

ANALYSE DES BESOINS & MAQUETTES

Innov'Events

Dossier de conception

ECF — Concepteur Développeur d'Applications
Février 2026

1. Analyse des besoins

L'application Innov'Events répond aux besoins de Chloé Dubois, organisatrice d'événements professionnels. L'analyse ci-dessous recense l'ensemble des exigences utilisateur exprimées dans le cahier des charges.

1.1 Acteurs et besoins fonctionnels

Visiteur (non connecté)

- Consulter les pages publiques (accueil, événements, avis)
- Contacter l'agence via le formulaire de contact
- Effectuer une demande de devis (création d'un prospect)
- Créer un compte et se connecter
- Demander la réinitialisation de mot de passe

Client (connecté)

- Consulter ses devis et télécharger les PDF
- Accepter, refuser ou demander la modification d'un devis
- Modifier son profil et demander la suppression de ses données (RGPD)
- Déposer un avis après un événement (un seul par événement)

Employé (connecté)

- Consulter la liste des clients et des événements
- Ajouter des notes sur les événements
- Mettre à jour le statut des tâches
- Valider ou refuser les avis déposés par les clients

Administrateur (connecté)

- Gérer les prospects, clients et événements (CRUD complet)
- Convertir un prospect en client
- Créer et mettre à jour les devis
- Générer les devis en PDF et les envoyer par email
- Gérer les comptes employés
- Consulter les logs d'audit et statistiques

1.2 Exigences non fonctionnelles

Exigence	Description	Implémentation
Accessibilité	Navigation clavier, structure sémantique, alternatives textuelles	HTML sémantique, aria-labels, contrastes RGAA
Sécurité	Validation entrées, mots de passe robustes, confirmation email	bcrypt, tokens JWT, XSS/SQL injection prevention
RGPD	Consentement, droit à l'effacement, mentions légales	Pages CGU/CGV, suppression compte, rétention données
Responsive	Application utilisable sur tous les écrans	Angular responsive + application mobile Ionic
Traçabilité	Journalisation des actions sensibles	MongoDB audit_logs avec TTL 12 mois
Performance	Temps de réponse acceptable	Lazy loading, pagination, cache statique Nginx

2. Charte graphique

La charte graphique d'Innov'Events est conçue pour transmettre professionnalisme, confiance et dynamisme.

2.1 Palette de couleurs

Primaire	#FA8A27	CTA, accents, éléments actifs
Secondaire	#334659	Headers, texte important, fond sombre
Fond	#F5F6FA	Arrière-plan des pages
Texte	#111827	Corps de texte principal
Muted	#6B7280	Texte secondaire, labels

2.2 Typographie

Support	Police principale	Fallbacks	Usage
Web (Angular)	Segoe UI	Tahoma, Geneva, Verdana, sans-serif	Interface complète
Mobile (Ionic)	System	iOS: -apple-system ; Android: Roboto	Respect des conventions OS
Documents	Calibri	Helvetica, sans-serif	Documentation technique

2.3 Principes d'interface

- CTA en couleur primaire (#FA8A27), actions secondaires en fond neutre
- Cartes fond blanc, border-radius 10–14px, ombre légère (box-shadow)
- Formulaires : labels explicites, erreurs proches du champ, champs requis signalés
- Accessibilité : contrastes suffisants, focus visible, navigation clavier, textes alternatifs
- Navigation : header fixe (web), bottom tab bar (mobile)
- Responsive : grille 12 colonnes, breakpoints 768px / 1024px

3. Wireframes & Mockups Web

Les wireframes présentent la structure et la hiérarchie des éléments. Les mockups appliquent la charte graphique. Les fichiers HTML interactifs sont fournis en annexe.

3.1 WEB_WF_01 — Accueil (public)

Objectif : présenter l'offre d'Innov'Events et guider le visiteur vers la demande de devis.

Zone	Contenu	Interaction
Header	Logo + navigation (5 liens)	Lien vers chaque page
Hero	Titre + accroche + CTA principal	Clic CTA → page Demande de devis
Services	3 blocs (Sur mesure, Accompagnement, Excellence)	Informatif
Footer	© 2026 + liens légaux + contact	Liens vers CGU/CGV/Contact

3.2 WEB_WF_02 — Événements (public)

Objectif : permettre la découverte et le filtrage des événements.

