

# DEVIS DÉTAILLÉ - Système MakePlus

## Plateforme de Gestion d'Événements Multi-Rôles

**Date:** 23 Décembre 2025

**Projet:** MakePlus Backend & Admin Dashboard

**Client:** [Nom du Client]

**Type:** Développement Full-Stack (Backend Django + Dashboard Web + API REST)

### RÉSUMÉ EXÉCUTIF

Développement complet d'une plateforme de gestion d'événements professionnels permettant la gestion de multiples événements simultanés avec système de badges QR, contrôle d'accès multi-niveaux, paiement en ligne, et interface d'administration complète.

### Technologies Déployées

- **Backend:** Django 5.2.7 + Django REST Framework
- **Base de données:** PostgreSQL (production) / SQLite (dev)
- **Authentification:** JWT (JSON Web Tokens)
- **Documentation API:** Swagger/OpenAPI (drf-yasg)
- **Compatibilité:** Web + Mobile (Flutter) via API REST

### FONCTIONNALITÉS DÉVELOPPÉES

#### 1. SYSTÈME DE GESTION D'ÉVÉNEMENTS

##### 1.1 Création et Configuration d'Événements

###### Wizard de création multi-étapes:

- Informations de base (nom, dates, lieu, description)
- Upload de fichiers (logo, bannière, programme PDF, guide PDF)
- Configuration des thèmes et catégories
- Paramètres avancés (JSON metadata)
- Calculs automatiques (participants, exposants, salles)

###### Gestion du cycle de vie:

- Statuts: À venir / Actif / Terminé / Annulé
- Mise à jour des dates et horaires
- Gestion des contacts organisateurs
- Modification et suppression contrôlées

###### Fichiers PDF de l'événement:

- Programme de l'événement (schedule/agenda)
- Guide du participant (handbook)
- Stockage optimisé avec lazy loading
- API multipart/form-data pour upload
- Système extensible vers cloud storage (S3, Azure)

**Temps de développement:** ~40 heures

---

## 2. SYSTÈME D'AUTHENTIFICATION & CONTRÔLE D'ACCÈS

### 2.1 Multi-rôles & Permissions

**4 Rôles implémentés:**

**Organisateur (Organizer):**

- Contrôle total sur l'événement
- Création/édition de tout le contenu
- Gestion des utilisateurs et rôles
- Accès aux statistiques complètes

**Gestionnaire des Salles (Room Manager):**

- Gestion des salles et sessions
- Validation des questions/réponses
- Assignation du staff aux salles
- Statistiques de fréquentation

**Contrôleur des Badges (Badge Controller):**

- Scan et vérification des QR codes
- Gestion des accès aux salles
- Enregistrement des entrées/sorties
- Statistiques de scan en temps réel

**Participant & Exposant:**

- Accès au contenu de l'événement
- Badge QR unique et permanent
- Inscription aux ateliers payants
- Questions/réponses sur les sessions
- (Exposant: scan des visiteurs au stand)

### 2.2 Système de QR Code Unifié

**One QR per User:**

- Un seul QR code par utilisateur (permanent)
- Valide pour tous les événements auxquels l'utilisateur est inscrit

- Format: MKPL-{user\_id\_hash}
- Génération automatique à la création de compte

**Contrôle d'accès multi-niveaux:**

- Niveau 1: Accès à l'événement (UserEventAssignment)
- Niveau 2: Accès aux sessions payantes (SessionAccess)
- Niveau 3: Accès aux salles spécifiques (Participant.allowed\_rooms)

## 2.3 JWT Authentication

**Authentification sécurisée:**

- Tokens access + refresh
- Context événement dans le token
- Switch entre événements sans reconnexion
- Expiration et renouvellement automatique
- Support CORS pour applications mobiles

**Temps de développement:** ~60 heures

---

## 3. GESTION DES SALLES & SESSIONS

### 3.1 Salles (Rooms)

**Fonctionnalités complètes:**

- Création avec capacité et localisation
- Calcul automatique du taux d'occupation
- Gestion des participants actuels
- Historique des accès
- Assiguation du personnel (time slots)

### 3.2 Sessions & Conférences

**Types de sessions:**

- Conférences (gratuites)
- Ateliers payants (avec prix)
- Statuts en français: "pas\_encore", "en\_cours", "terminé"

