

Rapport développement python et frameworks

3ème Année en Ingénierie Informatique et Réseaux

Application de sondages

Réalisé par :

*Abouelfath Saad
Zairi Yassine
Cherkaoui yassine
Bouchra Ilyas*

Année universitaire : 2024/2025

Table des matières

Table des matières

1. **Introduction**
 - 1.1. Présentation du Projet
 - 1.2. Objectifs pédagogiques
 - 1.3. Contexte et motivation
2. **Analyse des besoins**
 - 2.1. Description fonctionnelle
 - 2.2. Acteurs de l'application
 - 2.3. Contraintes techniques et fonctionnelles
 - 2.4. Diagrammes UML
3. **Architecture et technologies**
 - 3.1. Architecture de l'application : le modèle MVT
 - 3.2. Schéma global de fonctionnement
 - 3.3. Technologies principales utilisées
4. **Implémentation**
 - 4.1. Gestion des utilisateurs
 - 4.2. Création de sondages
 - 4.3. Diffusion et participation
 - 4.4. Analyse et visualisation des résultats
 - 4.5. Interface d'administration
5. **Captures d'écran de l'application**
 - 5.1. Page d'accueil
 - 5.2. Inscription et connexion
 - 5.3. Tableau de bord utilisateur
 - 5.4. Interface de création de sondage
 - 5.5. Participation à un sondage
 - 5.6. Visualisation des résultats
 - 5.7. Interface d'administration
6. **Résultats et analyse**
 - 6.1. Fonctionnement général de l'application
 - 6.2. Analyse des données collectées
 - 6.3. Feedback utilisateur
 - 6.4. Limites observées
7. **Conclusion et perspectives**
 - 7.1. Bilan du projet
 - 7.2. Difficultés rencontrées
 - 7.3. Perspectives d'évolution

Introduction

1.1 Présentation du projet

Le projet intitulé "**Application de sondages**" s'inscrit dans le cadre d'un apprentissage pratique du développement web avec le framework **Django**. Il a pour objectif la conception d'une application interactive permettant à des utilisateurs de créer, diffuser et analyser des sondages ou questionnaires personnalisés. Cette solution vise à offrir une interface intuitive et responsive, accessible aussi bien aux créateurs de sondages qu'aux participants. Le projet se veut complet, intégrant la gestion des utilisateurs, la logique conditionnelle dans les formulaires, la visualisation des résultats en temps réel et un tableau de bord d'administration.

1.2 Objectifs pédagogiques

Ce projet répond à plusieurs objectifs pédagogiques fixés dans le cadre de l'apprentissage du développement d'applications web complètes avec Django. Il permet notamment de :

- Maîtriser le **modèle MVT** (Modèle - Vue - Template) de Django, qui structure l'ensemble de l'application.
- Implémenter des **formulaires dynamiques** en utilisant les Django Forms pour collecter les réponses des utilisateurs.
- Assurer la **gestion sécurisée des utilisateurs** via Django Authentication, incluant l'inscription, la connexion et la personnalisation des profils.

- Mettre en œuvre des outils d'**analyse de données** pour extraire des statistiques exploitables à partir des réponses récoltées.
- Réaliser des **visualisations graphiques interactives** grâce à l'intégration de Chart.js.
- Développer une interface utilisateur moderne avec **Bootstrap** et du JavaScript pour une meilleure interactivité.

1.3 Contexte et motivation

À l'ère du numérique, les sondages et enquêtes en ligne jouent un rôle crucial dans la prise de décision, que ce soit dans le domaine de l'enseignement, de la recherche, du marketing ou encore dans la gestion des ressources humaines. Disposer d'un outil personnalisé et évolutif pour concevoir ses propres sondages est un atout majeur. Le choix de Django pour le développement de cette application s'explique par sa robustesse, sa facilité de mise en œuvre rapide et sa riche documentation. Par ailleurs, ce projet permet de combiner plusieurs compétences clés : conception logicielle, architecture web, gestion de base de données, UI/UX, et visualisation de données.

Analyse des besoins

2.1 Description fonctionnelle

L'application de sondages vise à répondre à un ensemble de besoins concrets autour de la création, la diffusion et l'analyse de questionnaires en ligne. Elle doit permettre à des utilisateurs authentifiés de créer différents types de sondages comportant des questions à choix unique, multiple, à échelle de valeur ou ouvertes. Ces sondages doivent pouvoir être diffusés via des liens, protégés par mot de passe si nécessaire, et inclure des restrictions de participation telles qu'une soumission unique ou une période de validité.

Les participants, quant à eux, doivent bénéficier d'une interface claire, fluide et responsive, leur permettant de répondre facilement aux questionnaires. Côté administrateur ou créateur, l'application doit fournir un suivi en temps réel des réponses, une interface de gestion des sondages existants, ainsi que des outils d'analyse statistique pour interpréter les résultats collectés.

2.2 Acteurs de l'application

L'analyse des besoins fonctionnels a permis d'identifier les principaux acteurs de l'application :

- **Utilisateur anonyme** : Peut accéder à des sondages publics et y répondre si aucune restriction ne l'en empêche.

- **Utilisateur authentifié** : Peut créer, modifier et consulter ses propres sondages, ainsi que participer à ceux d'autres utilisateurs.
- **Créateur de sondage** : Un utilisateur ayant la possibilité de générer des questionnaires personnalisés, gérer leur diffusion et accéder aux résultats.
- **Administrateur** (*si applicable*) : Peut accéder à un tableau de bord global de l'application, gérer les utilisateurs ou surveiller l'activité globale.

2.3 Contraintes techniques et fonctionnelles

L'application doit être développée avec Django, selon le modèle MVT, et s'appuyer sur une base de données MySQL pour le stockage des utilisateurs, sondages, réponses et statistiques. Le frontend doit utiliser Bootstrap pour assurer la réactivité de l'interface sur tous les types d'écrans. Des bibliothèques comme Chart.js et JavaScript doivent être intégrées pour l'interactivité et la visualisation dynamique des données.

