Otantist
MVP Technical Specification

Version 2.0 — Updated February 2026

Status: Ready for Development

Scope: Minimum Safe and Ethical Viable Product

Platform	Web Application (MVP) — Mobile in Phase 2
Languages	Fully Bilingual (French & English)
Target Users	Autistic Individuals (14+) & Parents/Guardians
Launch Strategy	Invite-Only Beta

Table of Contents

1. MVP Overview

What We're Building

A bilingual (FR/EN) web application for 1:1 messaging between autistic individuals, with:

- Invite-only access (controlled beta)
- Parent-managed accounts for minors
- Communication preference profiles
- Time boundary enforcement
- Calm Mode for sensory breaks
- Human-moderated safety with AI-assisted flagging

What We're NOT Building (Yet)

Feature	Phase
Mobile apps (iOS/Android)	Phase 2
User discovery/matching	Phase 2
Group messaging	Phase 2
AI message suggestions	Phase 2
Voice messages	Phase 2
Payment processing	Phase 2C
Professional integrations	Phase 2D

2. Technology Stack (Confirmed)

Currently Running

Component	Technology	Port	Status
Backend	NestJS (Node.js 20)	3001	Running
Web Frontend	Next.js 14	3000	Running
Database	PostgreSQL 15	5432	Running
ORM	Prisma	-	Configured
Cache	Redis 7	6379	Running
Email Testing	Mailhog	8025	Running
DB Admin	Prisma Studio	5555	Available

To Implement

Component	Technology	Priority
Real-time	Socket.io	High (messaging)
AI Flagging	Claude API	Medium
File Storage	Cloudflare R2	Low (profile images)
i18n	i18next	High

3. Database Schema (Prisma)

The database uses PostgreSQL with Prisma ORM. Key models are organized into logical groups:

3.1 Authentication & Accounts

Model	Purpose	Key Fields
Account	Authentication credentials	email, passwordHash, accountType, status
User	Profile information	displayName, ageGroup, profileVisibility
ParentManagedAccount	Parent-child relationships	parentAccountId, memberAccountId, relationship
InviteCode	Beta access control	code, maxUses, currentUses, expiresAt

3.2 Preferences

Model	Purpose
CommunicationPreferences	Tone, modes, rhythm preferences
SensoryPreferences	Animations, colors, sounds, notifications
ConversationStarters	"How to Talk to Me" - good/avoid topics, tips
TimeBoundary	Available hours per day of week

3.3 Messaging

Model	Purpose
Conversation	1:1 conversation between two users
Message	Individual messages with status tracking
BlockedUser	User blocking relationships

3.4 Safety & Moderation

Model	Purpose
ModerationQueue	Items flagged for human review
UserReport	User-submitted reports
ParentAlert	Alerts sent to parents of managed accounts

MemberIndicator	Aggregated daily activity indicators

4. API Endpoints

4.1 Authentication

Method	Endpoint	Description
POST	/api/auth/register	Register with invite code
POST	/api/auth/login	Login, returns JWT
POST	/api/auth/logout	Invalidate refresh token
POST	/api/auth/refresh	Refresh access token
POST	/api/auth/verify-email	Verify email with token
POST	/api/auth/accept-terms	Accept ToS/legal declarations

4.2 Users & Preferences

Method	Endpoint	Description
GET	/api/users/me	Get current user profile
PATCH	/api/users/me	Update profile
GET	/api/users/me/onboarding-status	Get onboarding progress
PATCH	/api/users/me/language	Change language (FR/EN)
GET	/api/users/:id/how-to-talk-to-me	Get user communication guide
GET/PATCH	/api/preferences/communication	Communication preferences
GET/PATCH	/api/preferences/sensory	Sensory preferences
GET/PATCH	/api/preferences/time-boundaries	Time boundaries

4.3 Messaging

Method	Endpoint	Description
GET	/api/conversations	List conversations
POST	/api/conversations	Start new conversation
GET	/api/conversations/:id	Get conversation details
GET	/api/conversations/:id/messages	Get messages (paginated)

POST	/api/conversations/:id/messages	Send message
DELETE	/api/messages/:id	Soft delete message

4.4 Parent Dashboard

Method	Endpoint	Description
GET	/api/parent/members	List managed members
GET	/api/parent/members/:id/indicators	Get member indicators
GET	/api/parent/members/:id/alerts	Get alerts
PATCH	/api/parent/members/:id/alerts/:alertId/acknowledge	Acknowledge alert

5. Real-Time Events (Socket.io)

Client to Server Events

Event	Payload	Purpose
join	{ userId }	Join personal room on connect
typing	{ conversationId }	Send typing indicator
markRead	{ conversationId, messageId }	Mark messages as read
presence	{ status }	Update online/away status

Server to Client Events

Event	Payload	Purpose
newMessage	{ message, conversation }	New message received
messageStatus	{ messageId, status }	Message delivery/read update
userTyping	{ conversationId, userId }	Typing indicator from other user
presenceChange	{ userId, status, lastSeen }	User presence update
calmModeAlert	{ userId, activated }	Calm mode alert (parent dashboard)

6. Core Feature Specifications

6.1 Invite Code System

Registration requires a valid invite code. Codes are generated by admins or existing users (if permitted). Each code has a maximum number of uses and optional expiration date. Used codes are tracked via a counter.

