



الكلية متعددة التخصصات - وورزازات
+oX+€M€+ - Loo%o%+
FACULTÉ POLYDISCIPLINAIRE DE OUARZAZATE



الكلية متعددة التخصصات - وورزازات
+oX+€M€+ - Loo%o%+
FACULTÉ POLYDISCIPLINAIRE DE OUARZAZATE

Université Ibn Zohr
Faculté Polydisciplinaire de
Ouarzazate
Filière Master : Intelligence
Artificielle et Applications

Développement d'une application
Système management qualité

Projet Réalisé par :

Hanan Bassou
Yasser El mouhib
Maryame Khouya
Abdelkrim Boussag

Encadré par :

Pr. M. BENADDY
Professeure à la Faculté
Polydisciplinaire de Ouarzazate

Devant le jury :

Pr. A. HADRI Professeure à la Faculté Polydisciplinaire de Ouarzazate

Pr. A. ELLAHYANI Professeure à la Faculté Polydisciplinaire de Ouarzazate

Présenter le : 8 septembre 2025

Année universitaire : 2024-2025

Table des matières

Remerciement	2
Résumé	3
Abstract	4
Introduction générale	5
1 Système de management de la qualité	6
1.1 Généralités sur la Qualité	6
1.2 Le système de management de la qualité (SMQ)	7
2 La Méthode DevOps	9
2.1 Définition	9
2.2 Principes fondamentaux	9
2.3 Outils et technologies DevOps	10
2.4 Avantages de DevOps	11
2.5 Caractéristiques innovantes de DevOps	11
2.6 Application dans ce projet	12
3 Réalisation de l'Application	13
3.1 Technologies utilisées	13
3.2 Architecture générale	15
3.3 Fonctionnalités principales	15
3.4 Organisation et acteurs	16
3.5 Visualisation de l'application	16
3.6 Visualisation de résultat	20
Conclusion générale	22
Références	23

Remerciement

Tout d'abord, nous remercions Dieu Tout-Puissant pour les privilèges qu'Il nous a accordés : l'opportunité d'étudier, ainsi que les capacités, les compétences et la patience qu'Il nous a données pour mener à bien ce travail.

*Nous exprimons également notre profonde gratitude à notre superviseur, le Docteur **M. BENADDY**, pour sa direction, ses conseils précieux et son accompagnement tout au long de ce projet. Nous remercions aussi chaleureusement les deux professeurs **Pr. A. HADRI** et **Pr. A. ELLAHYANI** pour leur présence parmi nous aujourd'hui, ainsi que pour leurs conseils et leurs remarques enrichissantes.*

Nous n'oublions pas non plus de témoigner notre reconnaissance à tous nos enseignants, à chaque étape de notre parcours. Sans leur dévouement et la transmission de leur savoir, nous ne serions pas arrivés là où nous en sommes aujourd'hui.

Résumé

Dans un contexte où la concurrence est de plus en plus forte, la qualité est devenue un facteur essentiel pour assurer la réussite durable des entreprises. Pour améliorer continuellement leurs processus, les organisations doivent mettre en place des systèmes de management de la qualité (SMQ) efficaces. Ces systèmes permettent une meilleure organisation, une traçabilité des documents, une conformité aux normes (comme l'ISO 9001), et une amélioration constante des performances.

Afin de faciliter cette gestion, une application web de gestion de la qualité a été développée dans ce projet à l'aide du framework Flask. L'application permet de gérer les domaines, les champs, les références normatives et les fiches documentaires (ou critères). Elle propose également des fonctionnalités d'export des données au format PDF, le tout dans une interface simple, responsive et accessible.

Le projet s'appuie aussi sur la méthode DevOps, qui favorise l'automatisation des processus de développement et de déploiement. Grâce à des outils comme GitHub, Docker, ou encore CI/CD, le code est facilement maintenu et mis à jour, garantissant une meilleure fiabilité et un gain de temps. Techniquement, l'application utilise Flask (Python) pour le backend, aussi HTML5, CSS3, et Bootstrap pour le frontend, en fin MySQL comme système de base de données.

Les mots clés :

Qualité, SMQ, traçabilité, conformité, ISO 9001, Application web, Flask, fiches documentaires, export PDF, Interface responsive, DevOps, automatisation, CI/CD, GitHub, Docker, MySQL.

Abstract

In a context where competition is becoming increasingly fierce, quality has become an essential factor for ensuring the sustainable success of businesses. To continuously improve their processes, organizations must implement effective quality management systems (QMS). These systems allow for better organization, document traceability, compliance with standards (such as ISO 9001), and ongoing performance improvement.

In order to facilitate this management, a web application for quality management has been developed in this project using the Flask framework. The application allows for the management of domains, fields, normative references, and documentation sheets (or criteria). It also offers data export functionalities in PDF format, all within a simple, responsive, and accessible interface.

The project also relies on the DevOps method, which promotes the automation of development and deployment processes. Thanks to tools like GitHub, Docker, and CI/CD, the code is easily maintained and updated, ensuring better reliability and time savings. Technically, the application uses Flask (Python) for the backend, as well as HTML5, CSS3, and Bootstrap for the frontend, and finally MySQL as the database system.

