

AlterValue V4.1 - Moteur d'Enquête SaaS

Vue d'Ensemble

Le moteur d'enquête V4.1 est une refonte majeure du module Survey existant (Méthode B présentisme) vers une architecture **JSON-driven** permettant de définir des types d'enquêtes réutilisables, de lancer des campagnes multi-entreprises, et de générer des rapports PDF professionnels.

Objectifs

1. **Flexibilité** : Définir des types d'enquêtes via des fichiers JSON réutilisables
2. **Multi-campagnes** : Lancer plusieurs campagnes simultanées pour différentes entreprises
3. **Anonymisation** : Conformité RGPD + BNQ 9700-800 (seuil d'agrégation 15 réponses)
4. **Calcul automatique** : Scores, indicateurs critiques, métriques financières (ROI, coût caché)
5. **Livrables PDF** : Rapports exécutifs, radar QVCT, feuilles de route

Architecture Technique

Stack (intégrée à Next.js existant)

- **Backend** : Next.js API Routes (TypeScript)
- **Base de données** : PostgreSQL via Prisma ORM
- **Génération PDF** : API Abacus HTML2PDF (playwright)
- **Graphiques** : Recharts (radar charts, bar charts)
- **Authentification** : NextAuth.js (existant)
- **RBAC** : Système v4.0 existant

Schéma de Base de Données (Prisma)

```

// =====
// SURVEY ENGINE V4.1 - JSON-DRIVEN SURVEYS
// =====

enum SurveyTypeCategory [] {
    PRESENTEEISM      // Diagnostic présentisme
    QVCT              // Qualité de Vie et Conditions de Travail
    RPS               // Risques Psycho-Sociaux
    CLIMATE           // Climat social
    CUSTOM            // Type personnalisé
}

enum CampaignStatus [] {
    DRAFT             // Brouillon, pas encore lancée
    SCHEDULED         // Planifiée, date de lancement future
    ACTIVE            // En cours de collecte
    CLOSED            // Clôturée, calculs en cours
    COMPLETED         // Terminée, résultats disponibles
    ARCHIVED          // Archivée
}

// Types d'enquêtes (définitions JSON réutilisables)
model SurveyType {
    id                String          @id @default(cuid())
    typeId            String          @unique // "PRESENTEEISM_METHOD_B",
"RADAR_QVCT_FLASH"
    name              String          // "Diagnostic Présentisme - Méthode B"
    description       String?
    version           String          @default("1.0")
    category          SurveyTypeCategory
    definition        Json            // Structure complète JSON
    isActive          Boolean         @default(true)
    isSystem          Boolean         @default(false) // Types système non modifiables
    estimatedDuration Int            @default(10) // Minutes
    anonymityThreshold Int           @default(15) // Seuil d'anonymat
    dataRetentionDays Int           @default(730) // 2 ans par défaut
    createdById       String?
    createdBy         User?
    references[: [id]] SurveyCampaign[]
    campaigns         SurveyCampaign[]
    createdAt         DateTime        @default(now())
    updatedAt         DateTime        @updatedAt

    @@index([category])
    @@index([isActive])
}

// Campagnes d'enquête (instances lancées)
model SurveyCampaign [] {
    id                String          @id @default(cuid())
    companyId         String
    company           Company         @relation(fields: [companyId], references[: [id]], onDelete: Cascade)
    surveyTypeId      String
    surveyType        SurveyType      @relation(fields: [surveyTypeId], references[: [id]])
    name              String          // "Diagnostic Q1 2026"
    status            CampaignStatus  @default(DRAFT)
    token             String          @unique @default(cuid()) // Token public pour accès anonyme
    configuration     String
}

