**Developer Brief: Eco Lotto Mobile-First Lottery Platform**

## **1. Project Overview & Goals**

Eco Lotto is a national, mobile-first lottery platform for Zimbabwe that leverages both smartphone and USSD channels. It offers daily (US $0.50) and weekly (US $1.00) lottery draws, transparent blockchain-recorded results, seamless EcoCash payments, and a scalable agent network. The primary goals are:

* **Accessibility:** Reach urban and rural users via app + USSD/SMS
* **Transparency:** Use blockchain or audited RNG for provably fair draws
* **Scalability:** Support millions of users and hundreds of agents nationwide
* **Social Impact:** Drive financial inclusion and inject revenue into community projects

## **2. Project Scope**

| **Component** | **Description** |
| --- | --- |
| **User App** | Android + iOS mobile app for ticket purchase, results, account. |
| **USSD/SMS Interface** | Lightweight menu for feature-phone users to register, buy tickets, view results. |
| **Blockchain Draw Engine** | Smart contract or oracle-driven system recording ticket hashes and draw outcomes immutably. |
| **Admin Dashboard** | Web-based backend for controlling draws, viewing reports, payout approvals, compliance/KYC. |
| **Agent Portal** | Web or mobile-web interface for retail agents to enter ticket sales, track commissions, assist customers. |

## **3. Functional Requirements**

### **3.1 User App (Android & iOS)**

* **Onboarding & KYC:**
  + Sign up with Econet-linked phone number
  + Basic KYC: ID number or SIM-registration check, age ≥ 18
* **Wallet Integration:**
  + EcoCash API for ticket payments and prize payouts
  + Optional fallback: airtime deduction via Zimswitch or Telecash
* **Ticket Purchase:**
  + Daily draw tickets (US $0.50) & Weekly jackpot tickets (US $1.00)
  + Manual number selection or “Quick Pick” auto-generate
* **Draw Viewing & Notifications:**
  + Real-time display of results
  + Push notifications for winning numbers and prize credits
* **Account Dashboard:**
  + View balance, ticket history, winnings, spent amounts
  + Responsible-gaming tools: spending limits, self-exclusion

### **3.2 USSD/SMS Version**

* **USSD Menu Flow:**
  1. Dial code (e.g. \*345#) → Main menu
  2. Register → Confirm KYC via SMS link or OTP
  3. “Buy Daily Ticket,” “Buy Weekly Ticket,” “View Last Draw,” “Balance”
  4. Payment prompt → EcoCash confirmation
* **SMS Fallback:** Shortcodes for key actions if USSD unsupported

### **3.3 Blockchain / RNG System**

* **Ticket Hashing:** At purchase, generate cryptographic hash of ticket data and store on chain or in a Merkle tree.
* **Draw Execution:**
  + Use a certified on-chain RNG oracle (e.g. Chainlink VRF) or audited off-chain RNG logged immutably.
  + Record draw results on chain with timestamp and hash.
* **Audit Interface:** Simple API endpoint or web view for anyone to verify ticket vs. draw hash.

### **3.4 Admin Dashboard**

* **Draw Management:** Schedule draws, trigger manual or automated draws, halt draws if needed.
* **User & Agent Management:** Approve KYC, view user status, freeze/ban accounts.
* **Financials & Reporting:**
  + Dashboard: daily sales, payouts, agent commissions
  + Exportable CSV/Excel reports for accounting and regulator submissions
* **Compliance/KYC Tools:** View uploaded IDs, flag suspicious patterns, integrate with national ID APIs if available.

### **3.5 Agent Portal**

* **Sales Entry:**
  + Agent logs in, enters customer phone & numbers, confirms payment.
  + Generate digital ticket receipt (SMS or print).
* **Commission Tracking:**
  + Real-time display of agent’s accumulated commissions
  + Weekly summary and payout scheduling
* **Support Tools:**
  + Quick-reference FAQ
  + Incident/ticket logging for technical or customer issues

## **4. UX/UI Expectations**

* **Design Principles:**
  + Minimalist, clear layouts with large touch targets
  + Low–data usage: optimized images & bundles, local caching
  + Bilingual support: English and Shona menu labels/text
  + Consistent branding: green/gold palette
* **Responsibility of Developer:**
  + If no designs provided, propose wireframes in Figma or equivalent
  + Ensure accessibility: contrast, font size, offline/poor-network handling

## **5. Integrations & Infrastructure**

| **Integration** | **Purpose** |
| --- | --- |
| **EcoCash API** | Ticket payments & prize payouts |
| **Telecel Telecash** | Secondary mobile-money option |
| **SMS/USSD Gateway** | USSD menu implementation & notifications |
| **Blockchain Oracle** | Verifiable RNG and draw-recording (e.g. Chainlink VRF) |
| **Cloud Hosting** | Scalable servers (AWS/Azure/GCP) & auto-scaling |
| **Database** | PostgreSQL or similar, with daily backups |
| **Push Notifications** | Firebase Cloud Messaging (Android/iOS) |
| **Analytics** | Firebase Analytics / custom dashboards |

## **6. Timeline & Milestones**

| **Phase** | **Duration** | **Key Deliverables** |
| --- | --- | --- |
| **MVP** | Months 0–2 | Core mobile app & USSD flow; basic RNG & blockchain logging; EcoCash integration; internal beta test |
| **Pilot** | Month 3 | Pilot with 100–200 users & 20 agents; bug fixes & performance tuning |
| **Full Launch** | Month 4 | Nationwide rollout; agent network onboarding; marketing support |
| **Feature Rollout** | Months 5–6 | Admin dashboard v1; agent portal; responsible-gaming tools; advanced reporting |
| **Scale & Optimize** | Months 7–12 | Performance optimization; UI refinements; new game modes; compliance audits |

## **7. Deployment & Security**

* **Hosting:**
  + Use a managed Kubernetes or container service for microservices
  + Auto-scale API & draw-engine pods based on load
* **CI/CD Pipeline:**
  + Automated testing (unit, integration) on each push
  + Blue/green or canary deployments to minimize downtime
* **Data Security:**
  + TLS everywhere (API, web, mobile app)
  + Encrypt sensitive data at rest (e.g. KYC documents)
  + Regular penetration testing, third-party security audits
* **Backups & DR:**
  + Nightly DB backups; weekly offsite copies
  + Disaster-recovery run-through every quarter

**Note to Developer:** This brief outlines end-to-end requirements. Adjust tech choices (frameworks, blockchain provider) based on your expertise and cost constraints, but adhere to core goals: **accessibility**, **transparency**, **scalability**, and **security**. Regularly demo progress and align with the LottoVerse leadership team for feedback at each milestone.