Keywords :

Quality, QMS, traceability, compliance, ISO 9001, web application, Flask, documentation sheets, PDF export, responsive interface, DevOps, automation, CI/CD, GitHub, Docker, MySQL.

Introduction générale

1. Contexte général

Comment prospérer dans un monde où la concurrence est toujours plus féroce ?

Certaines entreprises et établissements sont dirigées par des leaders charismatiques, d'autres s'appuient sur le pouvoir du collectif. Or, il existe un ingrédient indispensable pour une réussite saine et pérenne, et cela vaut pour l'épicerie au coin de la rue comme pour les grandes entreprises ou établissement : la qualité.

C'est pourquoi une gestion de la qualité efficace est désormais un impératif pour la réussite de toute entreprise. À l'ère de l'innovation et face à l'évolution rapide des attentes, l'adoption d'une démarche axée sur l'amélioration continue pour rester dans la course s'impose. Et pour atteindre cet objectif, il faut s'appuyer sur des systèmes de management de la qualité robustes.

Un système de management de la qualité (SMQ) efficace permet de répondre à tout moment aux attentes des consommateurs et de fournir des produits et services en évitant au maximum le rebut. Dans une économie mondiale hautement compétitive, la mise en place d'un SMQ est la condition sine qua non du succès sur le long terme.

Afin de mieux gérer les documents et processus qualité, il est nécessaire d'utiliser des outils numériques adaptés.

2. Objectif de l'Application

Ce projet a pour objectif de développer une application web de gestion de la qualité en utilisant le framework Flask. Cette application permet la gestion des domaines, champs, références et fiches, tout en offrant des fonctionnalités d'import et d'export.

L'objectif de l'application est de faciliter la gestion des différents éléments liés à la qualité dans une organisation. Elle permet notamment :

- La gestion des **domaines** de qualité
- La gestion des **champs** associés
- La gestion des **références normatives**
- La gestion des **fiches documentaires**(Critères)
- l'exportation des données (au format PDF)

Cette approche centralisée assure une meilleure traçabilité et organisation des informations.

Chapitre 1

Système de management de la qualité

Introduction

Dans cette chapitre, nous allons définir en générale le concept de la qualité, après nous voyons ce qu'est un système de management de la qualité (SMQ), présenter les principes fondamentaux sur lesquels il repose, expliquer son importance pour les entreprises ou établissements, ainsi que les conditions de sa mise en œuvre réussie. Enfin, nous aborderons les avantages qu'il offre aussi bien aux entreprises qu'à leurs clients, et les principales normes internationales qui encadrent son application.

1.1 Généralités sur la Qualité

a. Définition

La qualité peut être définie comme l'ensemble des caractéristiques d'un produit ou d'un service qui lui confèrent l'aptitude à satisfaire les besoins exprimés ou implicites des clients. Elle concerne à la fois la conformité aux exigences, la fiabilité, la performance, la sécurité et la satisfaction du client.

b. Évolution de la qualité

Historiquement, la qualité a évolué du simple contrôle des produits finis vers une approche plus globale et préventive. De la qualité de fabrication (inspection) à la qualité totale (implication de toute l'organisation), on est passé à un véritable management stratégique de la qualité, intégré à tous les niveaux de l'entreprise.

c. Importance de la qualité

Une bonne gestion de la qualité permet :

- La satisfaction des clients
- La réduction des coûts liés aux non-conformités
- Une meilleure organisation interne
- Le respect des normes internationales (ISO 9001, etc.)

1.2 Le système de management de la qualité (SMQ)

a. Définition du SMQ

Un **système de management de la qualité (SMQ)**, ou système de gestion de la qualité (SGQ), désigne l'ensemble des activités coordonnées permettant à une organisation de définir, planifier, mettre en œuvre et améliorer en continu sa politique qualité et ses objectifs, en cohérence avec sa stratégie globale.

Un SMQ repose sur des processus corrélés et interactifs, mobilisant des ressources pour produire des résultats attendus (produits ou services), tout en créant de la valeur pour les clients et les parties prenantes.

Il comprend généralement :

- Un système de **documentation** (procédures, instructions, modes opératoires) ;
- Un système de **vérification** (audits internes, contrôles qualité) ;
- Un système **d'évaluation** (revue de direction, indicateurs qualité).

Plus concrètement, le SMQ permet de :

- Assurer des résultats **cohérents** et **standardisés** ;
- **Réduire les erreurs** et d'augmenter l'**efficacité opérationnelle** ;
- Améliorer la **satisfaction des clients** grâce à la qualité des produits/services ;
- Favoriser l'**amélioration continue** et la **conformité réglementaire**.

