UCLouvain

École polytechnique de Louvain

Développement d'une plateforme de crowdsourcing pour la transcription et la labellisation de documents anciens

Auteur : **Jeremy Sidgwick**
Promoteur : **Yves Deville**
Lecteurs : **Aurore François**, **Xavier Gillard**
Année académique 2024–2025
Master [120] in Computer Science

Remerciements

Je tiens à remercier toutes les personnes qui m'ont aidé à réaliser ce mémoire.

En premier, Monsieur Deville, mon promoteur, pour m'avoir proposé ce sujet et pour m'avoir conseillé tout au long de l'année.

Merci à Monsieur Gillard, pour son expertise et ses connaissances des Archives.

Je remercie également Monsieur Gielis et Monsieur Van Gelder pour avoir pris le temps de m'expliquer le fonctionnement des Archives de l'État et pour avoir testé ma plateforme.

Enfin, je remercie ma famille et mes amis pour le soutien qu'ils m'ont apporté lors de ce mémoire.

Table des matières

1 Abstract

La transcription de documents anciens est une tâche essentielle pour les historiens, les linguistes et les chercheurs qui s'intéressent à la préservation et à la compréhension de l'histoire de l'humanité. Aujourd'hui, avec des millions de documents scannés, les chercheurs rencontrent des difficultés à trouver des informations précises et à analyser les différents documents de manière transversale.

L'objectif de ce mémoire est de développer une plateforme, nommée Archio, conçue pour faciliter la transcription et la labellisation de documents anciens. Cette plateforme vise, grâce au crowdsourcing, à accélérer et à améliorer le traitement des textes historiques. Le crowdsourcing est une forme d'externalisation qui permet de sous-traiter une tâche à un groupe de personnes quelconques. Cette plateforme vise également à faciliter l'accès aux connaissances anciennes pour la recherche universitaire et l'éducation du public. L'objectif est également de fournir des données utiles pour entraîner un modèle d'intelligence artificielle à transcrire des documents anciens.

En adoptant une approche globale et généraliste, cette plateforme vise à améliorer la transcription de tout type de document, indépendamment du sujet, de la langue et de la mise en page.

2 Introduction

Depuis des milliers d'années, les Hommes ont accumulé et conservé un nombre important de documents. Ces documents sont des traces du passé qui forment un socle de connaissances important pour la recherche. Leur analyse est une tâche essentielle pour les historiens, les linguistes et les chercheurs qui s'intéressent à la préservation et à la compréhension de l'histoire de l'humanité.

Depuis peu, les Archives ont commencé à adopter un virage numérique, notamment en numérisant ses documents physiques. Ceux-ci sont scannés pour être convertis en images.

Les Archives de l'État en Belgique possèdent plus de 26 millions d'archives numérisées [1]. Cette grande quantité de documents conservés est une richesse, mais l'analyse d'une telle quantité est également une difficulté. De même, il reste très compliqué et fastidieux d'effectuer des recherches manuelles dans ces images. Pour faciliter et automatiser ces recherches, il faut transcrire le texte de ces images afin de pouvoir directement l'analyser.

Malgré les progrès récents de l'intelligence artificielle pour transcrire des documents dactylographiés, la transcription de ces textes écrits à la main reste actuellement un défi trop complexe pour l'intelligence artificielle. Ces transcriptions doivent encore être réalisées par des humains possédant des compétences spécifiques en paléographie. La transcription de ces millions de documents, pourtant essentielle, est un travail trop important pour être réalisé au sein des Archives. Elles font donc appel à des bénévoles pour réaliser ce travail.

Ce mémoire propose une solution qui améliore et accélère le processus de transcription. Pour organiser et gérer ce travail collaboratif, cette solution utilise la pratique du crowdsourcing. Il s'agit une forme d'externalisation qui permet de sous-traiter une tâche à un groupe de personnes quelconques. Une plateforme dédiée à la transcription de documents anciens est développée dans ce mémoire afin de répondre aux besoins spécifiques des Archives. Le code de cette plateforme Open Source est disponible sous licence MIT sur GitHub (https://github.com/JeremySidgwick/thesis).

Ce mémoire est structuré en trois grandes parties.

La première partie présente toutes les réflexions et analyses qui ont été réalisées avant d'entamer le développement. Le contexte dans lequel ce mémoire évolue, le cahier des charges, l'étude des outils existants et l'analyse fonctionnelle y sont présentés.

La seconde partie explique le développement réalisé ainsi que les différents choix d'implémentation.

Pour finir, la troisième partie présente les analyses réalisées après le développement, à savoir une analyse des retours des utilisateurs, une analyse de sécurité, des recommandations pour les développements futurs et des perspectives d'évolution.

3 Contexte

Ce mémoire propose une solution pour aider la transcription de documents d'archives en utilisant une solution de crowdsourcing à partir du projet PARDONS. Ce chapitre explique ces trois concepts.

3.1 L'archivage

L'archivage est une série d'actions visant à collecter, conserver, classer et communiquer des documents, appelés archives. Selon les Archives de l'État en Belgique, « *Une archive reflète l'activité antérieure d'une personne, d'un groupe de personnes ou d'un organisme […], peu importe la date […] et la forme. Une archive peut être imprimée, manuscrite, audiovisuelle ou numérique. Les cartes, dessins, photos, enregistrements sonores, fichiers informatiques, etc., sont susceptibles de devenir des archives.*» [2]. Dans le cadre de ce mémoire, nous nous limiterons à des archives manuscrites qui ont été préalablement numérisées. Des Archives publiques sont présentes dans chaque pays, chaque région, chaque ville. Leurs tailles peuvent varier de quelques documents à plusieurs centaines de kilomètres de rayonnage. Il existe également des Archives privées dans les entreprises ou chez des particuliers.

L'archivage joue un rôle important dans la constitution de corpus de documents pouvant faire l'objet de recherches scientifiques et historiques. Les Archives conservent également des documents administratifs (jugements, propositions de loi, actes notariés, etc.) pendant de nombreuses années. Cela permet de garder une trace des événements et des décisions prises par les différentes institutions.

3.2 Le projet PARDONS

Ce mémoire s'appuie sur les problématiques pratiques rencontrées dans le projet « PARDONS » [3], par l'intermédiaire du Dr Gert Gielis, pour comprendre les besoins et les limitations de la transcription d'anciens documents. Ce projet est un des projets des Archives de l'État en Belgique réalisé en collaboration avec l'UCLouvain et la KULeuven. L'objectif de ce projet est de numériser, de transcrire et de rendre accessible au public des lettres de rémission accordées entre le XV^{e} et le XVII^{e} siècle par les souverains bourguignons et habsbourgeois qui régnaient sur la Belgique.

Une lettre de rémission, aussi appelée lettre de grâce, est une lettre accordée par le roi pour amnistier un individu d'un crime commis. Pour obtenir cette clémence, le condamné devait introduire une demande de grâce dans laquelle il expliquait les circonstances de son crime. Ces témoignages de citoyens ordinaires apportent de nombreuses informations sur la vie quotidienne à cette époque (bagarres de taverne, adultères, disputes entre voisins, …). De plus, ces lettres exposent un point de vue intéressant sur le pouvoir et les relations sociales dans les Pays-Bas méridionaux du début de l'Époque moderne, notamment lors des périodes de tensions politiques et religieuses, comme la Révolte hollandaise.

Les Archives de l'État possèdent des milliers de lettres de rémission. Numériser et transcrire ces archives est donc un projet de grande ampleur. Pour ce faire, le projet fait appel à des bénévoles pour réaliser la transcription de ces lettres. Pour le moment, le processus de transcription et sa gestion sont réalisés manuellement sans aucune automatisation.

Le processus manuel actuel est présenté ci-dessous en suivant les indications données dans le *guide de transcription* [4] du projet PARDONS de mai 2022.

Lorsqu'un bénévole souhaite participer, il doit s'inscrire et attendre de recevoir l'accès à un fichier partagé. Ce fichier Excel partagé fait office de « base de données de répartition ». Dans celui-ci, les différents documents à transcrire sont listés avec un certain nombre de métadonnées : la date, la langue dans laquelle le document est écrit, le lieu du crime, le nom du criminel et de la victime, etc.

Le bénévole choisit le document qu'il souhaite transcrire. Pour éviter que deux bénévoles ne travaillent sur le même document, le bénévole doit marquer son nom et la date de début de transcription dans les colonnes correspondantes du fichier Excel (en veillant à ne pas modifier les autres colonnes). Pour le choix des documents, le principe du « premier arrivé, premier servi » s'applique. Un utilisateur peut réserver jusqu'à cinq documents simultanément. Une fois le document réservé, le bénévole a trois mois pour réaliser sa transcription. Passé ce délai, si aucune transcription n'a été envoyée, un responsable doit contacter le bénévole et lever la réservation si la transcription n'est finalement pas réalisée.

