Flick! Application Development Agreement

Good Monkeys LLC & Codeflow Studios

This Application Development Agreement is made between Good Monkeys LLC and CodeFlow Studios. The parties agree to cooperate for the purpose of developing the Flick MVP mobile application.

1. Project Overview

The Developer agrees to design, develop, and deliver a complete Minimum Viable Product (MVP) application titled “**Flick MVP Development”**. This project will include mobile app development, backend implementation, cloud deployment, and a free single-page landing website for goodmonkeys.com. The purpose of the MVP is to validate the Flick concept: a social, QR-enabled lighter tracking platform designed for community engagement and brand interaction.

2. Scope of the Work

The MVP will include the following core functionalities and modules, derived from the Flick MVP

Proposal and integrated with CodeFlow Studios' development structure:

a. QR scanning & registration of lighters
b. User Profiles and Onboarding
c. Lost & Found Workflow
d. Trading and Gifting Features
e. Basic Gamification and Rewards (Lite Version)
f. Admin Dashboard with basic analytics
g. Backend APIs and Cloud Integration (AWS / Vercel / GCP)
h. Mobile App (iOS & Android) using Flutter or React Native
i. Database integration using Supabase or Firebase
j. OAuth Authentication (Google, Email, or Social)
h. Additionally, Developer agrees to design and deliver a single-page landing website for goodmonkeys.com free of charge.

A detailed description of each module, including all functionalities, components, and user interactions, shall be documented in ***Annex 1 – Detailed Scope of Work****, which forms an integral part of this Agreement.*

3. Development Timeline

The total project duration will be **twelve (12) weeks,** starting from the Effective Date. -

Weeks 1–2:

Planning & Design

- *Requirements analysis*
- *UI/UX wireframing*
- *Technical architecture planning*
- *Project setup and environment configuration*

Weeks 3–8:

Core Development

- *Frontend development*
- *Backend API development*
- *Database implementation*
- *Integration testing*

Weeks 9–10:

Testing & Optimization

- *Comprehensive testing*
- *Performance optimization*
- *Security audit*
- *Bug fixes and refinements*

Weeks 11–12:

Deployment & Launch

- *Production deployment*
- *Final testing*
- *Documentation completion*
- *Launch support*

*The Developer shall also prepare and deliver a comprehensive **Application Flow Diagram**, describing the overall user journey, data interactions, and backend structure.*

*This document shall be attached as **Annex 2 – Application Flow Diagram** once finalized, and shall form an integral part of this Agreement.*

***Weekly progress meetings shall be held at least once per week** between the Developer and the Client to review milestones, deliverables, and pending tasks.*

Additional meetings may be scheduled as needed for clarifications, design reviews, or technical discussions.

4. Payment Terms

The total cost for the project is **fixed at USD 13,500**, payable as follows:

- 40% **(USD 5,400)** upon project kickoff
- 40% **(USD 5,400)** upon completion of the core development phase
- 20% **(USD 2,700)** upon final delivery and approval Payments are due within thirty (30) days of each invoice. **Full ownership and source code rights are transferred to the Client upon receipt of the final payment.**

*The **first payment** shall be made **between November 1–7, 2025**, marking the official project kickoff date.*

5. Revisions and Support

The Agreement includes up to three **(3) major revision** rounds during the development phase. Any additional revisions or scope extensions shall be quoted separately.

The Developer will also provide **three (3) months** of post-launch support, which covers bug fixes, security patches, and minor optimizations. Feature additions or redesign requests beyond the original MVP scope will require a new agreement or amendment.

6. Ownership and Intellectual Property

All intellectual property rights, including but not limited to source code, databases, user interface designs, and documentation, shall be **transferred to Good Monkeys LLC** upon full payment. The Developer retains the right to showcase non-confidential aspects of the project in its portfolio or marketing materials with prior written consent from the Client.

7. Confidentiality

Both parties agree to keep all shared data, materials, and business information strictly confidential. Confidential information shall not be disclosed or used for any purpose other than the execution of this Agreement without prior written consent from the other party.

8. Termination

Either party may terminate this Agreement by written notice if the other party materially breaches any of its obligations and fails to remedy such breach within fourteen (14) days after written notification. Upon termination, the Client shall compensate the Developer for all completed work up to the termination date.

9. Governing Law and Dispute Resolution

This Agreement shall be governed by the laws of the United Arab Emirates. Any disputes shall be resolved through online or in-person arbitration under the rules of the Dubai International Arbitration Centre (DIAC).

