Software Engineering Seminar

ProTicket

Business Context, User Requirements and Initial Planning Artifacts

Tutor: Carlos Andres Sierra

Students:

- Juan Daniel Vanegas Mayorquín — *Cod:* 20222020077
- Juan Sebastian Villalba Roa — *Cod:* 20201020066
- Daniel Santiago Arcila Martínez — *Cod:* 20191020075

Universidad Francisco José de Caldas
School of Engineering
Systems Engineering

Bogotá D.C. 2025

1 Objective and Scope

Objective

Design and implement *ProTicket*, a web application that enables event organizers to publish events and buyers to purchase digital tickets securely, delivering an end-to-end flow from event discovery to ticket delivery and post-purchase access.

Product Vision

Provide a simple, reliable, and secure ticketing experience with digital delivery (QR/PDF), basic sales tracking for organizers, and clear self-service paths for buyers.

In Scope

- **Buyer experience:** account creation/login, browse events, event detail, basic search/filter, checkout via a single payment provider, digital ticket generation (QR/PDF), order/ticket history.
- **Organizer experience:** create/edit events (core fields), publish/unpublish, view sales summary and remaining capacity, download basic CSV report.
- **Support/content:** static Help/FAQ page and a `mailto:` contact link.
- **Platform foundations:** role-based access (buyer/organizer/admin minimal), basic validation, error handling, and consistent fee/total calculations.

Out of Scope

- Seat maps or assigned seating; multi-currency or multiple payment providers.
- Automated refunds/cancellations, dispute handling, or anti-fraud advanced features.
- Full support ticketing platform, SLAs, or NLP chatbot.
- Mobile apps; push notifications; SMS delivery.
- Complex analytics dashboards beyond counts/sums and CSV export.

Assumptions & Constraints

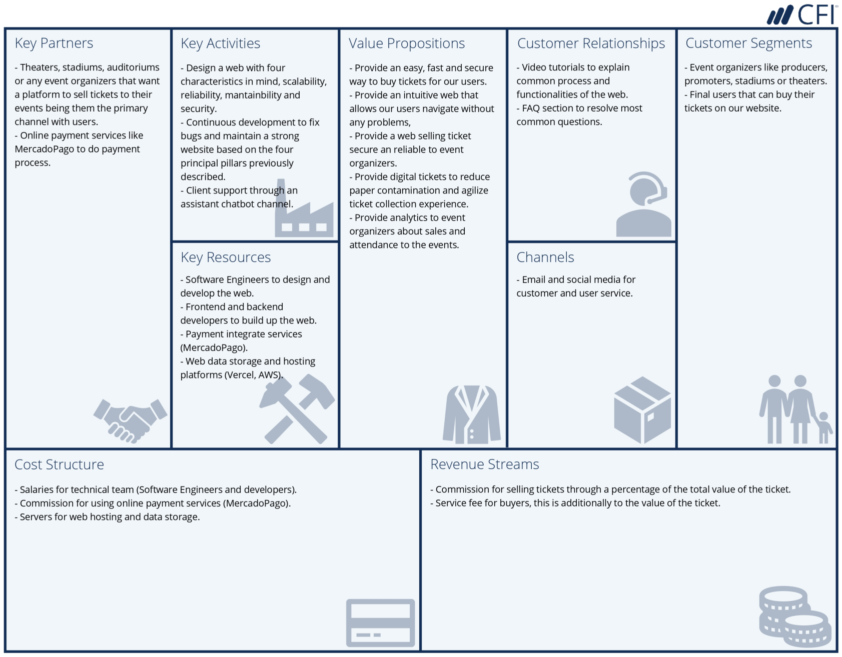
- Events are general-admission with finite capacity; time information is stored in UTC and displayed in the event's timezone.
- Payments are processed by a single gateway; webhook delays may occur, with manual status refresh available.

- Email is the only notification channel; if email fails, tickets remain accessible in “My Tickets”.
- Non-functional goals: acceptable page responsiveness, basic accessibility (labels, error states), and privacy (users only access their own orders/tickets).

Success Criteria

- A buyer can complete the full flow: discover an event → select quantity → pay → receive/download a unique QR/PDF ticket → find it later in “My Tickets”.
- An organizer can publish an event and see sales/remaining capacity; CSV export produces correct headers and totals.
- Capacity is consistent under concurrency (no overselling) and only decreases on Paid; failed/cancelled orders do not issue tickets.

2 Business Model Canvas



CFI

Key Partners

- Theaters, stadiums, auditoriums or any event organizers that want a platform to sell tickets to their events being them the primary channel with users.
- Online payment services like MercadoPago to do payment process.

Key Activities

- Design a web with four characteristics in mind, scalability, reliability, mantainbility and security.
- Continuous development to fix bugs and maintain a strong website based on the four principal pillars previously described.
- Client support through an assistant chatbot channel.