Le bénévole doit ensuite obtenir l'image du document scanné. Pour ce faire, il doit récupérer le numéro d'archive du document dans le fichier Excel et se rendre sur le site des Archives de l'État (*search.arch.be*) pour rechercher l'image. Une fois l'image trouvée, elle peut être visualisée sur le site ou téléchargée.

Pour réaliser la tâche en elle-même, le bénévole doit réaliser sa transcription dans un *template* Word fourni sur le site du projet (*pardons.eu*). La transcription doit respecter les règles et les standards des notations afin de garder une cohérence entre les transcriptions, sur le fond et la forme du texte. Par exemple, il faut faire un retour à la ligne lorsqu'il y en a un dans le document original, il faut indiquer les abréviations avec les symboles « [ ] », etc. Ce travail est complexe : des connaissances en paléographie sont nécessaires pour être capable de lire des textes anciens manuscrits. La transcription d'une page prend environ une heure pour des personnes expertes.

Lorsque la transcription est finie, le bénévole l'envoie par mail. Cette transcription est ensuite vérifiée, corrigée et ajoutée dans une base de données par les gestionnaires du projet. Le fichier Excel est ensuite manuellement complété pour indiquer que la transcription du document est terminée. Le document et sa transcription sont ensuite mis sur le site du projet où ils peuvent être consultés.

Ce processus archaïque possède de nombreux problèmes. Il est très chronophage pour les administrateurs du projet qui doivent vérifier les transcriptions, gérer les inscriptions et manuellement relancer les utilisateurs après trois mois. Cela est également peu pratique pour les bénévoles qui doivent récupérer les différentes informations réparties sur différents sites. Le suivi des documents dans un fichier Excel partagé est également une grande source potentielle d'erreurs ou d'actes malveillants. La gestion de ce projet repose uniquement sur la bienveillance des bénévoles. À cause de ces problèmes, il faut parfois attendre plusieurs mois pour qu'une transcription soit réalisée alors que la tâche en elle-même ne dure que quelques heures.

Depuis la publication du *guide de transcription* [4], certains liens ou sites présents dans ce guide sont devenus obsolètes, notamment le site des Archives de l'État où se trouvent les images. Malgré cela, le processus reste inchangé.

3.3 Le crowdsourcing

Ce mémoire s'appuie sur la notion de « crowdsourcing ». Le crowdsourcing est un anglicisme constitué des mots, « crowd » et « outsourcing ». On peut le traduire littéralement par « externalisation par la foule ».

Les institutions francophones s'accordent davantage sur des traductions telles que « production participative » ou « externalisation ouverte ».

L'Office québécois de la langue française définit le crowdsourcing comme une «*pratique qui consiste, pour une organisation, à externaliser une activité par l'entremise d'un site web, en faisant appel à la créativité, à l'intelligence et au savoir-faire de la communauté des internautes pour créer du contenu, développer une idée, résoudre un problème ou réaliser un projet innovant, et ce, à moindre coût.*» [5].

En d'autres termes, le crowdsourcing est une forme d'externalisation qui permet à une entreprise de sous-traiter une tâche à un groupe de personnes quelconques et externes à l'entreprise (la « foule »), au lieu de la faire en interne. Le crowdsourcing peut être réalisé de manière bénévole ou en échange de rétributions.

Par définition, le bénéfice de cette pratique doit revenir à une organisation et non à une personne physique. Par exemple, si une personne demande de l'aide sur un groupe Facebook pour retrouver un objet perdu, cela ne peut pas être considéré comme du crowdsourcing, bien qu'une foule réalise une tâche.

3.3.1 Bref historique

Le mot « crowdsourcing » a été défini et popularisé par le journaliste Jeff Howe en juin 2006 dans un article du magazine *WIRED* [6]. Sa popularité a continué à grandir dans les années qui ont suivi. Malgré tout, cette notion reste assez méconnue du grand public. Pourtant, l'utilisation du crowdsourcing sur Internet est quotidienne.

Bien que cette notion ait été définie et popularisée au XXI^{e} siècle, il est possible de trouver d'anciens exemples de crowdsourcing a posteriori.

Les chasseurs de primes lors de la conquête de l'Ouest aux États-Unis, peuvent être considérés comme des sous-traitants de l'État, qui externalise la traque de criminels en faisant appel à la foule en échange d'une récompense. Dans les années 50, la vente à domicile popularisée par *Tupperware* pouvait s'apparenter à du crowdsourcing, car la vente des produits était externalisée à une foule de clients en échange de rémunération.

Il est donc difficile de dater avec précision sa première apparition au-delà du mot en lui-même.

3.3.2 Typologie du crowdsourcing

Il est possible de diviser le crowdsourcing en différentes sous-catégories en fonction de la nature de la tâche demandée ainsi que de la motivation de la foule pour réaliser cette tâche.

Selon la nature de la tâche [7]

- Tâches d'activités inventives :
  Ce type de crowdsourcing propose à la foule des tâches créatives ou plus complexes qui requièrent généralement des compétences plus spécifiques. L'entreprise propose la tâche à réaliser, ensuite les différentes personnes vont proposer une solution et la meilleure solution gagne et est fortement rémunérée. (*Winner takes all*). Ici, la taille de la foule est moins importante que sa diversité, car le nombre de tâches et de solutions retenues est assez faible. Le *Bug Bounty* (système de signalement de vulnérabilités en cybersécurité) est un bon exemple pour ce type de tâche. Une foule ouverte d'experts cherche une vulnérabilité. Seul celui qui en trouve une reçoit une récompense financière.

  Pour l'aspect créatif, cela prend plutôt la forme d'un sondage ou d'un concours dans lequel la foule propose de nouvelles idées de produits, slogans, etc.

- Tâches routinières :
  Ces tâches sont généralement très chronophages et ne demandent pas beaucoup de compétences pour être réalisées. Pour inciter les personnes à participer, une faible rémunération est souvent mise en place pour chaque tâche réalisée.

  L'exemple le plus connu pour ce type de tâche est *reCAPTCHA* (Figure 1), le système de Google de protection contre les *bots* informatiques. Ce système a un double objectif : en plus de pouvoir différencier un utilisateur humain d'un *bot*, il est utilisé par Google pour entrainer ses modèles d'intelligence artificielle. C'est ce deuxième objectif qui fait appel au crowdsourcing. Cela est notamment utilisé pour réaliser de l'OCR (*optical character recognition*) et pour améliorer la détection d'objets sur les images de Google Street View.

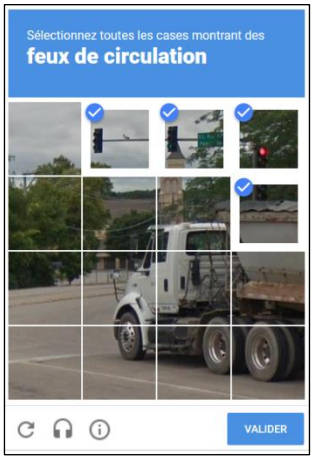


Figure 1. Exemple de reCAPTCHA

Dans cette catégorie, on peut également citer le projet GIMPS, Great Internet Mersenne Prime Search [8]. Ce projet recherche le plus grand nombre premier de *Mersenne* (ayant la forme 2^N -1). En octobre 2024, cette initiative a découvert un nouveau plus grand nombre premier ($2^{136279841}$-1). La tâche effectuée par des bénévoles consiste à partager la puissance de calcul de leur ordinateur, pour que celui-ci puisse chercher un nouveau nombre premier. Les tâches peuvent donc prendre de nombreuses formes.

Dans le cas du partage de puissance de calcul, tout ce qui est lié aux cryptomonnaies ne peut pas s'inscrire dans le crowdsourcing car la gouvernance n'est pas hiérarchisée et il n'y a pas une entreprise qui coordonne les tâches. Dans ce cas, l'externalisation prend plutôt la forme d'une *communauté.*

- Tâches de contenu :
  Le rôle de ces tâches est d'apporter des données, de l'information. Ce type de tâche ne propose généralement pas de rémunération pour le contenu fourni. L'exemple certainement le plus populaire est *Wikipédia*. Ce site invite une foule d'internautes à apporter du contenu et des informations à travers les différents articles du site.

Selon la motivation de la foule [9]

- Le travail mécanique
  Ici, la motivation est uniquement monétaire. La réalisation des tâches peut être vue comme un travail. Dans cette optique, de nombreuses tâches sont réalisées avec comme seule motivation les micropaiements, notamment les tâches de NLP (traitement du langage naturel), de détection d'émotion et d'opinion.

- L'aspect ludique (gamification)
  Certains projets proposent des tâches sous la forme de jeu. L'utilisateur participe principalement pour l'aspect ludique et non pas pour la finalité de la tâche. L'application *Waze* utilise les données envoyées par les utilisateurs afin d'avoir une carte détaillée et à jour. Pour motiver les utilisateurs à participer, *Waze* a mis en place un système de points, avec des défis, des récompenses et un classement pour transformer la collecte d'informations en un jeu.