10. Signatures

IN WITNESS WHEREOF, the parties here to have executed this Application Development

11. Annexes (Integral Parts of the Agreement)

- **Annex 1:** Detailed Scope of Work (including full feature breakdown to be provided by the Developer)
  **Referenced at the end of Section 2 (Scope of the Work)*
- **Annex 2:** Application Flow Diagram (to be provided by the Developer)
  ***Referenced at the end of Section 3 (Development Timeline)*

These annexes form an integral and inseparable part of this Agreement.

This Agreement shall not be considered complete, valid, or enforceable until Annex 1 and Annex 2 are prepared, reviewed, and duly signed by both parties.

For and on behalf of **Good Monkeys LLC**
Name: **Deniz Ordulu**
Title: Co-Founder & General Manager
Date: 08.10.2025

____

For and on behalf of **CodeFlow Studios**
Name: ____________________
Title: ____________________
Date: ____________________

____

Saddam

Name: Saddam
Title: Founder
Location: Pakistan
Date: 10.10.2025

ANNEX 1

Detailed Scope of Work

Flick MVP - iOS Application Development

October 8, 2025

1. Introduction

This document provides a comprehensive breakdown of all features, modules, and functionalities to be developed as part of the Flick MVP (Minimum Viable Product) for iOS platform. This annex forms an integral part of the Application Development Agreement between Good Monkeys LLC and CodeFlow Studios.

1.1 Project Scope

Platform: iOS (iPhone and iPad compatible)

Development Framework: Flutter or React Native (cross-platform capable, optimized for iOS)

Backend: Node.js with Express.js framework

Database: Supabase or Firebase (cloud-hosted, scalable)

Cloud Infrastructure: AWS, Vercel, or Google Cloud Platform

Note: Android development is excluded from this MVP scope. Focus is exclusively on iOS platform delivery.

2. Core Modules & Features

2.1 QR Scanning & Lighter Registration

Purpose: Enable users to scan QR codes on lighters and register them in their collection.

Features:

- QR code scanner using device camera (iOS native integration)
- Unique lighter ID generation and validation
- Lighter registration with metadata (brand, color, design, acquisition date)
- Photo capture functionality to document the lighter
- Duplicate detection to prevent multiple registrations of same lighter
- Registration confirmation with success notification
- History log of all scanned/registered lighters

2.2 User Profiles & Onboarding

Purpose: Create personalized user accounts with profile management and seamless onboarding.

Features:

- User registration with email or social OAuth (Google Sign-In)
- Profile creation with username, avatar, bio, and location
- Email verification and password recovery
- Onboarding tutorial (first-time user walkthrough)
- Profile editing and customization
- Privacy settings (public/private profile)
- View personal lighter collection
- Activity feed showing recent scans, trades, and achievements
- Push notification preferences

2.3 Lost & Found Workflow

Purpose: Allow users to report lost lighters and facilitate their recovery through community engagement.

Features:

- Report a lighter as 'Lost' with description and location
- Browse lost lighters in the community
- Mark a lighter as 'Found' when scanned by another user
- Notification system to alert original owner when lighter is found
- In-app messaging to coordinate lighter return
- Optional reward system (user can offer reward for return)

- Lost & Found history tracking
- Map view showing locations of reported lost lighters

2.4 Trading and Gifting Features

Purpose: Enable peer-to-peer lighter exchanges and gifts within the community.

Features:

- Browse available lighters for trade in community marketplace
- Send trade requests to other users
- Accept/reject trade offers with messaging
- Gift lighters to friends or community members
- Transfer ownership through secure transaction
- Trade history and transaction log
- Rating system for successful trades
- Search and filter lighters by brand, rarity, or design
- Wishlist functionality to track desired lighters

2.5 Basic Gamification and Rewards (Lite Version)

Purpose: Encourage user engagement through achievements, points, and leaderboards.

Features:

- Point system for various activities (scans, trades, referrals)
- Achievement badges (e.g., 'First Scan', 'Collector', 'Trader', 'Helper')
- Leaderboard showing top collectors and active users
- Level progression system (Bronze, Silver, Gold, Platinum tiers)
- Daily login rewards
- Referral bonus program
- Milestone celebrations (e.g., 10th lighter scanned)
- Exclusive digital collectibles for top users

2.6 Admin Dashboard

Purpose: Provide administrators with tools to manage the platform, users, and content.