Key Resources

- Software Engineers to design and develop the web.
- Frontend and backend developers to build up the web.
- Payment integrate services (MercadoPago).
- Web data storage and hosting platforms (Vercel, AWS).

Value Propositions

- Provide an easy, fast and secure way to buy tickets for our users.
- Provide an intuitive web that allows our users navigate without any problems,
- Provide a web selling ticket secure an reliable to event organizers.
- Provide digital tickets to reduce paper contamination and agilize ticket collection experience.
- Provide analytics to event organizers about sales and attendance to the events.

Customer Relationships

- Video tutorials to explain common process and functionalities of the web.
- FAQ section to resolve most common questions.

Channels

- Email and social media for customer and user service.

Customer Segments

- Event organizers like producers, promoters, stadiums or theaters.
- Final users that can buy their tickets on our website.

Cost Structure

- Salaries for technical team (Software Engineers and developers).
- Commission for using online payment services (MercadoPago).
- Servers for web hosting and data storage.

Revenue Streams

- Commission for selling tickets through a percentage of the total value of the ticket.
- Service fee for buyers, this is additionally to the value of the ticket.

Figure 1: Business Model Canvas for ProTicket.

3 User Stories

3.1 Role: Buyer (End User)

3.1.1 US-01 — Sign Up & Login (Priority: MVP)

As a buyer **I want** to create an account and log in **so that** I can purchase and view my tickets.

Acceptance Criteria

- Unique email; duplicates are rejected with a clear message.
- Minimum password rules are enforced.
- Successful login redirects to “My Tickets” or “Events”.
- Logout clears the session.

Edge Cases

- Invalid email format, weak password, wrong credentials (generic error), expired session.
- *Enhancements:* email verification, lockout/rate-limit after many failed attempts.

3.1.2 US-02 — Browse Events (Priority: MVP)

As a buyer I want to see upcoming events so that I can choose one to buy tickets.

Acceptance Criteria

- List shows title, date, location, price-from, availability.
- Event detail page includes description and remaining capacity.
- Basic search/filter (text or date).
- Empty state when no results.

Edge Cases

- Empty catalog; past events are hidden; simple pagination for long lists; display dates in the event’s timezone.
- *Enhancements:* advanced filters (price/city), custom sorting or search engine optimization.

3.1.3 US-03 — Checkout with Online Payment (Priority: MVP)

As a buyer I want to purchase tickets using a payment gateway so that I can complete my order securely.

Acceptance Criteria

- Quantity cannot exceed available stock; totals include platform fees.
- Payment redirects to the provider and back with success/failure status.
- Successful payment decreases capacity; failed/cancelled does not issue tickets.
- Order shows final status after returning from the gateway.

Edge Cases

- **Concurrency:** two users try to buy the last ticket → only one succeeds.
- **Idempotency:** double-click on "Pay" does not duplicate orders.
- **Webhook fallback:** if gateway callback is delayed/fails, allow "Refresh payment status".
- Consistent totals; no negative amounts; fees applied uniformly.
- *Enhancements:* cart timeout/reservation, automatic webhook retries, multi-currency, refunds.

3.1.4 US-04 — Receive Digital Ticket (Priority: MVP)

As a buyer I want a digital ticket after payment so that I can present it at the venue.

Acceptance Criteria

- After successful payment, generate a unique ticket (code/QR).
- Confirmation email includes event, order id, and a ticket link.
- "My Tickets" lists paid tickets with a PDF download link.
- Only the buyer can access their ticket(s).

Edge Cases

- Atomic ticket generation; if email sending fails, the ticket remains available in "My Tickets".
- *Enhancements:* watermark/anti-fraud, QR revocation/rotation, Wallet pass.

3.1.5 US-05 — Reset Password (Priority: MVP)

As a user I want to reset my password via email so that I can regain access.

Acceptance Criteria

- Submitting a registered email sends a reset link that expires.
- The link is single-use; after reset, only the new password works.
- Non-registered emails get a neutral success message.

Edge Cases

- Expired/invalid token paths and messaging.
- *Enhancements:* rate-limit to avoid abuse; invalidate all active sessions after change.

3.1.6 US-10 — View My Orders & Tickets (Priority: MVP)

As a buyer I want to see my order/ticket history so that I can access them anytime.

Acceptance Criteria

- List orders with status (Paid/Failed/Pending).
- Paid orders display ticket codes and PDF links.
- Sorted by most recent first.

Edge Cases

- Empty history state; downloads disabled unless status = Paid.
- *Enhancements:* filters (by status/date); "resend confirmation email".

3.2 Role: Organizer (B2B)

3.2.1 US-06 — Create Event (Core) (Priority: MVP)

As an organizer I want to create an event post (title, date/time, location, description, price, capacity) so that it can be listed for sale.