Zone	Contenu	Interaction
Titre page	"Nos événements"	—
Filtres	Chips : Tous, Conférence, Séminaire, Soirée, Team building	Filtrage par type
Grille	Cartes : image + badge type + nom + date/lieu + CTA	Clic → détail événement
Pagination	Navigation pages (si > 8 résultats)	Changement de page

3.3 WEB_WF_03 — Espace client (Mes devis)

Objectif : permettre au client de consulter et gérer ses devis.

Zone	Contenu	Interaction
Topbar	Nom utilisateur + entreprise + déconnexion	Déconnexion
Sidebar	Menu : Dashboard, Événements, Mes devis (actif), Profil	Navigation entre sections
Liste devis	Cartes : référence + montant + statut (badge)	—
Actions	Boutons : PDF, Accepter, Refuser, Modifier	Actions sur chaque devis

3.4 WEB_MK_01 — Mockup Accueil

Application de la charte : hero en gradient secondaire (#334659), CTA primaire (#FA8A27), cartes services fond blanc avec icônes, footer sombre.

3.5 WEB_MK_02 — Mockup Événements

Application de la charte : filtres en chips (actif = primaire), cartes avec image gradient + badge type en primaire, bouton « Voir détails » en lien orange.

3.6 WEB_MK_03 — Mockup Espace client

Application de la charte : sidebar claire avec item actif en primaire, badges statut sémantiques (orange = en attente, vert = accepté), boutons d'action colorés.

Les fichiers HTML interactifs des 3 wireframes et 3 mockups web sont fournis en annexe (maquettes_web.html).

4. Wireframes & Mockups Mobile

L'application mobile Ionic/Angular est destinée aux employés et administrateurs pour la gestion terrain des événements.

4.1 MOB_WF_01 — Dashboard (employé)

Objectif : accéder rapidement aux événements à venir.

Zone	Contenu	Interaction
Topbar	Titre "Événements à venir"	—
Liste	Vignette + nom + date + lieu + statut	Tap → fiche événement

Tab bar	Événements (actif), CGU, CGV	Navigation entre onglets
---------	------------------------------	--------------------------

4.2 MOB_WF_02 — Fiche événement

Objectif : afficher les informations clés et permettre l'ajout de notes.

Zone	Contenu	Interaction
Image	Photo + bouton retour + badge type	Retour à la liste
Infos	Nom, dates, lieu, client, budget	Tap client → fiche client
CTA	"Ajouter une note"	Ouvre la zone de saisie
Notes	Liste chronologique des notes	Scroll

4.3 MOB_WF_03 — Fiche client

Objectif : contact en un clic depuis le terrain.

Zone	Contenu	Interaction
Avatar	Initiales + nom + entreprise	—
Coordonnées	Téléphone, email, adresse	—
Actions	Appeler, Email, Itinéraire	Ouvre l'app native correspondante

4.4 MOB_MK_01 — Mockup Dashboard

Topbar fond secondaire (#334659), cartes blanches avec vignette emoji colorée, badges statut sémantiques, tab bar avec icônes.

4.5 MOB_MK_02 — Mockup Fiche événement

Image header en gradient, badge type primaire, infos avec icônes, CTA orange "Ajouter une note", notes avec auteur en orange.

4.6 MOB_MK_03 — Mockup Fiche client

Avatar avec initiales sur fond primaire, coordonnées avec icônes, 3 boutons d'action pleine largeur (Appeler vert, Email secondaire, Itinéraire primaire).

Les fichiers HTML interactifs des 3 wireframes et 3 mockups mobile sont fournis en annexe (maquettes_mobile.html).

5. Schéma d'enchaînement des maquettes

Ce schéma formalise les parcours utilisateur principaux à travers les maquettes.

5.1 Parcours Web — Visiteur (demande de devis)



5.2 Parcours Web — Client (gestion devis)



5.3 Parcours Mobile — Employé



Chaque flèche représente une interaction utilisateur (clic, tap, soumission de formulaire). Les couleurs correspondent aux sections du parcours.

6. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation ci-dessous représente l'ensemble des fonctionnalités du système Innov'Events, organisées par acteur.