L'aspect ludique peut également prendre une forme plus artistique et communautaire, comme dans le projet *r/place* organisé par *Reddit*. Dans ce projet, plus de sept millions d'internautes ont pu réaliser une immense toile (Figure 2) en posant chacun un pixel toutes les cinq minutes. En dessous de l'aspect ludique, cette action marketing a permis à *Reddit*, en utilisant la foule, d'atteindre de nouveaux utilisateurs, de créer de l'engagement sur leur plateforme et de bénéficier d'une certaine retombée médiatique.

Figure 2. Extrait de la toile du r/place 2022

- La motivation altruiste

  Certains projets reposent uniquement sur la bonne volonté de bénévoles pour réaliser les tâches. Il n'y a pas de rémunération ou de contrepartie. Pour cette motivation, on peut reprendre l'exemple de Wikipédia dans lequel les bénévoles n'ont pas de contrepartie financière ni un système de gamification pour les inciter à participer au projet.

Dans le cas du projet de transcription de documents, il s'agit d'un crowdsourcing de contenu. Les bénévoles apportent du contenu en réalisant la transcription sans rémunération en contrepartie. Contrairement à l'explication théorique fournie ci-dessus, ce projet demande malgré tout, aux bénévoles, une certaine expertise dans un domaine assez spécifique. Bien que la plateforme soit ouverte au monde entier, les tâches requièrent des compétences en paléographie (étude des écritures manuscrites anciennes), en ancien français et en ancien néerlandais. Cela rend la transcription d'anciens documents assez particulière dans l'écosystème du crowdsourcing. La motivation des bénévoles qui vont participer à ce projet est strictement altruiste. Il n'est donc pas nécessaire de prévoir un système de rémunération ou de gamification.

3.3.3 Avantages

Le crowdsourcing peut apporter de nombreux avantages à l'entreprise :

- Cela permet de diminuer le coût des tâches concernées, car le crowdsourcing est principalement bénévole ou peu rémunéré.
- Si la foule a une taille assez grande, il peut y avoir un gain d'efficacité par rapport à une équipe d'employés plus réduite.
- Cette solution permet également une plus grande flexibilité que des employés en interne. Le nombre de tâches à réaliser par la foule peut varier énormément en un temps très court.
- Cela peut apporter des idées créatives et une plus grande diversité. Il est possible, notamment pour les tâches d'activités inventives, d'obtenir une diversité plus importante en faisant appel à des bénévoles partout dans le monde.
- Cela permet d'externaliser l'échec et le risque. L'entreprise ne paie que pour ce qui est réalisé, ou uniquement pour la meilleure solution dans le cas du crowdsourcing d'activités inventives, contrairement aux équipes en interne que l'entreprise doit payer même en cas d'échec. Pour reprendre l'exemple des Bug Bounty, l'entreprise ne paie pas les experts qui ne trouvent pas de vulnérabilités.

3.3.4 Risques et inconvénients

Les premiers risques que le crowdsourcing peut apporter sont liés aux travailleurs.

Pour les employés de l'entreprise qui met en place du crowdsourcing, cela peut remplacer certains emplois précédemment réalisés en interne, et donc impliquer à terme des licenciements.

Pour les personnes qui participent au crowdsourcing, elles risquent de faire face à une forme d'exploitation par le travail. Les entreprises tirent un bénéfice d'un travail qui a été fait gratuitement ou en échange d'une rémunération très faible. De plus, ces travailleurs n'ont pas de statut légal vis-à-vis de l'entreprise et ils ne sont soumis à aucune loi du travail. Il n'y a aucune limitation sur le temps de travail maximal, aucun contrat, pas de salaire minimal, pas de stabilité, aucune protection sociale, etc. D'autant plus qu'une grande partie de la foule peut être originaire de pays en développement où le statut d'indépendant est peu règlementé.

D'autres risques peuvent aussi se porter sur les données en elles-mêmes.

La foule peut commettre des erreurs volontaires ou involontaires. Dans ce cas, l'entreprise doit être capable de les corriger ou d'en assumer la responsabilité, car on ne peut pas imputer la faute à la foule.

Comme toutes méthodes d'externalisation, il y a aussi un risque de divulgation de données confidentielles, car celles-ci sont partagées avec de nombreuses personnes. Dans le cadre de ce projet, les données d'archives ne sont pas confidentielles. Il n'y a donc pas de risque à ce niveau-ci.

3.3.5 Alternatives au crowdsourcing

D'autres moyens d'externalisation existent comme présenté dans le Tableau 1 [10]. Une entreprise peut faire appel à :

- Une communauté : il s'agit d'un groupe ouvert où tout le monde peut apporter des problèmes et des solutions. La gouvernance est faite par la communauté et il n'y a pas d'entreprise qui chapeaute le projet. Le monde de l'Open Source en est un bon exemple.
- De la consultance : il s'agit d'un groupe fermé d'experts qui apporte une solution au problème posé par une entreprise.
- Un consortium : il s'agit d'un groupe fermé dans lequel chaque membre contribue au choix du problème à résoudre.

Comme vu précédemment, le crowdsourcing est une solution dont la participation est ouverte à tous et où l'entreprise choisit la solution à son problème.

<table>
<tr><td>Crowdsourcing
Un endroit où une entreprise propose un problème. N'importe qui peut proposer une solution et l'entreprise choisit celle qu'elle préfère.</td><td>Communauté
Un réseau ouvert où tout le monde peut proposer des problèmes et des solutions.
Exemple : Communauté Open Source</td><td rowspan="3">Participation</td><td>Ouverte</td></tr>
<tr><td>Consultance
Un groupe d'experts choisis par l'entreprise. L'entreprise choisit le problème et la solution.
Exemple : Capgemini</td><td>Consortium
Un groupe privé de partenaires qui choisissent un problème, décident comment organiser le travail et sélectionnent la solution conjointement.
Exemple : partenariats d'IBM</td><td>Fermée</td></tr>
<tr><td colspan="2">Gouvernance</td></tr>
<tr><td>Structure hiérarchique</td><td>Structure plate</td><td colspan="2"></td></tr>
</table>

Tableau 1. Formes d'externalisation d'après Pisano et Verganti [10]

4 Cahier des charges

Ce cahier des charges dresse les fonctionnalités attendues de la plateforme. Il a été réalisé sur base du processus existant et des besoins exprimés par Dr Gert Gielis dans le projet PARDONS des Archives de l'État.

Le but principal de la plateforme souhaitée est de simplifier et d'accélérer le processus de transcription de documents anciens. Cette plateforme doit adopter une approche altruiste du crowdsourcing afin de permettre à des utilisateurs de participer bénévolement au projet. Aucun système de rémunération ou de gamification ne doit être mis en place.

Les coûts de mise en place et d'exploitation de cette plateforme doivent rester faibles.

La plateforme doit être accessible via un site web pour être disponible au plus grand nombre. Ce site doit être conçu pour être utilisé sur un ordinateur. En effet, une interface verticale, conçue pour téléphones ou tablettes, ne conviendrait ni à la tâche de transcription ni aux habitudes des utilisateurs. L'interface doit être claire et accessible à des utilisateurs sans expertise informatique.

4.1 Fonctionnalités liées aux documents

Il doit être possible pour les gestionnaires du projet d'ajouter des documents à transcrire sur la plateforme, en prenant en compte qu'un document d'archives peut avoir plusieurs pages scannées séparément. La plateforme doit accepter des images au format JPEG et PNG.

Ces documents sont répartis dans différents projets. Chaque projet doit indiquer les différents documents disponibles et l'état de leur transcription. Il doit également y avoir des informations concernant l'avancement du projet dans sa globalité.

La plateforme doit permettre aux utilisateurs de transcrire le texte du document. Lorsqu'un bénévole commence la transcription d'un document, celui-ci ne doit plus être accessible aux autres utilisateurs afin d'éviter le travail redondant. Les transcriptions doivent être segmentées ligne par ligne pour pouvoir entrainer une intelligence artificielle à transcrire des textes manuscrits.

Il faut également mettre en place un système de vérification afin de garantir la qualité des transcriptions proposées. Un utilisateur ne peut pas réaliser la transcription et la vérification d'un même document.

Pour finir, une fois un document transcrit, la plateforme doit rendre le document et sa transcription accessibles à tous. Le résultat des transcriptions doit également être exportable dans des formats adaptés à l'utilisation des archivistes et à l'entrainement d'une intelligence artificielle de reconnaissance de texte.