Features:

- Web-based admin panel (accessible via browser)
- User management (view, edit, suspend, delete accounts)
- Lighter database management (view all registered lighters)
- Analytics dashboard (user growth, activity metrics, engagement rates)
- Content moderation tools (review flagged content, messages)
- Push notification broadcast system
- Report management (handle user reports and disputes)
- System configuration and settings
- Export data (CSV/Excel reports for analytics)

2.7 Backend APIs and Cloud Integration

Purpose: Build robust, scalable backend infrastructure to support all app functionalities.

Features:

- RESTful API endpoints for all app features
- User authentication API (OAuth, JWT tokens)
- QR code validation and lighter registration API
- Real-time messaging API for trades and lost & found
- Push notification service integration (Firebase Cloud Messaging)
- Cloud storage for user photos and lighter images
- Database schema design and implementation
- API rate limiting and security measures
- Error handling and logging system
- Automated backup and data recovery system

2.8 Mobile App (iOS)

Purpose: Develop native-quality iOS application with intuitive UI/UX.

Features:

- iOS-optimized user interface (iPhone 12 and above)
- iPad compatibility with responsive design
- Native iOS design patterns and gestures
- Smooth animations and transitions
- Offline mode for viewing personal collection
- Dark mode support
- In-app camera integration for QR scanning
- Location services integration (for lost & found features)
- Push notifications
- App Store deployment and optimization
- TestFlight beta testing setup

2.9 Landing Website for goodmonkeys.com

Purpose: Single-page marketing website (complimentary deliverable).

Features:

- Responsive single-page website design
- Hero section with app overview and download link
- Feature highlights section
- Screenshots gallery (carousel)
- Call-to-action buttons (Download from App Store)

• Contact form integration
• Social media links
• Mobile-optimized layout
• Fast loading and SEO optimization
• Hosting on Vercel or Netlify (included)

3. Technical Specifications

3.1 Technology Stack

Component	Technology
Mobile Framework	Flutter or React Native
Target Platform	iOS (iPhone & iPad)
Backend Framework	Node.js + Express.js
Database	Supabase or Firebase
Authentication	OAuth 2.0 (Google) + Email/Password
Cloud Infrastructure	AWS / GCP / Vercel
Push Notifications	Firebase Cloud Messaging (FCM)
File Storage	AWS S3 or Firebase Storage
Version Control	GitHub

4. Project Deliverables

iOS Application:

- Production-ready app submitted to Apple App Store
- TestFlight build for beta testing
- Complete source code repository

Backend Infrastructure:

- Deployed and configured cloud backend
- API documentation (Swagger/Postman collection)
- Database schema documentation

Admin Dashboard:

- Web-based admin panel (deployed and accessible)
- Admin user manual and documentation

Landing Website:

- Single-page website for goodmonkeys.com
- Deployed and live on production server

Documentation:

- Technical documentation (architecture, API specs)
- User manual and onboarding guide
- Deployment and maintenance guide

5. Exclusions from Scope

- Android application development (excluded from MVP)
- Advanced AI/ML features (e.g., lighter image recognition)
- Payment gateway integration (no monetization in MVP)
- Third-party brand partnerships or integrations
- Advanced analytics and business intelligence tools
- Multi-language support (English only for MVP)
- Video content or streaming features

6. Testing & Quality Assurance

6.1 Testing Approach

- Unit testing for backend APIs and critical functions
- Integration testing for end-to-end workflows
- iOS device testing (iPhone 12, 13, 14 series)
- iPad compatibility testing
- Performance testing (load times, API response times)
- Security testing (authentication, data protection)
- User acceptance testing (UAT) with Good Monkeys LLC team
- Beta testing via TestFlight with selected users

7. Post-Launch Support (3 Months)

- Bug fixes and critical issue resolution
- Security patches and updates
- Performance monitoring and optimization
- Minor UI/UX improvements based on user feedback
- iOS version updates (compatibility with latest iOS releases)
- Cloud infrastructure monitoring and maintenance
- Monthly progress reports and analytics review

This document forms Annex 1 of the Application Development Agreement between Good Monkeys LLC and CodeFlow Studios, dated October 8, 2025.

ANNEX 2

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. It provides a comprehensive view of how different modules interact with each other and with the backend infrastructure.

