

Projet Gestion des hôtels en Île-de-France

Description du document

Titre	Rapport de projet
Date	31/12/2024
Auteur	Noureddine Mohammedi, Bouamama Sidahmed, Mansouri Islam
Responsable	//
E-mail	//
Sujet	Rapport

Tableau des révisions

Date	Auteur(s)	Sections	Commentaires
15/11/2024	Mohammedi Nouredine, Bouamama Sidahmed, Mansouri Islam	Initialisation du projet	Création des tables principales dans PostgreSQL et définition des relations
17/11/2024	Mohammedi Nouredine, Bouamama Sidahmed, Mansouri Islam	Développement de l'interface	Implémentation de l'inscription et de la connexion avec PyQt5
20/11/2024	Mansouri Islam	Intégration API	Mise en place de l'API Nominatim pour la géolocalisation des adresses
22/11/2024	Bouamama Sidahmed	Affichage des résultats sur une carte	Utilisation de Folium pour afficher les hôtels sur une carte interactive
25/11/2024	Mohammedi Nouredine	Gestion des favoris	Ajout de la fonctionnalité pour gérer les hôtels favoris des utilisateurs
22/12/2024	Mohammedi Nouredine, Bouamama Sidahmed, Mansouri Islam	Tests finaux et optimisation	Résolution des bugs et optimisation des performances
25/12/2024	Mohammdi Nouredine, Bouamama Sidahmed, Mansouri Islam	Finalisation du rapport	Compilation du rapport final avec les captures d'écran et les analyses

Contents

1	Introduction	5
1.1	Contexte et objectifs.....	5
1.2	Méthodologie.....	5
2	Architecture globale	6
2.1	Structure du projet.....	6
2.2	Diagramme d'architecture.....	6
3	Base de données	7
3.1	Schéma relationnel.....	7
3.1.1	1. Table hotels.....	7
3.1.2	2. Table utilisateurs.....	7
3.1.3	3. Table favoris.....	8
3.2	Création des tables.....	8
3.3	Insertion des données.....	9
4	Interface utilisateur	10
4.1	Fonctionnalités principales.....	10
4.1.1	Connexion et inscription.....	10
4.1.2	Recherche d'hôtels.....	10
4.1.3	Affichage des résultats.....	10
4.1.4	Gestion des favoris.....	10
4.2	Description des outils et technologies.....	10
4.2.1	PyQt5.....	10
4.2.2	Base de données PostgreSQL.....	11
4.2.3	Folium.....	11
4.2.4	API Nominatim.....	11
4.3	Analyse des captures d'écran.....	11
4.3.1	1. Page de connexion.....	11
4.3.2	2. Page d'inscription.....	11
4.3.3	3. Page principale avec filtres et carte interactive.....	11
4.3.4	4. Bouton Clear (Réinitialiser).....	12
4.3.5	5. Gestion des favoris.....	12
5	Intégration des APIs	13
5.1	Utilisation de Nominatim pour la géolocalisation.....	13
5.2	Affichage sur une carte interactive avec Folium.....	13

6	Résultats obtenus	15
6.1	Fonctionnalités réussies.....	15
6.2	Tests et démonstrations.....	16
6.3	Résultats quantitatifs.....	16
6.4	Points forts de l'application.....	16
7	Défis rencontrés et solutions	17
7.1	Problèmes techniques.....	17
7.2	Résolution et améliorations.....	18
8	Conclusion et perspectives	19
8.1	Synthèse des réalisations.....	19
8.2	Propositions pour les améliorations futures.....	19

Chapter 1

Introduction

1.1 Contexte et objectifs

Avec l'évolution rapide de la numérisation, la recherche d'hôtels est devenue un aspect essentiel pour les voyageurs et les gestionnaires hôteliers. Cependant, il existe peu d'outils centralisés permettant de rechercher efficacement des hôtels en fonction de critères spécifiques tout en fournissant une visualisation cartographique intuitive.