**Informations détaillées:**

- Titre, description, horaires
- Intervenant (nom, titre, bio, photo)
- Thème et image de couverture
- Metadata personnalisées (JSON)

**Intégration YouTube Live:**

- URL de streaming en direct
- Support pour événements hybrides
- Affichage dans l'application mobile

**Système Q&A (Questions/Réponses):**

- Participants posent des questions sur les sessions
- Gestionnaires répondent en temps réel
- Horodatage et traçabilité complète
- Filtres: répondu/non-répondu

**Actions sur les sessions:**

- Démarrer une session (mark\_live)
- Terminer une session (mark\_completed)
- Annuler une session

**Temps de développement:** ~50 heures

---

## 4. SYSTÈME DE PARTICIPANTS & BADGES

### 4.1 Gestion des Participants

**Profils complets:**

- Lien avec compte utilisateur
- Badge unique par événement
- QR code pour accès
- Statut de check-in
- Horodatage des actions

**Contrôle d'accès aux salles:**

- Liste des salles autorisées (ManyToMany)
- Vérification en temps réel
- Historique des accès (RoomAccess)
- Statistiques de fréquentation

### 4.2 Ateliers Payants

**Système de paiement:**

- Sessions marquées comme payantes
- Prix configurables
- Vérification d'accès (SessionAccess)
- Statut de paiement
- Date d'octroi d'accès

**Temps de développement:** ~35 heures

---

---

## 5. SYSTÈME DE CAISSE & TRANSACTIONS

### 5.1 Point de Vente (POS)

**Gestion des caisses:**

- Création de multiples caisses par événement
- Authentification email/password indépendante
- Interface dédiée pour opérateurs
- Isolation des données par caisse

### 5.2 Articles Payables

**Configuration flexible:**

- Ateliers payants (sync avec sessions)
- Articles personnalisés (dîner, goodies, etc.)
- Prix configurables
- Activation/désactivation

### 5.3 Transactions

**Processus de paiement:**

- Recherche participant (nom, email, QR code)
- Sélection multiple d'articles
- Traitement du paiement
- Génération de badge imprimable avec QR
- Marquage automatique de présence

**Gestion des transactions:**

- Historique complet
- Annulation/remboursement avec raison
- Statistiques en temps réel
- Filtres avancés

**Sécurité:**

- Sessions indépendantes par caisse
- Hashing des mots de passe
- Protection CSRF
- Audit trail complet

**Temps de développement:** ~45 heures

---

## 6. SYSTÈME D'ANNONCES

---

### **Communication ciblée:**

- Création d'annonces par événement
- Ciblage par rôle:
  - Tous les participants
  - Participants seulement
  - Exposants seulement
  - Contrôleurs seulement
  - Gestionnaires seulement
- Horodatage de création
- Permissions: propriétaire ou gestionnaire

### **API complète:**

- Liste avec filtres
- Création/édition/suppression
- Recherche full-text (titre, description)

**Temps de développement:** ~20 heures

---

## 7. SYSTÈME EXPOSANTS

### 7.1 Gestion des Stands

#### **Fonctionnalités exposants:**

- Badge QR dédié
- Scan des visiteurs au stand
- Capture de notes sur chaque visite
- Export des données visiteurs (Excel)

### 7.2 Statistiques de Visites

#### **Analytics exposants:**

- Nombre total de visites
- Visites du jour
- Détails par visiteur
- Horodatage précis
- Notes enregistrées

#### **Endpoints API:**

- Liste des scans
- Mes scans (exposant only)
- Création de scan
- Filtres par exposant/événement

**Temps de développement:** ~25 heures

---

---

## 8. ASSIGNATION DU PERSONNEL AUX SALLES

### Planification du staff:

- Assignation utilisateurs → salles
- Plages horaires (start\_time, end\_time)
- Rôles configurables
- Statut actif/inactif

### Filtres avancés:

- Par salle
- Par utilisateur
- Par événement
- Par rôle
- Assignations actuelles (time-based)

