

Behavioral Health Connect - Development Scope

Full-Stack B2B SaaS Platform

What We're Building

A **two-sided marketplace** connecting referring clinicians (ED social workers, case managers) with behavioral health treatment facilities. Real-time bed availability directory that eliminates hours of phone tag.

Type: B2B SaaS web application

Users: Clinicians searching for patient placement + Treatment providers managing availability

Core Value: Real-time bed availability, verified providers, peer ratings

The Problem

Clinicians spend **2-3 hours daily** calling facilities to find patient placement. They rely on outdated spreadsheets with no current availability information. Providers have empty beds but no visibility to referring clinicians.

Result: Delayed patient care, clinician burnout, missed revenue for providers.

Three User Portals

1. Clinician Portal

Search and discover treatment facilities

2. Provider Portal

Manage facility profiles and bed availability

3. Admin Portal

Verify providers, manage users, monitor platform

Complete Feature Scope

PROVIDER PORTAL

Registration & Verification

Reference: [Provider Registration Mockup](#)

- Multi-step registration with facility information
- Recovery.com/LegitScript verification requirement
- Admin verification queue and approval workflow
- Email notifications for verification status

Profile Management

Reference: [Edit Profile Mockup](#)

Comprehensive facility profile editor:

- Basic info: Name, address, phone, email, website, logo
- Levels of care: Detox, Residential, PHP, IOP, Outpatient, Sober Living, MAT
- Insurance accepted: Multi-select from standardized list (Medicaid, Medicare, commercial, self-pay)
- Demographics served: Age ranges, gender, special populations (LGBTQ+, Veterans, First Responders)
- Specialties: SUD, Mental Health, Dual Diagnosis, Eating Disorders, Trauma
- Amenities: MAT, family programming, aftercare planning
- Profile preview mode
- Profile completion progress indicator

Bed Availability Management

Reference: [Provider Dashboard Mockup](#)

Critical feature - must be frictionless:

- Granular bed counts by level of care AND bed type
 - Example: "Residential (Male): 5 beds, Residential (Female): 2 beds, PHP: 8 beds"
- Simple +/- buttons for quick updates (touch-friendly)
- Auto-save (no submit button)
- Last updated timestamp (visible to clinicians)
- Visual status indicators: Available (green), Limited (yellow), Full (red)
- Bulk update capability
- Update history log

Magic Link Updates (Passwordless)

NEW FEATURE - Core requirement:

Providers can update beds via link sent to email or SMS without logging in.

How it works:

- System sends unique link daily at provider's preferred time
- Link format: `app.com/update-beds/{providerId}/{token}`
- Provider clicks → sees simple update interface → updates beds → submit
- **Token expires when:** Bed count is updated OR 48 hours passes (whichever comes first)
- No authentication required
- Delivery method: Email AND/OR SMS (provider configurable)
- Security: IP logging, rate limiting, notification to provider on update

Why this matters:

- Reduces friction for daily updates (critical for data freshness)
- Mobile-friendly for staff without computer access (especially SMS)
- Configurable schedule per provider (timezone-aware)
- Providers can enable/disable, set preferred update time and delivery method
- Single-use token prevents reuse after update

Admin controls:

- View token usage statistics
- Configure email/SMS schedules per provider
- Enable/disable email vs SMS per provider
- Manual token generation for testing

Dashboard

Reference: [Provider Dashboard Mockup](#)

- Bed availability widget (prominent, front-and-center)
- Quick update shortcuts
- Recent activity feed
- Profile views and search appearance stats
- Recent reviews and ratings
- Subscription status
- Last updated compliance indicator

File Upload

- Logo upload with cropping
 - Image optimization and resizing
 - Drag-and-drop interface
-

CLINICIAN PORTAL

Registration & Authentication

Reference: [Clinician Registration Mockup](#)

- Organization email verification (must use recognized healthcare org email)
- Organization lookup and matching system
- Domain verification for known organizations
- Manual verification queue for unknown orgs
- User profile setup (credentials, license info)