6.1 Acteurs du système

Acteur	Rôle	Accès
Visiteur	Utilisateur non connecté	Pages publiques uniquement
Client	Utilisateur connecté (rôle client)	Espace client (devis, profil, avis)
Employé	Utilisateur connecté (rôle employee)	Consultation + notes + tâches + avis
Administrateur	Utilisateur connecté (rôle admin)	Accès complet (CRUD + gestion)

6.2 Cas d'utilisation par acteur

Visiteur

ID	Cas d'utilisation	Description
UC1	Consulter les événements (public)	Voir la liste et le détail des événements visibles
UC2	Consulter les avis (public)	Lire les avis approuvés
UC3	Contacter l'agence	Envoyer un message via le formulaire
UC4	Effectuer une demande de devis	Créer un prospect via le formulaire
UC5	Créer un compte	Inscription avec email + mot de passe
UC6	Se connecter	Authentification par email/mot de passe
UC7	Mot de passe oublié	Réinitialisation par email

Client

ID	Cas d'utilisation	Description
UC8	Consulter ses devis	Voir la liste de ses devis + statuts
UC9	Télécharger un devis (PDF)	Télécharger le PDF généré
UC10	Accepter / Refuser un devis	Changer le statut du devis
UC11	Demander une modification	Envoyer une contre-proposition

UC12	Déposer un avis	Noter (1-5) + commenter après événement
------	-----------------	---

Employé

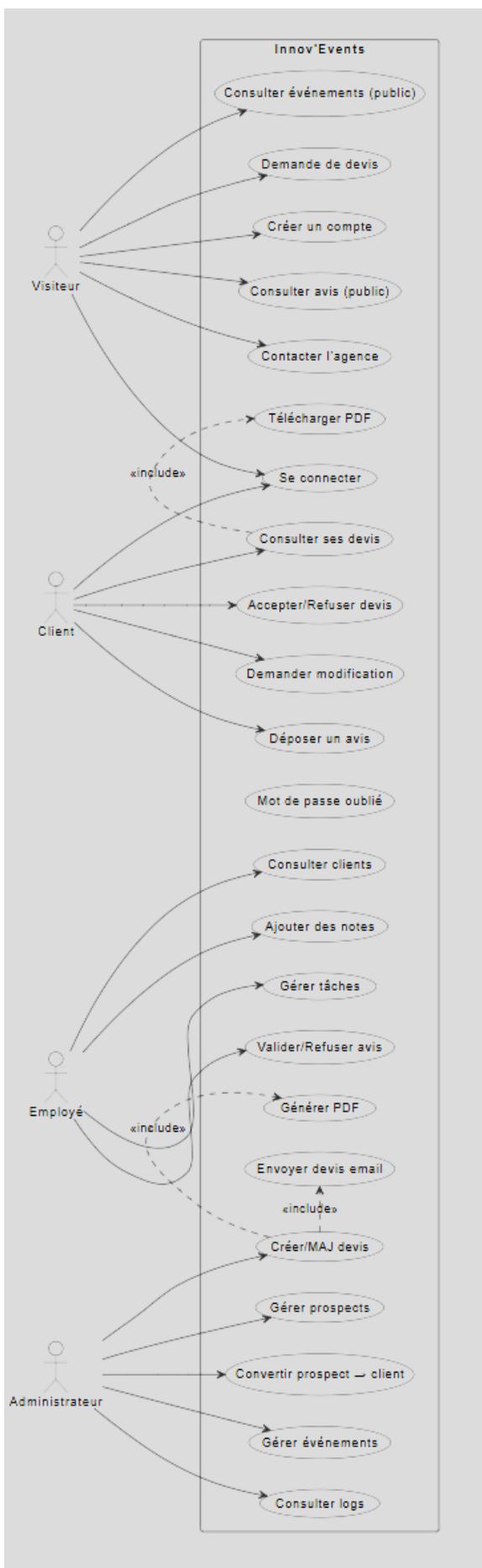
ID	Cas d'utilisation	Description
UC13	Consulter clients	Voir fiches clients et coordonnées
UC14	Consulter événements	Voir détails des événements
UC15	Ajouter des notes	Créer des notes sur un événement
UC16	Gérer statut des tâches	Mettre à jour (à faire / en cours / terminé)
UC17	Valider / Refuser avis	Modérer les avis clients

Administrateur

ID	Cas d'utilisation	Description
UC18	Gérer clients	CRUD complet sur les clients
UC19	Gérer prospects	CRUD + suivi statut des prospects
UC20	Convertir prospect en client	Créer le client depuis le prospect
UC21	Gérer événements	CRUD + statuts + images
UC22	Gérer employés	Créer/modifier/supprimer comptes employés
UC23	Créer / Mettre à jour devis	Édition complète des devis + services
UC24	Générer PDF devis	Génération TCPDF du document
UC25	Envoyer devis par email	Envoi via PHPMailer
UC26	Consulter logs / statistiques	Audit trail MongoDB

6.3 Relations entre cas d'utilisation

- UC23 « include » UC24 (Générer PDF) et UC25 (Envoyer email)
- UC19 « include » UC20 (Convertir prospect en client)
- UC8 « include » UC9 (Télécharger PDF)



Le diagramme UML complet est fourni en annexe au format PlantUML (fichier .puml).