**Temps de développement:** ~20 heures

---

## 9. STATISTIQUES & ANALYTICS

### 9.1 Dashboard Organisateur

#### Vue d'ensemble:

- Nombre total d'événements
- Participants inscrits/présents
- Salles et sessions actives
- Activité récente

### 9.2 Statistiques Contrôleur

#### Endpoint dédié: `GET /api/my-room/statistics/`

- Auto-détection de la salle assignée
- Scans totaux et du jour
- Accès accordés vs refusés
- Participants uniques
- Scans récents avec détails

### 9.3 Statistiques Exposant

#### Endpoint dédié: `GET /api/exposant-scans/my_scans/`

- Visites totales et du jour
- Liste détaillée des visiteurs
- Notes et commentaires

## 10. DASHBOARD ADMINISTRATEUR WEB

### 10.1 Interface d'Administration

#### **Dashboard complet développé en Django:**

- Interface web responsive
- Navigation intuitive
- Formulaires de création/édition
- Tableaux avec filtres et recherche
- Pagination automatique

### 10.2 Pages Implémentées

#### **Gestion des événements:**

- Liste des événements avec statistiques
- Wizard de création multi-étapes
- Page détails avec metrics
- Édition et suppression

#### **Gestion des utilisateurs:**

- Création rapide avec QR automatique
- Assignation de rôles par événement
- Affichage et download du QR code
- Historique des assignations

#### **Gestion des salles:**

- Configuration complète
- Visualisation des sessions
- Assignation du personnel

#### **Gestion des sessions:**

- Création avec tous les champs
- Support YouTube live
- Configuration ateliers payants

#### **Gestion des caisses:**

- CRUD complet des caisses
- Gestion des articles payables
- Historique des transactions
- Statistiques par caisse

**Templates développés:** ~15 fichiers HTML/CSS

**Formulaires Django:** ~10 forms avec validation

**Temps de développement:** ~70 heures

---

## 11. API REST COMPLÈTE

### 11.1 Documentation Interactive

**Swagger UI intégré:**

- Documentation auto-générée
- Interface de test interactive
- Schémas de données
- Exemples de requêtes/réponses

### 11.2 Endpoints Développés (60+ endpoints)

**Authentification:**

- POST /api/auth/register/
- POST /api/auth/login/
- POST /api/auth/logout/
- GET /api/auth/profile/
- GET /api/auth/me/
- POST /api/auth/change-password/
- POST /api/auth/select-event/
- POST /api/auth/switch-event/
- GET /api/auth/my-events/

**Événements:**

- GET/POST /api/events/
- GET/PUT/PATCH/DELETE /api/events/{id}/
- GET /api/events/{id}/statistics/
- POST /api/events/{id}/upload\_programme/
- POST /api/events/{id}/upload\_guide/

**Salles:**

- GET/POST /api/rooms/
- GET/PUT/PATCH/DELETE /api/rooms/{id}/
- GET /api/rooms/{id}/current\_participants/
- GET /api/rooms/{id}/access\_history/

**Sessions:**

- GET/POST /api/sessions/
- GET/PUT/PATCH/DELETE /api/sessions/{id}/

- POST /api/sessions/{id}/mark\_live/
- POST /api/sessions/{id}/mark\_completed/
- POST /api/sessions/{id}/cancel/
- GET /api/sessions/{id}/participants/

#### **Participants:**

- GET/POST /api/participants/
- GET/PUT/PATCH/DELETE /api/participants/{id}/
- GET /api/participants/{id}/qr\_code/
- POST /api/participants/verify\_qr/
- GET /api/participants/{id}/access\_rooms/
- POST /api/participants/{id}/grant\_room\_access/

#### **Accès aux salles:**

- GET/POST /api/room-access/
- GET /api/room-access/?participant\_id={id}
- GET /api/room-access/?room\_id={id}

#### **Accès aux sessions:**

- GET/POST /api/session-access/
- GET /api/session-access/?participant\_id={id}
- GET /api/session-access/?session\_id={id}

#### **Annonces:**

- GET/POST /api/annonces/
- GET/PUT/PATCH/DELETE /api/annonces/{id}/
- GET /api/annonces/?target=participants
- GET /api/annonces/?event\_id={id}