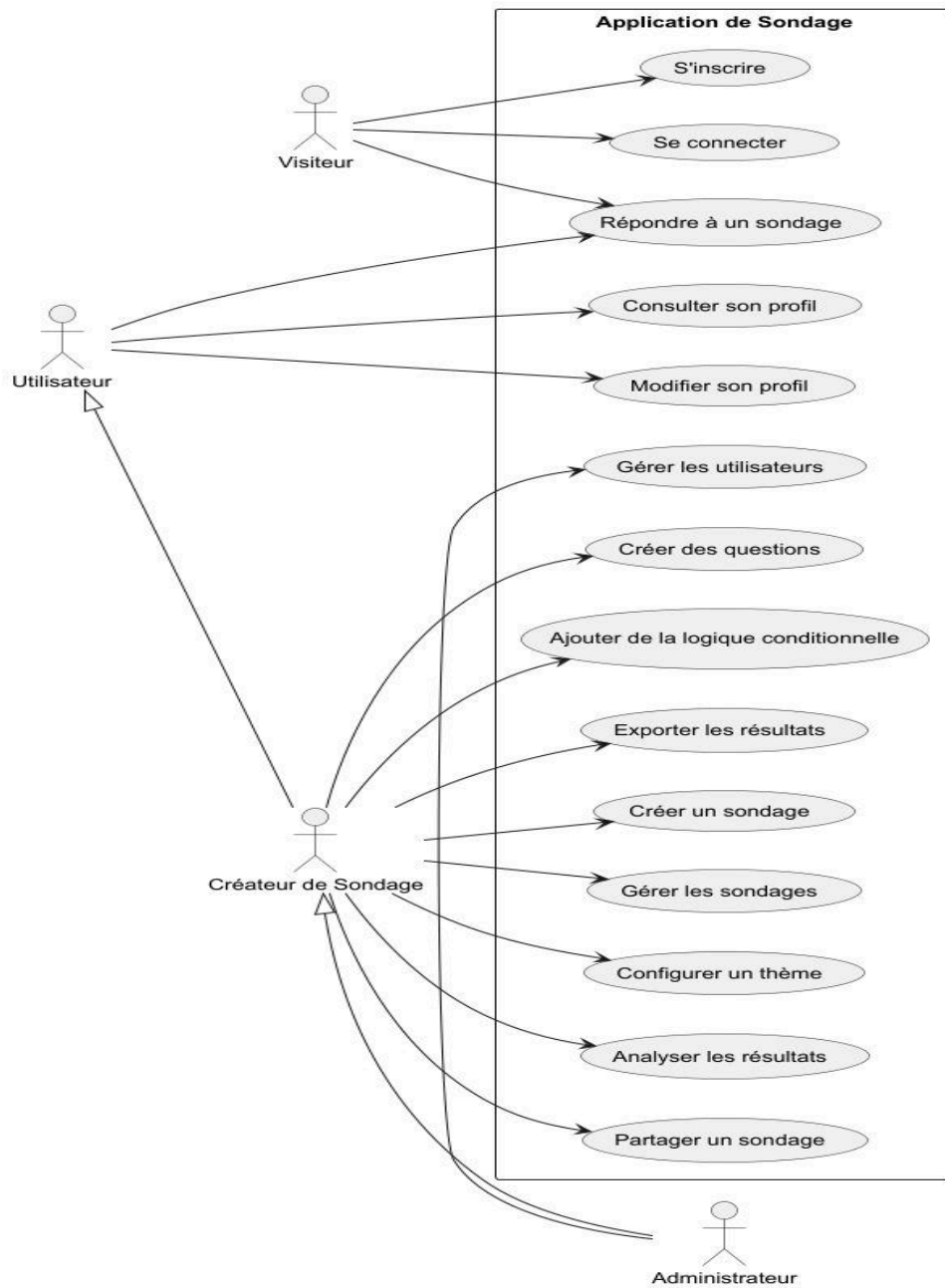
Parmi les contraintes fonctionnelles, on retrouve :

- Une réponse unique par participant (en option)
- L'impossibilité de modifier une réponse après soumission
- Un rendu optimisé pour smartphones et tablettes
- Une logique conditionnelle entre les questions, selon les réponses précédentes

