

Kylian Di Gaetano

Alexandre Pitarch

Rue de Madretsch 74
2503 Biel/Bienne

Au Village 63d
2904 Bressaucourt

Cahier des charges NightBuddy

Création API intuitive



(Travail réalisé dans le cadre du projet interdisciplinaire C732)
Année 2024-2025

lundi 27 janvier 2025

V.0.1

Table des matières

1	Table des matières.....	1
2	Introduction et contexte	2
2.1	Objet du document	2
2.2	Contexte	2
2.3	Enjeux et objectifs	2
3	Description du besoin fonctionnel	2
3.1	Besoins exprimés	2
3.2	Problématique à résoudre.....	3
3.3	Critères de succès.....	3
4	Fonctionnalités attendues	3
4.1	Nom de la fonctionnalité : Création d'un événement.....	Erreur ! Signet non défini.
4.2	3.3 Nom de la fonctionnalité : Gestion des invitations	Erreur ! Signet non défini.

1 Introduction et contexte

1.1 Objet du document

Ce document présente le cahier des charges du projet **NightBuddy**, une API dédiée à l'organisation et à la gestion d'événements nocturnes. Son objectif est de faciliter la planification d'événements nocturnes en centralisant des fonctionnalités comme les invitations, le covoiturage, et la gestion des thèmes musicaux. Ce document définit les objectifs, les fonctionnalités attendues et les livrables associés.

Note :

Il est important de préciser que ce projet et ce cahier des charges se concentrent uniquement sur le développement d'une API. Il ne s'agit ni de créer un site web ni une interface utilisateur, mais bien d'une infrastructure technique permettant à d'autres applications de gérer des événements nocturnes via cette API.

1.2 Contexte

Le projet **NightBuddy** est réalisé dans le cadre de l'atelier 732, "Cahier des charges". Cet atelier vise à apprendre comment rédiger un cahier des charges tout en développant une API pour mieux gérer le travail. L'objectif est de simuler un contexte professionnel et de combiner les compétences acquises à l'école dans différentes branches.

NightBuddy répond à ces enjeux en centralisant les informations essentielles pour l'organisation d'événements sociaux, notamment le covoiturage, les invitations, et la gestion des thèmes musicaux.

1.3 Enjeux et objectifs

L'objectif principal est de fournir une API intuitive qui :

- **Gestion des covoiturages** : Permettre aux conducteurs de créer des offres de covoiturage et aux participants de s'y inscrire.
- **Organisateurs d'événements** : Fournir une plateforme pour créer et partager des événements nocturnes (invitations, lieu, thème musicale).
- **Développeurs tiers** : Offrir une API ouverte pour intégrer des fonctionnalités sociales liées aux événements dans leurs applications.

2 Description du besoin fonctionnel

2.1 Besoins exprimés

NightBuddy doit répondre aux besoins suivants :

- **Développeurs tiers** : Besoin d'une API ouverte pour intégrer des fonctionnalités d'événements sociaux dans leurs applications.
- **Organisateurs d'événements** : Besoin d'une plateforme pour créer, gérer et partager des événements nocturnes.

- **Invités** : Besoin de recevoir des invitations et de consulter les détails (lieu, covoiturage, musique).

2.2 Problématique à résoudre

Le principal problème est le manque d'un outil unifié qui centralise toutes les informations nécessaires à l'organisation et à la participation à des événements nocturnes. Actuellement, les organisateurs et participants doivent souvent utiliser plusieurs plateformes distinctes (invitations, trajets, musique) sans possibilité de tout gérer au même endroit. Cela rend la planification d'événements fastidieuse, augmentant les risques de mauvaise coordination et d'oubli d'informations importantes.

Les critères suivants détermineront le succès du projet :

- Nombre d'organiseurs utilisant la plateforme.
 - Satisfaction des utilisateurs mesurée via des enquêtes.
 - Taux d'adoption par les développeurs tiers pour l'intégration de l'API.
-

3 Fonctionnalités attendues

3.1 Liste des fonctionnalités

Les fonctionnalités principales attendues de **NightBuddy** sont :

1. Covoiturage :

- Gestion des covoiturages avec création d'offres par les conducteurs.
- Inscription des participants à un covoiturage en fonction des places disponibles.
- Intégration de l'API Google Maps pour la gestion des rendez-vous pour covoiturage

2. Gestion des utilisateurs :

- Inscription et connexion des utilisateurs.
- Gestion des profils utilisateurs (informations personnelles, préférences).

3. Création et gestion des événements :

- Formulaire de création d'événements avec nom, date, lieu, et thème musical.
- Visualisation des événements créés et possibilité de les modifier ou supprimer.

4. Invitations aux événements :

- Envoi d'invitations à des participants.
- Suivi des réponses (acceptation ou refus) des invités.

5. Thèmes musicaux :

- Sélection et affichage d'un thème musical pour chaque événement.