Chaque entreprise adapte son SMQ à son contexte spécifique, en tenant compte de sa taille, de son secteur, de ses enjeux et de ses objectifs stratégiques.

b. Les principes fondamentaux du SMQ

Selon la norme **ISO 9000 :2015**, la mise en place d'un SMQ repose sur **sept principes fondamentaux** :

1. **Orientation client** : comprendre et répondre aux besoins actuels et futurs des clients.
2. **Leadership** : établir une vision claire, aligner les objectifs, inspirer et motiver.
3. **Implication du personnel** : valoriser les compétences, l'engagement et la responsabilisation des collaborateurs.
4. **Approche processus** : structurer les activités comme un système de processus interdépendants.
5. **Amélioration continue** : rechercher constamment des opportunités de progrès.
6. **Prise de décision fondée sur des preuves** : s'appuyer sur des données fiables pour prendre des décisions efficaces.
7. **Management des relations avec les parties intéressées** : instaurer des relations bénéfiques et durables avec les parties prenantes.

c. Objectifs et avantages du SMQ

La mise en place d'un SMQ vise plusieurs objectifs clés :

- **Améliorer la performance globale** de l'organisation ;
- **Réduire les non-conformités et les pertes** grâce à une meilleure maîtrise des processus ;
- **Identifier les risques et opportunités** pour anticiper et corriger les défaillances ;
- **Garantir la traçabilité**, la conformité réglementaire et la transparence documentaire ;
- **Renforcer la crédibilité** et la satisfaction client.

Un SMQ bien conçu peut également :

- Centraliser la gestion documentaire (procédures, enregistrements, audits, CAPA, etc.) ;
- Faciliter la traçabilité des actions correctives et préventives ;
- Piloter efficacement les indicateurs qualité ;
- Impliquer toutes les parties prenantes dans une démarche d'amélioration continue.

En somme, un SMQ performant permet à une entreprise de :

- Assurer une **qualité constante** ;
- Gagner en **efficacité opérationnelle** ;
- **Fidéliser ses clients** en leur offrant une expérience cohérente et satisfaisante ;
- Se conformer aux **normes internationales** telles que l'**ISO 9001**.

Conclusion

Le système de management de la qualité (SMQ) constitue un levier essentiel pour garantir la performance, la conformité et la satisfaction client au sein des organisations. Sa mise en œuvre avec ses principes solides, adaptée au contexte de chaque entreprise ou établissement, contribue directement à sa crédibilité, notamment à travers le respect des normes internationales comme l'ISO 9001.

Chapitre 2

La Méthode DevOps

Introduction

Dans un contexte de développement logiciel de plus en plus exigeant, la rapidité, la fiabilité et la collaboration entre équipes sont devenues essentielles. La méthode DevOps répond à ces enjeux en combinant développement et opérations dans une approche intégrée. Ce chapitre présente les principes fondamentaux de DevOps, les outils associés, ainsi que ses bénéfices pour les projets modernes, en particulier dans le cadre du développement d'applications web.

2.1 Définition

Que signifie le terme **DevOps**[1] ?

Le mot *DevOps* est issu de la contraction de **development** (développement) et **operations** (exploitation). Cette méthodologie vise à renforcer la collaboration entre les équipes de développement logiciel et celles en charge des opérations informatiques.

DevOps cherche à instaurer une culture et un environnement dans lesquels la conception, les tests et la mise en production de logiciels peuvent être réalisés de manière rapide, fréquente et fiable. Il ne s'agit pas uniquement d'une méthode, mais d'une véritable philosophie de travail axée sur l'agilité, l'automatisation et l'amélioration continue.

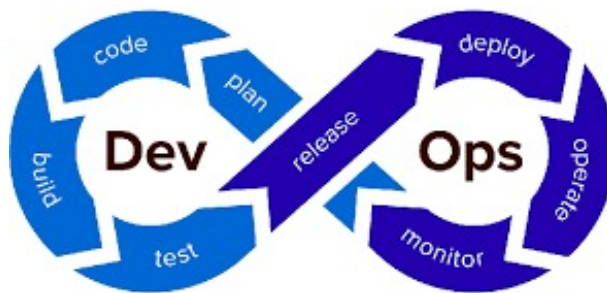


FIGURE 2.1 – La méthode DevOps

2.2 Principes fondamentaux

Les principaux piliers de DevOps sont les suivants :

- **Intégration continue (CI)** : fusion régulière du code dans un référentiel partagé, avec exécution de tests automatisés pour détecter rapidement les erreurs.
- **Livraison continue (CD)** : déploiement automatique et fréquent du logiciel en production, avec un minimum d'intervention humaine.
- **Infrastructure as Code (IaC)** : automatisation de la gestion et de la configuration des serveurs et ressources informatiques à l'aide de code.
- **Surveillance et feedback** : supervision en temps réel des systèmes pour détecter les anomalies et améliorer les performances de manière proactive.

2.3 Outils et technologies DevOps

DevOps repose sur un ensemble d'outils facilitant la collaboration entre les équipes de développement et d'opérations, tout en automatisant les processus et en assurant une surveillance continue des systèmes.

1. Automatisation et gestion du code

- GitHub : plateforme de gestion de code source basée sur Git. Elle facilite le travail collaboratif grâce aux pull requests, branches, et actions GitHub pour l'automatisation des workflows (CI/CD).
- Jenkins, Ansible, Puppet : outils d'automatisation pour le déploiement, la configuration et les tests.

2. Conteneurisation

- Docker : permet d'empaqueter les applications avec toutes leurs dépendances.
- Kubernetes : orchestre et gère les conteneurs à grande échelle.

3. Monitoring

- Prometheus : collecte des métriques système et applicatives.
- Grafana : crée des tableaux de bord visuels et gère les alertes.