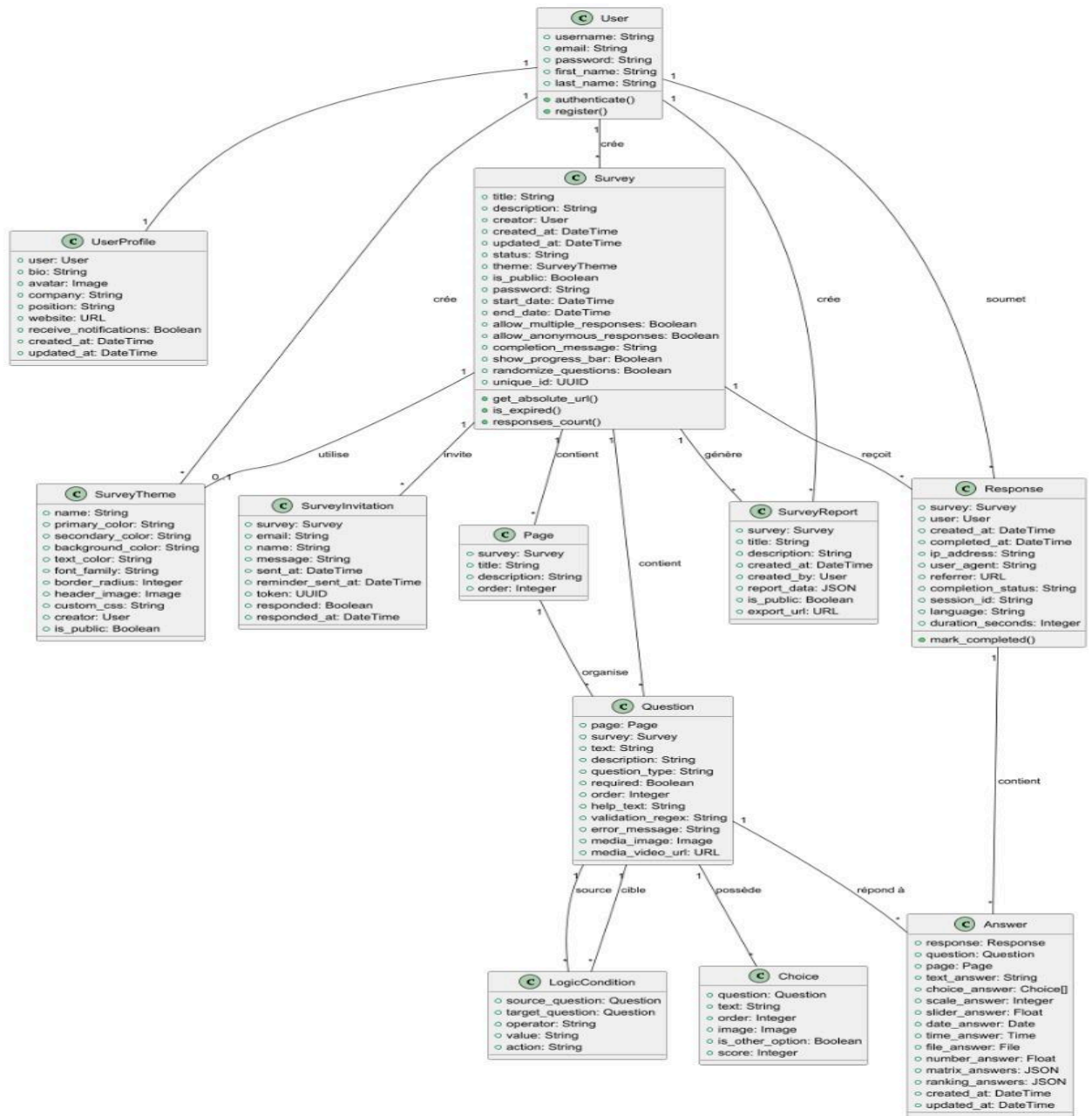
2.4 Diagrammes UML

Les diagrammes suivants permettent de représenter les principaux aspects techniques et fonctionnels de l'application :

