**Cahier des charges — Plateforme « Makona Awards »**

**Version** : 1.0  
**Dernière mise à jour**: 05 Septembre 2025  
**Objectif** : Conception d’une plateforme de gestion des candidatures et de vote en ligne pour l’événement culturel annuel « Makona Award » (Guéckédou).

**1) Contexte & vision**

Makona Award est une cérémonie annuelle de distinction des talents locaux (p. ex. musique, danse, arts, sport, initiatives sociales/innovation). L’édition 2025 souhaite se numériser pour élargir la participation du public, renforcer la transparence et moderniser l’image de l’événement.

**Objectif général** : mettre en place une plateforme web de vote fiable, sécurisée et accessible, permettant :

* la gestion des catégories et des candidats ;
* l’inscription/validation des candidatures ;
* le vote du public;
* le suivi en temps réel des résultats et l’édition de rapports.

**Cibles** :

* Public local et diaspora ;
* Candidats (artistes, équipes, projets) ;
* Comité d’organisation ;
* Partenaires/Sponsors.

**2) Périmètre fonctionnel (MVP + extensions)**

**2.1 MVP (phase 1)**

1. **Gestion des catégories** : créer, éditer, archiver (ex. Musique, Danse, Sport…).
2. **Candidats** : soumission par formulaire (nom, bio, photo, médias : lien vidéo/audio), rattachement à une catégorie, statut (brouillon → en revue → validé).
3. **Comptes utilisateurs (votants)** : inscription (téléphone + mot de passe )
4. **Vote public** : 1 vote par catégorie et par utilisateur (ou règle précise : cf. § 5).
5. **Anti‑fraude** : limitation par appareil/compte protection IP/Rate limite.
6. **Affichage des résultats** : compteur en temps réel (option d’occulter le score), classement par catégorie, page « Tendances ».
7. **Admin** : tableau de bord, gestion des comptes, candidats, catégories, période de vote, export CSV/Excel.

**2.2 Extensions (phase 2 et +)**

* **Vote jury** pondéré (pondération configurable par catégorie).
* **Monétisation des votes** (votes « boost » via paiements mobiles) avec quotas/limites et transparence.
* **USSD/SMS** (vote hors ligne) via intégration opérateurs locaux.
* **Application mobile** (PWA).
* **Pages sponsors** (branding, bannières, reporting audience).
* **Multi‑édition / Historique** (archiver chaque année, hall of fame).
* **Multilingue** (français + anglais pour les pays frontaliers ).
* **Streaming & Live** (embed vidéo du gala, sondages live).

**3) Parties prenantes & rôles**

* **Comité d’organisation** (admin): définit les catégories, valide les candidats, paramètre les règles de vote, supervise.
* **Candidats** : soumettent leur profil et leurs médias, animent leur communauté.
* **Votants** : créent un compte et votent selon les règles.
* **Administrateurs techniques** (superuser): gèrent l’hébergement, la sécurité, les sauvegardes.

**4) Parcours utilisateurs (User Journeys)**

**4.1 Votant**

1. Arrive sur le site → 2) Crée un compte (email/téléphone) → 3) Parcourt les catégories → 4) Ouvre la fiche d’un candidat → 5) Vote (si période ouverte et quota disponible) → 6) Reçoit confirmation → 7) Partage sur réseaux sociaux.

**4.2 Candidat**

1. Crée un compte → 2) Soumet candidature (formulaire + médias) → 3) Reçoit validation/retour du comité → 4) Page publique du candidat accessible → 5) Suivi de ses votes en temps réel (si autorisé).

**4.3 Administrateur**

1. Auth → 2) Paramètre l’édition (dates, catégories, règles) → 3) Modère/valide les candidats → 4) Supervise les votes, détecte anomalies → 5) Exporte résultats.

**5) Règles de vote (paramétrables)**

* **Fenêtre de vote** : date/heure d’ouverture et de clôture par catégorie ou globale.
* **Éligibilité** : compte vérifié (OTP).
* **Quota base** : 1 vote par catégorie **par utilisateur**
* **Anti‑double vote** : clé (user\_id + category\_id) unique ; contrôle device/IP ; rate‑limit
* **Transparence** : option « afficher le compteur » (on/off) ; publication d’un rapport signé (hash).
* **Contestations** : fenêtre de réclamation avec preuves, arbitrage comité.

