

- Linea – Solution Overview (What Exists Today)
  - Who uses the system (personas)
  - Key modules at a glance
- Authentication and sessions (how sign-in works)
  - Magic-link sign-in (Visitors and Owners)
  - Admin password sign-in (Admins only)
  - Session lifecycle
- Roles, access, and redirects (business behavior)
- Public Website – what users can do
  - Discover and browse events
  - Event details
  - Favorites
- Owner Portal – what owners can do
- Admin Portal – what admins can do
- Integrations – inbound webhooks
- Business rules (selected highlights already implemented)
- API endpoints (non-exhaustive, for validation/testing)
  - Auth
  - Public Events
  - Favorites (requires sign-in)
  - Owner
  - Admin
  - Webhooks
- How to test in Production (step-by-step)
- Configuration considerations for PROD
- What's not covered here

# Linea – Solution Overview (What Exists Today)

---

This document explains, in clear business terms, what already exists and works in the Linea application. It's intended for the solution team to validate current behavior and to use when testing in production.

## Who uses the system (personas)

- **Visitor:** Anyone who lands on the public site to discover events. Can sign in using an email magic link. Can save favorites after sign-in.
- **Owner:** A designer/organizer who manages their events and audiences (waitlists/registered users) via the Owner Portal.
- **Admin:** Internal team members who manage the platform through the Admin Portal (owners, users, events, categories, areas, products, overview stats).

## Key modules at a glance

- **Public Website:** Browse/search events, view event details, sign in with email, mark favorites.
- **Owner Portal:** View and manage owned events, see and contact registered users, manage profile/theme, view analytics.
- **Admin Portal:** Overview dashboard, manage owners, users, events, categories, areas, products.
- **Favorites:** Signed-in users can save/unsave events.
- **Webhooks:** Inbound integrations for Telegram, WhatsApp, SMS (Twilio), and Email (SendGrid) are wired to a unified webhook service.
- **User Preferences (API):** Endpoints exist for getting/updating user preferences (auth required). Frontend integration may be partial.

## Authentication and sessions (how sign-in works)

---

### Magic-link sign-in (Visitors and Owners)

- From the UI (Owner Portal or Admin Portal login form), users enter their email and request a magic link.
- Backend creates a verification record and sends an email (SendGrid is used if configured; otherwise it safely logs the send in non-production configs).
- The user clicks the `/auth/callback?token=...` link in the email.
- The backend verifies the token, upserts the user if needed (default role: **VISITOR**), records audit activity, and creates a session.
- A secure session cookie (default name: `linea_session`) is set for subsequent requests.

Notes:

- In production, emails are sent when `SENDGRID_API_KEY` and sender settings are configured. If email is not configured, you will not receive the message and cannot complete magic-link login in PROD.
- The system supports role-based redirects: Admins go to Admin Portal, Owners to Owner Portal; Visitors remain on public site.

## Admin password sign-in (Admins only)

- Admins can sign in with email + password via an admin login endpoint.
- Password check uses the configured seed admin password (`SEED_ADMIN_PASSWORD`).
- On success, a session is created (cookie), and last-login is updated.

## Session lifecycle

- The session is stored server-side (via the session service) and referenced by a secure cookie in the browser.
- `GET /auth/me` returns whether the current browser session is authenticated and the user's email/role.
- Sign-out is done with `POST /auth/signout` (frontend calls this and refreshes state).

## Roles, access, and redirects (business behavior)

---

- **Visitor**

- Can browse all public events.
- After sign-in (magic link), can create and manage their favorites.
- Cannot access Owner or Admin functionality.

- **Owner**

- After sign-in, gains access to the Owner Portal.

- Can see only their events and their audiences (registered users), send bulk messages/emails to them, and update event details and profile/theme.

- **Admin**

- After sign-in, is redirected to the Admin Portal.
- Can view overview stats and manage owners, users, events, categories, areas, and products.

- **Redirect rules** (already implemented)

- If an authenticated user with role **OWNER** tries to access Admin Portal, they are redirected to Owner Portal.
- If an authenticated non-admin tries to access Admin Portal, they are redirected to the home page.