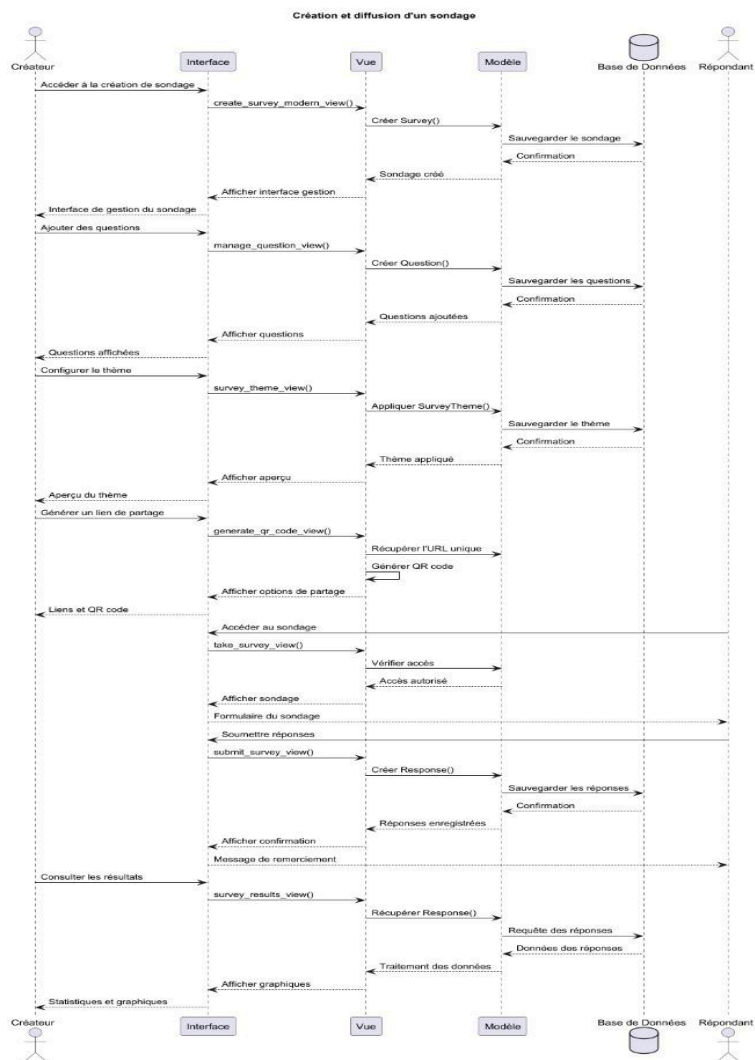
- Diagramme de cas d'utilisation



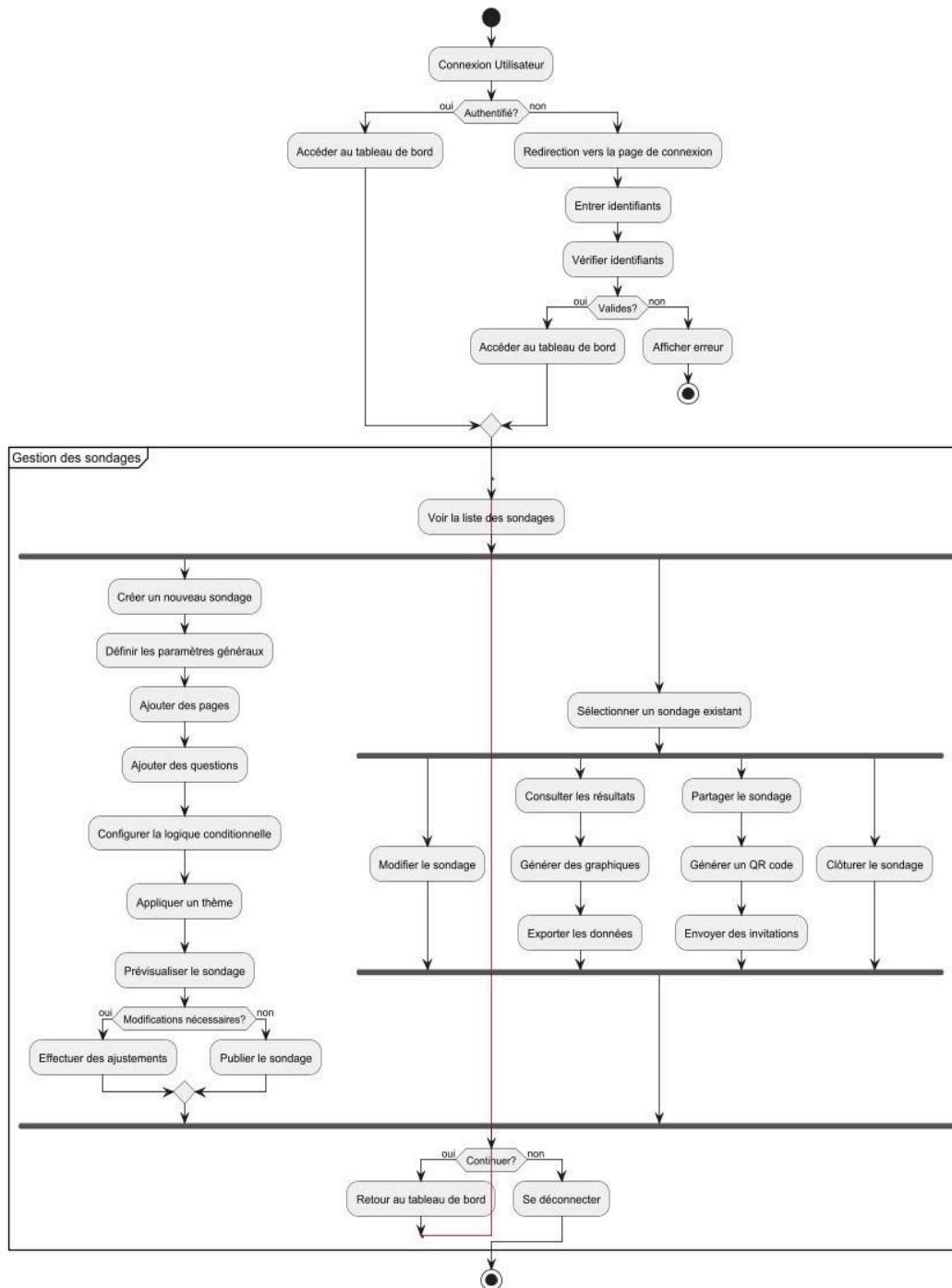
- Diagramme de classes



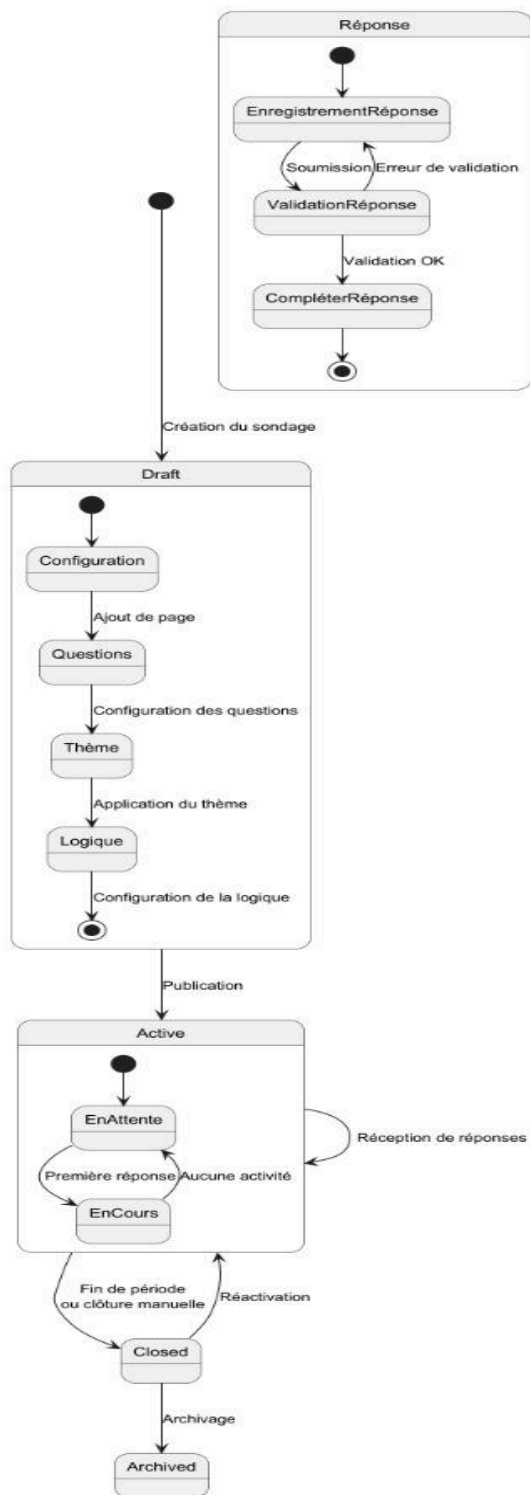
- Diagramme de séquence



- Diagramme d'activités



- Diagramme états-transitions



Architecture et technologies

3.1 Architecture de l'application : le modèle MVT

L'application repose sur l'architecture **MVT (Modèle - Vue - Template)** proposée par le framework Django. Ce modèle structure de manière claire le développement web en trois couches :

- **Modèle (Model)** : Définit la structure des données et leur représentation dans la base de données à l'aide de classes Python. Dans le cas présent, les modèles représentent les utilisateurs, les sondages, les questions, les choix de réponse et les réponses collectées.
- **Vue (View)** : Gère la logique métier. Les vues traitent les requêtes des utilisateurs, interagissent avec les modèles, et transmettent les données nécessaires aux templates pour l'affichage.
- **Template** : Gère la présentation. Les templates HTML, enrichis par le moteur de template de Django, permettent de générer dynamiquement les pages en fonction des données fournies par les vues.