#### **Questions de session:**

- GET/POST /api/session-questions/
- GET/PUT/PATCH/DELETE /api/session-questions/{id}/
- POST /api/session-questions/{id}/answer/
- GET /api/session-questions/?is\_answered=false

#### **Assignations de salles:**

- GET/POST /api/room-assignments/
- GET/PUT/PATCH/DELETE /api/room-assignments/{id}/
- GET /api/room-assignments/?current=true
- GET /api/room-assignments/?user\_id={id}

#### **Scans exposants:**

- GET/POST /api/exposant-scans/
- GET /api/exposant-scans/my\_scans/

- GET /api/exposant-scans/?exposant\_id={id}

#### **Statistiques:**

- GET /api/my-room/statistics/ (contrôleur)
- GET /api/exposant-scans/my\_scans/ (exposant)

#### **Assignations utilisateur-événement:**

- GET/POST /api/user-assignments/
- GET/PUT/PATCH/DELETE /api/user-assignments/{id}/

### **11.3 Fonctionnalités API**

#### **Standards REST:**

- Verbes HTTP appropriés (GET, POST, PUT, PATCH, DELETE)
- Codes de statut corrects (200, 201, 400, 403, 404, etc.)
- Pagination automatique
- Filtres avancés via query params
- Recherche full-text
- Tri des résultats

#### **Sécurité:**

- JWT authentication
- Permissions granulaires par rôle
- CORS configuré pour mobile/web
- Validation des données
- Protection CSRF

#### **Performance:**

- Select\_related et prefetch\_related
- Requêtes optimisées
- Lazy loading pour fichiers
- Indexation base de données

**Temps de développement:** ~80 heures

---

## **12. INTÉGRATION FLUTTER (Documentation)**

#### **Guides complets fournis:**

- Guide d'intégration Flutter (1721 lignes)
- Exemples de code Dart
- Modèles de données
- Gestion des erreurs
- Gestion du stockage sécurisé (tokens)

- Gestion du cache
- Exemples d'écrans

#### Configuration CORS:

- Headers configurés pour mobile
- Support des requêtes OPTIONS
- Domaines multiples autorisés

Temps de développement (documentation): ~20 heures

---

## MODÈLES DE DONNÉES (12 Modèles)

Modèles Implémentés:

1. **UserProfile** - Profil étendu utilisateur avec QR code unique
2. **Event** - Événement avec fichiers PDF, metadata, statistiques
3. **UserEventAssignment** - Liaison utilisateur-événement-rôle
4. **Room** - Salles avec capacité et occupation
5. **Session** - Sessions/ateliers avec streaming YouTube et Q&A
6. **Participant** - Badges avec QR et accès aux salles
7. **RoomAccess** - Logs d'accès aux salles avec audit
8. **SessionAccess** - Contrôle d'accès ateliers payants
9. **Annonce** - Annonces ciblées par rôle
10. **SessionQuestion** - Questions/réponses sur sessions
11. **RoomAssignment** - Assigntion personnel aux salles
12. **ExposantScan** - Tracking des visites aux stands

Modèles Caisse:

13. **Caisse** - Points de vente avec authentification
14. **PayableItem** - Articles payables configurables
15. **CaisseTransaction** - Transactions avec traçabilité

**Total:** 15 modèles Django avec relations complexes

---

## INFRASTRUCTURE & DÉPLOIEMENT

Technologies & Packages

#### Requirements.txt (22 packages):

- Django 5.2.7
- Django REST Framework 3.15.2
- djangorestframework-simplejwt 5.4.0
- django-cors-headers 4.6.0
- django-filter 24.3

- drf-yasg 1.21.10 (Swagger)
- psycopg2-binary 2.9.11 (PostgreSQL)
- Pillow 12.0.0 (Images)
- qrcode 8.2 (QR code generation)
- openpyxl 3.1.5 (Excel export)
- gunicorn 23.0.0 (Production server)
- whitenoise 6.8.2 (Static files)
- python-decouple 3.8 (Config management)
- Et autres dépendances...