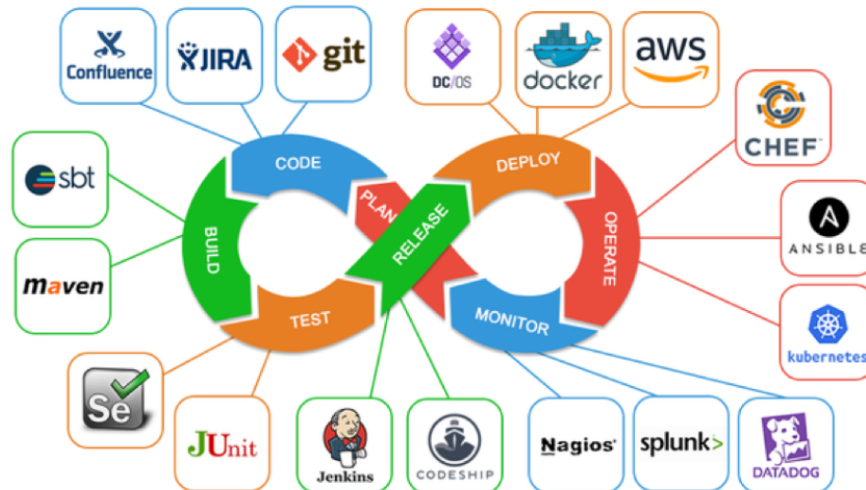


FIGURE 2.2 – Les outils DevOps

2.4 Avantages de DevOps

Les principaux bénéfices de DevOps pour toutes les parties prenantes sont :

- **Orientation client** : DevOps favorise une adaptation continue aux besoins des utilisateurs. Il aide les équipes à éviter de développer uniquement le “meilleur logiciel” en théorie, et à se concentrer sur ce qui apporte réellement de la valeur au client.
- **Collaboration entre les équipes** : toutes les équipes (développement, exploitation, tests, sécurité...) travaillent de manière plus intégrée et agile. Les équipes opérationnelles adoptent les pratiques agiles déjà éprouvées côté développement.
- **Automatisation** : DevOps permet d’automatiser de nombreux processus répétitifs (tests, déploiement, configuration...), ce qui libère du temps pour l’innovation et améliore la productivité.
- **Responsabilisation** : les parties prenantes sont impliquées dans toutes les étapes du cycle de vie du logiciel. Les mises en production fréquentes permettent au client de suivre l’évolution du produit et aux équipes de récolter des retours en continu.
- **Vitesse et qualité** : grâce à l’automatisation et à la collaboration renforcée, la livraison est plus rapide, les erreurs sont moins fréquentes et la qualité du produit s’améliore.

2.5 Caractéristiques innovantes de DevOps

Voici quelques innovations que DevOps introduit dans les organisations :

• Équipe intégrée

Les développeurs et les ingénieurs des opérations ne travaillent plus séparément. Ils sont réunis au sein d’une seule équipe cohérente, partagée entre le développement (Dev) et l’exploitation (Ops).

• Modèle de livraison unifié

DevOps favorise un code source unique qui intègre à la fois le logiciel, le middleware et l'environnement d'exécution. Il regroupe ce qui était auparavant géré de manière distincte (infrastructures, cycles de publication, processus de contrôle).

• CI/CD performant

L'association de l'intégration continue (CI) et de la livraison continue (CD) permet d'accélérer le cycle de développement, tout en assurant une qualité constante. Cela permet de mieux répondre aux besoins du marché avec des délais réduits.

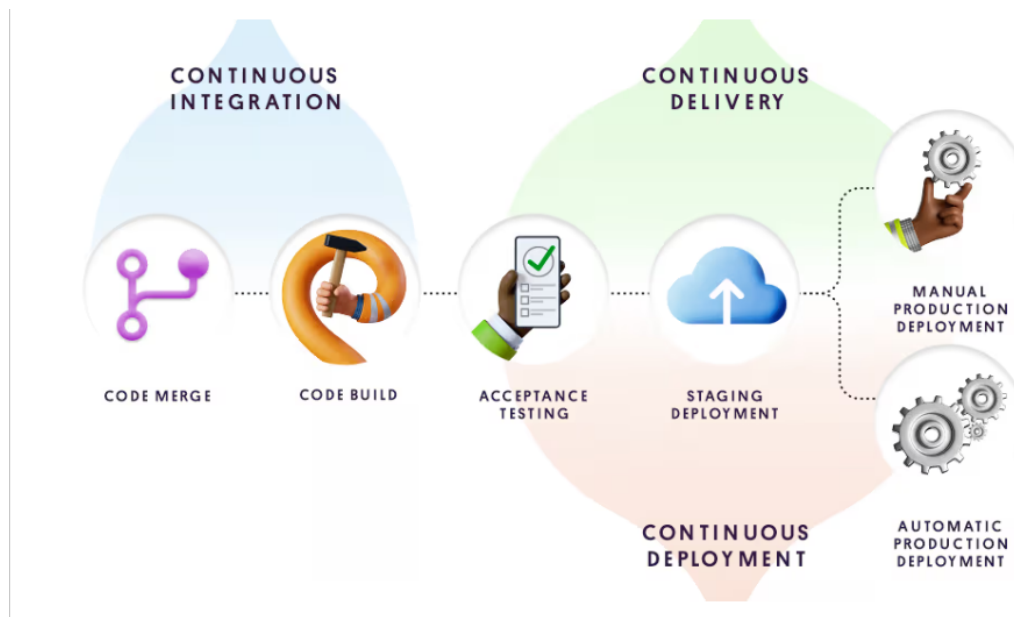


FIGURE 2.3 – CI/CD

2.6 Application dans ce projet

Dans le cadre de ce projet de gestion de la qualité, la méthode DevOps peut être appliquée de la manière suivante :

- Les mises à jour du code **Flask** peuvent être intégrées automatiquement via une pipeline **CI/CD**.
- Le déploiement de l'application peut être automatisé à l'aide d'outils comme **Docker** et **Kubernetes**.
- Une **surveillance** de l'application permettrait de garantir sa disponibilité et ses performances en continu.