Cette séparation des responsabilités facilite la maintenance, l'évolutivité et le test de l'application.

3.2 Schéma global de fonctionnement

L'utilisateur interagit avec l'interface via un navigateur. Lorsqu'il envoie une requête (ex. : création d'un sondage ou soumission de réponses), celle-ci est traitée par une vue Django. La vue interagit avec les modèles (en lecture ou écriture), puis transmet les données nécessaires à un template HTML. Ce dernier est rendu et renvoyé comme réponse au client.

3.3 Technologies principales utilisées

Django

Django est le **framework principal** utilisé pour ce projet. Il offre une structure robuste, une sécurité intégrée, une gestion efficace des sessions et une forte extensibilité. Il simplifie également la création de formulaires dynamiques et la gestion des utilisateurs.

Base de données MySQL

Les données sont stockées dans une base **MySQL**, reconnue pour sa performance et sa compatibilité avec Django via l'ORM (Object-Relational Mapping). Les relations entre les entités (utilisateur → sondage → questions → réponses) sont modélisées avec des clés étrangères.

Django Forms

Les **Django Forms** ont été largement utilisés pour la création des interfaces de saisie, notamment lors de la création des sondages, la participation et la validation des données en entrée. Ils permettent de créer des formulaires robustes avec des contrôles de validation côté serveur.

Django Authentication

La gestion des utilisateurs (inscription, connexion, sessions, permissions) repose sur le système d'**authentification intégré de Django**. Il a été personnalisé pour inclure un historique des participations, un profil utilisateur et des rôles (participant, créateur).

Bootstrap

Le **framework CSS Bootstrap** a été utilisé pour concevoir une interface responsive et moderne. Il permet un rendu cohérent sur différents supports (ordinateur, tablette, smartphone) avec un minimum d'effort de mise en page.

Chart.js

Pour la **visualisation graphique des résultats**, la bibliothèque **Chart.js** a été intégrée. Elle permet de générer dynamiquement des graphiques interactifs (camemberts, histogrammes, etc.) à partir des réponses collectées.

JavaScript

Le **JavaScript** natif a été utilisé pour améliorer l'interactivité, notamment pour les comportements conditionnels dans les formulaires (ex. : affichage de certaines questions en fonction des réponses précédentes).

Django REST Framework (*optionnel*)

Une réflexion a été menée autour de l'intégration de **Django REST Framework** pour exposer certaines fonctionnalités de l'application sous forme d'API RESTful. Cela permettrait à terme une ouverture vers des applications mobiles ou des systèmes tiers.

Implémentation

4.1 Gestion des utilisateurs

La gestion des utilisateurs repose sur le système d'authentification fourni par Django. Deux types d'utilisateurs sont distingués : les participants et les créateurs de sondages. Chaque utilisateur peut s'inscrire via un formulaire personnalisé, se connecter, et accéder à un tableau de bord contenant son historique de participations ou de créations.

Le profil utilisateur inclut des informations telles que l'adresse email, la date d'inscription, ainsi que la liste des sondages créés ou complétés. Un système de permissions pourrait être étendu afin de différencier les rôles ou restreindre certaines actions.

4.2 Création de sondages

L'interface de création de sondages a été conçue pour être simple, dynamique et flexible. Un créateur peut :

- Ajouter un **titre** et une **description** à son sondage.
- Définir plusieurs types de **questions** : à choix unique, à choix multiple, à échelle de valeur (ex. : 1 à 5), ou question ouverte (champ texte).
- Appliquer une **logique conditionnelle** : certaines questions peuvent apparaître uniquement si une réponse précise a été donnée à une question antérieure.
- **Personnaliser** la présentation (thème de couleur, disposition, etc.).

Chaque sondage est sauvegardé avec un identifiant unique, qui permet d'y accéder via un lien.

4.3 Diffusion et participation

Une fois un sondage créé, il peut être partagé grâce à un **lien public** ou restreint par :

- Un **mot de passe** à fournir avant de répondre,
- Une **limite de participation** (ex. : un seul enregistrement par utilisateur ou par adresse IP),
- Une **durée de validité** configurable (date de début et de fin).

L'interface de participation est responsive et intuitive. Chaque question est affichée selon le type défini et la progression est guidée question par question. Une fois le questionnaire complété, la réponse est enregistrée de manière sécurisée dans la base de données.

4.4 Analyse et visualisation des résultats

Dès qu'un sondage commence à recevoir des réponses, les résultats sont disponibles sous forme de :

- **Graphiques en temps réel**, générés via Chart.js, représentant la répartition des réponses (ex. : camemberts pour les choix multiples, barres pour les échelles, etc.).
- **Statistiques détaillées** par question : nombre de participants, réponses dominantes, moyenne, écart-type selon le type de données.
- **Filtres** permettant d'analyser les réponses selon des critères définis (date, profil utilisateur, etc.).
- **Exportation** des réponses sous format CSV ou Excel, pour exploitation externe.

