which forces manual management and generates inconsistencies.
- **Impact:** Time wasted, errors due to duplicate scheduling, outdated information, and no automation for user notifications.

User Need

- Intuitive interface for scheduling.
- Automatic conflict validation.
- Ability to schedule multiple screenings efficiently.
- Clear visualization of room availability.

Proposed Solution

A **frontend form** connected to the Cinema Management microservice:

- Dropdown selectors for movie and theater room.
- Date/time pickers with validation.
- Real-time conflict detection.
- Preview before saving.
- Clear error messages.

Expected Prototype

- **Main panel:** Screening registration form (movie, room, date, time, optional price).
- **Secondary panel:** Daily view of screenings with timeline.
- **Interaction flow:** Fill → validate → confirm → update schedule.

Validation

- Success: Register 5 screenings in < 3 minutes.
- Error rate < 5%.
- Screenings update immediately in schedule view.
- 100% conflict detection.

Business Value

- 70% faster scheduling.
- Eliminates schedule conflicts.
- Updated, reliable schedule.
- Traceability and scalability.

figure5_mockup function 5

Definition of Done (DoD)

- Code reviewed.
- Unit/integration tests (>80% coverage).
- Frontend/backend validations.
- API documentation updated.
- Usability testing passed.
- Deployment in staging environment.

US6 – Movie–Screening Relationship

User Story

As a cinema administrator
I want to associate a movie with multiple screenings
So that the system correctly manages the schedule and users see all options available

Acceptance Criteria

1. A movie can be linked to multiple screenings (1:N).
2. Each screening must belong to one movie.
3. Deleting a movie warns about its screenings.
4. The system allows querying all screenings per movie.
5. Relationship persisted in DB.
6. Possible to modify the movie of an existing screening.
7. Visual summary of screenings per movie.

Agile Process – Product Discovery

- **Problem:** Currently, movies and screenings exist independently, causing orphan data and inconsistent schedules.
- **User Needs:** Admins need quick assignment and overview; users need all showtimes in one place.
- **Solution:**
    - Data model with FK (screening → movie).
    - REST endpoints (assign, bulk assign, query).
    - Batch creation for weekly templates.
- **Prototype:** Admin view with quick scheduling, accordion view of screenings, filters by date/room/status.

Validation

- Test: Create “Avengers 5” → 42 screenings (2 rooms × 3 times × 7 days).
- Verify schedule, deletion warning, and queries.
- Metrics: < 5 min weekly scheduling, no orphan screenings, queries < 200ms.

Business Value

- Weekly scheduling 10x faster.
- Data consistency.

- User-friendly showtimes.
- Supports analysis by title.

DoD

- Data model updated.
- Endpoints implemented and documented.
- Unit/integration tests (>85% coverage).
- Frontend quick scheduling functional.
- Load tests (100 movies × 50 screenings).
- Documentation delivered.

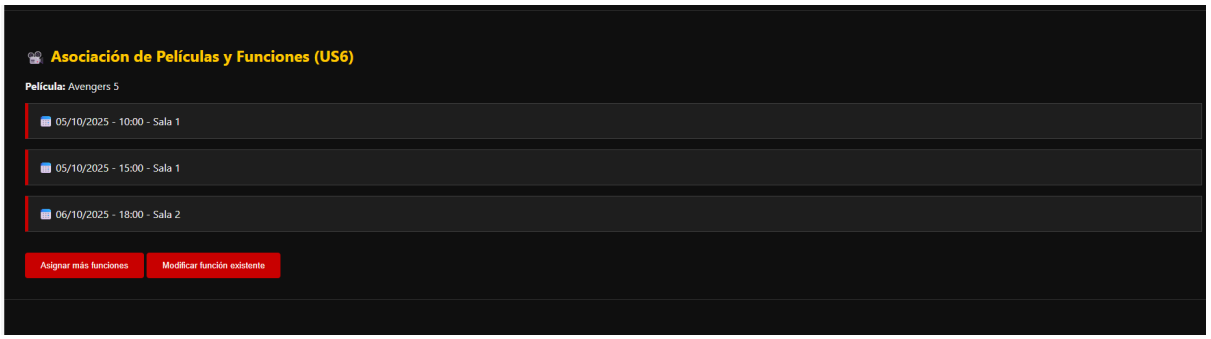

figure6_mockup function 6

US7 – Screening Query (End User)

User Story

As a cinema user
I want to see all screenings for a movie with times, rooms, and availability
So that I can choose the most convenient option

Acceptance Criteria

1. Accessible from movie detail view.
2. Screenings grouped by date.
3. Show time, room, seat availability, price.
4. No past screenings shown.
5. Low availability (<20%) highlighted.
6. Filters: date, time slot, room.
7. Loading < 1s.
8. Responsive design.

Agile Process – Product Discovery

- **Problem:** Users abandon purchases if showtimes aren't clear.
- **Needs:** Fast, filtered access to all screenings, with real-time availability.
- **Solution:** Optimized GET endpoint with caching + filters.
- **Prototype:** Movie detail view with tabs for dates, cards with showtime info, responsive design.

