Active Context: LoveNotes

Current Work Focus

Phase 1: Setup & Infrastructure ☑ COMPLETED

Phase 2: Core Systems - IN PROGRESS

Recent Changes (Phase 1 Completion)

Completed Tasks

1. ☑ Memory bank structure created with all core files
2. ☑ Expo project initialized with TypeScript template
3. ☑ Firebase project structure configured:
   - Firestore security rules (comprehensive, encryption-aware)
   - Storage security rules (2MB limit, encrypted files only)
   - Firestore indexes defined
   - Cloud Functions structure initialized
4. ☑ GitHub repository structure created:
   - .gitignore (excludes Firebase configs and sensitive data)
   - README.md with comprehensive setup instructions
   - GitHub Actions CI/CD workflow (test, build Android, deploy Firebase rules)
   - Issue templates (bug report, feature request, security issue)
5. ☑ Development environment configured:
   - TypeScript configuration
   - ESLint setup
   - Jest testing configuration
   - Project directory structure (app, components, services, contexts, types, utils, assets)
   - Dependencies defined in package.json (Firebase, Expo, encryption libraries)

Documentation Created

- `docs/firebase-setup.md`: Complete Firebase setup guide
- `docs/encryption-guide.md`: Detailed encryption implementation guide
- `docs/environment-setup.md`: Environment variable configuration guide

Recent Changes (Phase 2 Progress)

Completed Tasks

1. ☑ **TypeScript Types**: Created comprehensive type definitions (User, Card, Pair, etc.)
2. ☑ **Firebase Initialization**: Set up Firebase service with environment variables
3. ☑ **Encryption Service**: Implemented TweetNaCl.js encryption service with:
   - Keypair generation and secure storage
   - Shared secret derivation via ECDH

   - Text encryption/decryption
   - Voice file encryption/decryption
   - Invite code generation
4. ☑ **Authentication Service**: Created auth service with:
   - User registration with keypair generation
   - Login/logout functionality
   - Password reset
   - User data caching
5. ☑ **AuthContext**: React Context for global auth state management
6. ☑ **Authentication Screens**: Login and Register screens with form validation
7. ☑ **Navigation Setup**: React Navigation integrated with AuthProvider

Next Steps (Phase 2: Core Systems - Remaining)

1. **Partner Connection System**:

   - Invite code generation and sharing
   - QR code generation/scanning
   - Partner pairing logic
   - Connection state management

2. **Card Data Models & Service**:

   - Firestore service for card CRUD operations
   - Card rotation algorithm implementation
   - Deck management

Active Decisions Made

Encryption Library Choice

- **Decision**: TweetNaCl.js selected
- **Rationale**: Smaller bundle size, simpler API, sufficient for our needs
- **Status**: ☑ Implemented - Full encryption service with key generation, ECDH, text/voice encryption

Firebase Security Rules

- **Status**: ☑ Completed
- **Approach**: Strict rules from the start, encryption-aware
- **Key Features**: Pair-based access control, encrypted content support

CI/CD Strategy

- **Status**: ☑ Workflow template created
- **Features**: Automated testing, Android APK builds, Firebase rule deployment
- **Note**: Requires EXPO_TOKEN and FIREBASE_TOKEN secrets in GitHub

Development Workflow

- **Status**: ☑ Configured
- **Tools**: Expo dev server, TypeScript, ESLint, Jest
- **Structure**: Clear separation of concerns (services, contexts, components)

Current Blockers

None. Ready to proceed with Phase 2.

Notes

- Phase 1 foundation is solid and production-ready
- All sensitive data excluded from repository (.gitignore configured)
- Firebase configuration uses environment variables (.env)
- Security-first approach maintained throughout
- Documentation is comprehensive and ready for team onboarding