7. Diagramme de séquence — Demande de devis

Fonctionnalité choisie : la demande de devis par un visiteur. Ce flux couvre la saisie du formulaire, la validation, la création du prospect en base et l'envoi d'une notification email.

7.1 Acteurs et composants

Participant	Type	Technologie
Visiteur	Acteur humain	Navigateur web
Frontend Web	Interface utilisateur	Angular 20
API PHP	Backend	PHP 8.2 (endpoint prospects/create.php)
MySQL	Base de données	Table prospects
SMTP	Service email	MailHog (dev) / Brevo (prod)

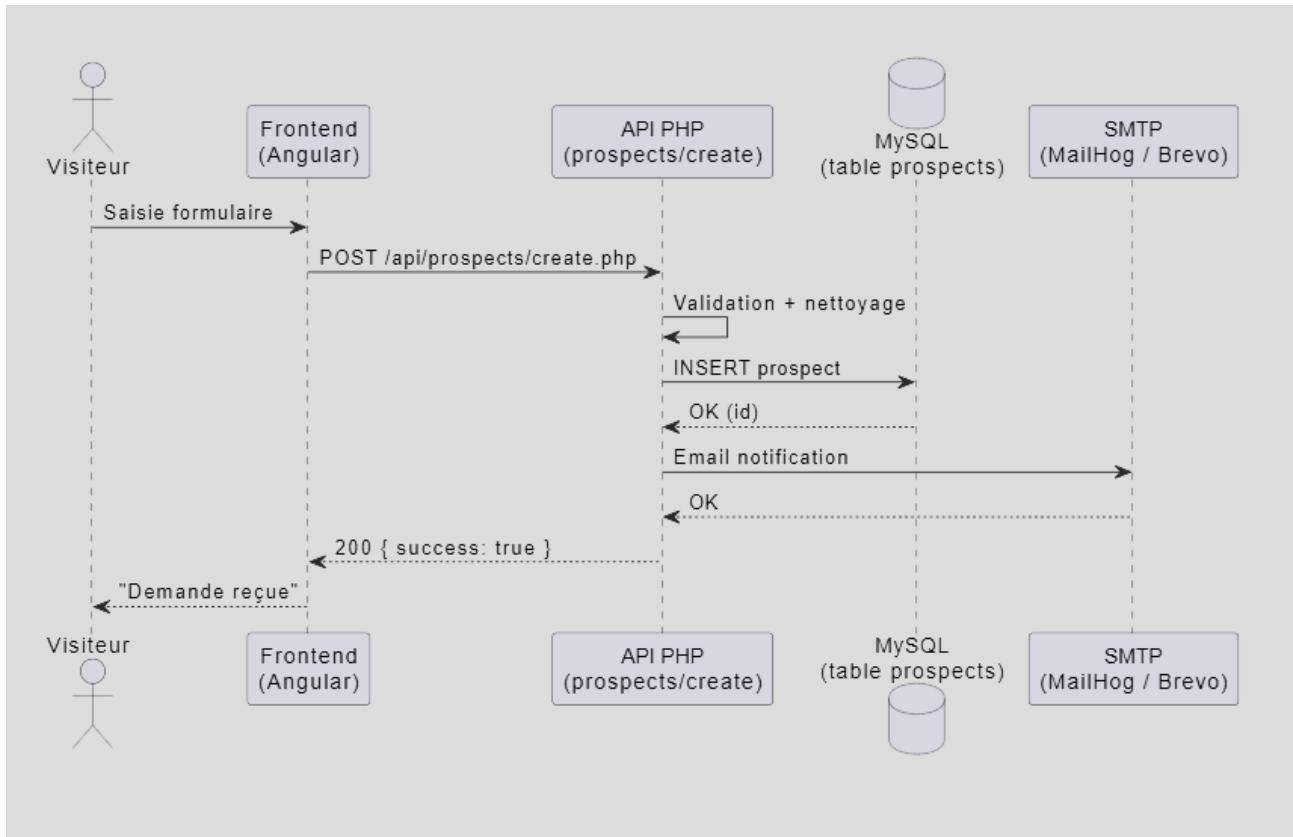
7.2 Flux séquentiel

#	Communication	Action	Détail
1	Visiteur → Frontend	Saisie et envoi du formulaire de demande de devis	L'utilisateur remplit : entreprise, contact, type d'événement, date, lieu, participants, description, image optionnelle
2	Frontend → API	POST /api/prospects/create.php (JSON)	Le frontend envoie les données en JSON via fetch/HttpClient
3	API → API	Validation des champs + nettoyage	Vérification : champs requis, format email, taille image, htmlspecialchars
4	API → MySQL	INSERT INTO prospects (...)	Création du prospect avec statut 'to_contact'
5	MySQL → API	OK (id prospect)	Retour de l'ID auto-incrémenté
6	API → SMTP	Envoi email notification	Notification à l'administrateur : "Nouvelle demande de devis"
7	SMTP → API	OK	Email envoyé avec succès
8	API → Frontend	200 JSON { success: true }	Réponse avec message de confirmation
9	Frontend → Visiteur	Affichage message de confirmation	"Votre demande a bien été reçue. Nous vous recontacterons sous 48h."

7.3 Cas alternatifs

Condition	Action	Réponse
Champs requis manquants	Validation échoue côté API	400 JSON { error: 'Champs obligatoires manquants' }
Email invalide	Regex email échoue	400 JSON { error: 'Format email invalide' }
Image trop lourde (> 5 Mo)	Vérification taille fichier	400 JSON { error: 'Image trop volumineuse' }
Erreur serveur MySQL	Catch PDOException	500 JSON { error: 'Erreur interne' }

Erreur envoi email	Catch PHPMailer exception	Le prospect est créé, mais l'email n'est pas envoyé (log erreur)
--------------------	---------------------------	--



Le diagramme UML de séquence complet est fourni en annexe au format PlantUML (fichier .puml).