**6) Exigences non fonctionnelles**

* **Scalabilité** : support pic de trafic (campagnes virales) ; horizontalisation possible.
* **Traçabilité** : audit logs des actions critiques (admin, validation, vote).
* **Confidentialité** : minimisation des données, RGPD‑like (consentement, droit d’accès/suppression).
* **Accessibilité** : WCAG AA (contrastes, clavier, alt text).

**7) Architecture technique (Django)**

* **Backend** : Django 4+, Python 3.11+, Django REST Framework (API), Django Admin.
* **DB** : PostgreSQL 14+ (UUID, contraintes uniques, vues matérialisées pour stats).
* **Cache/Queue** : Redis (sessions, rate‑limit, file asynchrone Celery pour jobs OTP, exports, emails).
* **Frontend** : react js et tailwind css
* **Auth** : Django auth + telephone
* **Déploiement** : conteneurisation (Docker), CI/CD (GitHub Actions), traefik + Gunicorn, TLS (Let’s Encrypt).

**Contraintes clés** :

* unique(user\_id, candidate\_id) pour vote unique (ou clé dérivée si mode daily : unique(user\_id, candidate\_id, date\_jour)).
* Intégrité référentielle sur toutes les FK.
* Index sur Vote(candidate\_id), Vote(created\_at), User(phone).

**9) API (extraits — REST)**

* POST /api/auth/register : inscription (email/phone)
* GET /api/editions/:year/categories : lister catégories.
* GET /api/categories/:id/candidates : lister candidats.
* POST /api/votes : {candidate\_id} → enregistre vote si éligible.
* GET /api/candidates/:id/stats : total, tendances (si autorisé).
* GET /api/admin/exports/votes.csv : export sécurisé (staff only).
* POST /api/candidates : soumission candidate (auth requise).
* PATCH /api/admin/candidates/:id : valider/refuser.
* Throttling : POST /api/votes limité (p.ex. 5 req/min/IP + règles supplémentaires).

**10) Sécurité & anti‑fraude**

* **Vérification OTP** sur email/téléphone avant tout vote.
* **Rate‑limit** multi‑niveaux : IP, user\_id, device signature douce.
* **Journaux d’audit** (signés/hachés) et **hash des exports** pour vérifiabilité.
* **Liste de blocage** (IP/plages, agents, emails jetables) et blocage pays si nécessaire.

**Politique de confidentialité** claire, transparence des règles de vote.

**11) Back‑office (Admin)**

* Dashboard : nb. comptes, candidats validés, votes (global/catégorie), anomalies.
* Gestion : éditions, catégories, candidats (statuts), règles de vote, jury, sponsors.
* Outils : export CSV/Excel/PDF, re‑calcul scores (si pondération), verrouillage d’une catégorie.
* Modération : validation médias, édition contenu (bios, libellés).
* Paramètres : ouverture/clôture du vote par catégorie, visibilité des compteurs.

**12) Contenus & UI (MVP)**

* **Accueil** : présentation Makona Award, CTA « Voter » / « Proposer une candidature ».
* **Catégories** : grille filtrable, badges « Ouvert au vote ».
* **Fiche candidat** : photo, bio courte, médias (embed YouTube/Audio), bouton « Voter ».
* **Classements** : top par catégorie (compteur visible si activé).
* **Mon compte** : gestion profil, statut vérification, historique de votes (si utile).
* **Mobile‑first** : PWA (icône, offline minime pour shell, notifications Web Push facultatives).

**13) Intégrations externes**

* **Email** : transactional (SendGrid/Mailgun/SMTP local fiable).
* **SMS/USSD** : via agrégateur régional ou opérateurs locaux (environnement de test + sandbox).
* **Paiement (phase 2)** : intégration Mobile Money (agrégateur) pour votes « boost » / dons, avec reçus.
* **Analytics** : Matomo/GA4 (consentement cookies), pixels sponsor optionnels.

**14) Indicateurs clés (KPIs)**

* Comptes créés / comptes vérifiés.
* Taux de conversion visite → vote.
* Votes totaux par catégorie et par canal (web/SMS/USSD/jury).
* Pics de charge (req/min), latence p95.
* Incidents de sécurité, taux de fraude bloquée.
* Audience par pays/appareil, partages sociaux.