4.2 Fonctionnalités liées aux utilisateurs

Cette plateforme sera utilisée par deux profils d'utilisateurs : les bénévoles qui réalisent les différentes tâches et les gestionnaires du projet qui dirigent l'ensemble. Ces deux profils ont des besoins et des utilisations différentes de la plateforme. On peut définir trois cas d'utilisation (*use cases*) :

1. Le bénévole qui réalise une transcription :
   Avant de commencer la transcription d'un document en tant que telle, le bénévole doit pouvoir le prévisualiser. Le bénévole sélectionne un des documents disponibles selon le principe du « premier arrivé, premier servi ». Un utilisateur ne peut réaliser qu'une seule transcription à la fois.
   Cette étape demande une expertise et des compétences spécifiques. L'accès à la transcription doit donc être limité à un groupe restreint de personnes afin de garantir la qualité du travail fourni. Dans ce travail, la foule (du crowdsourcing) est restreinte à un groupe d'experts, mais la candidature pour intégrer cette foule doit rester ouverte à tous.

2. Le bénévole qui vérifie une transcription :
   La plateforme doit permettre la comparaison entre l'image et la transcription précédemment effectuée. Le bénévole doit pouvoir apporter des modifications si la transcription n'est pas correcte. Cette tâche doit être uniquement proposée à un sous-groupe d'utilisateurs de confiance pour garantir la qualité de la vérification.

3. Les gestionnaires de projets :
   La plateforme doit simplifier la gestion de différents aspects : gérer les utilisateurs, gérer les documents en permettant d'en ajouter ou d'en retirer et gérer les productions faites par les bénévoles en ayant la possibilité de supprimer une contribution erronée ou de remettre un document dans un état précédent.

Un système de feed-back entre les utilisateurs doit également être mis en place afin d'améliorer leur expertise et la qualité du résultat final.

5 Étude de l'existant

De nombreux sites web proposent déjà des solutions de crowdsourcing. Cependant, le crowdsourcing est un domaine vaste avec beaucoup d'approches différentes, d'autant plus que ce projet a des besoins très spécifiques.

Idéalement, on cherche un outil qui pourrait répondre au cahier des charges et qui possède au moins les grandes caractéristiques suivantes :

- Adopter une approche altruiste du crowdsourcing.
- Permettre la transcription ligne par ligne de documents.
- Posséder un système de gestion des utilisateurs pour garantir la qualité du travail et l'expertise des bénévoles.
- Être personnalisable et idéalement Open Source.
- Avoir un faible coût de mise en place et d'exploitation.

Dans cette section, différentes solutions vont être présentées : des outils plus généralistes, puis des solutions conçues pour la transcription d'archives.

5.1 Solutions généralistes

Les solutions les plus populaires n'adoptent pas une approche altruiste du crowdsourcing. Les sites, tels que *Amazon Mechanical Turk* [11] et *Microworkers* [12] sont utilisées pour réaliser un travail mécanique en échange de micropaiements. Le site *Wazoku Crowd* [13], développé par *InnoCentive,* est conçu pour la réalisation de tâches d'activités inventives. Une grosse rémunération, plusieurs milliers d'euros, est donnée à la meilleure proposition (*Winner takes all*).

D'autres sites, comme C*lickworker* [14] ou *DataAnnotation* [15], sont des solutions uniquement conçues pour entrainer leurs modèles d'intelligence artificielle. Il n'est donc pas possible d'utiliser ceux-ci pour notre projet.

De plus, les cinq solutions ci-dessus sont peu personnalisables et ne sont pas Open Source. Il n'est pas possible de gérer les utilisateurs comme souhaité.

5.2 Solutions de transcription

Il existe également des solutions de crowdsourcing conçues spécifiquement pour la transcription de documents.

Malheureusement, une grande partie d'entre elles sont des solutions développées par des entreprises uniquement pour leur propre besoin de transcription. Il n'est donc pas possible de se baser sur ces solutions. On peut citer les sites : T*ranscribathon* [16], *DigiVol* [17] et *Smithsonian Digital Volunteers* [18].

Certains sites, comme *e-manuscripta* [19] ou *FromThePage* [20], pourraient être utilisés. Ils utilisent une approche altruiste mais les coûts d'exploitation sont trop élevés pour le budget du projet PARDONS. La solution proposée n'est pas personnalisable ni en adéquation avec le cahier des charges.

Les sites *DoeDat* [21] et *Zooniverse* [22] ne peuvent également pas être utilisés car la collecte de données est présentée sous la forme d'un formulaire. De plus, il n'est pas possible de gérer les utilisateurs comme souhaité.

Le site *Transkribus* [23] est déjà utilisé dans le projet PARDONS. Il permet d'entrainer un modèle d'intelligence artificielle capable de faire de la reconnaissance d'écriture manuscrite. Pour ce faire, il a besoin de documents et de leur transcription. Ce site n'est pas conçu pour du crowdsourcing mais pour réaliser l'étape suivante, une fois la transcription terminée.

Il existe plusieurs projets Open Source :

Les projets *Scribe* [24], W*4P* [25] et *Pybossa* [26] sont trois projets qui ne sont plus maintenus depuis plusieurs années. De plus, ils ne correspondent pas assez au cahier des charges. Il n'est donc pas possible de les utiliser comme base de ce projet.

Pour finir, le projet *Concordia* [27] peut être mentionné. Ce projet de crowdsourcing est développé en interne par la *Bibliothèque du Congrès des États-Unis*. Il s'agit de la solution la plus proche de celle recherchée. Le projet est Open Source et utilise une approche altruiste. Malheureusement, il manque des fonctionnalités importantes. En effet, les utilisateurs peuvent participer sans créer de comptes, ce qui rend le suivi et le contrôle des utilisateurs impossibles. Il n'est pas possible non plus de transcrire les documents ligne par ligne. Ce projet est plus adapté pour des documents dactylographiés, pour lesquels l'OCR (*optical character recognition*) est facilement utilisable. Ces différences sont importantes et structurelles. De plus, *Concordia* est un vaste projet complexe développé par une organisation au fil des années, prévu pour accueillir simultanément un grand nombre d'internautes. Le travail à fournir pour adapter ce projet est trop important dans le cadre de ce mémoire et n'apporterait pas une plus-value suffisante.

Après avoir réalisé une vaste analyse des outils et projets existants, on peut constater qu'il n'existe pas de solution que nous pouvons utiliser comme base pour ce mémoire. La solution est donc de développer une nouvelle plateforme spécifique pour ce projet.

6 Analyse fonctionnelle

L'analyse fonctionnelle a pour objectif de définir les fonctions attendues du programme. Elle a permis de formaliser les rôles des utilisateurs et le processus suivi par un document.

6.1 Les utilisateurs

Pour permettre un suivi des utilisateurs et gérer leurs permissions, un système d'authentification va être implémenté sur la plateforme.

Cela va également permettre de donner différents rôles à chaque utilisateur pour chaque projet. Les différents rôles sont les suivants :

- **Transcripteur** : ce rôle permet uniquement à l'utilisateur de réaliser les tâches de transcription.
- **Vérificateur** : un vérificateur peut réaliser les tâches de vérification ainsi que les tâches de transcription.
- **Gestionnaire** : ce rôle permet de gérer les documents et les utilisateurs du projet. Ce rôle peut également réaliser les tâches de transcriptions et de vérifications.
- **Candidat** : il s'agit d'un rôle temporaire pour les utilisateurs qui ont fait la demande pour intégrer le projet et qui sont en attente d'acceptation.
- **Bloqué** : il s'agit d'un rôle pour les utilisateurs qui ont été exclus du projet. Un utilisateur banni d'un projet ne peut plus se porter *candidat*.

Les utilisateurs qui n'ont pas créé de compte ne peuvent pas accéder aux informations des différents projets.

Les utilisateurs qui sont connectés mais qui n'ont aucun rôle dans le projet ne peuvent pas y participer. Ils peuvent uniquement consulter les documents déjà transcrits et postuler pour intégrer ce projet.

Un système de feed-back sera mis en place entre les utilisateurs sous la forme d'un forum. Les sujets de ces discussions peuvent porter sur un document ou sur une ligne spécifique d'un document.

6.2 Les documents

Cette section va expliquer le processus (*workflow)* que va suivre un document pour être transcrit. Ce processus commence à la mise en ligne sur la plateforme par le gestionnaire et se termine lorsque le document est disponible à la consultation.

Pour chaque étape, un schéma reprend les actions que les différents utilisateurs pourront effectuer ainsi que les actions qui seront faites par le système. Le schéma complet se trouve dans l'Annexe A. Le schéma complet contient également des informations sur les modifications apportées aux données. Cette partie sera traitée dans le chapitre *Développement*.

6.2.1 Phase d'initialisation

Le gestionnaire peut mettre en ligne un document à transcrire dans un projet créé au préalable. Dès que le document est chargé sur la plateforme, celui-ci rejoint les documents prêts pour la phase de transcription (Figure 3).