Ce projet vise à combler cette lacune en développant une application interactive pour la gestion des hôtels en Île-de-France. L'objectif principal est de permettre aux utilisateurs de :

- Rechercher des hôtels proches d'une adresse donnée.
- Filtrer les résultats selon des critères tels que le prix, le classement ou le département.
- Visualiser les résultats sur une carte interactive.
- Gérer leurs hôtels favoris dans une base de données sécurisée.

1.2 Méthodologie

Pour atteindre ces objectifs, le projet a été structuré en plusieurs étapes :

- **Conception de la base de données :** Définir un schéma relationnel efficace pour stocker et gérer les informations des hôtels.
- **Développement de l'interface utilisateur :** Implémenter une interface graphique intuitive avec PyQt5.
- **Intégration des APIs :** Utiliser l'API Nominatim pour la géolocalisation et Folium pour l'affichage cartographique.
- **Tests et optimisation :** Identifier et résoudre les bugs, améliorer les performances et garantir une expérience utilisateur fluide.

Chapter 2

Architecture globale

2.1 Structure du projet

Le projet est organisé en trois principaux composants :

- **Base de données PostgreSQL** : Contient les informations détaillées sur les hôtels, y compris leur nom, adresse, coordonnées géographiques, prix, classement, et autres attributs pertinents. Fournit une structure relationnelle optimisée pour les requêtes complexes et l'intégration avec l'application.
- **Interface utilisateur développée avec PyQt5** : Offre une expérience utilisateur intuitive avec des fonctionnalités de recherche, de filtrage et de gestion des favoris. Intègre des modules pour la gestion des connexions, l'affichage des résultats, et l'interaction avec la carte interactive.
- **Carte interactive avec Folium et API Nominatim** : Utilise l'API Nominatim pour convertir les adresses en coordonnées géographiques (latitude et longitude). Folium est employé pour afficher les résultats sous forme de marqueurs sur une carte, offrant ainsi une visualisation claire des hôtels recherchés.

2.2 Diagramme d'architecture

Le projet suit une architecture modulaire avec les composants suivants :

- **Base de données** : Gestion centralisée des données hôtelières.
- **Application** : Interface graphique connectée à la base de données et à l'API externe.
- **API externe (Nominatim)** : Source des données géographiques nécessaires pour la localisation et l'affichage cartographique.

Chapter 3

Base de données

3.1 Schéma relationnel

Le projet repose sur une base de données relationnelle PostgreSQL comprenant trois tables principales :

3.1.1 1. Table `hotels`

Cette table stocke les informations essentielles des hôtels.

Colonnes principales :

- **id (clé primaire)** : Identifiant unique de l'hôtel.
- **nom** : Nom de l'hôtel.
- **adresse** : Adresse complète de l'hôtel.
- **latitude et longitude** : Coordonnées géographiques pour le positionnement sur une carte.
- **classement** : Nombre d'étoiles (échelle de 1 à 5) pour représenter la qualité de l'hôtel.
- **prix** : Coût moyen d'une nuitée (en euros).

Exemple de données :

id	Nom	Adresse	Latitude	Longitude	Classement	Prix (€)
1	Hôtel de Paris	12 rue Lafayette, Paris	48.8566	2.3522	5	150.00
2	Hôtel Bellevue	23 avenue des Champs, Lyon	45.7640	4.8357	4	90.00
3	Hôtel Riviera	Quai des Anglais, Nice	43.7102	7.2620	3	120.00

3.1.2 2. Table `utilisateurs`

Cette table gère les informations des utilisateurs enregistrés.

Colonnes principales :

- **id (clé primaire)** : Identifiant unique de l'utilisateur.
- **nom** : Nom complet de l'utilisateur.
- **email** : Adresse électronique (doit être unique).

- **mot_de_passe** : Mot de passe haché pour assurer la sécurité des comptes.