Validation

- User test: Buy 2 tickets for Saturday evening screening.
- Metrics: Task time < 45s, >95% success, SUS > 80/100.
- A/B test on card vs table design.

Business Value

- +25% screening views.
- +15% conversion to purchase.
- –60% support calls.

DoD

- Endpoint implemented and optimized.
- Cache with invalidation.
- Responsive frontend with filters.
- Tests and analytics metrics.
- Product Owner approval.

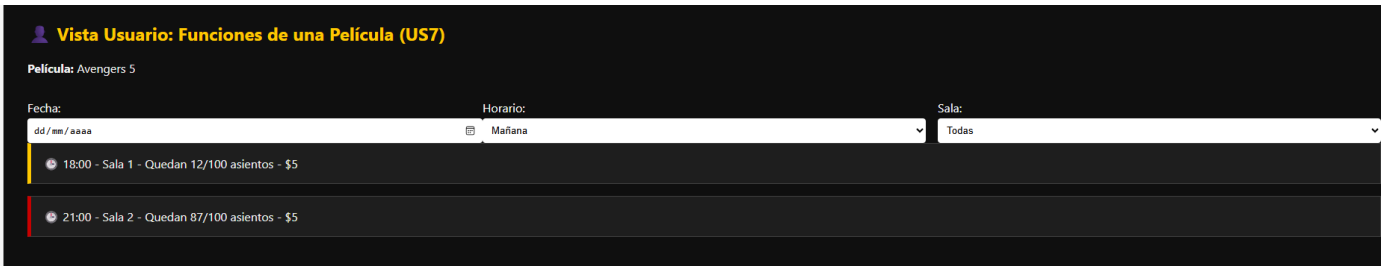

figure7_mockup function 7

US8 – Screening Deletion

User Story

As a cinema administrator
I want to delete screenings with confirmation and ticket handling
So that the schedule stays accurate and users aren't misled

Acceptance Criteria

1. Deletion from admin panel.
2. Confirmation required.
3. If tickets sold:
    - Show number of affected tickets.
    - Require cancellation reason.
    - Notify affected users.
4. Deleted screening disappears from public view.
5. Soft delete maintained in DB (historical record).
6. Audit trail (who/when).
7. Cannot delete past/ongoing screenings.

8. Batch deletion option.

Agile Process – Product Discovery

- **Problem:** Cancelled screenings remain public, creating complaints, refunds, and legal risks.
- **Needs:** Fast deletion, impact transparency, user notifications, auditability.
- **Solution:** Admin panel with deletion flow, soft delete, notification system, role-based restrictions.

Validation

- Scenario: Projector failure → delete all Room 3 screenings same day.
- Metrics: Deletion < 30s, 100% user notifications, audit log generated.

Business Value

- Preserves user trust.
- Reduces operational complaints.
- Ensures legal compliance.
- Provides complete traceability.

DoD

- Soft delete implemented.
- Notifications integrated (email/SMS).
- Role-based restrictions.
- Tests (unit, integration, usability).
- Logs/audit trail available.
- Approved by Product Owner.

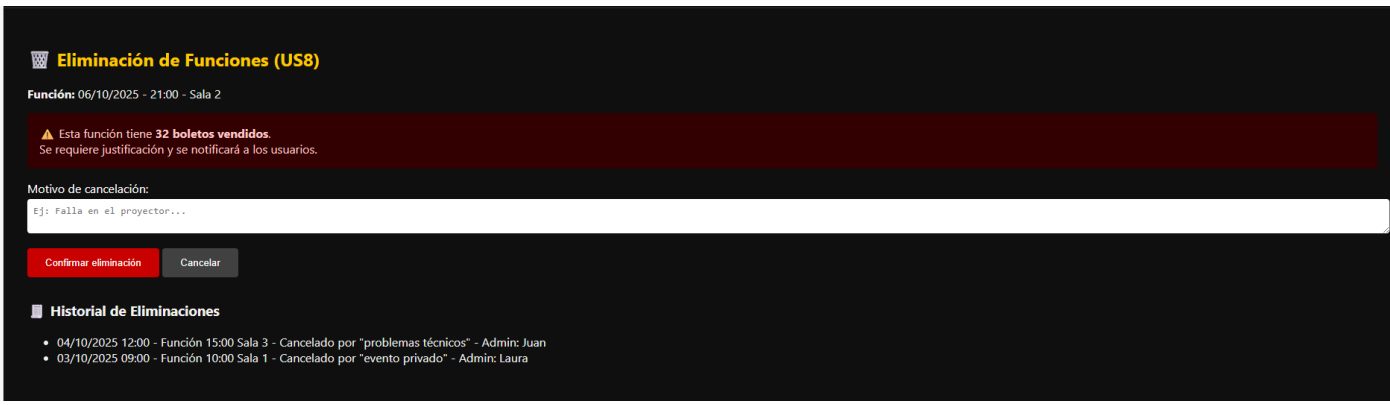

figure8_mockup function 8