## Configuration Déploiement

### **Ready for production:**

- Settings.py configuré pour prod/dev
- Variables d'environnement (.env support)
- Static files avec WhiteNoise
- Gunicorn WSGI server
- PostgreSQL support
- Migrations complètes

## Scripts Utilitaires Fournis

### **Management commands et scripts:**

- reset\_admin.py - Reset mot de passe admin
- check\_db.py - Vérification base de données
- create\_test\_event.py - Génération données de test
- test\_controller\_stats.py - Test stats contrôleurs
- update\_event\_dates.py - Mise à jour dates événements
- assign\_controller\_room.py - Assignation contrôleurs
- create\_paid\_ateliers.py - Création ateliers payants
- Et autres scripts de maintenance...

**Temps de développement:** ~30 heures

---

## DOCUMENTATION FOURNIE

Fichiers Documentation (30+ fichiers Markdown):

### 1. **BACKEND\_DOCUMENTATION.md** (3642 lignes)

- Documentation API complète
- Architecture système
- Exemples d'utilisation

### 2. **FLUTTER\_INTEGRATION\_GUIDE.md** (1721 lignes)

- Intégration mobile complète
- Code Dart exemples
- Best practices

### 3. **DATABASE\_STRUCTURE\_AND\_WORKFLOW.md** (557 lignes)

- Structure de la base de données
- Relations entre entités
- Workflows

### 4. **USER\_ACCESS\_CONTROL\_SYSTEM.md** (848 lignes)

- Système QR code
- Contrôle d'accès multi-niveaux
- Flows d'authentification

### 5. **ADMIN\_DASHBOARD\_DOCUMENTATION.md** (871 lignes)

- Guide complet dashboard
- Captures d'écran
- Instructions d'utilisation

### 6. **CAISSE\_SYSTEM\_IMPLEMENTATION.md** (314 lignes)

- Système de caisse
- Point de vente
- Transactions

### 7. **CONTROLLER\_STATISTICS\_GUIDE.md** (539 lignes)

- API statistiques contrôleurs
- Exemples Flutter
- Implémentation

### 8. **EVENT\_PDF\_DOCUMENTATION\_INDEX.md**

- Upload fichiers PDF
- Gestion documents

### 9. **YOUTUBE\_AND\_QA\_INTEGRATION.md**

- Streaming live
- Système Q&A

### 10. **NEW\_API\_ENDPOINTS.md** (277 lignes)

- Référence rapide API
- Nouveaux endpoints
- Exemples requêtes

### 11. Et 20+ autres fichiers de documentation

## ⚡ FEATURES AVANCÉES

### 1. Système de Recherche

Recherche full-text sur:

- Événements (nom, description, lieu)
- Utilisateurs (nom, email, username)
- Sessions (titre, description, intervenant)
- Annonces (titre, description)

### 2. Filtrage Avancé

Filtres sur tous les endpoints:

- Par date/période
- Par statut
- Par rôle
- Par événement
- Par utilisateur
- Combinaisons multiples

### 3. Export de Données

Export Excel:

- Liste des participants
- Statistiques exposants
- Transactions caisse
- Rapports personnalisés

### 4. Upload de Fichiers

Gestion fichiers:

- Images (logo, bannière, photo intervenant)
- PDFs (programme, guide)
- QR codes (génération automatique)
- Système extensible vers S3/Azure

### 5. Gestion Multi-Événements

Événements parallèles:

- Isolation complète des données
- Switch entre événements
- Rôles différents par événement

- QR code unique multi-événements

## 6. Audit & Traçabilité

### Tracking complet:

- created\_at, updated\_at sur tous les modèles
  - created\_by, verified\_by pour audit
  - Historique des accès
  - Logs des transactions
- 

## FORMATION & SUPPORT

### Documentation Technique

#### Fournie:

- Documentation API complète (Swagger)
- Guides d'intégration (10,000+ lignes)
- README et guides de démarrage rapide
- Exemples de code
- Scripts de test

### Support Post-Livraison

#### À définir selon contrat:

- Période de garantie
  - Corrections de bugs
  - Support par email/ticket
  - Mises à jour mineures
- 

## STATISTIQUES DU PROJET

### Lignes de Code

- **Backend Python/Django:** ~15,000 lignes
- **Templates HTML/CSS:** ~3,000 lignes
- **Documentation Markdown:** ~10,000 lignes
- **Scripts utilitaires:** ~1,500 lignes
- **Total:** ~29,500 lignes