3.2 Priorisation des fonctionnalités

Les fonctionnalités sont priorisées selon leur importance pour l'expérience utilisateur :

3.2.1 Priorité 1 :

- **Covoiturage** : Central dans l'organisation d'événements nocturnes, il facilite la logistique pour les invités et les organisateurs.

3.2.2 Priorité 2 :

- Gestion des utilisateurs (inscription, connexion).
- Création et gestion des événements.
- Gestion des invitations.

3.2.3 Priorité 3 :

- Intégration avec Google Maps.
- Thèmes musicaux.

3.3 Cas d'utilisation

Cette section présente les principaux cas d'utilisation de l'API NightBuddy, mettant en évidence les interactions essentielles entre les utilisateurs, organisateurs et conducteurs. Le diagramme ci-dessous illustre comment ces acteurs utilisent l'API pour gérer les soirées et le covoiturage.

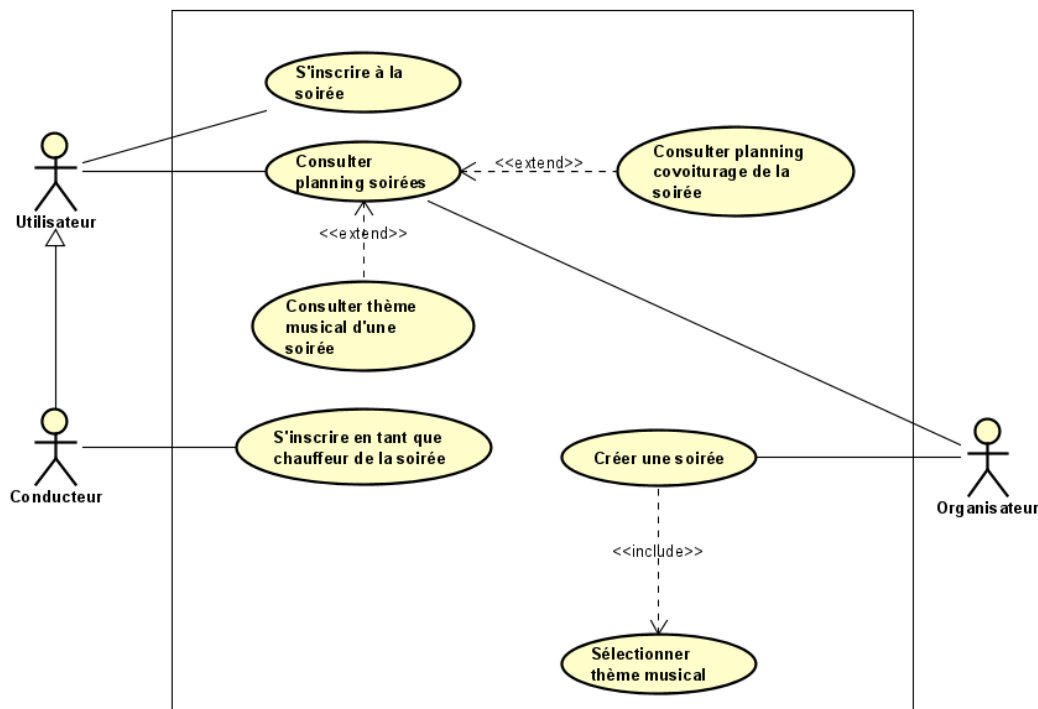


Figure 1, UseCase NightBuddy

3.3.1 Cas d'utilisation général

Les principaux cas d'utilisation sont les suivants :

1. **S'inscrire à la soirée** : Un utilisateur peut s'inscrire à une soirée organisée en consultant le planning des événements. Cette action est essentielle pour confirmer sa participation.
2. **Consulter le planning des soirées** : Les utilisateurs peuvent consulter les soirées disponibles, incluant les informations sur l'événement et les éventuels trajets de covoiturage disponibles.
3. **Consulter le thème musical d'une soirée** : L'utilisateur peut voir le thème musical associé à chaque soirée, ce qui est une information précieuse pour le choix de la soirée.
4. **S'inscrire en tant que chauffeur de la soirée** : Un conducteur peut proposer ses services en tant que chauffeur et offrir des places aux autres participants. Cela facilite la gestion du covoiturage.
5. **Créer une soirée** : L'organisateur peut créer une nouvelle soirée en remplissant un formulaire avec les informations pertinentes (date, lieu, etc.). Cette fonctionnalité inclut la sélection d'un thème musical.
6. **Consulter le planning de covoiturage de la soirée** : Les utilisateurs peuvent voir quelles voitures sont disponibles et s'inscrire pour un covoiturage.