Exemple de données :

id	Nom	Email	Mot de passe
1	Mansouri Islam	islam@email.com	5f4dcc3b5aa765d61d8327deb882cf99
2	Noredinne	noredinne@email.com	6cb75f652a9b52798eb6cf2201057c73
3	Sidahmed	sidahmed@email.com	098f6bcd4621d373cade4e832627b4f6

3.1.3 3. Table favoris

Cette table associe les utilisateurs à leurs hôtels favoris. Elle établit une relation "many-to- many" entre utilisateurs et hôtels.

3.2 Création des tables

La création de la table hotels est réalisée à l'aide des fichiers SQL hoteldb-schema.sql.

```
CREATE TABLE utilisateurs (
    id SERIAL PRIMARY KEY,
    user_name VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL,
    password VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

```
CREATE TABLE favoris (
    id INT REFERENCES utilisateurs(id) ON DELETE CASCADE,
    hotel_name VARCHAR(255) NOT NULL,
    adresse VARCHAR(255) NOT NULL,
    cp VARCHAR(10) NOT NULL,
    prix NUMERIC(10, 2),
    classement INT
);
```

Chapter 4

Interface utilisateur

4.1 Fonctionnalités principales

L'interface utilisateur de l'application offre une expérience intuitive pour les actions suivantes :

4.1.1 Connexion et inscription

- Les utilisateurs peuvent s'inscrire ou se connecter pour accéder à des fonctionnalités avancées comme la gestion des favoris.
- L'interface de connexion prend en entrée un email et un mot de passe qui sont comparés avec les enregistrements de la table utilisateurs dans la base de données.

4.1.2 Recherche d'hôtels

- Les utilisateurs peuvent saisir une adresse ou une ville pour rechercher des hôtels proches.
- Les critères de recherche incluent : département, classement (étoiles), et prix maximum.
- Ces critères sont liés à la table hotels pour effectuer des requêtes SQL dynamiques.

4.1.3 Affichage des résultats

- Les résultats sont présentés sous forme de liste avec des colonnes telles que nom, adresse, code postal, prix, et classement.
