



Cahier des charges pour la conception et le développement d'une application web de commande des bulletins, attestations de réussite, et diplômes en ligne pour l'Université du Burundi

1. Contexte du projet

L'Université du Burundi souhaite développer une application web permettant aux étudiants de commander en ligne leurs bulletins de notes, attestations de réussite, et diplômes. Cette initiative vise à améliorer l'efficacité des services administratifs, à réduire les files d'attente, et à offrir une accessibilité accrue aux étudiants.

2. Objectifs du projet

- Automatiser le processus de commande : Faciliter et dématérialiser les demandes de documents académiques par les étudiants.
- Réduire les délais de traitement : Assurer une gestion plus rapide et transparente des demandes.
- Améliorer l'expérience utilisateur : Offrir une interface intuitive et accessible à tous les étudiants, quel que soit leur niveau de compétence informatique.
- Garantir la sécurité des données : Protéger les informations personnelles et académiques des étudiants.

3. Périmètre fonctionnel

3.1. Authentification et gestion des utilisateurs

- Système de connexion sécurisé avec identifiant et mot de passe.
- Gestion des rôles : Étudiant, administrateur, service académique.
- Fonctionnalités de récupération de mot de passe.

3.2. Tableau de bord

- Vue d'ensemble des commandes en cours et passées.
- Accès rapide aux différentes sections : Commande de bulletin, attestation, diplôme.

- Notifications des actions nécessaires (paiement, téléchargement, etc.).

3.3. Commande de documents

- Formulaire de demande pour chaque type de document :
 - Bulletin de notes : Sélection des années académiques concernées.
 - Attestation de réussite : Sélection des années ou semestres concernés.
 - Diplôme : Demande d'édition du diplôme final.
- Vérification automatique des données saisies par l'étudiant.
- Système de validation par les services académiques.

3.4. Suivi de commande

- Suivi en temps réel de l'état de la demande : En cours de traitement, en attente de paiement, prêt à télécharger, etc.
- Historique des demandes passées.

3.5. Paiement en ligne

- Intégration avec les systèmes de paiement locaux (ex : Bancobu-inoti, Lumicash).
- Génération de reçus de paiement.
- Vérification automatique du paiement avant traitement de la commande.

3.6. Notifications

- Notifications par email ou SMS à chaque étape du processus.
- Alertes pour les actions nécessaires (par exemple, lorsque le document est prêt à être téléchargé).

3.7. Téléchargement de documents

- Téléchargement des documents en format PDF une fois approuvés et disponibles.
- Signature numérique pour garantir l'authenticité des documents.

3.8. Administration

- Tableau de bord pour les administrateurs avec statistiques (nombre de demandes, état des demandes, délais moyens de traitement, etc.).
- Gestion des utilisateurs et des droits d'accès.

- Outils de reporting et d'exportation des données.

4. Périmètre technique

4.1. Technologies front-end

- Langages : HTML5, CSS3, JavaScript.
- Frameworks : Vue.js, React ou Angular.
- Responsive design : Compatibilité avec les appareils mobiles (smartphones, tablettes).

4.2. Technologies back-end

- Langage : Python (Django) ou PHP (Laravel).
- Base de données : PostgreSQL ou MySQL.
- API REST : Pour les interactions entre le front-end et le back-end.

4.3. Sécurité

- Chiffrement des données : Utilisation de SSL pour sécuriser les communications.
- Authentification : OAuth2 ou JWT pour la gestion des sessions.
- Protection contre les vulnérabilités : Mise en place de mesures contre les attaques XSS, CSRF, et SQL Injection.

4.4. Hébergement

- Serveur : Serveurs cloud (AWS, Azure, DigitalOcean) ou serveurs internes à l'université.
- Sauvegardes : Système de sauvegarde quotidien des données.
- Redondance : Mise en place de redondance pour garantir la disponibilité du service.

5. Exigences non fonctionnelles

5.1. Performance

- Temps de réponse : Moins de 2 secondes pour la plupart des actions courantes.
- Capacité à gérer un grand nombre de connexions simultanées (scalabilité horizontale si nécessaire).

5.2. Accessibilité

- Conformité avec les normes d'accessibilité WCAG 2.1.
- Compatibilité avec les lecteurs d'écran et autres aides techniques.

5.3. Maintenance

- Documentation détaillée pour les développeurs et les administrateurs système.
- Système de logs pour la surveillance et la détection des erreurs.

6. Planning prévisionnel

6.1. Phase d'analyse et conception : 15 jours

- Analyse des besoins.
- Conception de l'architecture.
- Prototypage.

6.2. Développement : 2 mois

- Développement front-end et back-end.
- Intégration des services tiers.
- Tests unitaires et d'intégration.

6.3. Tests utilisateurs : 10 jours

- Tests avec un groupe d'étudiants et de personnel administratif.
- Ajustements basés sur les retours.

6.4. Déploiement et présentation mémoire : 5 jours

- Déploiement sur l'infrastructure choisie.
- Formation du personnel.
- Documentation et support.

Ce cahier des charges servira de guide tout au long du projet pour assurer la conformité aux objectifs fixés et garantir le succès de la mise en place de l'application web à l'Université du Burundi.