

CONFIDENTIAL



PROJECT OVERVIEW

This guide is for developers building VOX. It contains week by week sprint details, system architecture, database schema, API specifications, and accessibility requirements. The five week timeline is aggressive but achievable with focused engineering.

SYSTEM ARCHITECTURE

Frontend Layer (React Native with Expo)

Cross platform iOS and Android using React Native and Expo. Screen reader optimization using AccessibilityInfo API. Offline messaging queue for reliability. Real time updates via Firebase listeners.

Backend Layer (Node.js and Express)

RESTful API endpoints. JWT based authentication. Real time features via Firebase. Automated KYC workflow. Rate limiting and DDoS protection. Error logging and monitoring.

Data Layer (PostgreSQL and Firebase)

PostgreSQL for persistent data: users, profiles, groups, events, messages. Firebase Realtime Database for real time messaging and presence. AWS S3 for file storage. Redis for caching and session management.

DATABASE SCHEMA

Users Table user_id (UUID, primary key), phone_number (unique), password_hash, created_at, verified (boolean), verification_date
Profiles Table profile_id (UUID, primary key), user_id (FK), bio, interests (JSON array), location, looking_for (dating, friendship, hobby), voice_bio_url, created_at, updated_at
Groups Table group_id (UUID, primary key), name, description, creator_id (FK), category, member_count, created_at
Friendships Table friendship_id (UUID, primary key), user_a_id (FK), user_b_id (FK), status (pending, accepted, blocked), created_at
Events Table event_id (UUID, primary key), group_id (FK), creator_id (FK), title, description, date_time, location, attendee_count, created_at
Messages Table (Firebase Realtime) message_id, sender_id, recipient_id, content, created_at, read_at
KYC Verification Table verification_id (UUID, primary key), user_id (FK), method, document_type, status (pending, approved, rejected), created_at, reviewed_at

FIVE WEEK DEVELOPMENT SPRINT

WEEK ONE: FOUNDATION

Days 1 and 2: Project Setup Create React Native Expo project with TypeScript. Initialize Node.js backend with Express. Set up PostgreSQL database locally and on AWS RDS. Configure Firebase project. Set up AWS Rekognition for document processing. Days 3 and 4: Authentication System Implement JWT token generation and validation. Create user registration endpoint. Phone based signup workflow. Password hashing with bcrypt. Session management with Redis. Database migration scripts. Days 5: User Profile Schema Create users and profiles tables. Build profile creation and update endpoints. Implement profile picture upload to S3. Interest and preference storage (JSON fields). Location based filtering preparation.

WEEK TWO: KYC AND MESSAGING

Days 1 and 2: KYC Verification Workflow Implement document upload and AWS Rekognition processing. Create manual review dashboard for moderators. Video call scheduling system. Build trusted referral workflow. Status tracking (pending, approved, rejected). Days 3 and 4: Messaging System Set up Firebase Realtime Database structure. Implement text messaging endpoints. Build message encryption (TLS in transit). Typing indicators. Read receipts. Offline message queue (local storage, then sync). Day 5: Voice Call Preparation Integrate Twilio or similar for voice call infrastructure. Build call initiation and acceptance flow. Call recording (with consent). Call history tracking.

WEEK THREE: GROUPS AND DISCOVERY

Days 1 and 2: Groups System Create groups table and endpoints. Group creation and management. Member management. Group messaging (multiple recipient handling). Admin and moderation features. Days 3 and 4: Friendship Discovery Algorithm Build profile recommendation algorithm based on interests and location. Implement like and match system. Build match notification flow. Design privacy controls (anonymous browsing until mutual like). Day 5: Voice Call Integration Complete voice call endpoints. Add call quality monitoring. Handle missed calls and call logs. Implement call notifications (push and in app).