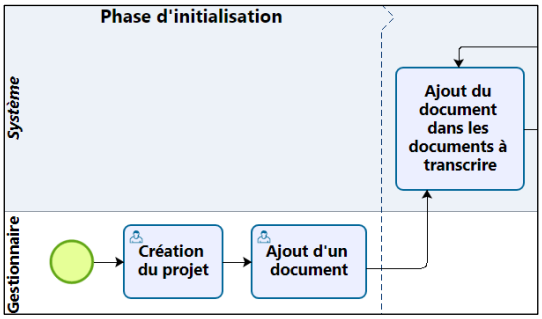


Figure 3. Schéma de la phase d'initialisation

6.2.2 Phase de transcription

N'importe quel utilisateur faisant partie de ce projet, quel que soit son rôle (transcripteur, vérificateur ou gestionnaire) peut démarrer la transcription d'un document (Figure 4). Dès que l'utilisateur commence la tâche, le document n'est plus disponible pour les autres tant que la tâche n'est pas finie. L'utilisateur a un délai maximal d'un mois pour réaliser la transcription. À la fin de ce délai (*timeout*), le document sera de nouveau disponible pour les autres transcripteurs.

Lors de la transcription, l'utilisateur doit labelliser chaque ligne de texte du document. Cela permet de segmenter la tâche ainsi que de former des données plus précises pour l'entraînement de modèles d'intelligence artificielle. Pour chaque ligne labellisée sur l'image, l'utilisateur doit transcrire le texte dans un champ texte. Dès qu'il a transcrit toutes les lignes du document, l'utilisateur peut soumettre sa transcription pour la phase suivante de vérification.

Les tâches de transcription peuvent être sauvegardées afin que l'utilisateur puisse garder sa progression et la reprendre ultérieurement.

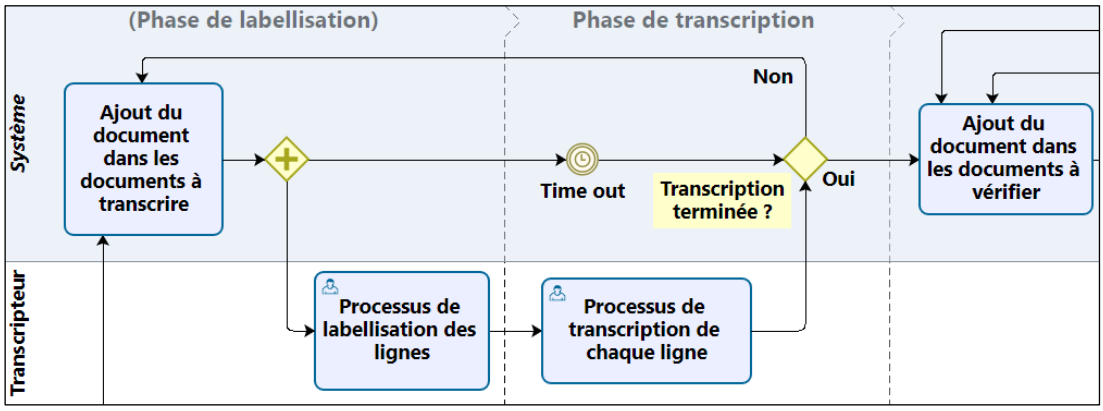


Figure 4. Schéma de la phase de transcription

6.2.3 Phase de vérification

Cette phase consiste à comparer le document avec la transcription proposée lors de la première étape. Un utilisateur ayant le rôle de vérificateur (ou gestionnaire) peut vérifier un document parmi ceux déjà transcrits (Figure 5). Cependant, l'utilisateur ne peut pas vérifier sa propre transcription. Comme pour la phase précédente, il a un temps limité (une semaine) pour vérifier la transcription.

Pour chaque ligne, le vérificateur peut accepter la transcription, apporter une modification ou supprimer le label et sa transcription (dans le cas où la labellisation n'a pas été faite correctement). Une fois chaque ligne contrôlée, le vérificateur peut finaliser sa tâche.

Si l'étape de vérification a apporté une modification par rapport à l'étape précédente, seules les lignes modifiées vont devoir être revues par un autre vérificateur. Cette étape de vérification sera répétée jusqu'à ce que toutes les lignes soient validées. Chaque ligne aura donc été validée par deux et seulement deux utilisateurs : soit le transcripteur et le premier vérificateur si la transcription était directement valide, soit par les deux derniers vérificateurs si des modifications ont été apportées.

Si la labellisation et la transcription sont particulièrement de mauvaise qualité, le vérificateur peut décider de renvoyer le document entier à l'étape de transcription.

Comme pour la transcription, les tâches de vérification peuvent être sauvegardées afin que l'utilisateur puisse garder sa progression et la reprendre ultérieurement.

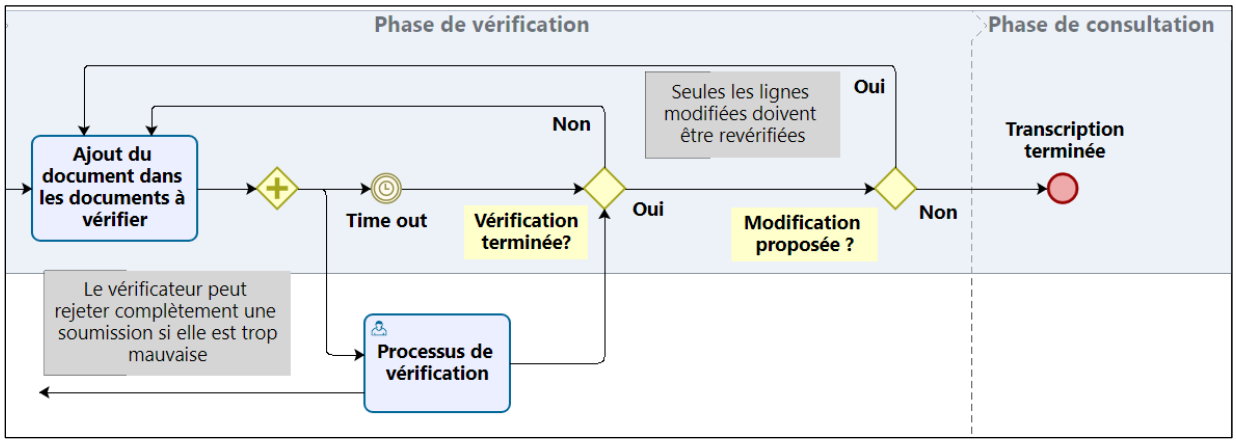


Figure 5. Schéma de la phase de vérification

6.2.4 Phase de consultation

Pour finir, lorsque le traitement du document est terminé, celui-ci et sa transcription sont rendus accessibles à tous. Il est possible de voir, pour chaque ligne, la transcription finale ainsi que les propositions précédentes. C'est également à ce moment-là qu'il est possible d'exporter la transcription sous forme de fichiers texte et *CSV*. Le format texte est adapté à l'utilisation des archivistes en contenant uniquement la transcription finale. Le format *CSV* contient des métadonnées supplémentaires pour chaque ligne du document.

7 Développement

Dans ce chapitre, l'implémentation, la structure de la base de données et les choix de développement sont expliqués. Le fonctionnement de chaque page web est également détaillé. Le code de cette plateforme Open Source est disponible sous licence MIT sur GitHub (https://github.com/JeremySidgwick/thesis).

7.1 Architecture

Pour le développement de cette application, il faut dans un premier temps choisir un langage et un framework back-end. Un framework back-end est un ensemble d'outils et de structure de code pré-écrit qui facilitent le développement. Son rôle est de générer les pages web et de gérer la base de données.

Python est un langage puissant qui permet d'écrire des programmes complexes. Il a été choisi car il possède de nombreux avantages. Son architecture modulaire facilite le développement de traitements d'images avancés et d'intelligences artificielles, ce qui est directement lié à ce projet. Python est l'un des langages de programmation les plus utilisés avec environ 30% de part de marché selon *cybersecuritynews* [28]. Sa grande popularité en fait un bon candidat pour ce projet Open Source. De plus, de nombreux frameworks web sont développés en Python.

Différents frameworks Open Source écrits en Python ont donc été analysés afin de déterminer le plus adéquat pour ce projet : Flask, Quart et Django.

7.1.1 Flask

Flask [29] est un framework Python très simple qui peut permettre de développer rapidement des petits prototypes de sites web. Concrètement, Flask est un module Python qui écoute les requêtes sur un port réseau spécifique. Lorsqu'une requête arrive sur ce port, Flask exécute simplement la fonction qui correspond à l'URL de la requête et renvoie une page HTML.