Conclusion

Appliquée à ce projet, la méthode DevOps permet de faciliter les mises à jour, d'optimiser les déploiements et d'assurer une meilleure qualité logicielle, tout en renforçant la collaboration entre les différentes parties prenantes du développement.

Chapitre 3

Réalisation de l'Application

Introduction

Ce chapitre présente la phase de réalisation technique de l'application web développée dans le cadre de ce projet. Il détaille les technologies utilisées, l'architecture générale du système, ainsi que les principales fonctionnalités mises en œuvre.

3.1 Technologies utilisées

Le projet a été réalisé avec les technologies suivantes :

- Backend :



Flask est un micro framework open-source de développement web en Python. Il est classé comme microframework car il est très léger. Flask a pour objectif de garder un noyau simple mais extensible. Il n'intègre pas de système d'authentification, pas de couche d'abstraction de base de données, ni d'outil de validation de formulaires. Cependant, de nombreuses extensions permettent d'ajouter facilement des fonctionnalités. Il est distribué sous licence BSD.[2]

- Frontend :



Le HyperText Markup Language (HTML5) est le langage de balisage conçu pour écrire les pages web. Il permet de structurer le texte, de créer des formulaires et d'inclure des ressources multimédias (images, vidéos, programmes). Il reste aujourd'hui un standard incontournable du web.[3]



Les feuilles de style en cascade (CSS – Cascading Style Sheets) décrivent la présentation des documents HTML et XML. Standardisé par le W3C depuis les années 1990, CSS est devenu essentiel pour la conception de sites web modernes, offrant un design responsive et cohérent sur tous les navigateurs.[4]



Bootstrap est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub.[5]

- Système de gestion de base de données SGBD :



MySQL est un système de gestion de bases de données relationnelles (SGBDR) open source utilisé pour stocker et gérer des données. Sa fiabilité, ses performances, son évolutivité et sa facilité d'utilisation font de MySQL un choix plébiscité par les développeurs. .[6]

3.2 Architecture générale

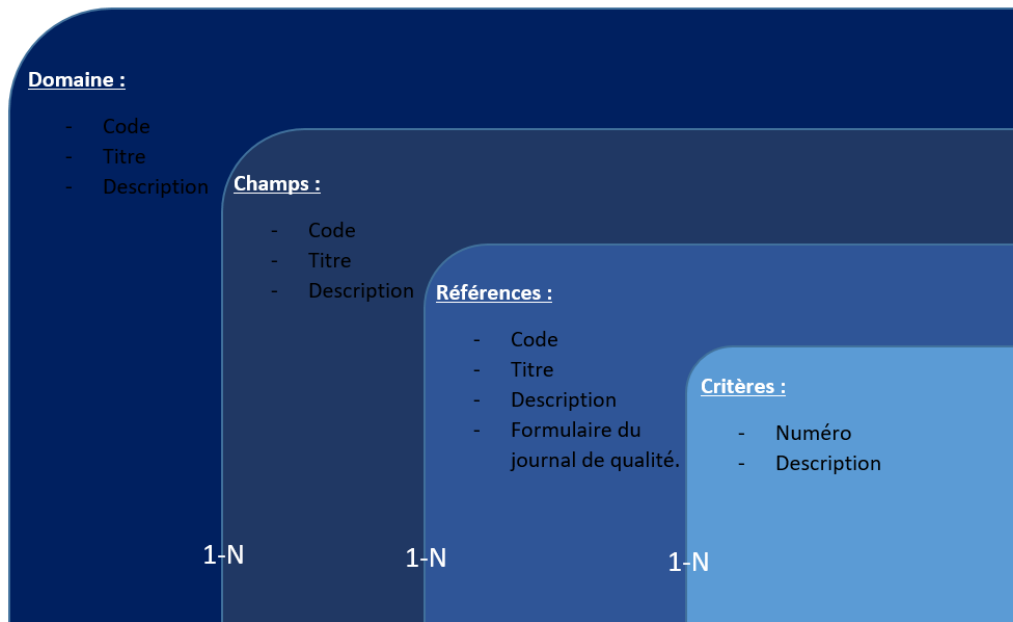


FIGURE 3.1 – Structure d'application SMQ

Schéma conceptuel montrant la structure du système de management de la qualité. Il présente les relations entre les entités : **Domaine**, **Champ**, **Référence** et **Critère**, avec une cardinalité 1-N (ce qui signifie que chaque entité peut contenir un ou plusieurs éléments de l'entité suivante). Chaque entité est décrite par des attributs clés tels que le code, le titre et la description. La Référence inclut également un formulaire de journal de qualité.