Dashboard

Reference: [Clinician Dashboard Mockup](#)

- Quick search interface (hero element)
 - Location + insurance + level of care
- Recently viewed providers with bed availability
- Saved searches management
- Notifications center
- Personalized recommendations

Advanced Search & Discovery

Reference: [Search Results Mockup](#)

Multi-filter search:

- **Location:** Zip code + radius (5, 10, 25, 50, 100 miles), city, state
- **Insurance:** Searchable multi-select
- **Level of care:** Multi-select (Detox, Residential, PHP, IOP, etc.)
- **Demographics:** Age, gender, special populations
- **Specialties:** SUD, mental health, dual diagnosis, etc.
- **Availability toggle:** Show only available beds

Search results display:

- Card-based layout with key info

- **Granular bed availability** (e.g., "Residential (M): 5 • PHP: 8 • IOP: 12")
- Distance from search location
- Rating stars and review count
- Accepted insurances (truncated with "+" indicator)
- Contact information with quick call button
- Save facility button
- Infinite scroll pagination
- Sort options: distance, rating, recently updated
- "No Results" state with suggestions

Map view:

- Interactive map with provider pins
- Clustered markers
- Map/list toggle
- "Search this area" functionality
- Marker click shows provider card
- Sync map bounds with results

Features:

- Real-time filter updates (no page reload)
- URL-based filter state (shareable links)
- Filter count badges
- "Clear All Filters" button

Provider Profile View

Reference: [Provider Detail Mockup](#)

Comprehensive detail page:

- Hero section with detailed bed availability breakdown
- Contact information (Call Now, Email buttons)
- Rating stars and review count
- Tabbed sections: Overview, Insurance, Reviews, Details
- Last updated timestamp
- Action buttons: Save, Call, Email, Log Contact
- Treatment specialties badges
- Insurance accepted list
- Breadcrumbs and navigation

Saved Facilities

Reference: [Saved Facilities Mockup](#)

Personal facility management:

- Save/unsafe providers
- Custom tags (High Priority, Follow Up, Recommended)
- Personal notes per facility
- Quick actions: View Details, Call, Log Contact, Edit Note
- Filter by tags
- Display when saved and last contacted
- Search saved facilities

Contact Log

Reference: [Contact Log Mockup](#)

Track all provider contacts:

- Log contact from provider detail page
- Capture:
 - Contact type (Phone, Email)
 - Contact date
 - Purpose and outcome
 - Contact person
 - Notes
 - Follow-up reminders
 - Status (Placement Made, Awaiting Response, No Availability, etc.)
- Chronological timeline organized by date
- Stats dashboard (today, this week, this month)
- Pending follow-ups counter
- Search and filter capabilities
- Export contact history

Rating & Review System

Reference: [Submit Review Mockup](#)

Review submission:

- Responsiveness rating (1-5 stars)
- Patient Acceptance (thumbs up/down/neutral)
- Overall Experience (1-5 stars)
- Optional written comment (moderated)
- Anonymous submission option
- Review guidelines and submission rules
- Thank you confirmation
- Edit/delete within 24 hours

Review display:

- List with filtering (most recent, highest rated)
- Pagination
- Summary statistics
- Provider responses inline
- Helpful/report buttons

ADMIN PORTAL

Dashboard

Reference: [Admin Dashboard Mockup](#)

Platform overview:

- Key metrics cards:
 - Active Providers
 - Active Clinicians

- Pending Verifications
- Total Searches
- System health monitoring
- Recent activity feed
- Pending actions table (verifications, reviews, support tickets)
- Quick action buttons

Provider Management

Reference: [Provider Management Mockup](#)

Verification queue:

- Filter by status: Pending, Verified, Suspended
- Review all provider details
- Check verification documents
- View contact information
- Recovery.com/LegitScript verification lookup
- Approve/reject with notes
- Contact provider option
- Display provider stats:
 - Beds available
 - Profile views
 - Response rate
 - Last updated
- Verification audit log
- Bulk management actions