## Public Website – what users can do

---

### Discover and browse events

- The public events list is available from the home page. It supports filters:
  - **Text search** across title, description, short description, owner name, business name
  - **Category** (by slug)
  - **Status** in {DRAFT, PUBLISHED, CANCELLED, COMPLETED}
  - **Featured** (true/false)
  - **City** (case-insensitive)
  - **Owner** (name/business name, case-insensitive)
  - **Date From / Date To** (filters by **startDate** range)
- Only events with **isPublic = true** and **deletedAt = null** appear.

### Event details

- Each event page shows owner details (name/business), venue information (name/city/country), category (name/icon/color), and counts (waitlist, etc.).

# Favorites

- Signed-in users can favorite/unfavorite events.
- The favorites list is paginated and returns the event objects with owner, venue, category, and counts.

## Owner Portal – what owners can do

---

- **Sign-in** via email magic link.
- **View My Events:** Owners can view and filter their events by search, status, featured, and date range.
- **Create/Edit Events:** Owners can create new events and edit existing ones (title, descriptions, dates, capacity, images, social links, press kit, contact info, QR code URL, public/featured flags, etc.).
- **Registered Users (Audience):** Owners can view registered users per event, filter them, and paginate results.
- **Bulk Email:** Owners can select registered users and send bulk messages (subject + body) via the platform's email helper (SendGrid if configured).
- **Profile & Theme:** Owners can update their business profile (name/logo) and theme settings.
- **Analytics:** Owners can access event analytics pages for their events.

## Admin Portal – what admins can do

---

- **Overview Dashboard:** High-level stats (total owners, total events, total waitlist, active events). Uses the newer `/api/admin/dashboard` endpoint with a backward-compatible fallback to `/api/admin/overview` if needed.
- **Manage Owners:** Search, filter by status, sort, paginate, and edit owner records (name, email, phone, business details, address).
- **Manage Users:** Search, list, paginate users.
- **Manage Events:** Admin list and editing for events.
- **Manage Categories, Areas, Products:** Full CRUD through dedicated admin pages.
- **Role-aware Access:** Admin-only routes are guarded in the UI via `GET /auth/me` role checks; the backend routes require valid admin session.

# Integrations – inbound webhooks

---

The backend exposes unified webhook handling across multiple platforms:

- **Telegram:** `POST /webhooks/telegram`
- **WhatsApp:** `POST /webhooks/whatsapp`
- **SMS (Twilio):** `POST /webhooks/sms`
- **Email (SendGrid):** `POST /webhooks/email`
- **Generic:** `POST /webhooks/:platform`

Each webhook request is routed through a centralized Webhook Service that validates secrets, normalizes payloads, and processes messages consistently. A status endpoint `GET /webhooks/status` reports the service health, and `GET /webhooks/messages` (if enabled) can list processed messages for review.

## Business rules (selected highlights already implemented)

---

- **Events listing** returns only public and not-deleted events; supports rich filtering (text, category, status, featured, city, owner name, date range).
- **Favorites** are unique by (`userId`, `eventId`); adding a favorite twice will not create duplicates; delete removes the favorite by that key.
- **Authentication** relies on secure session cookies tied to server-side session storage. `GET /auth/me` is the canonical check for UI role-based rendering and route guards.
- **Admin login** uses the configured seed admin password; this flow is reserved for admin users.
- **Email sending** uses SendGrid when configured; otherwise the backend logs the email operation (non-breaking behavior).
- **Audit trail:** Successful magic-link logins write a user activity record (action `LOGIN`, with metadata such as method, IP, user-agent, token reference).

## API endpoints (non-exhaustive, for validation/testing)

---

Production base URL: <https://linea-production.up.railway.app/>.