3.3 Fonctionnalités principales

L'application est proposée les fonctionnalités suivantes :

- **Gestion des domaines :**
 - Création, modification, suppression ces domaines.
- **Gestion des champs :**
 - Ajout, modification, suppression des champs liés à un domaine spécifique.
- **Gestion des références :**
 - Association de références normatives ou procédurales.
- **Gestion des critères (fiches documentaires) :**
 - Ajout, consultation, modification, suppression.
 - Possibilité d'associer des pièces jointes.
- **Export PDF :**
 - Possibilité d'exporter les données au format PDF pour migration ou sauvegarde.
- Réaliser une interface utilisateur en générale simple, responsive et accessible.

3.4 Organisation et acteurs

- Réaliser une interface utilisateur en générale simple, responsive et accessible.
- Les utilisateurs cibles :
 - **Administrateur** : gère tout (structure, utilisateurs...)
 - **Responsable qualité** : gère les fiches documentaires
 - **Consultant** : lecture seule
 - **Invité** (optionnel)

Remarque :

À ce stade du développement de notre application, un seul acteur, nommé « admin », a été implémenté. Il possède l'ensemble des privilèges associés à tous les profils utilisateurs (administrateur, responsable qualité, consultant, invité).

3.5 Visualisation de l'application

- **Lien de GitHub pour le code d'application[7] :**

https://github.com/hanan-ui/SMQ_app

- **l'authentification :**

Page de connexion au SMQ (Système Management Qualité). L'utilisateur doit saisir son nom d'utilisateur et son mot de passe. Des identifiants administrateur par défaut sont fournis : utilisateur "admin" et mot de passe "admin123".



FIGURE 3.2 – Page d'authentification

- **Page d'accueil :**

Cette page présente l'écran principal de navigation avec un accès direct aux domaines.

3. Réalisation de l'Application

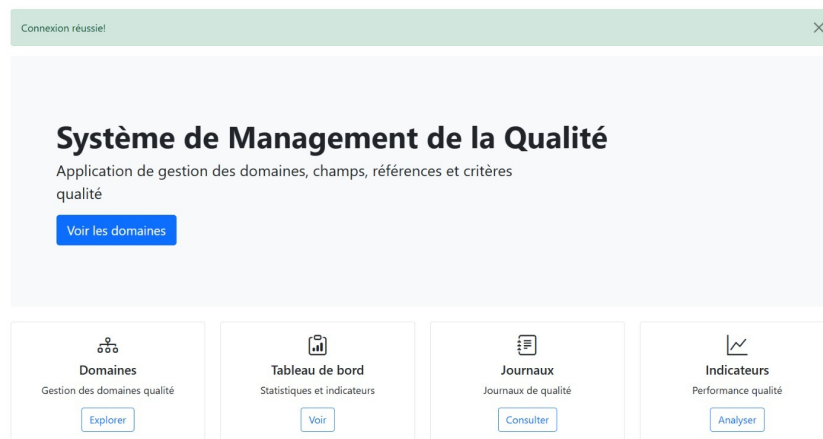


FIGURE 3.3 – Page d'accueil d'application SMA

- **Gestion de domaine :**

Page listant tous les domaines existants (exemples : b, D, F, 9) avec leurs codes, titres, descriptions et un bouton "Ciamps : FIX" pour chaque ligne.

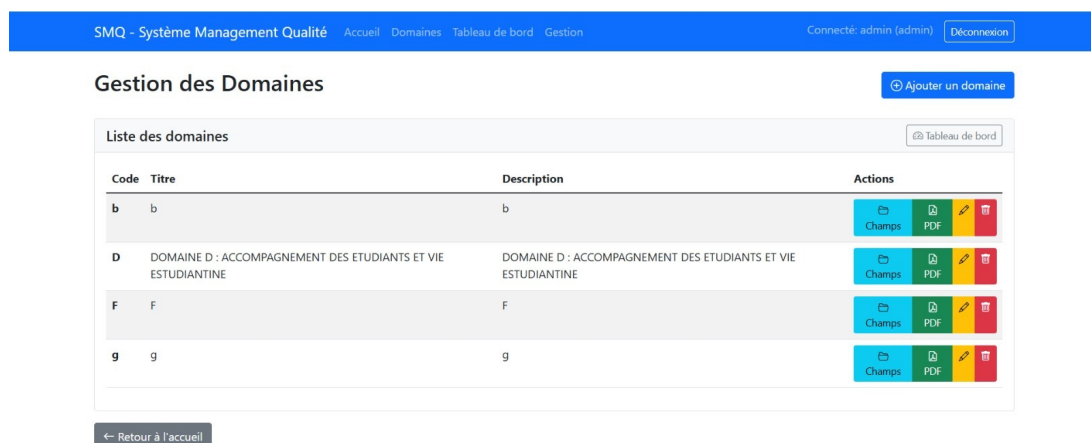


FIGURE 3.4 – Gestion de domaines

- **Gestion de champ :**

Page de gestion des champs appartenant à un domaine nommé "b". Affiche une liste des champs avec leurs codes, titres, descriptions et actions possibles (voir, modifier, supprimer).