User Management

Reference: [User Management Mockup](#)

Comprehensive user administration:

- User table with search/filter
- Filter by status, credentials, location
- User details:
 - Name, email, credentials, license
 - Organization
 - Joined date, last active
- Status indicators (Active, Inactive, Suspended)
- Pagination
- Bulk actions and export
- User detail view with full profile
- User activity logs
- Quick actions: View, Edit, Suspend

Platform Analytics

Reference: [Analytics Dashboard Mockup](#)

Data dashboard:

- Key metrics with trend indicators:
 - Total Searches

- Placements Made
- Reviews Submitted
- Interactive charts:
 - Search volume over time
 - User growth (clinicians vs providers)
 - Top providers by profile views
 - Popular search filters breakdown
 - Geographic distribution
- Time period selector (7/30/90 days, This Year, All Time)
- Export report functionality
- Real-time data updates
- Custom date range selection

Content Moderation

- Review moderation queue
- Flagged content management
- Moderation actions: approve, reject, hide, contact user
- Moderation audit log
- Moderation guidelines documentation

System Monitoring

- Error tracking and alerting
 - API performance metrics
 - Database query performance
 - Search success rates
 - Availability update compliance tracking
 - System health indicators
-

Design Mockups (Complete)

📍 **Location:** design-mockups/ folder
 🔗 **Access:** [Open Navigation Hub](#) | [View README](#)

18 Interactive HTML/CSS Prototypes

Authentication (3 pages):

- [Login Page](#) - Universal login
- [Clinician Registration](#) - Multi-step clinician registration
- [Provider Registration](#) - Provider registration with verification

Clinician Portal (6 pages):

- [Clinician Dashboard](#) - Quick search, recent views, saved searches
- [Search Results](#) - Advanced filters, map/list toggle
- [Provider Detail View](#) - Full provider profile view
- [Submit Review](#) - Review submission form
- [Saved Facilities](#) - Saved facilities with tags/notes
- [Contact Log](#) - Contact tracking timeline

Provider Portal (3 pages):

- [Provider Dashboard](#) - Bed availability widget, stats

- [Edit Profile](#) - Comprehensive profile editor
- [Provider Settings](#) - Account settings, billing

Admin Portal (4 pages):

- [Admin Dashboard](#) - Platform metrics, pending actions
- [Provider Management](#) - Verification queue
- [User Management](#) - User management table
- [Analytics Dashboard](#) - Charts and reports

Design System:

- [Design System](#) - Complete component library, colors, typography

All mockups are:

- Mobile-first responsive
 - WCAG 2.1 AA compliant
 - Interactive (functional forms, buttons, filters)
 - Built with HTML5, Tailwind CSS, vanilla JavaScript
-

Technical Requirements

Geospatial Search (Critical)

Must support:

- Search by zip code + radius (5-100 miles)
- Distance calculations from search location
- Map display with provider pins
- "Search this area" on map pan/zoom

Suggested approach: PostgreSQL with PostGIS extension

Real-Time Bed Availability

Requirements:

- Granular tracking by level of care AND bed type
- Last updated timestamp
- Status calculation (Available/Limited/Full)
- Update history/audit log
- Staleness detection (>24 hours)

Multi-Filter Search

Must support:

- Multiple simultaneous filters (AND logic)
- Fast response time (<500ms)
- Faceted search (show result counts per filter)
- URL-based state (shareable links)
- Sort and pagination

Suggested approach: Algolia or Elasticsearch for search layer

Authentication & Verification

Requirements:

- Email verification
- Organization email domain matching
- Role-based access control (Clinician, Provider Admin, Platform Admin)
- Session management
- Password reset

Suggested approach: Clerk, Auth0, or custom JWT implementation

File Upload

Requirements:

- Logo upload with cropping
- Image optimization
- Cloud storage
- CDN delivery

Suggested approach: Vercel Blob, AWS S3, or Cloudinary

Subscription & Billing

Requirements:

- Multiple subscription tiers (Clinician, Provider)
- Subscription management (upgrade/downgrade/cancel)
- Invoice generation
- Feature gating by tier
- Trial periods

Suggested approach: Stripe

Magic Link System

Requirements:

- Secure token generation (cryptographically secure, 64+ characters)
- Single-use tokens with expiration: **expires when bed count updated OR 48 hours (whichever first)**
- Scheduled delivery (cron job, timezone-aware)
- **Delivery options:** Email AND/OR SMS (provider configurable)
- Public update page (no auth required)
- Token usage tracking and analytics
- Security: Rate limiting, IP logging, notifications

SMS Integration:

- SMS gateway integration (e.g., Twilio)
 - Configurable per provider (email only, SMS only, or both)
 - Message template: "Update your bed availability: [link]"
 - Character limit considerations for SMS
-

Data Model (Core Entities)

User

- id, email, password_hash, role (CLINICIAN, PROVIDER_ADMIN, PLATFORM_ADMIN)
- organization_id, email_verified, created_at, updated_at

Organization (Clinician)

- id, name, type, email_domain, address
- subscription_tier, subscription_status, created_at

Provider (Facility)

- id, name, slug, description
- address, city, state, zip, lat, lng (geospatial)
- phone, email, website, logo_url
- verification_status, verification_source
- subscription_tier, is_active, created_at, updated_at

BedAvailability

- id, provider_id, level_of_care, bed_type (MALE, FEMALE, COED)
- total_beds, available_beds, status (AVAILABLE, LIMITED, FULL)
- updated_at, updated_by

BedUpdateToken (Magic Links)

- id, token (unique), provider_id
- expires_at, used_at (NULL until used, then token immediately invalidated)
- delivery_method (EMAIL, SMS, BOTH)
- ip_address, created_at

SavedProvider (Clinician Saved Facilities)

- id, clinician_id, provider_id
- tags (array), notes, created_at

ContactLog (Clinician Contact Tracking)

- id, clinician_id, provider_id
- contact_type (PHONE, EMAIL), contact_date
- purpose, contact_person, notes
- status, follow_up_date, created_at

Review

- id, provider_id, user_id
- responsiveness_rating (1-5), patient_acceptance (UP, DOWN, NEUTRAL)
- overall_rating (1-5), comment
- is_anonymous, is_moderated, provider_response, created_at

ProviderInsurance

- id, provider_id, insurance_id (many-to-many)

Insurance

- id, name, type (COMMERCIAL, MEDICAID, MEDICARE, SELF_PAY), state

Levels of Care (Enum)

- DETOX, RESIDENTIAL, PHP, IOP, OUTPATIENT, SOBER_LIVING, MAT, INPATIENT_PSYCH
-

Performance & Quality Requirements

Performance Targets

- Search results in <500ms (p95)
- Page load time <2s (p95)
- Lighthouse score >90 (all categories)

Mobile Requirements

- Mobile-first responsive design
- Touch-friendly targets (44x44px minimum)
- Works on iOS and Android
- Optimized for slow networks
- PWA capability (offline support)

Accessibility

- WCAG 2.1 AA compliance
- Keyboard navigation throughout
- Screen reader support
- Proper ARIA labels
- Color contrast compliance
- Focus indicators

Security

- No PHI (Protected Health Information) stored
- SQL injection prevention
- XSS protection
- CSRF protection for state-changing operations
- Rate limiting
- Secure session management
- API authentication and authorization
- Input validation on all forms

Testing

- Unit test coverage (recommended >80%)
 - Integration tests for critical paths
 - E2E tests for main user flows
 - Cross-browser testing (Chrome, Safari, Firefox)
 - Mobile device testing
-

Success Metrics (What "Done" Looks Like)

Technical Success

- All features functional as shown in mockups
- Performance targets met
- Accessibility compliant

- Security best practices implemented
- Comprehensive test coverage
- Production-ready deployment

User Experience Success

- Clinician can search → find providers → view availability → contact → log → review
- Provider can register → create profile → update beds (dashboard + magic link) → view analytics
- Admin can verify providers → manage users → view analytics → moderate content