Ce framework manque de fonctionnalités et de modularité par rapport aux autres frameworks analysés ci-dessous. En effet, il n'y a rien d'implémenté pour gérer les bases de données ou pour utiliser des *templates* HTML plus complexes. Il n'y a pas de structure de projet standardisée et proposée par défaut. De plus, il utilise la convention WSGI (*Web Server Gateway Interface*), ce qui ne permet de traiter que des requêtes synchrones.

7.1.2 Quart

Quart [30] est une version asynchrone de Flask développée par la même équipe. L'utilisation d'une convention de requêtes asynchrones, telle que ASGI (*Asynchronous Server Gateway*

Interface) permet de meilleures performances. Malheureusement, Quart possède les mêmes limitations que celles présentes dans Flask.

7.1.3 Django

Django [31] est un framework web Open Source écrit en Python. C'est l'un des frameworks les plus répandus [32] et il est, ou a été, utilisé par de nombreux sites populaires comme Instagram, Pinterest et Mozilla [33].

Ce framework est compartimenté en applications web [34]. L'idée est de séparer le code en différents blocs afin de permettre une meilleure réutilisation de ces applications dans différents projets. Grâce à la grande communauté de développeurs Django, il est possible de trouver plus de 4000 applications Django existantes qu'on peut intégrer au projet ou dont on peut s'inspirer [35].

Ce framework suit le *design pattern* MVT (Model-View-Template) [36]. Dans ce *pattern*, l'architecture d'un site est divisée en trois parties distinctes :

- les **Models** dans lesquels on gère les données et les bases de données
- les **Templates** dans lesquels on gère la logique liée à l'affichage et l'interface utilisateur (front-end).
- les **Views** dans lesquelles on traite toute la logique de l'application (back-end). Les Views font également le lien entre les Models et les Templates.

Afin de faciliter les échanges entre les Views et les Templates, Django intègre des *gabarits*. Ce sont des pages HTML qui intègrent des balises spéciales supplémentaires. Ces balises permettent aux Views d'envoyer des données dynamiques ou des objets Python aux Templates pour générer les pages HTML.

Django intègre également nativement un ORM [37]. Un ORM (object relational mapping) est une technique de programmation qui permet d'utiliser les bases de données relationnelles comme s'il s'agissait de bases de données orientées objet. Avec cette technique, il ne faut plus directement faire des requêtes SQL. On peut directement manipuler les données sous forme d'objet. L'ORM rend le code plus lisible et plus facilement maintenable.

Django supporte les deux standards ASGI et WSGI qui permettent de traiter des requêtes synchrones et/ou asynchrones.

Pour finir, Django possède également différentes protections activées par défaut contre les attaques de types « *Cross-site scripting* », « *Cross-site request forgery* », les injections SQL, les piratages de mots de passe ainsi que d'autres attaques courantes [38].

Pour l'implémentation de ce projet, le framework Django a été choisi car il a de nombreux avantages par rapport aux autres frameworks présentés.

7.2 Architecture globale

Cette section vise à définir les grandes notions utilisées dans le développement de sites web ainsi que les interactions entre les principaux éléments.

L'application est divisée en deux grandes parties, le front-end et le back-end (Figure 6).

Le front-end correspond à tout le code exécuté par le navigateur, directement sur la machine de l'utilisateur. Son rôle est principalement lié à l'affichage et l'interaction avec l'utilisateur. Les technologies utilisées pour cette partie sont le HTML, CSS et JavaScript.

Le back-end correspond à tous les traitements exécutés sur le serveur. Cela englobe à la fois la logique de l'application, le traitement de données et le traitement des échanges avec le front-end. Dans ce projet, le back-end utilise un framework Django, en Python.

Le front-end et le back-end utilisent des requêtes HTTP pour communiquer et échanger des informations. Le front-end peut envoyer deux types de requêtes. La méthode GET permet au front-end de récupérer les données venant du back-end, généralement le code HTML et JavaScript de la page ainsi que les ressources nécessaires pour afficher la page demandée. La méthode POST permet au front-end d'envoyer des informations vers le back-end. En retour de ses requêtes, le back-end renvoie une réponse avec les informations demandées ou un code d'erreur.

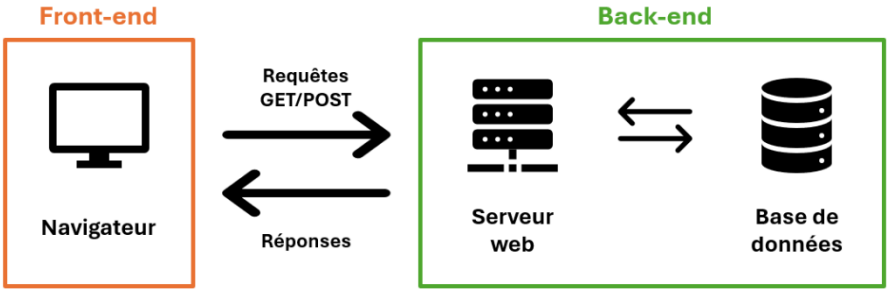


Figure 6. Schéma des interactions entre le back-end et le front-end

7.2.1 Architecture du back-end

Les fichiers de ce projet respectent la structure conventionnelle (Figure 7) pour un framework Django [39].

Un certain nombre de fichiers de configuration globaux du projet *my_project* se trouvent dans le sous-dossier *my_project*. Le nom de ce sous-dossier est par convention identique au nom du projet. Chaque application contient ses propres fichiers de configuration dans son sous-dossier.

Chaque application contient au minimum les fichiers suivants :

- admin.py : contient la configuration de la page d'administration
- models.py : définit les modèles de l'application
- urls.py : fait le lien entre les URL et les fonctions à exécuter
- views.py : contient la logique pour chaque page

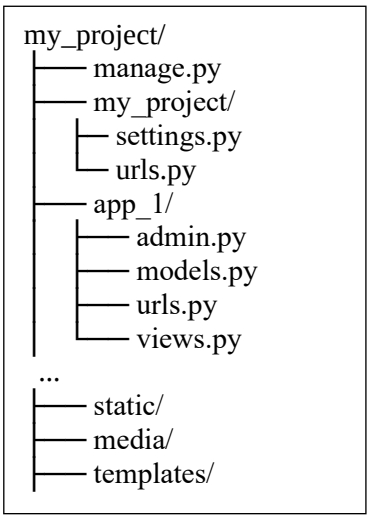


Figure 7. Structure d'un projet Django

Les fichiers média, correspondant aux scans des documents à transcrire, sont placés dans le dossier /*media*.

7.2.2 Architecture du front-end

Le front-end est simplement constitué de pages HTML envoyées par le back-end. Il n'y a pas de framework pour le front-end. Les *templates* (HTML, CSS, JavaScript) qui sont communs à tout le projet se situent dans les dossiers /*static* et /*templates* à la racine du projet. Les autres *templates*, spécifiques à une seule application, se trouvent dans les dossiers /*static* et /*templates* dans cette application. Cette structure permet d'éviter la duplication de code ainsi que de garder tout le code relatif à une application à l'intérieur de celle-ci.

7.2.3 Architecture de la base de données

Ce projet utilise une seule base de données en SQLite. Chaque application possède ses propres *modèles* Django. Un modèle est la définition d'un objet de données de l'ORM et correspond à une table en base de données. Chaque modèle contient un identifiant unique (ID) qui sert de clé primaire. L'ORM l'utilise implicitement pour établir les relations entre les modèles. L'explication détaillée des modèles utilisés par chaque application sera donnée ultérieurement dans les chapitres dédiés.

7.3 Application d'authentification

Le rôle de cette application est de gérer la connexion, la déconnexion et la création de comptes par les utilisateurs. L'authentification des utilisateurs permet une gestion des utilisateurs avec différents rôles, pour mieux assigner les tâches aux bons utilisateurs.

L'implémentation de cette application a été inspirée par l'application d'authentification présentée dans la formation « *Allez plus loin avec le framework Django* » d'Openclassroom [40].

7.3.1 Base de données

L'application d'authentification définit un modèle de données ***User*** qui étend le modèle ***AbstractUser***. Il s'agit d'un des modèles fournis par défaut dans Django pour stocker le compte d'un utilisateur.

Dans ce modèle, de nombreux champs sont déjà implémentés et réutilisés dans ce projet (Tableau 2).

- username : stocke le nom de l'utilisateur.
- password : stocke le hash du mot de passe pour ne pas le stocker en clair.
- superuser_status : un booléen qui permet de savoir si l'utilisateur est un *superuser* Django, ce qui lui permet notamment d'avoir accès au panneau administrateur.
- date_joined : représente la date de création du compte.
- last_login : représente la date de dernière connexion.

En plus de cela, le modèle ***User*** possède un champ supplémentaire appelé *completed_tasks* qui est un entier permettant de compter le nombre de tâches réalisées par l'utilisateur sur le site. Ce champ n'est pas utilisé pour le moment mais il pourra être utilisé dans un développement futur.