Acceptance Criteria

- Required fields: title, start date/time, location, price, capacity.
- Status: Draft or Published; only Published is visible to buyers.
- Capacity > 0; price ≥ 0.
- Draft is editable; Published allows editing of description/status only (no reducing capacity below tickets sold).

Edge Cases

- Invalid dates (end $<$ start); capacity $\leq$ sold; price < 0.
- *Enhancements:* auto-save draft, duplicate event, media validation.

3.2.2 US-07 — View Sales Summary (Priority: MVP)

As an organizer I want to see sales and remaining capacity so that I can track performance.

Acceptance Criteria

- For each event: sold count, remaining, gross sales, fees.
- Numbers update after successful payments.
- Download CSV with basic columns (event, date, sold, revenue, fees).

Edge Cases

- Event with zero sales; timezone consistency with event view; correct CSV headers.
- *Enhancements:* date filters, tax breakdown, payment-method splits.

3.3 Role: Admin (Platform)

3.3.1 US-08 — Approve Organizer (Priority: Future)

As an admin I want to approve/reject organizer onboarding so that only verified organizers can publish.

Acceptance Criteria (Future)

- Pending organizer list with basic information.
- Set status = Approved/Rejected with an optional note.
- Only Approved organizers can publish events.

3.4 Role: Support (Simulated)

3.4.1 US-09 — Help Center (Static) (Priority: MVP — simulated)

As a user I want a simple Help/FAQ page so that I can resolve common issues quickly.

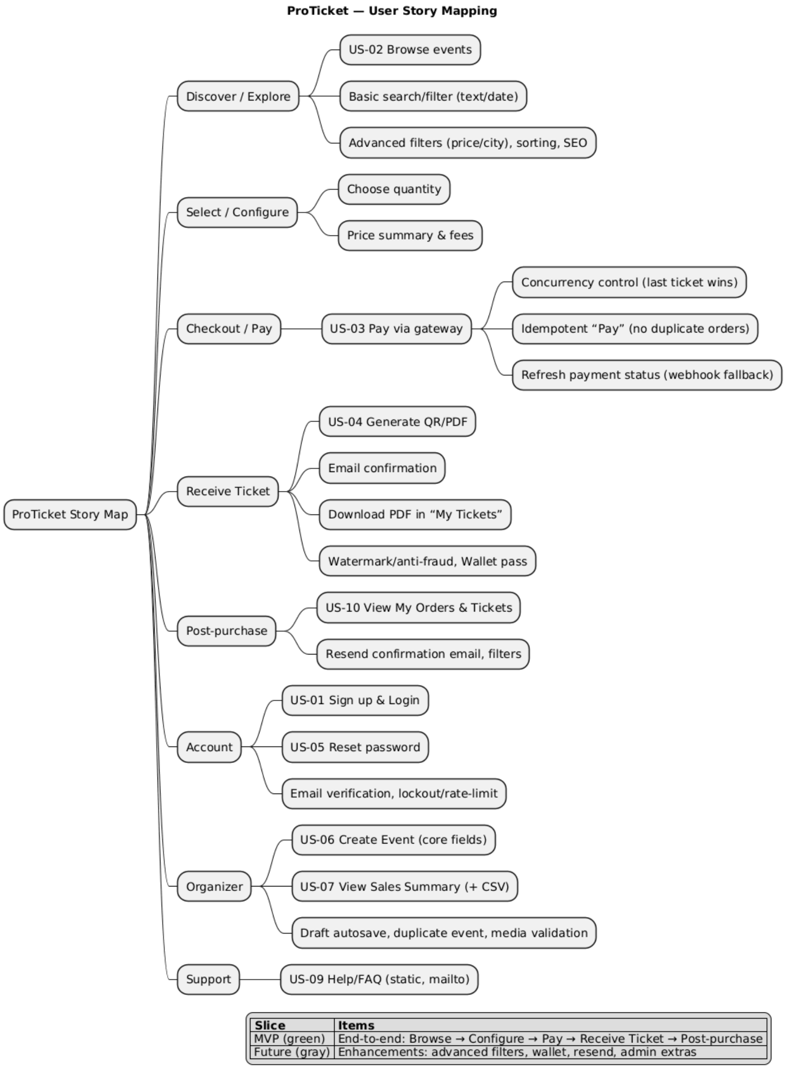
Acceptance Criteria

- Static “Help/FAQ” page linked in header/footer.
- 8–12 FAQs covering login, purchase, payment errors, tickets.
- “Contact support” is a `mailto:` link (simulated).
- (Optional) “Chatbot” stub with a few canned answers.