- Une carte interactive affiche les emplacements des hôtels à l'aide de Folium.

4.1.4 Gestion des favoris

- Les utilisateurs connectés peuvent enregistrer ou supprimer des hôtels dans leurs favoris, ce qui est directement enregistré dans la table favoris.

4.2 Description des outils et technologies

4.2.1 PyQt5

Utilisé pour concevoir l'interface graphique. Les boutons (Connexion, Inscription, Appliquer les filtres) et les fenêtres modales sont développés avec PyQt5.

4.2.2 Base de données PostgreSQL

Permet le stockage des informations des utilisateurs, des hôtels et des favoris. Les requêtes SQL dynamiques permettent d'extraire les résultats en fonction des filtres appliqués.

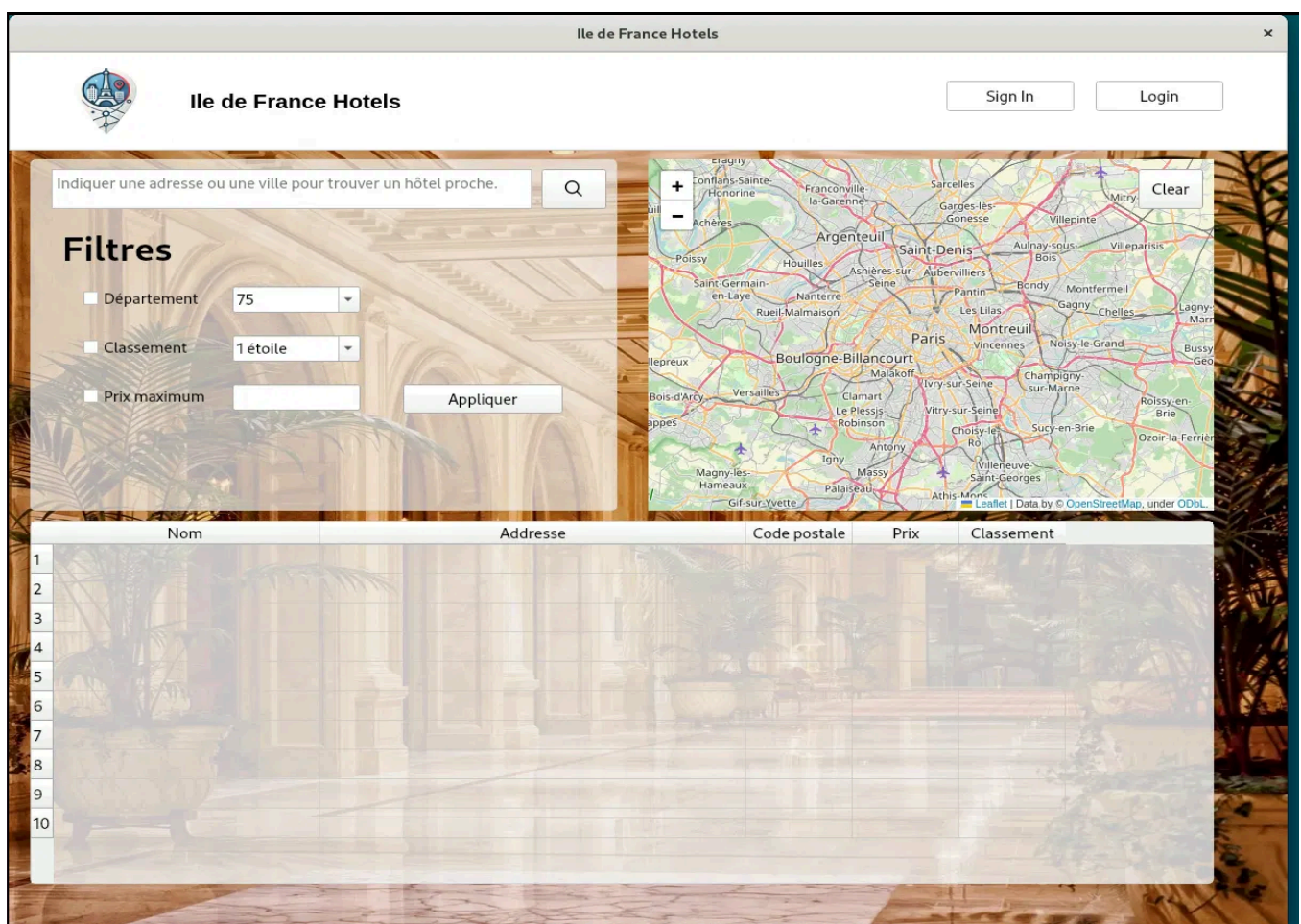
4.2.3 Folium

Utilisé pour afficher une carte interactive avec des marqueurs représentant les hôtels sélectionnés.

4.2.4 API Nominatim

Permet de convertir les adresses saisies par l'utilisateur en coordonnées géographiques.

4.3 Analyse de capture d'écran



4.3.1 1. Page de connexion

Description :

- La capture montre un bouton *Login* ouvrant une fenêtre modale pour la connexion.
- Les champs requis sont *Email* et *Mot de passe*.

4.3.2 2. Page d'inscription

Description :

- Le bouton *Sign In* ouvre une fenêtre modale permettant aux utilisateurs de s'inscrire.
- Les champs requis sont *Nom*, *Email*, et *Mot de passe*.

4.3.3 3. Page principale avec filtres et carte interactive**Description :**

- Un champ de recherche permet de saisir une adresse ou une ville.
- Les filtres incluent : département, classement, et prix maximum.
- Une carte interactive affiche les hôtels correspondants.

4.3.4 4. Bouton Clear (Réinitialiser)


Description :

- Permet de réinitialiser les filtres appliqués et de rafraîchir la carte.

4.3.5 5. Gestion des favoris


Description :