## Auth

- GET <https://linea-production.up.railway.app/auth/me> → {  
  authenticated: boolean, user?: { email, role } }
- POST <https://linea-production.up.railway.app/auth/request-magic-link> body { email } → sends email if configured
- GET <https://linea-production.up.railway.app/auth/callback?token=...> → finalizes magic-link login, sets session cookie, redirects
- POST <https://linea-production.up.railway.app/auth/signout> → clears session
- POST <https://linea-production.up.railway.app/auth/admin/login> body { email, password } → admin password login (requires configured admin + seed password)

## Public Events

- GET <https://linea-production.up.railway.app/api/events> with optional query params:
  - search, category, status, featured, city, owner, dateFrom, dateTo

## Favorites (requires sign-in)

- GET <https://linea-production.up.railway.app/api/favorites> → returns { favorites, pagination }
- POST <https://linea-production.up.railway.app/api/favorites> body { eventId } → adds to favorites
- DELETE <https://linea-production.up.railway.app/api/favorites/:eventId> → removes favorite
- GET <https://linea-production.up.railway.app/api/favorites/:eventId> → { isFavorited, favoriteId }

# Owner

- GET <https://linea-production.up.railway.app/api/owner/events> → owner's events list
- GET <https://linea-production.up.railway.app/api/owner/profile> → owner profile info for header and branding
- Additional endpoints exist for creating/updating events and communicating with registered users.

# Admin

- GET <https://linea-production.up.railway.app/api/admin/dashboard> → overview stats (new)
- GET <https://linea-production.up.railway.app/api/admin/overview> → overview stats (legacy)
- GET <https://linea-production.up.railway.app/api/admin/owners> → paginated owners list
- Similar endpoints exist for users, events, categories, areas, products.

# Webhooks

- POST <https://linea-production.up.railway.app/webhooks/telegram|whatsapp|sms|email>
- POST <https://linea-production.up.railway.app/webhooks/:platform>
- GET <https://linea-production.up.railway.app/webhooks/status>

# How to test in Production (step-by-step)

---

Use a normal browser session; the app uses a secure session cookie. Avoid incognito while switching roles.



## 1. Validate site is reachable

- Open <https://linea-production.up.railway.app/> and confirm the home page loads with events.

## 2. Test public events browsing

- Navigate categories and use search/filters (city, date range, owner name). Confirm results update and show public events only.

## 3. Test magic-link login (Visitor)

- From header or relevant page, enter your email to request a magic link.
- Check your inbox and click the link.
- After redirect, open <https://linea-production.up.railway.app/auth/me> in a new tab or via the app's UI sign-in state; you should see **authenticated: true** and your role.
- If emails are not configured in PROD, this step will not complete (expected). In that case, coordinate with engineering for a pre-seeded session or a temporary admin login to proceed.

## 4. Test favorites

- As a signed-in Visitor, open an event and click favorite/unfavorite.
- Navigate to your favorites page and confirm the event appears/disappears.

## 5. Test admin password login (Admins)

- Go to <https://linea-production.up.railway.app/admin>.
- Sign in with your admin email and password (seed admin password must be set by engineering).
- You should be redirected to the Admin Portal and see overview stats.

## 6. Validate role redirects

- As an Owner: trying to open </admin> should redirect you to </owner>.
- As a Visitor: trying to open </admin> should redirect you to </>.

## 7. Owner Portal smoke test

- As an Owner, open </owner>.
- Confirm your events list loads. Open one event and verify you can see registered users.

- Optionally try editing an event or opening analytics.

## 8. Admin Portal smoke test

- Confirm dashboard stats load.
- Open Owners/Users/Events lists. Use search/sort/paging; open edit dialogs where applicable.

## 9. Webhook service health (optional)

- **GET https://linea-production.up.railway.app/webhooks/status** should return status object.
- Inbound platform webhooks require correct secrets/config; exercise only if arranged with the integration partner.

# Configuration considerations for PROD

---

- Email delivery (magic links, bulk messages) requires **SENDGRID\_API\_KEY** and sender settings configured. Without it, login links won't be delivered.
- Session cookie is named **linea\_session** by default; it must be allowed by the PROD domain and not blocked by the browser.
- Admin password login relies on **SEED\_ADMIN\_PASSWORD** and a corresponding admin user record.
- Database must be reachable and seeded for full functionality (events, owners, users, favorites, waitlists, analytics data).

# What's not covered here

---

- Exact database schema/migrations and rollback plans (covered in engineering docs).
- Non-user-facing admin/dev tooling.
- Low-level implementation details (Prisma models, service classes, etc.).

---

If anything above doesn't match what you observe in production, please note the URL, the time of the test, and the role you were using. Share a screenshot if possible so engineering can verify and adjust.