2. System Architecture

Layer	Components	Technology
Presentation Layer	iOS Application (iPhone & iPad)	Flutter/React Native
API Layer	RESTful API Gateway	Node.js + Express.js
Business Logic Layer	Authentication, QR Processing, Trading Logic, Gamification Engine	Node.js
Data Layer	User DB, Lighter DB, Transaction DB	Supabase/Firebase
Storage Layer	Images, Documents, Backups	AWS S3 / Firebase Storage
Infrastructure Layer	Cloud Hosting, Load Balancing	AWS/GCP/Vercel

3. Key User Journey Flows

3.1 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)	Data stored in User DB
7	Onboarding tutorial displayed	User learns key features
8	User lands on home screen	Display personal collection (empty initially)

3.2 QR Scanning & Lighter Registration Flow

Step	Action	System Response
1	User taps "Scan QR" button	Camera opens with QR scanner overlay
2	User scans QR code on lighter	App validates QR code format
3	QR validated successfully	Backend checks if lighter already registered
4	If new lighter: show registration form	User enters lighter details (brand, color, notes)
5	User takes photo of lighter	Photo uploaded to cloud storage
6	User confirms registration	Backend creates lighter record, links to user
7	Success notification displayed	Lighter added to user collection
8	Points awarded for registration	Gamification system updates user score

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

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 and owner profile
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

4. Data Flow Architecture

4.1 Authentication & Authorization Flow

→ User initiates login (Email/Google OAuth)
→ iOS app sends credentials to API Gateway
→ API validates credentials against User DB
→ If valid: JWT token generated and returned
→ iOS app stores JWT token securely (Keychain)
→ All subsequent API calls include JWT token in header
→ API validates token on each request
→ Token expires after 24 hours, refresh required

4.2 QR Code Validation & Registration Flow

→ User scans QR code via iOS camera
→ App extracts unique lighter ID from QR code
→ App sends lighter ID to backend API for validation
→ Backend checks if ID format is valid
→ Backend queries Lighter DB to check if already registered
→ If new: create new lighter record with user_id linkage
→ Upload lighter photo to cloud storage (S3/Firebase)
→ Update User DB to link lighter to user's collection
→ Return success response with lighter details to app

4.3 Real-Time Messaging Flow (Lost & Found / Trading)

→ User initiates chat (from lost & found or trade screen)
→ App sends message via API to backend
→ Backend stores message in Messages DB
→ Backend sends push notification to recipient via FCM
→ Recipient's app receives push notification
→ Recipient opens chat, app fetches message history from API
→ Real-time updates via WebSocket or polling (every 5 seconds)
→ All messages encrypted in transit (HTTPS)

5. Database Schema Overview

5.1 Core Database Tables

Table Name	Key Fields	Purpose
users	user_id, email, username, avatar_url, created_at, points, level	Stores user account information
lighters	lighter_id, qr_code, owner_id, brand, color, photo_url, status, registered_at	Store registered lighter data
trades	trade_id, requester_id, owner_id, lighter_offered_id, lighter_requested_id, status, created_at	Manage trade requests and history
lost_found	report_id, lighter_id, reporter_id, status, description, last_location, reported_at	Track lost and found lighters
messages	message_id, sender_id, recipient_id, content, timestamp, read_status	In-app messaging system
achievements	achievement_id, user_id, badge_type, earned_at	Store user achievements and badges
notifications	notification_id, user_id, type, content, read_status, created_at	Push notification records

6. Key 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/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

7. Security & Performance Considerations

7.1 Security Measures

- HTTPS encryption for all API communication
- JWT token-based authentication with secure storage
- Password hashing using bcrypt (salt rounds: 10)
- OAuth 2.0 implementation for Google Sign-In
- Input validation and sanitization on all endpoints
- Rate limiting to prevent API abuse (100 requests/minute per user)
- SQL injection prevention through parameterized queries
- XSS protection through content security policies
- Regular security audits and penetration testing

7.2 Performance Optimization

- Database indexing on frequently queried fields (user_id, lighter_id)
- Image compression before upload (max 1MB per image)
- Caching frequently accessed data (user profiles, leaderboard)
- Lazy loading for lighter collections (paginated results)
- CDN for static assets and images
- API response time target: < 200ms for most endpoints
- Background processing for non-critical tasks (analytics, email)
- Optimized database queries with proper joins
- Load balancing for high-traffic scenarios

8. Deployment Architecture

iOS App: Deployed to Apple App Store via Xcode + App Store Connect
Backend API: Hosted on AWS EC2 / Google Cloud Run / Vercel
Database: Supabase (managed PostgreSQL) or Firebase Realtime Database
File Storage: AWS S3 or Firebase Storage for user-uploaded images
Admin Dashboard: Deployed on Vercel or Netlify (static hosting)
Landing Website: Deployed on Vercel/Netlify with custom domain
CI/CD Pipeline: GitHub Actions for automated testing and deployment
Monitoring: AWS CloudWatch / Firebase Analytics for performance tracking
Backup Strategy: Daily automated database backups to secure cloud storage

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