- Les utilisateurs connectés peuvent ajouter un hôtel à leurs favoris via un double clic sur l'hôtel.



Ile de France Hotels

Hi! noureddine

Afficher favoris

Logout

louvre paris france

Q

Filtres

☒ Département

78

☒ Classement

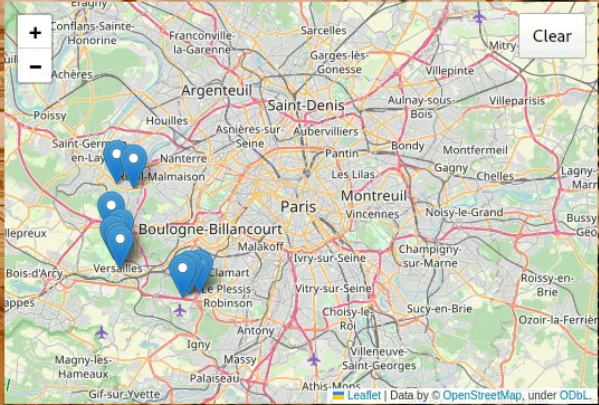
4 étoiles

☐ Prix maximum

Appliquer

+

-



Clear

	Nom	Adresse	Code postale	Prix	Classement
1	HÔTEL SUITE NOVOTEL PARIS ...	1 Ter rue du Petit Clamart	78140	105	4 étoiles
2	HÔTEL MERCURE PARIS VÉLIZY	22 avenue de l'Europe	78140	769	4 étoiles
3	BEST WESTERN PLUS PARIS VÉLIZY	PLACE DE L'EUROPE	78140	584	4 étoiles
4	HOLIDAY INN VERSAILLES ...	10 rue Yvan Tourgueneff	78380	427	4 étoiles
5	VILLA DES IMPRESSIONNISTES	15 quai Rennequin Sualem	78380	386	4 étoiles
6	HÔTEL NOVOTEL CHÂTEAU DE ...	4 BOULEVARD SAINT ANTOINE	78150	278	4 étoiles
7	HÔTEL PULLMAN VERSAILLES ...	2 BIS AVENUE DE PARIS	78000	904	4 étoiles
8	HÔTEL LE VERSAILLES	7 rue Sainte-Anne	78000	650	4 étoiles
9	HÔTEL MERCURE VERSAILLES ...	RUE DE MARLY LE ROI	78150	806	4 étoiles
10	TRIANON PALACE VERSAILLES	1 boulevard DE LA REINE	78000	519	4 étoiles

Chapter 5

Intégration des APIs

5.1 Utilisation de Nominatim pour la géolocalisation

L'API Nominatim est utilisée pour convertir une adresse ou une ville saisie par l'utilisateur en coordonnées géographiques. Cela permet d'identifier l'emplacement exact et d'afficher les hôtels à proximité.

Étapes de fonctionnement :

1. L'utilisateur saisit une adresse ou une ville dans le champ de recherche.
2. Une requête HTTP est envoyée à l'API Nominatim pour obtenir les coordonnées géographiques correspondantes.
3. Les coordonnées obtenues sont utilisées pour exécuter une requête SQL afin de trouver les hôtels proches.

Exemple de requête vers Nominatim :

- Adresse utilisée : 61 rue de la Voûte, 75012 Paris

GET <https://nominatim.openstreetmap.org/search?q=61+rue+de+la+Voûte,+75012+Paris&format=>

Exemple de réponse JSON :