8. Annexes — Code PlantUML

Les diagrammes UML peuvent être générés à partir du code PlantUML ci-dessous, via <https://www.plantuml.com/plantuml> ou tout outil compatible.

8.1 Diagramme de cas d'utilisation

```

@startuml
left to right direction

actor "Visiteur" as V
actor "Client" as C
actor "Employé" as E
actor "Administrateur" as A

rectangle "Innov'Events" {
    usecase "Consulter événements (public)" as UC1
    usecase "Consulter avis (public)" as UC2
    usecase "Contacter l'agence" as UC3
    usecase "Demande de devis" as UC4
    usecase "Créer un compte" as UC5
    usecase "Se connecter" as UC6
    usecase "Mot de passe oublié" as UC7
    usecase "Consulter ses devis" as UC8
    usecase "Télécharger PDF" as UC9
    usecase "Accepter/Refuser devis" as UC10
    usecase "Demander modification" as UC11
    usecase "Déposer un avis" as UC12
    usecase "Consulter clients" as UC13
    usecase "Ajouter des notes" as UC15
}

```

```

usecase "Gérer tâches" as UC16
usecase "Valider/Refuser avis" as UC17
usecase "Gérer prospects" as UC19
usecase "Convertir prospect → client" as UC20
usecase "Gérer événements" as UC21
usecase "Créer/MAJ devis" as UC23
usecase "Générer PDF" as UC24
usecase "Envoyer devis email" as UC25
usecase "Consulter logs" as UC26
}

V --> UC1
V --> UC2
V --> UC3
V --> UC4
V --> UC5
V --> UC6
C --> UC6
C --> UC8
C --> UC10
C --> UC11
C --> UC12
E --> UC13
E --> UC15
E --> UC16
E --> UC17
A --> UC19
A --> UC20
A --> UC21
A --> UC23
A --> UC26

UC23 .> UC24 : <<include>>
UC23 .> UC25 : <<include>>
UC8 .> UC9 : <<include>>
@enduml

```

8.2 Diagramme de séquence — Demande de devis

```

@startuml
actor "Visiteur" as V
participant "Frontend\n(Angular)" as FE
participant "API PHP\n(prospects/create)" as API
database "MySQL\n(table prospects)" as DB
participant "SMTP\n(MailHog / Brevo)" as SMTP

V -> FE : Saisie formulaire
FE -> API : POST /api/prospects/create.php
API -> API : Validation + nettoyage
API -> DB : INSERT prospect
DB --> API : OK (id)
API -> SMTP : Email notification
SMTP --> API : OK
API --> FE : 200 { success: true }
FE --> V : "Demande reçue"
@enduml

```