Data Integrity

- Geospatial search accurate within 1 mile
 - Bed availability updates reflected immediately
 - Last updated timestamps accurate
 - No data loss on updates
 - Audit logs for sensitive operations
-

Current State

What's Already Built

Partial implementation exists:

- Monorepo structure initialized (Turborepo + pnpm)
- Prisma database schema defined (packages/database/prisma/schema.prisma)
- API modules partially implemented (apps/api/src/modules/)
- Frontend pages created (apps/web/src/app/)
- Shared packages set up (packages/types/ , packages/validation/)
- Docker setup for local development

What's functional:

- Basic search and display
- Provider profile viewing
- Clinician dashboard
- Contact logging (with some bugs)
- Saved facilities (with some bugs)

What needs work:

- Complete all features to match mockups
 - Fix existing bugs
 - Implement magic link system (new feature)
 - Complete admin portal
 - Implement subscription/billing
 - Performance optimization
 - Testing infrastructure
 - Production deployment setup
-

Suggested Tech Stack

Note: These are suggestions based on the existing codebase. Open to discussion.

Layer	Current/Suggested	Alternative Options
Frontend	Next.js 14+ (App Router)	Remix, SvelteKit
UI	Tailwind CSS + shadcn/ui	Material-UI, Chakra
Backend	Node.js 20+ + Express	NestJS, Fastify
Database	PostgreSQL 15+ with PostGIS	MySQL (but PostGIS needed)
ORM	Prisma	TypeORM, Drizzle
Cache	Redis	Memcached
Search	Algolia	Elasticsearch, MeiliSearch
Auth	Clerk	Auth0, NextAuth
Maps	Mapbox GL JS	Google Maps
Payments	Stripe	Paddle, Lemon Squeezy
Email	Resend or SendGrid	Postmark, Mailgun
Hosting	Vercel (frontend) + Railway (API/DB)	AWS, DigitalOcean, Render
Monitoring	Sentry	Datadog, LogRocket

Business Context

Revenue Model

Clinician Subscriptions:

- Individual: \$5-10/month
- Organization: \$199-499/month

Provider Subscriptions:

- Launch Special: Free (6 months to seed marketplace)
- Basic: \$199-299/month
- Professional: \$499-699/month
- Enterprise: \$999+/month

Target Metrics (Year 1)

- 100+ active providers in launch state
- 500+ monthly active clinicians
- 80% daily bed update compliance
- \$50K MRR by Month 12

Future Features & Architecture Considerations

Important: While not part of MVP, the system should be architected with these future features in mind:

Phase 2 Features (Post-MVP)

1. EHR Integration for Bed Management

Critical future requirement:

- API connections to common EHR systems (Epic, Cerner, Allscripts, etc.)
- Automated bed availability sync (real-time or scheduled)
- Two-way data flow: EHR → Platform and Platform → EHR
- Standard HL7/FHIR protocols
- OAuth authentication for EHR systems

Architecture implications:

- Design data model to be EHR-compatible
- Build abstraction layer for bed updates (manual, magic link, API, EHR)
- Event-driven architecture for bed availability changes
- Webhook support for real-time updates
- API versioning from day 1

2. Advanced Notification System

- Clinicians subscribe to alerts when matching beds available
- Real-time notifications (email, SMS, push)
- Saved search alerts
- Customizable notification preferences

Architecture implications:

- Notification service as separate module
- Queue system for async notifications
- Template-based notification system

3. Referral Tracking & Analytics

- Track referral outcomes (admitted, declined, waitlisted)
- Provider conversion metrics
- Clinician placement success rates
- End-to-end referral analytics

Architecture implications:

- Event tracking throughout user journey
- Analytics data warehouse considerations
- HIPAA-compliant data storage (if tracking referrals)

4. Multi-Facility Management

- Enterprise providers managing multiple locations
- Centralized bed updates across facilities
- Roll-up analytics and reporting
- Bulk operations