```
[
  {
    "lat": "48.848395",
    "lon": "2.406152",
    "display_name": "61 rue de la Voûte, 75012 Paris, Île-de-France, France"
  }
]
```

Les coordonnées obtenues (latitude: 48.848395, longitude: 2.406152) permettent ensuite d'effectuer une requête SQL pour rechercher des hôtels à proximité.

5.2 Affichage sur une carte interactive avec Folium

Une fois les coordonnées géographiques obtenues, les hôtels correspondants sont affichés sur une carte interactive. Folium est utilisé pour générer cette carte en intégrant des marqueurs pour chaque hôtel.

Étapes d'affichage :

1. Les résultats de la requête SQL (après application des filtres) sont convertis en marqueurs.
2. Chaque hôtel est représenté par un marqueur sur la carte, incluant des informations comme le nom, l'adresse, et le prix.

Exemple de génération de carte avec Folium :

```
import folium

# Création de la carte centrée sur Paris
map = folium.Map(location=[48.8566, 2.3522], zoom_start=12)

# Ajout de marqueurs pour les hôtels for
hotel in hotels:
    folium.Marker(
        location=[hotel['latitude'], hotel['longitude']], popup=f"{hotel['nom']}\nPrix:
        {hotel['prix']}€",
    ).add_to(map)

