

ANNEX 2 - COMPLETE

Application Flow Diagram

Flick MVP - iOS Application

October 8, 2025

1. Overview

This document describes the overall application flow, user journeys, data interactions, and backend architecture for the Flick MVP iOS application, including ALL additional features requested by Good Monkeys LLC. It provides a comprehensive view of how different modules interact with each other and with the backend infrastructure.

2. Enhanced System Architecture

| Layer | Components | Technology |
|----------------------|---|--|
| Presentation Layer | iOS Application (iPhone & iPad) + Web QR Fallback | Firebase React Native + HTML/CSS/JS |
| API Layer | RESTful API Gateway | Node.js + Express.js |
| Business Logic Layer | Authentication, QR Processing, Trading Logic, Notifications | Node.js, Notification Engine, Location Tracking, Ownership History |
| Data Layer | User DB, Lighter DB, Transaction DB, Ownership DB, Ship History DB, Location DB | Supabase/Firebase |
| Storage Layer | Images, Documents, Backups | AWS S3 / Firebase Storage |
| Infrastructure Layer | Cloud Hosting, Load Balancing, CDN for Web QRs | AWS/GCP/Vercel |

3. Key User Journey Flows

3.1 Enhanced QR Code Scanning Flow

| Step | Action | System Response |
|------|---|---|
| 1 | User scans QR code using native iOS camera | Camera app detects QR and attempts to open app |
| 2A | If app installed: Deep link to app | App opens directly to relevant screen |
| 2B | If app not installed: Open web page | Web page loads with notification-style interface |
| 3 | System validates QR code format | Backend checks if QR exists in database |
| 4A | If registered lighter: Show ownership status | Display "This lighter is registered - return or claim?" |
| 4B | If unregistered lighter: Show registration prompt | Display "New lighter - add to collection!" with download link |
| 5 | User takes action (return/claim/register) | System processes action and updates database |
| 6 | Success confirmation displayed | User receives confirmation and next steps |

3.2 User Registration & Onboarding Flow

| Step | Action | System Response |
|------|---|---|
| 1 | User downloads app from App Store | App launches with splash screen |
| 2 | User sees welcome screen | Display registration options (Email/Google OAuth) |
| 3 | User selects registration method | Navigate to respective auth flow |
| 4 | User provides credentials | Backend validates and creates account |
| 5 | Email verification sent (if email signup) | User verifies email via link |
| 6 | User completes profile (username, avatar, privacy settings) | Details stored in User DB |
| 7 | Onboarding tutorial displayed | User learns key features including QR scanning |
| 8 | User lands on home screen | Display personal collection (empty initially) |

3.3 Lost & Found Workflow

| Step | Action | System Response |
|------|------------------------------|---|
| 1 | User reports lighter as lost | Opens "Report Lost" form with lighter details |

| | | |
|---|---|--|
| 2 | User provides description & last known location | Data saved to Lost & Found DB |
| 3 | Lost lighter appears in community feed | Other users can see lost item listing |
| 4 | Another user finds and scans the lighter | System detects lighter is marked as lost |
| 5 | Finder sees "This lighter is lost" notification | Option to contact owner displayed |
| 6 | Finder initiates contact | In-app message sent to original owner |
| 7 | Owner and finder coordinate return | Chat system facilitates communication |
| 8 | Owner confirms return | Lighter marked as recovered, finder awarded points |
| 9 | Ownership history updated | Previous owner chain preserved |

3.4 Trading Workflow

| Step | Action | System Response |
|------|-------------------------------------|--|
| 1 | User browses marketplace | Display available lighters for trade |
| 2 | User selects desired lighter | Show lighter details, owner profile, and ownership history |
| 3 | User sends trade request | Select lighter from own collection to offer |
| 4 | Trade request sent | Owner receives push notification |
| 5 | Owner reviews trade offer | Can accept, reject, or counter-offer |
| 6 | If accepted: trade initiated | Backend updates ownership records |
| 7 | Both users confirm trade completion | Ownership transferred in database |
| 8 | Trade recorded in history | Both users awarded trade points |
| 9 | Ownership history chain updated | New owner can view previous ownership chain |

3.5 Smart Location Update Flow

| Step | Action | System Response |
|------|--------------------------------|--|
| 1 | System sends push notification | "Is your lighter with you right now?" |
| 2 | User responds Yes/No | One-tap response captured |
| 3A | If Yes: Location updated | Current GPS location saved to lighter record |
| 3B | If No: Follow-up question | "Is your lighter lost?" |
| 4A | If lost: Mark as lost | Lighter status updated, owner notified |
| 4B | If not lost: Normal status | "Just not with me right now" recorded |
| 5 | System logs interaction | Analytics updated, frequency tracking |

3.6 Ownership History Viewing Flow

| Step | Action | System Response |
|------|----------------------------|--|
| 1 | User views lighter details | Display current lighter information |
| 2 | User taps "View History" | Request ownership history from backend |

| | | |
|---|---------------------------------------|--------------------------------------|
| 3 | System retrieves ownership chain | Query database for previous owners |
| 4 | Filter by privacy settings | Only show users with public profiles |
| 5 | Display ownership timeline | Show "X → A → B → You" format |
| 6 | User can view previous owner profiles | Display public profile information |

4. Enhanced API Endpoints

| Endpoint | Method | Purpose |
|-------------------------------------|--------|---|
| /api/auth/register | POST | User registration |
| /api/auth/login | POST | User login |
| /api/auth/oauth/google | POST | Google OAuth login |
| /api/lighters/register | POST | Register new lighter |
| /api/lighters/my-collection | GET | Fetch user's lighter collection |
| /api/lighters/marketplace | GET | Browse available lighters for trade |
| /api/lighters/qr/validate | POST | Validate QR code and return status |
| /api/lighters/qr/web/action | POST | Handle QR web actions (return/claim/register) |
| /api/trades/request | POST | Send trade request |
| /api/trades/respond | PUT | Accept/reject trade |
| /api/lost-found/report | POST | Report lighter as lost |
| /api/lost-found/list | GET | Get list of lost lighters |
| /api/messages/send | POST | Send message |
| /api/messages/conversation/:userId | GET | Fetch conversation history |
| /api/users/profile/:userId | GET | Get user profile |
| /api/users/leaderboard | GET | Fetch leaderboard |
| /api/notifications/send | POST | Send push notification |
| /api/lighters/:id/ownership-history | GET | Get ownership history for lighter |
| /api/location/update | POST | Update lighter location |
| /api/location/request | POST | Request location update from user |
| /api/qr/web/register | POST | Handle web-based QR registration |

5. Future Expansion Readiness

5.1 Social Feed Architecture Foundation

- Event Logging: All user actions (trades, scans, location updates) logged for future social features
- Activity Streams: Database structure ready for activity feed generation
- User Interaction Tracking: All social interactions captured for future analysis
- Scalable Infrastructure: Built to handle increased social activity and engagement

5.2 Data Structure for Future Features

- Modular API Design: Easy to add new endpoints for social features
- Flexible Database Schema: Ready for additional social-related tables
- Event-Driven Architecture: Foundation for real-time social updates
- Analytics Foundation: Comprehensive tracking for future business intelligence

This document forms Annex 2 of the Application Development Agreement between Good Monkeys LLC and CodeFlow Studios, dated October 8, 2025.