Architecture implications:

- Hierarchical organization structure
- Parent-child facility relationships
- Permission inheritance

5. API for Partners

- Public API for health systems
- Partner integrations
- White-label capabilities
- Embedded search widgets

Architecture implications:

- API-first design
- Rate limiting per partner
- API key management
- Comprehensive API documentation

Architectural Principles (Critical)

1. Modularity

- Each feature as self-contained module
- Clear boundaries between domains
- Easy to add, remove, or replace features
- Minimal coupling, maximum cohesion

2. Flexibility

- Configurable per provider/organization
- Feature flags for gradual rollouts
- Multi-tenant architecture
- Extensible data models

3. Event-Driven

- Core events: bed_updated, provider_verified, review_submitted
- Event bus for module communication
- Webhook support for external systems
- Audit trail via events

4. API-First

- All functionality accessible via API
- Version all APIs from start (/v1/)
- RESTful design
- Comprehensive OpenAPI documentation
- Future GraphQL consideration

5. Scalability

- Stateless services (horizontal scaling)
- Database connection pooling
- Caching strategy (Redis)
- CDN for assets
- Background job processing (queues)

6. Integration-Ready

- Abstraction layers for external services
- Standard protocols (HL7, FHIR, OAuth)
- Webhook delivery system

- Retry logic and error handling
- Rate limiting and throttling

Data Model Extensibility

Design for:

- Custom fields per organization
- Additional bed types/levels of care
- Flexible metadata storage (JSONB)
- Audit logs for all changes
- Soft deletes (never hard delete)

Technical Debt Prevention

From day 1:

- Comprehensive logging
 - Performance monitoring
 - Error tracking
 - Automated testing
 - Code documentation
 - Database migrations tracked
 - Feature flags for risky changes
-

Questions for Dev Team

1. **Timeline:** What's realistic timeline to complete MVP?
 2. **Team Size:** How many engineers do you recommend?
 3. **Tech Stack:** Any concerns with suggested stack? Preferences?
 4. **Existing Code:** Continue with current implementation or start fresh?
 5. **Scope:** Any features that should be deferred to Phase 2?
 6. **Estimate:** What's the cost estimate (time & materials)?
 7. **Approach:** How would you structure the development phases?
 8. **Hosting:** Preference for infrastructure/hosting?
 9. **Testing:** What's your testing strategy?
 10. **Handoff:** What do you need for maintenance post-launch?
 11. **Architecture:** How will you ensure modularity and flexibility for future features (especially EHR integrations)?
 12. **Scaling:** What's your approach to ensure the system can handle multi-state expansion and API partners?
-

What We're Looking For

A development team that can:

- Take full ownership of technical architecture and implementation
- Provide realistic timeline and cost estimates
- Build production-ready, scalable web application
- Deliver clean, tested, maintainable code
- Work from design mockups to pixel-perfect implementation
- Make smart technical decisions independently
- Communicate progress and blockers clearly

What we'll provide:

- Complete design mockups (18 pages)
 - Feature specifications (this document)
 - Product decisions and prioritization
 - Business domain expertise
 - User feedback and testing coordination
 - Provider/clinician onboarding (post-launch)
-

Next Steps

1. **Review this document and mockups** (`design-mockups/index.html`)
 2. **Ask clarifying questions** about scope or features
 3. **Propose your approach:** timeline, team structure, tech decisions
 4. **Provide estimate:** time and cost
 5. **Discuss contract structure:** fixed-price, time & materials, milestones
-

Files Reference

Design Mockups:

- `design-mockups/index.html` - Start here, navigation to all mockups

Documentation:

- `BehavioralHealth_Connect_Project_Plan.md` - Full PRD (business context)
- `ARCHITECTURE.md` - Current architecture details
- `packages/database/prisma/schema.prisma` - Database schema

Current Codebase:

- `apps/web/` - Next.js frontend
 - `apps/api/` - Express backend
 - `packages/` - Shared packages
-

Let's discuss your approach to building this.