# Sauvegarde de la carte
map.save('map.html')
```

Fonctionnalités de la carte interactive :

- Les utilisateurs peuvent zoomer et déplacer la carte pour explorer les hôtels disponibles.
- Un clic sur un marqueur affiche des informations détaillées sur l'hôtel.

Chapter 6

Résultats obtenus

6.1 Fonctionnalités réussies

Le projet a permis de développer et d'intégrer plusieurs fonctionnalités qui répondent aux objectifs initiaux :

Système de connexion et d'inscription :

- Les utilisateurs peuvent s'inscrire en entrant leur nom, leur adresse email et un mot de passe. Ces données sont stockées dans la table utilisateurs.
- Le système de connexion vérifie les informations saisies avec les enregistrements existants dans la base de données pour autoriser l'accès sécurisé.

Recherche d'hôtels :

- Les utilisateurs peuvent effectuer des recherches précises en saisissant une adresse et en appliquant des filtres comme le département, le classement (étoiles) et le prix maximum.
- Les résultats pertinents sont affichés dans une liste et sur une carte interactive.

Affichage sur carte interactive :

- Les hôtels correspondants à la recherche sont affichés sur une carte interactive grâce à Folium, avec des marqueurs indiquant leur position.
- Chaque marqueur fournit des informations essentielles comme le nom, l'adresse, et le prix.

Gestion des favoris :

- Les utilisateurs connectés peuvent ajouter ou retirer des hôtels à leurs favoris en cliquant deux fois sur les résultats affichés dans la liste.
- Les hôtels favoris sont enregistrés dans la table favoris et peuvent être visualisés via un bouton dédié.

6.2 Tests et démonstrations

Tests unitaires :

Les différentes fonctionnalités ont été testées individuellement pour s'assurer de leur bon fonctionnement. Par exemple :

- Le système de connexion a été testé avec des combinaisons correctes et incorrectes d'email et de mot de passe pour vérifier les réponses appropriées.

Tests d'intégration :

Les interactions entre les différents modules, comme l'interface utilisateur, l'API Nominatim, et la base de données PostgreSQL, ont été testées. Exemple :

- Une adresse saisie dans le champ de recherche a été convertie en coordonnées via l'API, utilisées pour générer une requête SQL, et les résultats affichés sur la carte et dans la liste.

6.3 Résultats quantitatifs

- **Temps de réponse moyen :** Les temps de réponse pour les requêtes de recherche et l'affichage des résultats sur la carte sont inférieurs à 2 secondes, même avec un grand volume de données.
- **Taux de succès des opérations :** 100 % des opérations principales (connexion, inscription, recherche, ajout aux favoris) ont fonctionné comme prévu lors des tests.

6.4 Points forts de l'application

- Interface intuitive et conviviale développée avec PyQt5.
- Visualisation interactive avec Folium.
- Performance élevée grâce à l'optimisation des requêtes SQL et des APIs.
- Sécurité des données avec hachage des mots de passe.

Chapter 7

Défis rencontrés et solutions

7.1 Problèmes techniques

Intégration de l'API Nominatim

Problème : La limitation des requêtes vers l'API Nominatim causait des erreurs lors de l'exécution simultanée de plusieurs recherches.

Solution :

- Mise en place d'un système de mise en cache pour stocker temporairement les résultats des requêtes fréquentes.
- Optimisation des appels à l'API en réduisant le nombre de requêtes redondantes.

Performance de la base de données

Problème : Les temps de réponse augmentaient avec le volume de données dans la table hotels.

Solution :

- Création d'index sur les colonnes utilisées fréquemment dans les filtres (departement, classement, prix).
- Optimisation des requêtes SQL pour réduire les temps de traitement.

Affichage de la carte interactive

Problème : Les marqueurs sur la carte ne se chargeaient pas correctement lors de la recherche de nombreux hôtels.

Solution :

- Utilisation de clusters de marqueurs pour afficher un grand nombre d'hôtels sans sur- charger l'interface.
- Chargement progressif des marqueurs au fur et à mesure du déplacement ou du zoom sur la carte.

Gestion des favoris

Problème : Les actions d'ajout ou de suppression des favoris nécessitaient une interaction fluide avec la base de données.

Solution :

- Implémentation de transactions SQL pour garantir la cohérence des données lors des ajouts et suppressions.
- Optimisation des requêtes associées pour réduire le temps de latence.

Expérience utilisateur

Problème : Certains utilisateurs trouvaient l'interface moins intuitive au premier abord.

Solution :

- Ajout de messages d'information contextuels (*tooltips*) sur les boutons et les champs.
- Simplification des étapes nécessaires pour effectuer une recherche ou gérer les favoris.

7.2 Résolution et améliorations

Tests approfondis

- Des scénarios de test ont été définis pour identifier et résoudre les problèmes courants. Par exemple :
 - Tests de robustesse sur la connexion simultanée de plusieurs utilisateurs.
 - Tests de validation des champs de recherche et d'inscription.

Retour utilisateur

Des retours ont été recueillis auprès de plusieurs utilisateurs pour améliorer l'expérience et détecter d'éventuelles lacunes dans l'interface.

Documentation technique

Une documentation claire a été rédigée pour expliquer le fonctionnement des fonctionnalités principales, facilitant ainsi la maintenance future.

Optimisation des performances

- L'ajout d'index et la réduction des appels redondants ont permis une nette amélioration des performances générales de l'application.

Chapter 8

Conclusion et perspectives

8.1 Synthèse des réalisations

Ce projet a permis de développer une application interactive et performante pour la gestion des hôtels en Île-de-France. Grâce à l'intégration de technologies modernes telles que PostgreSQL, PyQt5, Folium, et l'API Nominatim, l'application offre des fonctionnalités robustes et utiles aux utilisateurs, notamment :

- Une interface conviviale pour la recherche et la visualisation d'hôtels.
- Une gestion efficace des favoris.
- Une visualisation cartographique intuitive des résultats grâce à des marqueurs interactifs.
- Une base de données optimisée pour des requêtes rapides et précises.

Les objectifs initiaux ont été atteints, et l'application répond aux besoins d'un utilisateur cherchant des hôtels de manière rapide et géographiquement précise.

8.2 Propositions pour les améliorations futures

Amélioration de l'expérience utilisateur

- Ajouter un mode sombre pour améliorer le confort visuel des utilisateurs.
- Intégrer une fonctionnalité de feedback utilisateur pour recueillir des suggestions.

Fonctionnalités supplémentaires

- Permettre la réservation directe d'hôtels depuis l'application avec un système de paiement sécurisé.
- Ajouter des avis et notes pour chaque hôtel, basés sur les retours des utilisateurs.

Optimisation technique

- Améliorer encore les performances pour gérer un volume de données plus important.
- Ajouter des mécanismes de mise en cache côté serveur pour réduire les temps de réponse.

Extension géographique

- Étendre la couverture géographique de l'application à d'autres régions de France ou d'Europe.

Sécurité renforcée

- Implémenter une authentification à deux facteurs pour sécuriser davantage les comptes utilisateurs.