4.5 Interface d'administration

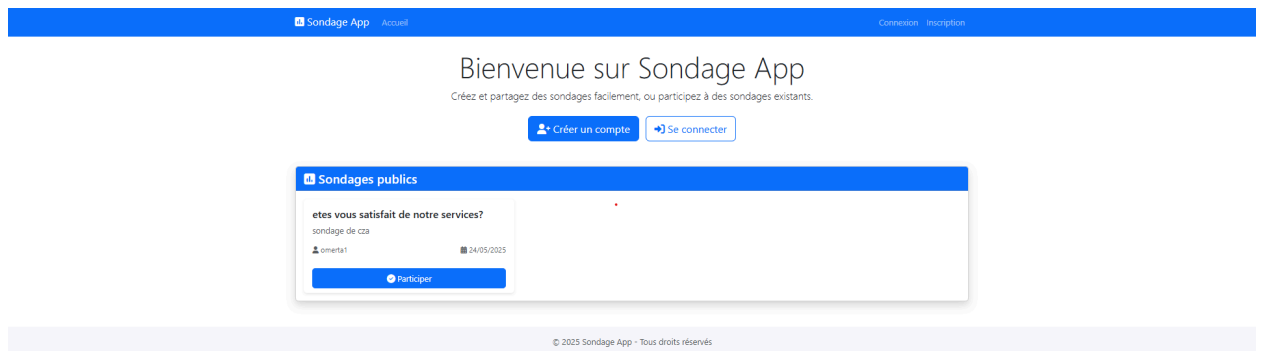
Une interface dédiée permet aux créateurs de gérer leurs sondages :

- **Tableau de bord** listant les sondages actifs, archivés ou en cours de création.
- Possibilité de **dupliquer un sondage** pour en créer une nouvelle version sans repartir de zéro.
- **Modification des paramètres** d'un sondage même après sa publication (tant qu'aucune réponse n'a été collectée).
- **Archivage** automatique des anciens sondages afin de conserver l'historique tout en évitant la surcharge visuelle.
- Accès à un ensemble de **modèles réutilisables** : types de sondages fréquemment utilisés pouvant être réemployés par d'autres créateurs.

Captures d'écran de l'application

Cette section présente les principales interfaces de l'application de sondages. Les captures d'écran sont accompagnées de brèves descriptions afin d'illustrer l'ergonomie, les fonctionnalités mises en place et l'expérience utilisateur.

5.1 Page d'accueil



La page d'accueil introduit l'application et propose des accès directs à la connexion ou à la création de compte. Elle est conçue de manière responsive avec Bootstrap, pour une navigation fluide sur tout type d'appareil.

5.2 Inscription et connexion

Sondage App

Accueil

Connexion

Inscription

Connexion

Nom d'utilisateur

Mot de passe

Se connecter

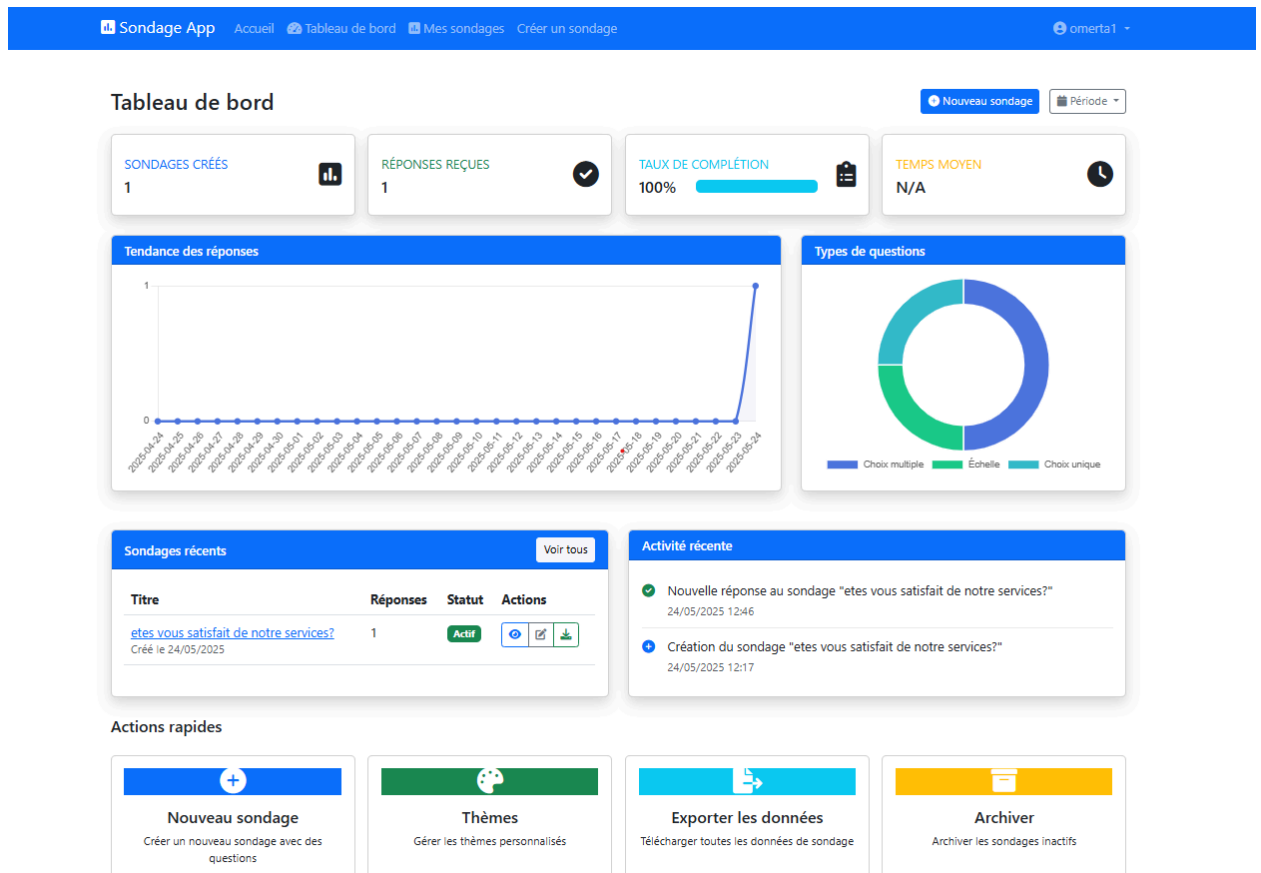
Vous n'avez pas de compte ?