Champ	**Type**	**Propriétés**
username	CharField	Maximum 150 caractères
password	Hash	Algorithme : PBKDF2, hash SHA256
is_superuser	BooleanField	Valeur par défaut : False
date_joined	DatetimeField	
last_login	DatetimeField	
completed_tasks	IntegerField	Valeur par défaut : 0

Tableau 2. Description des champs du modèle AbstractUser

7.3.2 Back-end

Cette application utilise des fonctions du package *django.contrib.auth* afin de faire le lien entre la requête venant du front-end et la base de données.

Pour la page de connexion et celle de déconnexion, on utilise directement les fonctions natives de Django (*authentication* et *logout*). Ces fonctions s'occupent directement de l'intégralité de la connexion et de la déconnexion à partir de la requête du front-end.

Pour la page permettant de créer un compte, le formulaire HTML envoyé par l'utilisateur est converti en un *Form* Django (étendu de la classe *UserCreationForm*). Cela permet d'utiliser des fonctions natives de Django pour vérifier si le *username* n'est pas déjà utilisé et si le mot de passe et la confirmation du mot de passe sont identiques et assez robustes.

Dans le fichier *settings.py*, il y a la possibilité de spécifier une liste de *validateurs* de mot de passe [41] pour s'assurer d'avoir un mot de passe robuste. La liste proposée par défaut par Django demande d'avoir un mot de passe pas trop ressemblant au *username*, avec au moins 8 caractères, de ne pas être un mot de passe répandu, ni d'être uniquement numérique.

Une fois la création du compte validée, l'utilisateur est automatiquement connecté et redirigé sur la page principale.

7.3.3 Front-end

Les pages de connexion et de création de comptes sont des simples pages HTML contenant un formulaire permettant de remplir les informations demandées (Figure 8 et Figure 9).

Pour la déconnexion, il n'y a pas de page dédiée. Lorsque l'utilisateur est connecté, le bouton qui renvoie vers la page de connexion est remplacé par un bouton pour se déconnecter.

La suppression d'un compte n'est pas possible par un utilisateur. Seul un *superuser* peut en supprimer via la page d'administration. Cette suppression effacera tout le travail fourni par cet utilisateur de manière rétroactive. (Une autre méthode permet de bloquer un utilisateur sans supprimer tout son travail.)

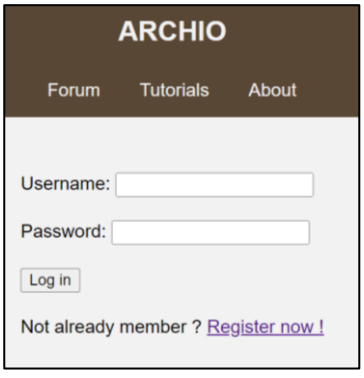


Figure 8. Page de connexion

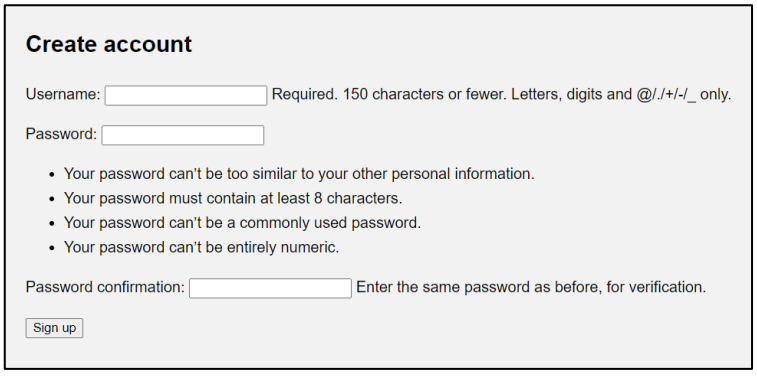


Figure 9. Page de création de compte

7.4 Application principale

Cette application Django est la partie principale du projet Archio. Elle gère tout le processus de transcription de documents, de l'ajout d'un document jusqu'à la finalisation de son traitement. Dans cette application, tout tourne autour de la notion de « document ». Il faut faire la distinction entre un document d'archives tel qu'utilisé aux Archives de l'État et un document d'un point de vue informatique. Dans ce projet, le terme « document » réfère à un document informatique et correspond à un scan d'une seule page manuscrite.

Comme cela a été décrit dans l'analyse fonctionnelle, un document va passer par deux grandes phases : la transcription et la vérification.

7.4.1 Base de données

Cette application contient de nombreux modèles permettant la gestion des documents et de leur transcription (Figure 10).

Le modèle central de cette application est le modèle ***Document*** et représente une image à transcrire. Ces documents sont compartimentés dans différents ***Projects***.

Chaque document est segmenté selon 2 axes. Un premier axe temporel à l'aide du modèle ***Task*** qui représente les différentes tâches réalisées par les utilisateurs au fil du temps.

Un second axe, spatial, qui divise le document en ***Rectangles***. Un ***Rectangle*** représente une zone du document équivalente à une ligne de texte.

Un modèle ***Subtask*** permet de stocker la transcription réalisée pour un ***Rectangle*** pendant une ***Task***. Les ***Subtasks*** permettent de garder un historique précis des évènements tout en permettant une architecture plus flexible sur le nombre de ***Rectangle*** et de ***Tasks***.

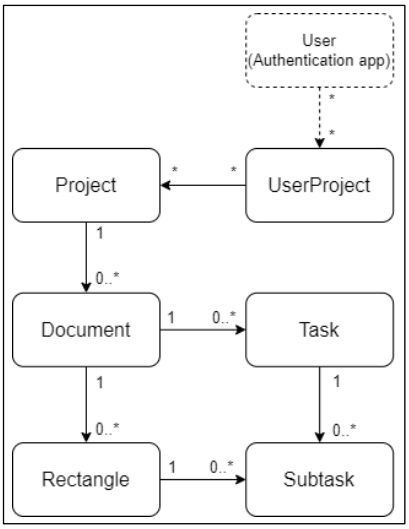


Figure 10. Schéma de relation des différents modèles de l'application principale

Le modèle ***UserProject*** permet de faire le lien entre les comptes venant de l'application d'authentification et les différents rôles que les utilisateurs ont pour chaque projet.

Description des modèles

Pour chaque modèle présenté, un tableau récapitulatif technique est repris dans l'annexe B.

Project

Ce modèle permet de regrouper des documents et des utilisateurs afin de mieux compartimenter le travail. Ce modèle contient deux champs :

- name : représente le nom du projet.
- description : contient la description du projet.

Document

Un ***Document*** correspond au scan d'une page manuscrite. Il contient les champs suivants :

- image : contient le chemin vers l'image du document.
- project : lie un document à son projet. Lorsqu'un projet est supprimé, tous les documents qui en font partie sont également supprimés en cascade.
- status : stocke l'état d'avancement du document. Ce champ peut prendre les valeurs : "to_transcribe", "in_transcription", "to_verify", "in_verification" ou "completed".
- name : représente le nom du document.
- archive_name : stocke le nom de l'archive. Ce champ permet de regrouper différents documents qui appartiennent à la même archive manuscrite. Un document papier peut avoir plusieurs pages scannées.

Task

Ce modèle représente une tâche de transcription ou de vérification réalisée par un utilisateur. Ce modèle contient les champs suivants :

- document : lie la tâche avec son document correspondant. Si le document est supprimé, la tâche est également supprimée en cascade.
- user : lie la tâche avec l'utilisateur qui réalise celle-ci. Si l'utilisateur est supprimé, la tâche est également supprimée en cascade.
- started : représente la date à laquelle la tâche a été commencée.
- ended : représente la date à laquelle la tâche a été terminée.
- status : indique l'état dans lequel la tâche se trouve. Ce champ peut avoir les valeurs "in_progress", "completed", "cancelled" ou "timeout".
- type : indique le type de tâche dont il s'agit : soit "transcription", soit "verification".

Rectangle

Un **Rectangle** représente une zone de l'image du document correspondant à une ligne de texte. Cela est utilisé pour fragmenter la tâche de transcription en plus petites parties. Ce modèle contient les champs suivants :

- document : lie le rectangle à son document. Si le document est supprimé, les rectangles associés le sont également.
- x : représente la coordonnée verticale du coin supérieur gauche du rectangle.
- y : représente la coordonnée horizontale du coin supérieur gauche du rectangle.
- width : représente la longueur du rectangle.
- height : représente la hauteur du rectangle.
- angle : représente l'angle d'inclinaison du rectangle.
- done : indique si la transcription du rectangle est terminée ou non.

Pour les champs *x, y, width* et *height*, les valeurs numériques représentent un nombre de pixels de l'image.

Subtasks

Une ***Subtask*** est l'intersection entre une ***Task*** et un ***Rectangle***. Une ***Subtask*** contient principalement la transcription réalisée pour un rectangle précis lors d'une tâche précise.