**15) Tests & acceptation**

* **Unitaires** (models, services), **intégration** (API), **E2E** (parcours vote).
* **Tests charge** (k6/Locust) : profils de pic réalistes.
* **Recette fonctionnelle** : cas de test par user story, PV de recette.
* **Cérémonie d’acceptation** : démonstration fin de sprint + checklist critères (cf. § 16).

**16) User stories & critères d’acceptation (exemples)**

* **US‑001** : En tant que votant, je crée un compte et je le vérifie par OTP.  
  **CA** : un OTP invalide empêche l’accès au vote ; un OTP valide active le compte.
* **US‑002** : En tant que votant, je ne peux voter qu’une fois par catégorie.  
  **CA** : tentative supplémentaire → message explicite, pas d’incrément.
* **US‑003** : En tant qu’admin, je définis les dates d’ouverture/fermeture du vote.  
  **CA** : hors fenêtre → bouton vote désactivé + message.

**17) Planning indicatif (MVP ~6–8 semaines)**

1. **Semaine 1** : cadrage détaillé, maquettes fil de fer, architecture.
2. **Semaines 2–3** : modèles Django, Admin, API auth/OTP, catégories/candidats.
3. **Semaines 4–5** : logique de vote, anti‑fraude, pages publiques, analytics.
4. **Semaine 6** : tests charge/sécu, recettes, contenus, formation.
5. **Buffer** : 1–2 semaines pour itérations/USSD‑SMS pilote si priorisé.

**18) Équipe & responsabilités**

* **Chef de projet** : planning, parties prenantes, reporting.
* **Tech lead Django** : architecture, code review, sécurité.
* **Développeur full‑stack** : features front/back.
* **DevOps** : CI/CD, infra, observabilité.
* **Designer UX/UI** : maquettes, design system.
* **QA** : stratégie de tests, E2E.
* **Responsable communication** : contenus, réseaux sociaux.
* **Relation opérateurs** (si SMS/USSD) : contrats/tests.

**19) Livrables**

* Code source (repo privé), documentation technique (README, schéma, scripts).
* Manuel d’administration (PDF), guide de modération.
* Dossier d’exploitation (runbook incidents, sauvegardes, monitoring).
* Jeux de données de test, cas de test, rapports de charge/sécu.
* Charte de vote publique (PDF) + Politique de confidentialité.

**20) Hypothèses & risques**

* Connexions réseau parfois instables : privilégier pages légères, caching, PWA.
* Pic de trafic en fin de période : prévoir autoscaling/over‑provisioning temporaire.
* Fraude organisée (bots, fermes de clics) : durcir règles, WAF, ajuster throttling.
* Dépendance opérateurs pour SMS/USSD : plan B email/WhatsApp OTP.
* Sensibilité des résultats : stratégie de communication et procédures de contestation.

**21) Budget (enveloppe indic.)**

* **Hébergement** : VPS/Cloud (2–3 instances + DB gérée) ; CDN.
* **Services** : email transactionnel, SMS/USSD (coût à l’envoi), CAPTCHA (si payant).
* **Développement** : jours‑hommes par rôle (cf. § 18).
* **Support événement** : garde pendant la fenêtre de vote.

**22) Conformité & Légal**

* CGU/Privacy rédigées, consentement cookies.
* Droit à l’image pour médias candidats.
* Archivage des logs et des preuves de vote (hash + horodatage) pour transparence.

**23) Annexes**

**23.1 Checklists**

* **Avant l’ouverture** :
  + Catégories et règles validées
  + Candidats vérifiés
  + OTP opérationnel (email/SMS)
  + Tests de charge OK
  + Plan de communication prêt
* **Pendant** :
  + Monitoring actif (latence, erreurs)
  + Alerte anomalies
  + Support niveau 1/2
* **Après** :
  + Exports signés
  + Publication résultats/rapport
  + REX (retour d’expérience)

**23.2 Esquisse de permissions (RBAC)**

* **Admin** : tout.
* **Staff** : gestion candidats/catégories, exports, sans accès aux secrets.
* **Jury** : interface de vote jury.
* **User** : voter, gérer profil.
* **Anonyme** : lecture publique uniquement.

**23.3 Style & identité**

* Visuel lié au **Fleuve Makona** : palette inspirée (bleus/verts, doré pour « award »), typographie lisible, icônes simples.
* Ton éditorial : célébration, fierté locale, transparence.