Inscrivez-vous

© 2025 Sondage App - Tous droits réservés

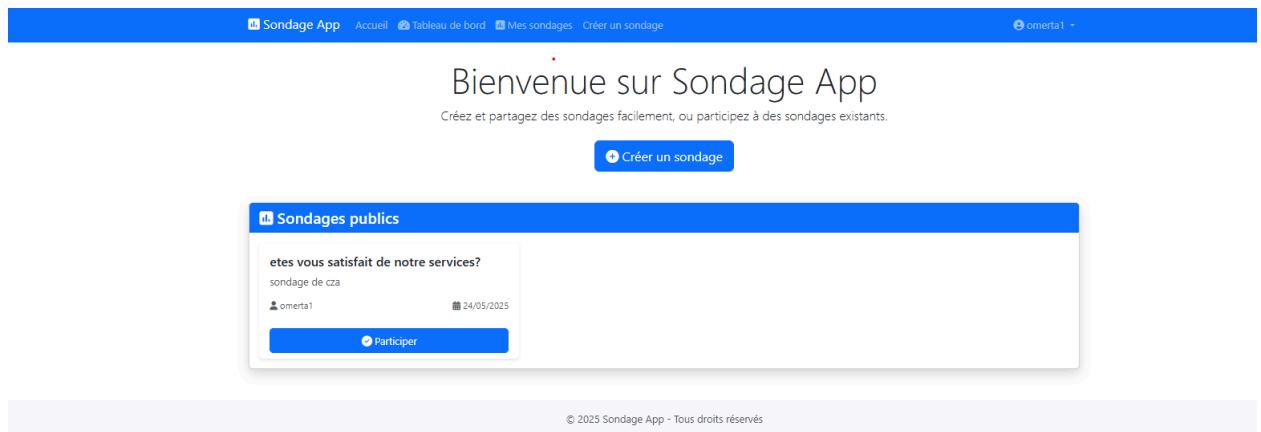
Les formulaires d’inscription et de connexion utilisent les Django Forms personnalisés avec des règles de validation intégrées. En cas d’erreur (mot de passe faible, utilisateur déjà existant), des messages d’alerte sont affichés.

5.3 Tableau de bord utilisateur



Après authentification, l'utilisateur accède à un tableau de bord personnel. Il peut y consulter ses sondages créés, ceux auxquels il a répondu, et lancer la création d'un nouveau sondage.

5.4 Interface de création de sondage

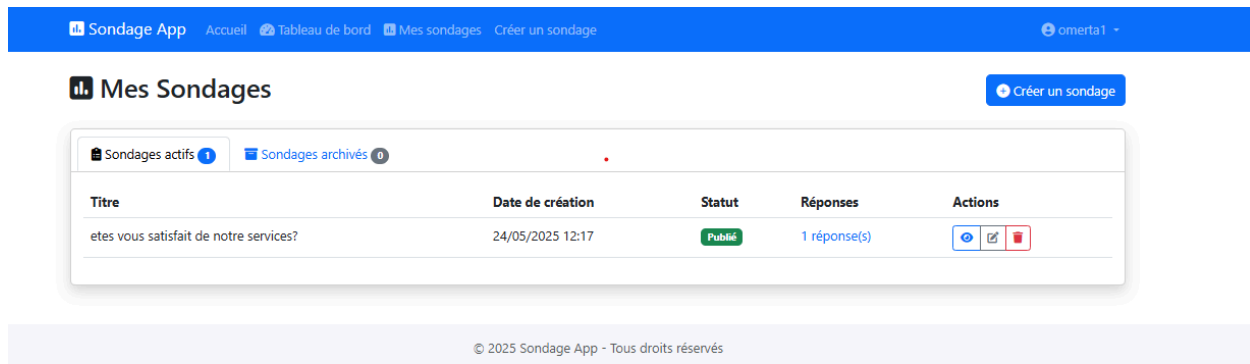


Le créateur peut définir les caractéristiques de son sondage via un formulaire dynamique. Il ajoute des questions, choisit leur type, configure la logique conditionnelle, et peut prévisualiser le rendu avant publication.

5.5 Participation à un sondage

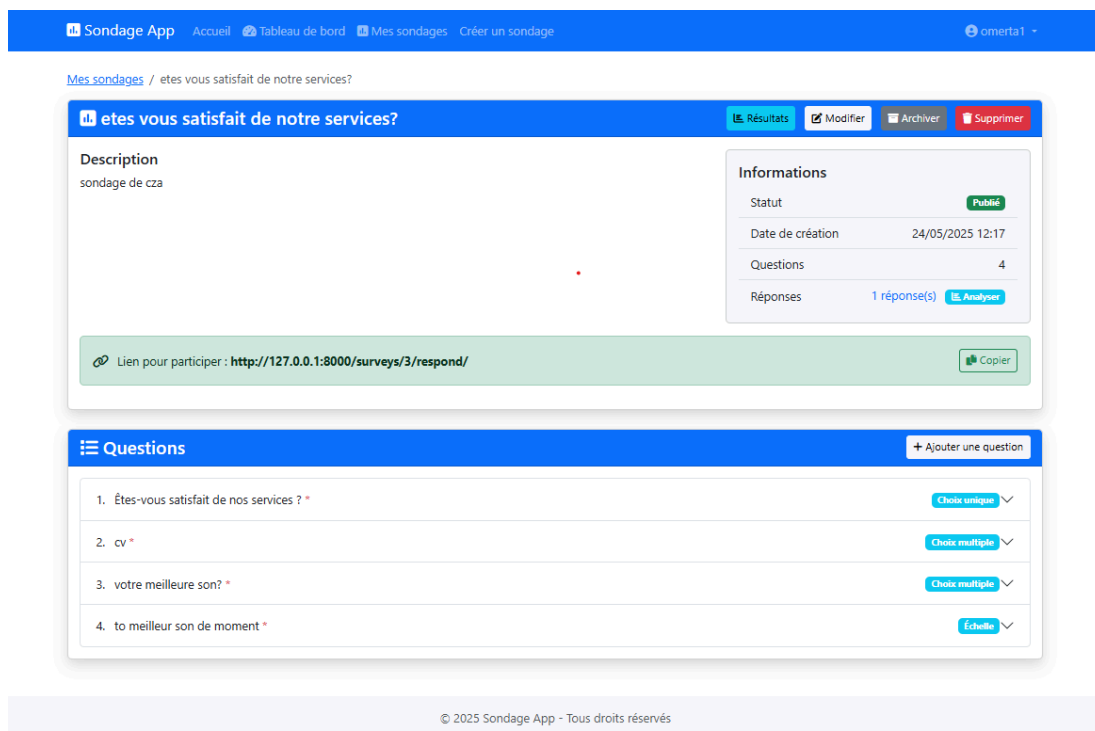
Cette interface, pensée pour la simplicité, permet aux participants de répondre rapidement et efficacement. L'affichage des questions s'adapte selon les types choisis et les réponses données.