### Fichiers Crées

- Modèles Django: 15
  - Views/ViewSets: 25+
  - Serializers: 20+
  - URLs endpoints: 60+
-

- Forms Django: 10+
- Templates HTML: 15+
- Scripts Python: 15+
- Documentation MD: 30+

## Temps de Développement Total

### Estimation basée sur les fonctionnalités:

Composant	Heures
Modèles de données & migrations	40h
Authentification & permissions	60h
API REST endpoints	80h
Système QR code	30h
Dashboard admin web	70h
Système de caisse	45h
Gestion événements	40h
Salles & sessions	50h
Statistiques & analytics	30h
Documentation	50h
Tests & debugging	60h
Configuration déploiement	30h
<b>TOTAL</b>	<b>~585 heures</b>

## ⌚ ESTIMATION BUDGÉTAIRE

### Option 1: Tarification au Forfait

#### Développement complet livré:

- Backend Django complet fonctionnel
- API REST complète (60+ endpoints)
- Dashboard administrateur web
- Système de caisse POS
- Documentation complète
- Configuration production
- Scripts utilitaires

**Prix forfaitaire suggéré:** À définir selon votre grille tarifaire

## Option 2: Tarification Horaire

### Basé sur 585 heures de développement:

- Taux horaire: [Votre taux] DZD/h ou €/h
- Total heures: 585h
- **Coût total:** [585 × Taux horaire]

## Option 3: Tarification par Module

### Modules décomposés:

Module	Heures	Prix Unitaire	Sous-total
Core Backend + API	120h	[Taux] × 120h	[...]
Authentification & Sécurité	60h	[Taux] × 60h	[...]
Gestion Événements	90h	[Taux] × 90h	[...]
Dashboard Admin	70h	[Taux] × 70h	[...]
Système Caisse POS	45h	[Taux] × 45h	[...]
QR Code & Accès	30h	[Taux] × 30h	[...]
Statistiques	30h	[Taux] × 30h	[...]
Documentation	50h	[Taux] × 50h	[...]
Tests & Déploiement	90h	[Taux] × 90h	[...]
<b>TOTAL</b>	<b>585h</b>		<b>[Total]</b>

## LIVRABLES

### Code Source

#### Repository complet:

- Code source Django complet
- Templates HTML/CSS
- Configuration settings
- Requirements.txt
- Scripts utilitaires
- Fichiers de migration
- .gitignore configuré

### Base de Données

#### Schema & migrations:

- Tous les fichiers de migration

- Script d'initialisation
- Données de test (optionnel)

## Documentation

### **30+ fichiers fournis:**

- README principal
- Documentation API (Swagger)
- Guides d'intégration Flutter
- Documentation technique complète
- Guides d'utilisation admin
- Diagrammes et workflows

## Configuration Production

### **Ready to deploy:**

- settings.py prod/dev
  - Gunicorn configuration
  - WhiteNoise pour static files
  - PostgreSQL setup
  - Variables d'environnement
  - Requirements.txt complet
- 

## MAINTENANCE & ÉVOLUTIONS

### Maintenance Suggérée

#### **Package recommandé:**

- Corrections de bugs
- Mises à jour de sécurité
- Support technique (X heures/mois)
- Mises à jour de documentation

### Évolutions Possibles

#### **Fonctionnalités additionnelles:**

- Application mobile Flutter complète
  - Système de notifications push
  - Système de messagerie in-app
  - Export rapports avancés (PDF, Excel)
  - Dashboard analytics avancé
  - Intégration paiement en ligne (CIB, Stripe)
  - Système de ticketing intégré
  - Check-in facial reconnaissance
-

- Application mobile caisse (offline)
- API webhooks pour intégrations tierces

Chaque évolution à estimer séparément selon la complexité.