6.2 Modular Onboarding

Users can complete onboarding in sections, saving progress and returning later:

Step	Required?	Fields
1. Account	Yes	email, password, inviteCode
2. Email Verify	Yes	verification token
3. Legal Accept	Yes	ToS agreement
4. Basic Profile	Yes	displayName, ageGroup
5. Communication	No	tone, modes, rhythm
6. Sensory	No	animations, colors, sounds
7. "How to Talk to Me"	No	topics, tips

6.3 Time Boundary Enforcement

Messages respect recipient's availability. When a message is sent outside the recipient's available hours, it is queued and delivered at the next available time. The sender sees a message indicating when delivery will occur.

6.4 Calm Mode

Users can activate Calm Mode manually or via an "I need a break" button. Effects include: reduced color palette, no animations, simplified UI, queued incoming messages, and suppressed notifications. For parent-managed accounts, an alert is generated. Deactivation is manual only (no auto-timeout by default).

6.5 AI Content Flagging

Background analysis only — AI never blocks messages. Messages are analyzed asynchronously. If concerning content is detected with high confidence, it is added to the moderation queue for human review. The AI accounts for autistic communication patterns and errs on the side of not flagging unclear cases.

7. Bilingual Implementation

All user-facing content must support French and English. Translation files are organized by feature:

- /locales/fr/ - French translations
- /locales/en/ - English translations
- Files: common.json, auth.json, onboarding.json, messaging.json, settings.json, errors.json

Database Content Strategy:

- System messages: Use translation keys
- User content: Store as-is in original language
- Parent alerts: Store both message_fr and message_en

8. Security Checklist

Authentication

✓ Password hashing (bcrypt, cost 12)
✓ JWT with short expiry (15min access, 7d refresh)
■ Rate limiting on auth endpoints
■ Email verification required

Data Protection

■ All API endpoints require authentication
■ Users can only access their own data
■ Parent access limited to managed accounts
■ Soft delete for messages
■ No sensitive data in logs

Privacy (PIPEDA + Quebec Law 25)

■ Minimal data collection
■ No document/ID uploads
■ Clear consent flows
■ Data export capability
■ Account deletion with data removal

Minor Protection

■ Parent-managed accounts
■ Human moderation
■ No direct adult-minor contact (unrelated)
■ Parent alerts for concerning patterns

9. MVP Feature Checklist

Must Have for MVP Launch

■ Invite code validation
■ Account creation (adult + parent-managed)
■ Email verification flow
■ Legal acceptance (ToS)
■ Bilingual UI (FR/EN)
■ Modular onboarding with save-and-return
■ Basic profile (displayName, ageGroup)
■ Communication preferences
■ "How to Talk to Me" panel
■ Time boundary configuration & enforcement
■ 1:1 messaging (text, emoji)
■ Message queuing for boundaries
■ Calm Mode activation/deactivation
■ Social energy selector
■ User blocking & reporting
■ AI-assisted content flagging
■ Human moderation dashboard
■ Parent dashboard (indicators & alerts)

Explicitly NOT in MVP

✗ User discovery/matching
✗ Group messaging
✗ AI message suggestions
✗ Voice messages
✗ Pictograms
✗ Payment processing
✗ Mobile apps
✗ Professional integrations
✗ Continuity Protocol

10. Development Priorities

Sprint	Weeks	Focus Areas
Sprint 1-2	1-3	Foundation: Prisma schema, Auth module, Invite codes, Email verification, i18n setup
Sprint 3-4	4-6	User System: Profile CRUD, Communication/Sensory preferences, Time boundaries, Onboardin
Sprint 5-6	7-9	Messaging: Conversation model, Messages, Time boundary enforcement, Socket.io, Typing/Rea
Sprint 7-8	10-12	Safety: User state, Calm Mode, Blocking, Reporting, AI flagging, Moderation queue
Sprint 9-10	13-14	Parent Features: Parent-managed accounts, Indicators, Alerts, Dashboard UI
Sprint 11-12	15-16	Polish: Testing, Bug fixes, Performance optimization, Beta preparation

---

Document Version: 2.0 | Updated: February 2026 | Status: Ready for Development
Prepared for stakeholder review