3. Réalisation de l'Application

SMQ - Système Management Qualité Accueil Domaines Tableau de bord Gestion Connecté: admin (admin) Déconnexion

Gestion des Champs - Domaine b

Ajouter un champ

Informations du domaine

Code: b
Titre: b
Description: b

Liste des champs

Code	Titre	Description	Actions
b	b	b	Voir Modifier Supprimer

Retour aux domaines

FIGURE 3.5 – Gestion des champs d'un domain spécifique

• Ajout des références :

Interface d'ajout d'une nouvelle référence dans un champ spécifique. L'utilisateur doit renseigner un code unique, un titre et une description. Boutons pour annuler ou ajouter.

Ajouter une Référence

Retour au champ

Informations du champ

Domaine: b
Champ: b - b
Description: b

Nouvelle référence

Code de la référence *
Ex: D.I.1
Le code doit être unique pour ce champ

Titre de la référence *
Ex: Processus d'admission

Description
Décrivez cette référence...

Annuler Ajouter la référence

FIGURE 3.6 – Ajout des références d'un champ spécifique

• Formulaire journal de qualité :

Formulaire de journal qualité pour une référence. Contient des sections pour les informations de base, les objectifs, les engagements qualité et les actions de suivi et d'amélioration.

3. Réalisation de l'Application

The screenshot shows the SMQ - Système Management Qualité interface. At the top, there's a navigation bar with 'Accueil', 'Domaines', 'Tableau de bord', and 'Gestion'. A user is logged in as 'admin (admin)' with a 'Déconnexion' button. Below the navigation bar, there's a 'Retour aux références' button. The main content area is divided into two sections: 'Informations de la référence' and 'Formulaire du Journal de Qualité'. The 'Informations de la référence' section displays 'Référence: b', 'Domaine: b', 'Titre: b', and 'Champ: b'. The 'Formulaire du Journal de Qualité' section is divided into three parts: '1. Informations de base' with fields for 'Faculté *' (Faculté Polydisciplinaire de Ouazazate), 'Période début *' (jj/mm/aaaa), and 'Période fin *' (jj/mm/aaaa); '2. Objectifs et Engagement Qualité' with 'Objectifs *' (Ajouter un objectif) and 'Engagements *' (Ajouter un engagement); and '3. Actions de Suivi et d'Amélioration'.

FIGURE 3.7 – journal de qualité

• Gestion des critères :

Interface de gestion des critères pour la référence "b.b.b". Présente une liste numérotée des critères avec leurs descriptions et un bouton "Journal" pour chaque entrée.

The screenshot shows the SMQ - Système Management Qualité interface. At the top, there's a navigation bar with 'Accueil', 'Domaines', 'Tableau de bord', and 'Gestion'. A user is logged in as 'admin (admin)' with a 'Déconnexion' button. Below the navigation bar, there's a 'Retour aux références' button. The main content area is divided into two sections: 'Informations de la référence' and 'Liste des critères'. The 'Informations de la référence' section displays 'Référence: b.b.b - b' and 'Description: b'. The 'Liste des critères' section displays a table with columns 'Numéro', 'Description', and 'Actions'. The table has one row with '1', 'b', and 'Journal', 'Modifier', and 'Supprimer' buttons. There is also an 'Ajouter un critère' button in the top right corner.

FIGURE 3.8 – Gestion des critères d'un champ spécifique

• Interface mysql :

The screenshot shows the MySQL interface. On the left, there's a 'Navigator' pane showing the database structure. The 'domaines' table is selected. The main pane shows the 'domaines' table structure with columns: 'id', 'code', 'titre', 'description', and 'created_at'. The table contains 4 rows of data. The 'id' column has values 26, 30, 31, and 32. The 'code' column has values D, g, b, and F. The 'titre' column has values 'DOMAINE D : ACCOMPAGNEMENT DES ETUDIA...', 'g', 'b', and 'F'. The 'description' column has values 'DOMAINE D : ACCOMPAGNEMENT DES ETUDIA...', 'g', 'b', and 'F'. The 'created_at' column has values '2025-09-04 18:41:15', '2025-09-04 19:34:17', '2025-09-04 22:01:49', and '2025-09-05 16:59:48'.

id	code	titre	description	created_at
26	D	DOMAINE D : ACCOMPAGNEMENT DES ETUDIA...	DOMAINE D : ACCOMPAGNEMENT DES ETUDIA...	2025-09-04 18:41:15
30	g	g	g	2025-09-04 19:34:17
31	b	b	b	2025-09-04 22:01:49
32	F	F	F	2025-09-05 16:59:48

FIGURE 3.9 – structure de table domaines dans MySQL

3.6 Visualisation de résultat

SYSTÈME DE MANAGEMENT DE LA QUALITÉ

DOMAINE: D - DOMAINE D : ACCOMPAGNEMENT DES ETUDIANTS ET VIE ESTUDIANTE

Description du Domaine:

DOMAINE D : ACCOMPAGNEMENT DES ETUDIANTS ET VIE ESTUDIANTE

CHAMP: D.I - Champ D.I – Admission et orientation des étudiants

Description:

Champ D.I – Admission et orientation des étudiants

RÉFÉRENCE: D.I.1 - Référence D.I.1- L'institution définit les qualifications des étudiants ciblés, et ce conformément à sa mission et ses objectifs. Elle met en place pour chaque formation des procédures claires et appropriées d'admission et de sélection des étudiants afin de leurs garantir la réussite.