WEEK FOUR: EVENTS AND POLISH

Days 1 and 2: Events System Create events table and endpoints. Event creation by group or individuals. RSVP tracking and attendee lists. Calendar integration. Event reminders via push notification. Days 3 and 4: Frontend Integration Build all UI components with screen reader support. Implement VoiceOver and TalkBack testing. Connect all backend endpoints. Implement error handling and retry logic. Performance optimization. Day 5: Security and Optimization Rate limiting on all endpoints. Input validation and sanitization. CORS configuration. Database query optimization. CDN setup for file delivery.

WEEK FIVE: TESTING AND LAUNCH

Days 1 and 2: Accessibility Testing Full VoiceOver testing on iOS. Full TalkBack testing on Android. Screen reader label verification. Touch target size validation. Color contrast testing. Audio description verification. All features accessible. Days 3: Integration Testing End to end testing of all flows. User registration through friendship to messaging. Group creation and communication. Event creation and RSVP. KYC verification workflows. Days 4: Beta and Feedback Internal testing with team members. Feedback collection and bug fixing. Documentation finalization. API documentation completion. Deployment to staging environment. Day 5: Production Deployment Final security audit. Database backups. Monitoring and alerting setup. Performance baseline establishment. Launch and support.

CORE API ENDPOINTS

Authentication POST /auth/register - User registration with phone and password POST /auth/login - Login and JWT token generation POST /auth/refresh - Refresh JWT token POST /auth/logout - Session termination Profiles GET /profile/:userId - Retrieve user profile PUT /profile/:userId - Update profile GET /profiles/discover - Get profile suggestions (discovery algorithm) POST /profile/:userId/like - Like a profile GET /matches - Get mutual matches Messaging POST /messages - Send message to user GET /messages/:conversationId - Get conversation history GET /conversations - Get user conversations POST /messages/:messageId/read - Mark message as read POST /call/initiate - Initiate voice call GET /call/history - Call history Groups POST /groups - Create group GET /groups - List groups GET /groups/:groupId - Get group details POST /groups/:groupId/join - Join group POST /groups/:groupId/message - Send group message GET /groups/:groupId/messages - Get group messages Events POST /events - Create event GET /events - List events GET /events/:eventId - Event details POST /events/:eventId/rsvp - RSVP to event GET /events/:eventId/attendees - Event attendees KYC POST /kyc/initiate - Start verification process POST /kyc/upload-document - Upload ID document POST /kyc/schedule-call - Schedule video verification GET /kyc/status - Check verification status

ACCESSIBILITY CHECKLIST (WCAG 2.2 AA)

Screen Reader Compatibility All text buttons have accessible labels using accessibilityLabel All images have meaningful alt text via accessibilityLabel Form inputs have associated labels Groups and regions properly announced Dynamic content updates announced via AccessibilityInfo.announceForAccessibility Touch Targets All interactive elements minimum 48x48 dp Proper spacing between touch targets (minimum 8dp gap) No small or hard to hit buttons Color and Contrast Text contrast ratio minimum 4.5:1 for normal text Text contrast ratio minimum 3:1 for large text No information conveyed by color alone Icons have sufficient contrast Audio and Video Voice calls have real time captions (when possible) Voice messages can be transcribed Event audio descriptions available No audio autoplays Focus Management Clear focus indicators visible Focus order logical and predictable Modal dialogs trap focus properly Focus returns to trigger element after closing Voice Input All major functions accessible via voice commands No time based voice input requirements Navigation accessible without voice Voice input has confirmation step

CRITICAL DEVELOPER NOTES

Accessibility First: Every feature must be accessible to screen reader users. Test with VoiceOver and TalkBack before closing any task.

KYC Importance: User verification is the foundation of community trust. Implement it robustly. Manual review is essential. Never rush verification.

Data Privacy: All messages are between users only. No server side storage of message content. Use encryption. Comply with GDPR.

Offline Support: Users are often in situations with poor connectivity. All features should work offline and sync when back online.

Performance: Slow apps are inaccessible. Optimize relentlessly. Test on low end devices. Measure every change.

Testing Mindset: Test with actual screen readers. Do not rely on automated tools alone. Real blind users will find issues you miss.

Rollback Plan: Every deployment needs a rollback plan. Have database backups. Have feature flags for gradual rollout.