```

```

targetPopulation      Int?          // Nombre de personnes ciblées
minRespondents       Int          @default(15) // Minimum requis
maxRespondents       Int?          // Maximum (null = illimité)
// Dates
scheduledStartDate   DateTime?
scheduledEndDate     DateTime?
launchedAt           DateTime?
closedAt              DateTime?
// Créateur
createdById          String
createdBy             User          @relation(fields: [createdById],
references:[id])
// Relations
responses            CampaignResponse[]
result               CampaignResult?
deliverables         CampaignDeliverable[]
createdAt            DateTime        @default(now())
updatedAt            DateTime        @updatedAt
// Indexes
@@index([companyId])
@@index([status])
@@index([token])
}

// Réponses individuelles (anonymisées)
model CampaignResponse []
  id                  String        @id @default(cuid())
  campaignId          String
  campaign             SurveyCampaign
  @relation(fields: [campaignId], references:[id], onDelete: Cascade)
  respondentHash       String        // Hash anonyme pour éviter doublons
  responses            Json          // [{"Q1": 7, "Q2": ["A", "C"], "Q3": "Text libre"}]
  metadata             Json?         // Données démographiques agrégées optionnelles
  submittedAt          DateTime      @default(now())
  isComplete            Boolean      @default(true)
  userAgent            String?       // Pour détection fraude
  @unique([campaignId, respondentHash])
  @@index([campaignId])
}

// Résultats calculés par campagne
model CampaignResult []
  id                  String        @id @default(cuid())
  campaignId          String        @unique
  campaign             SurveyCampaign
  @relation(fields: [campaignId], references:[id], onDelete: Cascade)
  responseCount        Int          @default(0)
  participationRate   Float?       // % de participation
  scores               Json          // {"GESTION": 6.5, "CONCILIATION": 7.2}
  criticalIndicators  Json?       // "high_presenteeism": true triggered by
  true, severity: "high"
  financialMetrics    Json?       // "hidden_cost": 125000,
  "roi_estimate": 87500
  qualitativeInsights Json?       // Verbatims, ranking agrégés
  narrative            String?      // Texte narratif généré
  calculatedAt         DateTime
  createdAt            DateTime      @default(now())
  updatedAt            DateTime        @updatedAt
}

```

```
// Livrables générés (PDF)
model CampaignDeliverable []
  id                  String          @id @default(cuid())
  campaignId         String
  campaign           SurveyCampaign  @relation(fields: [campaignId], references:[]
[id], onDelete: Cascade)
  type                String          // "executive_report", "radar_chart", "ac-
tion_roadmap"
  title               String
  cloudStoragePath   String?
  fileFormat          String          @default("pdf")
  generatedAt        DateTime        @default(now())
  @@index([campaignId])
}
```



Structure JSON des Types d'Enquêtes

Template de Base

```
{
  "survey_type_id": "UNIQUE_ID",
  "survey_metadata": {
    "name": "Nom commercial",
    "version": "1.0",
    "framework": ["BNQ 9700-800"],
    "target_audience": "PME/ETI 50-2000 salariés",
    "decision_makers": ["CEO", "DAF", "DRH"],
    "estimated_duration_minutes": 15,
    "primary_objective": "Objectif business principal",
    "secondary_objectives": []
  },
  "data_governance": {
    "anonymity_threshold": 15,
    "rgpd_compliant": true,
    "sensitive_data": false,
    "data_retention_days": 730
  },
  "questionnaire_structure": {
    "modules": [
      {
        "id": "MODULE_ID",
        "title": "Titre du module",
        "bnq_reference": "BNQ 9700-800 § X.X",
        "questions": [
          {
            "id": "Q1",
            "type": "scale",
            "scale": "0-10",
            "text": "Question...",
            "required": true
          }
        ]
      }
    ]
  },
  "calculation_engine": {
    "scoring_dimensions": [
      {
        "id": "DIMENSION_ID",
        "questions": ["Q1", "Q2"],
        "aggregation": "mean",
        "weight": 0.35,
        "alert_threshold": 6
      }
    ],
    "critical_indicators": [
      {
        "id": "indicator_id",
        "condition": "DIMENSION_ID < 6",
        "severity": "high",
        "message": "Message d'alerte"
      }
    ],
    "financial_formulas": [
      {
        "id": "hidden_cost",
        "formula": "prevalence * headcount * daily_cost * productivity_loss * 220"
      }
    ]
  },
  "output_engine": {

```

```

    "deliverables": [],
    "visualizations": [],
    "narrative_templates": {}
}
}

```

Types de Questions Supportés

Type	Description	Valeur Stockée
consent	Case à cocher consentement	boolean
scale	Échelle numérique (0-10)	number
single_choice	Choix unique	string
multiple_choice	Choix multiples	string[]
rank	Classement par priorité	string[] (ordonné)
open-ended	Texte libre	string
numeric	Valeur numérique libre	number

Types d'Enquêtes V4.1

1. PRESENTEEISM_METHOD_B (Système)

Durée : 5 min | Catégorie : PRESENTEEISM

Questionnaire actuel de la Méthode B, migré vers le nouveau format JSON.