3.3.2 Diagramme de cas d'utilisation

Analyse du diagramme :

- Le diagramme met en évidence trois rôles principaux :
 - **Utilisateur** : Toute personne qui interagit avec la plateforme pour consulter ou participer à une soirée.
 - **Conducteur** : Un utilisateur qui propose un service de covoiturage pour une soirée.
 - **Organisateur** : Celui qui crée une soirée et gère les détails associés (lieu, heure, thème musical).
- Les relations <<include>> et <<extend>> montrent des dépendances ou des actions optionnelles, renforçant la modularité du système :
 - Par exemple, **Consulter le thème musical d'une soirée** est une extension possible lors de la consultation du planning des soirées. L'utilisateur peut ou non consulter cette information selon ses besoins.
 - **Sélectionner le thème musical** est une action incluse dans le processus de création d'une soirée, ce qui montre que c'est une étape obligatoire pour l'organisateur.

4 Interfaces et intégration

4.1 Interfaces utilisateurs

L'API **NightBuddy**, étant destinée à être utilisée par des développeurs tiers, mais nous allons tout de même **implémenter une interface utilisateur graphique (UI)**. Toutefois, elle pourra être exploitée dans diverses applications front-end pour créer des interfaces utilisateur personnalisées. Ces applications pourront proposer différentes formes d'interactions, telles que :

- **Interfaces graphiques** : Applications mobiles ou web permettant aux utilisateurs de créer, gérer des événements, consulter des cartes interactives via Google Maps, ou organiser des covoitages.

- **Notifications** : Les utilisateurs pourront recevoir des notifications push ou par e-mail pour les mises à jour d'événements ou les réponses aux invitations.

Bootstrap 

Design System

Événements

#	Nom fête	Lieu	Propriétaire	Privé	
1	Carnaval Bassecourt	Bassecourt	Association Carnaval	<input type="checkbox"/>	Liste covoiturage
2	Anniversaire Michael	Grandfontaine	Michael	<input checked="" type="checkbox"/>	Liste covoiturage
3	Souper de boîte	Bienne	Swatch Group	<input checked="" type="checkbox"/>	Liste covoiturage

Conducteur	Nom	Prénom	Heure départ	Lieu départ	Places libres
1	Mark	Otto	20:30	rue du canton 30 2900 Porrentruy	3/4
2	Jacob	Thornton	22:00	Grande rue 15 2740 Moutier	1/4

4.2 Interfaces techniques

L'intégration technique de NightBuddy repose principalement sur les interactions avec plusieurs systèmes et services tiers :

- **API REST** : L'API de NightBuddy expose des points de terminaison REST qui permettent aux développeurs d'interagir avec la gestion des événements, invitations, et covoiturages. Les requêtes sont traitées via des méthodes standard (GET, POST, PUT, DELETE).
- **Base de données** : NightBuddy intègre une base de données relationnelle pour stocker les informations sur les utilisateurs, événements, invitations et covoiturages. Les données sont accessibles via des appels API sécurisés.
- **Services externes** : Intégration avec des API tierces, comme Google Maps, pour la gestion des adresses et la génération d'itinéraires. Cela permet aux utilisateurs de visualiser les lieux des événements et de faciliter l'organisation des trajets. La clé API Google Maps est nécessaire pour ce service.

5 Sécurité et confidentialité

L'API doit inclure des mécanismes d'authentification et de contrôle d'accès pour garantir que seules les personnes autorisées peuvent utiliser les fonctionnalités et accéder aux données

6 Planning et délais

Le calendrier du projet NightBuddy est structuré autour de plusieurs phases clés, chacune marquée par des échéances spécifiques et des jalons de validation. Le diagramme de Gantt permettra de visualiser ces étapes de manière claire.

- **Échéances clés** : Chaque phase du développement (conception, développement, tests, déploiement) sera définie avec des dates cibles pour assurer une progression régulière du projet.

- **Jalons de validation** : Des points de contrôle sont prévus à des moments stratégiques pour valider les livrables et garantir la conformité avec les attentes du projet.

Le diagramme de Gantt fournira une vue d'ensemble des délais et des dépendances entre chaque étape.

7 Glossaire

- **API (Application Programming Interface)** : Interface de programmation permettant à deux applications de communiquer entre elles.
- **OAuth2** : Protocole d'authentification qui permet un accès sécurisé aux ressources d'un utilisateur sans partager ses identifiants.
- **Covoiturage** : Fonctionnalité permettant aux utilisateurs de partager un véhicule pour se rendre à un événement.
- **Clé API** : Code unique permettant d'authentifier et d'autoriser les requêtes envoyées à un service tiers, comme Google Maps.
- **Organisateur** : Utilisateur qui crée et gère les événements dans l'application.
- **Utilisateur** : Personne qui interagit avec l'API pour s'inscrire à des événements ou consulter les informations.
- **Jalon** : Point de contrôle dans le projet, correspondant à la validation d'une phase ou d'un livrable.
- **Diagramme de Gantt** : Outil de gestion de projet permettant de visualiser les étapes et les délais.