---

## 📞 CONDITIONS & MODALITÉS

Paiement

### 📝 À définir:

- Acompte: [X]%
- Livraison intermédiaire: [X]%
- Livraison finale: [X]%

Délais

### 📝 Déjà développé:

- Le système est **complètement développé et fonctionnel**
- jours
- jours

Garantie

### 📝 Support post-livraison:

- mois
- Corrections bugs incluses
- Support technique par email
- Mises à jour mineures incluses

Formation

### 📝 Optionnelle:

- heures
- heures
- Documentation vidéo

---

## ⌚ AVANTAGES COMPÉTITIFS DU SYSTÈME

### 1. Système Complet et Intégré

- Solution tout-en-un (backend + admin + API)
- Aucune dépendance externe complexe
- Prêt pour production immédiate

### 2. Scalable et Extensible

- Architecture modulaire
- API REST standard
- Extensible vers cloud (AWS, Azure)
- Support multi-événements simultanés

### 3. Sécurisé

- JWT authentication
- Permissions granulaires
- Audit trail complet
- Protection CSRF
- Hashing des mots de passe

### 4. Performant

- Requêtes optimisées
- Pagination automatique
- Caching possible
- Lazy loading fichiers

### 5. Compatible Mobile

- API REST complète
- CORS configuré
- Documentation Flutter fournie
- Exemples de code mobile

### 6. Maintenance Facile

- Code propre et documenté
- Structure Django standard
- Migrations automatiques
- Scripts utilitaires fournis

---

## TECHNOLOGIES UTILISÉES - DÉTAIL

### Backend Framework

- **Django 5.2.7:** Framework web Python moderne et sécurisé
- **Django REST Framework 3.15.2:** API REST puissante
- **Simple JWT 5.4.0:** Authentication JWT standard

### Base de Données

- **PostgreSQL:** Production (recommandé)
- **SQLite:** Développement et tests
- Support ORM Django complet

## Authentication & Sécurité

- **JWT Tokens:** Access + Refresh tokens
- **Django Permissions:** Système de permissions natif
- **CORS Headers:** Configuration multi-domaines
- **Password Hashing:** Bcrypt via Django

## Documentation API

- **drf-yasg 1.21.10:** Génération Swagger/OpenAPI automatique
- **ReDoc & Swagger UI:** Interfaces interactives

## Fichiers & Media

- **Pillow 12.0.0:** Traitement d'images
- **qrcode 8.2:** Génération QR codes
- **openpyxl 3.1.5:** Export Excel

## Déploiement

- **Gunicorn 23.0.0:** WSGI HTTP Server production
- **WhiteNoise 6.8.2:** Serving static files
- **python-decouple 3.8:** Variables d'environnement

## Outils Développement

- **django-filter 24.3:** Filtrage avancé
- **PyYAML 6.0.3:** Configuration YAML

---

## NOTES IMPORTANTES

1. **Système Déjà Développé:** Le système est complet et fonctionnel. Cette offre concerne la livraison du système existant avec éventuelles personnalisations.
2. **Personnalisations:** Toute personnalisation supplémentaire (branding, fonctionnalités spécifiques) sera estimée séparément.
3. **Hébergement:** Le prix n'inclut pas l'hébergement serveur (à prévoir séparément: VPS, cloud, etc.).
4. **Application Mobile:** L'API est prête pour Flutter, mais le développement de l'app mobile complète est une prestation séparée.
5. **Formation:** La formation des utilisateurs peut être incluse selon le package choisi.
6. **Code Source:** Le code source complet est livré (pas de location/SaaS).

---

## PROCHAINES ÉTAPES

1. **Validation du devis** par le client

2. **Signature du contrat** et versement acompte
  3. **Personnalisations** si nécessaires (logo, couleurs, etc.)
  4. **Configuration** sur serveur de production
  5. **Tests** en environnement client
  6. **Formation** des administrateurs
  7. **Mise en production** et livraison finale
  8. **Support** post-livraison selon contrat
- 

## CONTACT

**Développeur:** [Votre Nom]

**Email:** [Votre Email]

**Téléphone:** [Votre Téléphone]

**Site Web:** [Votre Site]

**Validité de l'offre:**  jours à partir du 23/12/2025

---

## CONFIDENTIALITÉ

Ce document et toutes les informations qu'il contient sont confidentiels et destinés uniquement au client désigné. Toute reproduction ou divulgation sans autorisation est interdite.

---

**Document généré le:** 23 Décembre 2025

**Version:** 1.0

**Statut:** Proposition Commerciale

---

*MakePlus - Plateforme de Gestion d'Événements Professionnels*

*Développé avec Django 5.2.7 + Django REST Framework*