Questions :

- Q1 (single_choice) : Fréquence de travail en efficacité réduite
- Q2 (scale 50-100) : Niveau d'efficacité perçu
- Q3 (multiple_choice) : Facteurs contribuant
- Q4 (multiple_choice) : Impacts observés
- Q5 (single_choice) : Heures de travail hebdomadaires

Calculs : Prévalence, Perte de productivité, Coût caché, ROI

2. RADAR_QVCT_FLASH

Durée : 8 min | Catégorie : QVCT

Diagnostic rapide des 4 sphères BNQ 9700-800.

Modules :

- Pratiques de Gestion (4 questions)
- Conciliation Vie Pro/Perso (3 questions)
- Environnement de Travail (3 questions)

- Habitudes de Vie (2 questions)
- Priorisation (ranking + verbatim)

Calculs : Score par sphère, Radar chart, Matrice de priorité

3. BNQ_DATA_COLLECTION

Durée : 15 min | **Catégorie** : QVCT

Collecte de données pour conformité BNQ complète.



RBAC - Permissions

Rôle	Créer Type	Créer Cam-pagne	Lancer	Voir Ré-sultats	Générer PDF
SU-PER_ADMIN	✓	✓	✓	✓	✓
EXPERT	✗	✓	✓	✓	✓
PILOTE_QVCT	✗	✓	✓	✓	✓
OBSERVATEUR	✗	✗	✗	✓ (lecture)	✗



Pages Next.js

Structure des Routes

/dashboard/surveys/	
└── page.tsx	# Liste des campagnes
└── types/	
└── page.tsx	# Liste des types d'enquêtes
└── [typeId]/	
└── page.tsx	# Détail/édition d'un type
└── campaigns/	
└── new/	
└── page.tsx	# Créer une campagne
└── [campaignId]/	
└── page.tsx	# Dashboard campagne
└── preview/	
└── page.tsx	# Prévisualisation questionnaire
└── results/	
└── page.tsx	# Résultats détaillés
└── deliverables/	
└── page.tsx	# Livrables générés
└── respond/	
└── [token]/	
└── page.tsx	# Page de réponse publique (anonyme)

Composants Clés

1. **SurveyTypeCard** : Affiche un type d'enquête avec métadonnées
 2. **CampaignCard** : Affiche une campagne avec statut et progression
 3. **SurveyRenderer** : Rend dynamiquement un questionnaire depuis JSON
 4. **ResultsDashboard** : Affiche les scores, charts, indicateurs
 5. **RadarChart** : Graphique radar des 4 sphères BNQ
 6. **PdfPreview** : Prévisualisation du rapport avant génération
-



API Routes

Types d'Enquêtes

```
// GET /api/surveys/types - Liste des types actifs
// POST /api/surveys/types - Créer un nouveau type (SUPER_ADMIN)
// GET /api/surveys/types/[typeId] - Détail d'un type
// PUT /api/surveys/types/[typeId] - Modifier un type
// DELETE /api/surveys/types/[typeId] - Supprimer un type
```

Campagnes

```
// GET /api/surveys/campaigns - Liste des campagnes (filtré par rôle)
// POST /api/surveys/campaigns - Créer une campagne
// GET /api/surveys/campaigns/[id] - Détail campagne
// PUT /api/surveys/campaigns/[id] - Modifier campagne
// PATCH /api/surveys/campaigns/[id]/launch - Lancer la campagne
// PATCH /api/surveys/campaigns/[id]/close - Clôturer la campagne
// POST /api/surveys/campaigns/[id]/calculate - Déclencher le calcul
```

Réponses (Public)

```
// GET /api/surveys/respond/[token] - Charger le questionnaire
// POST /api/surveys/respond/[token] - Soumettre une réponse
// GET /api/surveys/respond/[token]/count - Nombre de réponses actuelles
```

Résultats & Livrables

```
// GET /api/surveys/campaigns/[id]/results - Résultats calculés
// POST /api/surveys/campaigns/[id]/deliverables/generate - Générer PDF
// GET /api/surveys/campaigns/[id]/deliverables - Liste des livrables
// GET /api/surveys/deliverables/[id]/download - Télécharger un livrable
```



Moteur de Calcul

Classe SurveyCalculationEngine

```
class SurveyCalculationEngine {
    constructor(surveyDefinition: SurveyTypeDefinition);

    calculate(responses: ResponseData[], companyParams?: CompanyParams): CalculationResult;

    private calculateDimensionScores(responses: ResponseData[]): Record<string, number>;
    private evaluateCriticalIndicators(scores: Record<string, number>): CriticalIndicator[];
    private calculateFinancialMetrics(responses: ResponseData[], params: CompanyParams): FinancialMetrics;
    private aggregateQualitativeData(responses: ResponseData[]): QualitativeInsights;

    generateNarrative(results: CalculationResult): string;
}
```