5.6 Visualisation des résultats



Les résultats sont affichés en temps réel sous forme de graphiques interactifs (camemberts, barres, lignes). Ces visualisations permettent une compréhension rapide des tendances. Un tableau synthétique complète l'analyse.

5.7 Interface d'administration



Accessible aux créateurs, cette interface permet de gérer les sondages existants (modification, duplication, archivage) et d'accéder aux options avancées comme l'exportation des résultats ou l'activation de restrictions.

Résultats et analyse

6.1 Fonctionnement général de l'application

L'application a été testée avec plusieurs scénarios de création et de participation à des sondages. L'ensemble des fonctionnalités principales — création de questionnaires, logique conditionnelle, participation, affichage des résultats — fonctionne correctement. Le système assure la cohérence des données, empêche les doublons lors de la participation, et gère la sécurité via le système d'authentification Django.

Les utilisateurs peuvent naviguer de manière fluide entre les différentes sections. Les tests ont montré que l'interface est suffisamment intuitive pour être utilisée sans tutoriel explicite.

6.2 Analyse des données collectées

Les résultats des sondages sont affichés de manière visuelle à l'aide de **graphiques générés par Chart.js**. Chaque type de question bénéficie d'une représentation adaptée :

- **Choix unique** : camemberts ou barres horizontales.
- **Choix multiple** : histogrammes groupés.
- **Échelles** : courbes ou barres empilées.
- **Réponses ouvertes** : tableau de texte brut ou export CSV.

Des filtres permettent de cibler l'analyse sur une plage de dates ou un sous-ensemble de participants. Cela rend l'interprétation des résultats plus fine et pertinente, notamment

dans les contextes professionnels (ressources humaines, études de marché) ou pédagogiques (feedback de cours, auto-évaluation).

6.3 Feedback utilisateur

Un retour qualitatif a été recueilli auprès de plusieurs utilisateurs tests :

- Les créateurs ont apprécié la simplicité de la création de sondages et la diversité des types de questions.
- Les participants ont souligné la clarté de l'interface et la rapidité de réponse.
- Certains ont suggéré d'ajouter un système de notifications par email ou une fonctionnalité de rappel.

Ces retours ouvrent la voie à des améliorations futures (voir Partie Conclusion).

6.4 Limites observées

Malgré les résultats globalement positifs, quelques limites techniques ont été identifiées :

- La logique conditionnelle devient complexe à gérer pour les très longs sondages.
- Les réponses ouvertes ne sont pas encore analysées automatiquement (pas de traitement de texte libre).
- Le support de la participation anonyme est limité si la restriction par session/IP est activée.
- L'exportation en Excel est parfois lente lorsque le volume de réponses est élevé.

Ces limites sont dues à des choix techniques initiaux ou au périmètre de développement défini, mais elles peuvent être corrigées ou étendues dans de futures versions.

Conclusion et perspectives

7.1 Bilan du projet

Le développement de cette application de sondages a permis de mettre en œuvre de manière concrète les compétences acquises en programmation web avec **Django**, en suivant une architecture **MVT** bien structurée. Le projet a couvert l'ensemble du cycle de vie d'une application web : de la modélisation des besoins à la mise en ligne d'une interface fonctionnelle.

Les objectifs pédagogiques ont été atteints :

- Maîtrise des **modèles, vues et templates** avec Django,
- Création de **formulaires dynamiques** avec logique conditionnelle,
- Intégration de **visualisations graphiques** en temps réel,
- Mise en place d'un **système d'authentification sécurisé**,
- Stockage et gestion des données via **MySQL**,
- Utilisation de **Bootstrap** et **JavaScript** pour une interface moderne et responsive.

Le projet s'est révélé à la fois formateur et stimulant, en impliquant des problématiques concrètes de développement web et d'expérience utilisateur.

7.2 Difficultés rencontrées

Quelques défis techniques ont jalonné la réalisation :

- L'implémentation de la logique conditionnelle entre questions a nécessité une réflexion avancée sur le traitement dynamique des formulaires.
- La génération de graphiques en temps réel impliquait une bonne coordination entre la collecte des données, leur agrégation, et leur restitution graphique.
- L'adaptation de l'interface aux mobiles a nécessité des ajustements CSS avec Bootstrap pour garantir un bon rendu sur tous les supports.

Ces obstacles ont été surmontés grâce à une veille technique constante, à l'expérimentation, et à la documentation officielle de Django et des bibliothèques utilisées.

7.3 Perspectives d'évolution

Plusieurs pistes peuvent être envisagées pour enrichir et pérenniser l'application :

- **Ajout d'un éditeur de sondage en glisser-déposer (drag-and-drop)** pour simplifier encore la création de questionnaires.
- **Mise en place d'une API REST** avec Django REST Framework pour permettre la communication avec des applications mobiles ou tierces.
- **Système de notification par email** à destination des participants (invitation, rappel, confirmation).
- **Analyse automatique des réponses ouvertes** via des techniques de traitement du langage naturel (NLP).
- **Amélioration de l'administration centrale**, avec possibilité de regrouper les résultats par campagne, session, ou groupe de participants.

Ces évolutions permettraient d'ouvrir l'application à un public plus large, d'enrichir l'expérience utilisateur, et de la rendre exploitable dans des contextes professionnels, académiques ou commerciaux.