3.5 Cross-cutting Edge Cases

- **Authorization:** users can only view/download their own orders/tickets.
- **Timezones:** store times in UTC; display in the event's timezone.
- **Payment resilience:** if the webhook is delayed, allow manual "Refresh payment status" from the order.
- **Capacity consistency:** decrease only on *Paid*; restore on *Failed/Cancelled*.
- **Rounding/fees:** consistent rounding for totals and fees.
- **Accessibility/UX:** labeled inputs, clear error states, keyboard-friendly flows.
- **Privacy:** do not expose other users' emails/orders; PDFs include minimal personal data.

4 User Story Mapping


ProTicket — User Story Mapping
ProTicket Story Map
Discover / Explore
US-02 Browse events
Basic search/filter (text/date)
Advanced filters (price/city), sorting, SEO
Select / Configure
Choose quantity
Price summary & fees
Checkout / Pay
US-03 Pay via gateway
Concurrency control (last ticket wins)
Idempotent “Pay” (no duplicate orders)
Refresh payment status (webhook fallback)
Receive Ticket
US-04 Generate QR/PDF
Email confirmation
Download PDF in “My Tickets”
Watermark/anti-fraud, Wallet pass
Post-purchase
US-10 View My Orders & Tickets
Resend confirmation email, filters
Account
US-01 Sign up & Login
US-05 Reset password
Email verification, lockout/rate-limit
Organizer
US-06 Create Event (core fields)
US-07 View Sales Summary (+ CSV)
Draft autosave, duplicate event, media validation
Support
US-09 Help/FAQ (static, mailto)

Slice	Items
MVP (green)	End-to-end: Browse → Configure → Pay → Receive Ticket → Post-purchase
Future (gray)	Enhancements: advanced filters, wallet, resend, admin extras

Figure 2: User Story Mapping for ProTicket.

5 CRC Cards (Class–Responsibility–Collaborator)

Class: Buyer

Responsibilities: Register and log in to the platform, browse available events and select tickets to purchase, complete online payments securely, view order and ticket history, request password reset via email.

Collaborators: AuthService, EventCatalog, PaymentGateway, Order, Ticket, EmailService.

Class: AuthService

Responsibilities: Validate unique emails and enforce password rules, manage login sessions and handle expiration or logout, send verification and password reset emails, generate and validate temporary reset tokens.

Collaborators: Buyer, EmailService.

Class: EventCatalog

Responsibilities: Display list of available events with filters and pagination, provide search functionality by title, date, or location, show event details such as description, capacity, and price, hide past or sold-out events.

Collaborators: Event, Buyer, Organizer.

Class: PaymentGateway

Responsibilities: Process transactions through external providers, redirect users to the payment provider and return with status, confirm payment via webhook callback, ensure idempotency to prevent duplicate orders.

Collaborators: Buyer, Order, Event, Ticket.

Class: Order

Responsibilities: Store order information such as event, quantity, total, and status, update status according to payment result (Paid, Failed, Pending), maintain stock consistency for events, provide a list of all orders per user.

Collaborators: Buyer, Event, PaymentGateway, Ticket.

Class: Ticket

Responsibilities: Generate a unique code or QR after successful payment, link the ticket to its event and buyer, provide a downloadable PDF version, ensure that only the correct buyer can access their tickets.

Collaborators: Order, Event, Buyer, EmailService.

Class: EmailService

Responsibilities: Send confirmation and ticket delivery emails, send password recovery and verification links, handle basic support or contact messages.

Collaborators: Buyer, Ticket, AuthService, SupportCenter.

Class: Organizer

Responsibilities: Create, edit, and publish event listings, define event details such as price, capacity, and dates, review sales summaries and performance reports.

Collaborators: Event, Admin, SalesReport.

Class: Event
Responsibilities: Store essential event information, control available capacity and sold tickets, manage event status (Draft, Published).
Collaborators: Organizer, EventCatalog, Order, Ticket.

Class: SalesReport
Responsibilities: Calculate tickets sold and remaining capacity, generate CSV reports including revenue and fees, update figures after successful payments.
Collaborators: Event, Organizer, PaymentGateway.

Class: Admin
Responsibilities: Approve or reject organizer onboarding requests, restrict event publishing to verified organizers.
Collaborators: Organizer, Event.

Class: SupportCenter
Responsibilities: Provide a static Help/FAQ page for common questions, handle basic support contact via email or chatbot.
Collaborators: Buyer, EmailService.

References

[1] *Corporate Finance Institute.* (n.d.). *Business Model Canvas Examples.* [Online]. Available: `https://corporatefinanceinstitute.com/resources/management/business-model-canvas-examples`

[2] *Planio.* (2019, Enero 14). *A Guide to User Story Mapping: Templates and Examples (How to Map User Stories).* [Online]. Available: `https://plan.io/blog/user-story-mapping/`