Algorithmes

1. **Scores par dimension** : Moyenne (pondérée ou non) des questions
2. **Indicateurs critiques** : Évaluation de conditions (ex: GESTION < 6)
3. **Métriques financières** : Formules paramétrables avec variables entreprise
4. **Narratif** : Templates avec placeholders dynamiques



Génération PDF

Templates HTML/Jinja2

1. **executive_report.html** : Rapport exécutif COMEX (2-3 pages)
 - KPIs clés (score global, coût caché, ROI)
 - Alertes critiques
 - Radar chart
 - Top 3 recommandations
2. **detailed_report.html** : Rapport détaillé RH (10+ pages)
 - Scores par dimension avec barres de progression
 - Distribution des facteurs
 - Verbatims clés
 - Plan d'action détaillé
3. **bnq_compliance_report.html** : Rapport conformité BNQ
 - Checklist des exigences
 - Preuves documentaires
 - Plan de remédiation

Charte Graphique

- **Primaire** : Or (#d4af37)
- **Secondaire** : Teal (#20b2aa)
- **Alertes** : Rouge (#dc3545), Orange (#ffc107), Vert (#28a745)

Plan d'Itérations

Phase 1 : v4.1-alpha - Schéma & Modèles

Objectif : Fondations de la base de données

- [] Ajouter les nouveaux modèles Prisma (SurveyType, SurveyCampaign, CampaignResponse, CampaignResult, CampaignDeliverable)
- [] Créer les migrations
- [] Seed avec le type PRESENTEEISM_METHOD_B
- [] Relations avec Company et User existants

Phase 2 : v4.1-beta - API Routes & Moteur de Calcul

Objectif : Backend fonctionnel

- [] API CRUD pour SurveyType
- [] API CRUD pour SurveyCampaign
- [] API de soumission de réponses (publique)
- [] Moteur de calcul TypeScript (`lib/survey-calculation-engine.ts`)
- [] Intégration RBAC

Phase 3 : v4.1-gamma - Interface Utilisateur

Objectif : Pages Next.js complètes

- [] Page liste des campagnes (`/dashboard/surveys`)
- [] Page création de campagne
- [] Page dashboard campagne (statut, progression, lien de partage)
- [] Page de réponse publique (`/survey/[token]`)
- [] SurveyRenderer dynamique

Phase 4 : v4.1-delta - Résultats & Visualisations

Objectif : Affichage des résultats

- [] Page résultats avec scores
- [] Composant RadarChart (Recharts)
- [] Indicateurs critiques avec alertes visuelles
- [] Distribution des facteurs (bar chart)

Phase 5 : v4.1-epsilon - Génération PDF

Objectif : Livrables professionnels

- [] Templates HTML (`executive_report`, `detailed_report`)
- [] Intégration API Abacus HTML2PDF
- [] Page liste des livrables
- [] Téléchargement des PDF

Phase 6 : v4.1-zeta - Types Supplémentaires & Polish

Objectif : Types RADAR_QVCT_FLASH et BNQ_DATA_COLLECTION

- [] Seed des 2 types supplémentaires

- [] Tests des différents questionnaires
 - [] Optimisations UX
 - [] Documentation utilisateur
-

Critères de Succès

- [] Un Expert peut créer une campagne PRESENTEEISM_METHOD_B en moins de 2 minutes
 - [] Les réponses sont collectées via un lien public anonyme
 - [] Le calcul des résultats s'exécute en moins de 5 secondes pour 500 réponses
 - [] Un PDF exécutif professionnel est généré avec radar chart
 - [] L'Observateur peut consulter les résultats en lecture seule
 - [] Le seuil d'anonymat (15 réponses) est respecté avant affichage des résultats
-

Dépendances avec V4.0

- **RBAC** : Utilise le système de rôles existant
 - **Company** : Les campagnes sont liées aux missions (companies)
 - **User** : Le créateur de la campagne est tracé
 - **ActivityLog** : Les actions sont auditées (création, lancement, clôture)
-

Fichiers de Référence

Les types d'enquêtes JSON sont stockés dans :

- lib/survey-types/PRESENTEEISM_METHOD_B.json - Type présentéisme Méthode B
- lib/survey-types/RADAR_QVCT_FLASH.json - Type Radar QVCT 4 sphères BNQ

Les spécifications originales ont été fournies via les fichiers uploadés lors de la conception.