Description:

Référence D.I.1- L'institution définit les qualifications des étudiants ciblés, et ce conformément à sa mission et ses objectifs. Elle met en place pour chaque formation des procédures claires et appropriées d'admission et de sélection des étudiants afin de leurs garantir la réussite.

CRITÈRES:

1. L'institution admet les étudiants dans le respect des exigences d'éthiques et conformément au principe d'égalité des chances à l'éducation.

JOURNAUX DE QUALITÉ:

Faculté:	Faculté Polydisciplinaire de Ouarzazate
Période:	2025-09-27 au 2025-09-23
Statut:	brouillon
Créé le:	2025-09-04 19:12:21
Responsable:	Administrateur SMQ

1. OBJECTIFS ET ENGAGEMENTS QUALITÉ

Objectifs de qualité pour la période:

- Assurer l'actualisation et la diffusion du règlement intérieur

Engagements de la direction:

- Veiller à ce que les procédures d'admission respectent les principes d'égalité des chances et d'éthique

3. Réalisation de l'Application

2. ACTIONS DE SUIVI ET D'AMÉLIORATION

Date	Processus	Description	Responsable	Statut
2025-09-09	Admission	Mise en place d'un système transparent et éthique pour l'admission des étudiants	Vice Doyen chargé de la pédagogie, Service de scolarité	planifié

3. NON-CONFORMITÉS ET ACTIONS CORRECTIVES

Date	Description	Cause	Action	Responsable	Statut
2025-09-02	Manque de clarté sur les critères d'admission	Défaut de communication	Publication des critères sur le site web et affichage physique	Vice Doyen chargé de la pédagogie, Service de scolarité	ouvert

4. INDICATEURS DE PERFORMANCE

Indicateur	Valeur cible	Valeur actuelle	Analyse
Taux d'admission conforme aux critères éthiques et équitables	90	90	Des efforts sont nécessaires pour améliorer la transparence et l'équité des admissions. Renforcement des contrôles et formations des personnels recommandés.

5. REVUES ET DÉCISIONS

Décisions prises:

- Mise en place d'une charte éthique pour l'admission des étudiants. - Renforcement des critères de sélection pour assurer une meilleure adaptation aux
- formations. - Amélioration des canaux d'information sur les procédures d'accès et de réinscription. - Mise en place d'un comité de concertation entre enseignants et administration pour les
- transferts.

Date de la revue: 2025-09-26

Participants: Vice Doyen chargé de la pédagogie, chefs de départements, chefs de filières, Service de scolarité, Service de communication, Service des étudiants

6. PLANS D'AMÉLIORATION CONTINUE

Action	Objectif	Responsable	Délai
Transparence des admissions	Garantir l'équité et l'éthique des procédures	Vice Doyen chargé de la pédagogie, Service de scolarité	

OBSERVATIONS GÉNÉRALES:

. L'institution veille à garantir un accès équitable à l'éducation, en respectant les principes d'éthique et d'égalité des chances lors du processus d'admission des étudiants. Cette approche renforce la transparence et l'inclusivité,

FIGURE 3.10 – Les images présentent un exemple de résultats pour un domaine nommé 'D'

Conclusion

En résumé, cette application constitue une solution complète pour la gestion centralisée de la qualité, intégrant des modules pour les domaines, champs, références et fiches documentaires. Grâce à l'utilisation de technologies modernes comme Flask, MySQL et Bootstrap, elle offre une interface efficace et accessible.

Conclusion générale

Ce projet présente la conception et la réalisation d'une application web de gestion de la qualité, développée avec le framework Flask. L'objectif principal est de faciliter la gestion des domaines, champs, références et fiches documentaires, tout en assurant une meilleure traçabilité des informations et une organisation optimale.

L'application permet l'exportation des données, une gestion centralisée et un accès simplifié aux documents qualité. De plus, l'approche DevOps intégrée à ce projet favorise l'automatisation des processus, la rapidité de déploiement et l'amélioration continue du système.

En résumé, ce projet démontre comment une solution numérique bien conçue peut contribuer à améliorer et digitaliser la gestion de la qualité dans les établissements modernes, en s'appuyant sur les bonnes pratiques du SMQ et sur les outils modernes de développement logiciel, tels que ceux proposés par la méthode DevOps.

Références

1. Définition de la Méthode DevOps : <https://www.salesforce.com/fr/platform/devops-tools/what-is-devops/>
2. Définition de Flask : [https://fr.wikipedia.org/wiki/Flask_\(framework\)](https://fr.wikipedia.org/wiki/Flask_(framework))
3. Définition de HTML : https://fr.wikipedia.org/wiki/Hypertext_Markup_Language
4. Définition de CSS : https://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade
5. Définition de Bootstrap : [https://fr.wikipedia.org/wiki/Bootstrap_\(framework\)](https://fr.wikipedia.org/wiki/Bootstrap_(framework))
6. Définition de Mysql : <https://www.oracle.com/ma/mysql/what-is-mysql/>
7. Code github : https://github.com/hanan-ui/SMQ_app