Cette approche permet de garder un historique précis de chaque rectangle lors de chaque tâche avec la possibilité d'avoir un nombre variable de tâches.

Ce modèle possède les champs suivants :

- rectangle : lie la Subtask avec son rectangle correspondant. Si le rectangle est supprimé, la Subtask est également supprimée en cascade.
- task : lie la Subtask avec sa Task correspondante. Si la Task est supprimée, la Subtask est également supprimée en cascade.
- text : contient la transcription du rectangle faite lors de cette tâche.
- time : représente la date de création de la Subtask.
- status : indique si la Subtask est en cours ou finie. Elle est considérée comme en cours lorsqu'un utilisateur a déjà proposé une transcription pour un rectangle et qu'il a sauvegardé sans soumettre son travail.

UserProject

UserProject fait le lien entre les comptes des utilisateurs provenant de l'application d'authentification et les différents projets. Ce modèle contient les champs suivants :

- user : lie le UserProject avec son User correspondant. Si l'utilisateur est supprimé, ses UserProject sont également supprimés en cascade.
- projet : lie le UserProject avec son Projet correspondant. Si le projet est supprimé, le UserProject est également supprimé en cascade.
- role ; indique le rôle de l'utilisateur dans ce projet. Les rôles possibles sont : "manager", "verifier","transcriber", "candidate" et "block".
- last_participation : indique la date de la dernière participation de l'utilisateur dans ce projet.
- participation_amount : compte le nombre de tâches effectuées par cet utilisateur au sein du projet.

Les champs last_participation et participation_amount ne sont pas utilisés pour le moment mais ils pourront être utilisés dans de futurs développements.

7.4.2 Pages web

Dans cette section, les différentes pages de l'application principale vont être présentées, à la fois du point de vue de l'interface, mais aussi du point de vue du back-end qui construit ces pages HTML. Les pages principales sont illustrées avec des captures d'écran. Les captures d'écran en haute résolution de chaque page sont reprises dans l'annexe C.

Page d'accueil

La page d'accueil est une page générique affichant la liste des projets existants (Figure 11). Un bouton, généré pour chaque projet, redirige l'utilisateur vers la page du projet. L'utilisateur n'a pas besoin d'être connecté pour avoir accès à cette page. Lorsqu'un utilisateur fait un appel à cette page, le back-end récupère tous les projets existants et renvoie la page HTML correspondante avec la liste des projets.

L'en-tête de la page est commun à toutes les autres pages du projet. On peut y retrouver des liens vers les pages principales des différentes applications ainsi que les pages annexes d'information.

Figure 11. Page d'accueil

Page d’un projet

Chaque projet est accessible via une page **/project** (Figure 12) qui permet d’avoir toutes les informations le concernant.

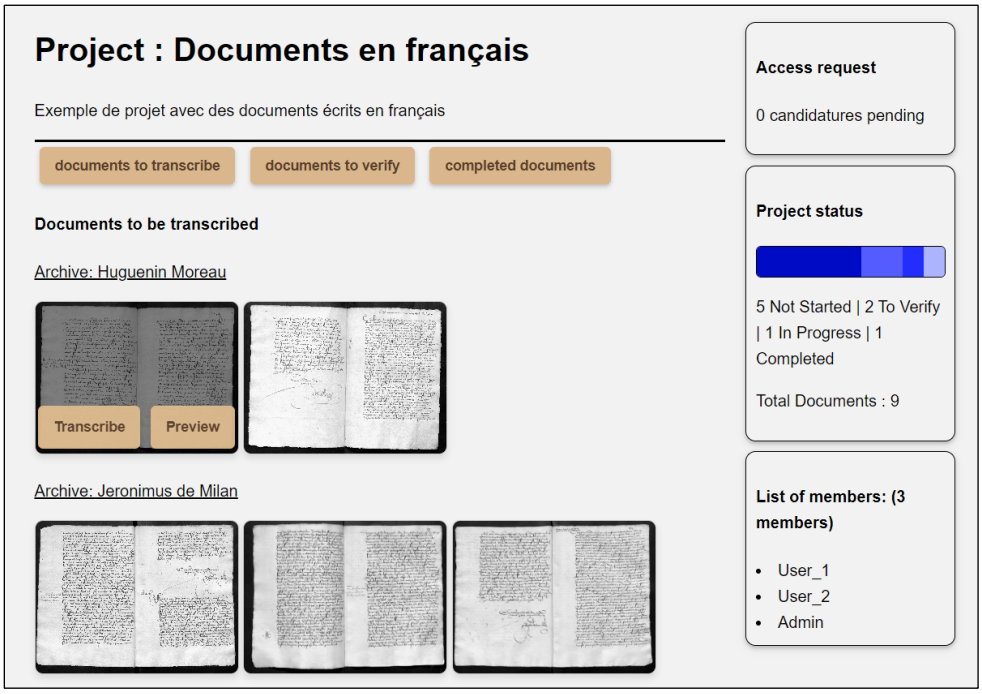


Figure 12. Page d'un projet

La page est divisée en trois grandes parties :

La partie supérieure (zone rouge de la Figure 13) contient le titre et la description du projet. C’est dans cette partie qu’on pourrait ajouter dans le futur des consignes, des informations ou des métadonnées sur le projet.

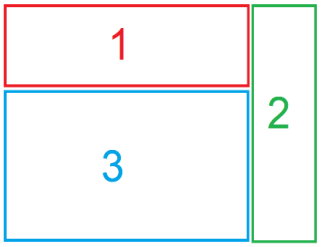


Figure 13. Division de la page d'un projet

La partie droite de la page (zone verte de la Figure 13) contient des informations relatives au projet. L’état d’avancement du projet est affiché avec le nombre de documents à transcrire, à vérifier et ceux terminés. Une liste des membres du projet est également affichée.
Dans cette partie, une section permet de gérer les demandes d’accès au projet. Un bouton de demande d’accès est affiché pour les utilisateurs ne faisant pas partie de ce projet. Pour les gestionnaires, cette zone indique le nombre de demandes en attente. Pour accepter une candidature, un *superuser* doit créer un accès dans la table *UserProject* via l’interface d’administration. Ce système d’accès est réduit à son minimum dans ce prototype mais il pourra faire l’objet d’une amélioration lors d’un développement futur.

Pour finir, la partie inférieure (zone bleue de la Figure 13) affiche les documents appartenant au projet. Ceux-ci sont répartis dans différents onglets en fonction de leur état d'avancement : *to transcribe*, *to verify* et *completed*.

Dans ces onglets, chaque document est représenté par une miniature de son image.

Les documents ayant le même archive_*name* sont regroupés afin de permettre à l'utilisateur de trouver plus facilement des documents provenant du même document manuscrit. L'utilisateur n'a plus accès aux documents pour lesquels il a déjà réalisé une tâche.

Dans les onglets relatifs à la transcription et à la vérification, il y a une sous-section qui montre les documents qui sont en cours de traitement par d'autres utilisateurs.

Quand l'utilisateur a déjà un document en cours de traitement dans ce projet, une section en haut de la page va s'afficher pour l'inviter à reprendre la tâche commencée car il n'est pas possible pour un utilisateur d'avoir en même temps plusieurs tâches différentes en cours.

Lorsqu'on passe la souris sur une miniature, des boutons apparaissent :

Le premier bouton, *transcribe* ou *verify* en fonction de l'onglet, redirige l'utilisateur vers la page dans laquelle il peut réaliser la tâche demandée. Ce bouton n'est visible que pour les utilisateurs ayant le rôle requis pour réaliser cette tâche. Il n'est également pas visible pour les utilisateurs qui ont déjà une tâche en cours.

Le second bouton *preview* permet de rediriger l'utilisateur vers la page ***/preview*** pour lui permettre de visualiser le document sans devoir commencer la tâche. La prévisualisation du document est disponible pour tous les utilisateurs.

Dans l'onglet des documents terminés, il n'y a qu'un seul bouton *consult* permettant d'aller sur la page de consultation du document achevé. Ce bouton est disponible pour tous les utilisateurs.

Pour générer cette page, le back-end récupère de la base de données tous les documents du projet regroupés en fonction de leur statut de progression et de leur *archive_name*. Il récupère également les informations concernant le document déjà commencé par l'utilisateur, le cas échéant. Pour l'affichage de la partie de droite, le back-end envoie la liste des membres du projet ainsi que les statistiques sur la progression du projet.

La page de visualisation

Cette page ***/preview*** est une page générique qui permet de visualiser un document sans devoir commencer la tâche en elle-même. Lorsque le back-end reçoit cette requête, il retourne simplement le *template* HTML avec l'image correspondante.

Afin de pouvoir visualiser le document au mieux, l'image est intégrée dans un *canvas* HTML dans lequel il est possible de zoomer et de se déplacer. Ce point sera plus détaillé dans la page de transcription.

La page de transcription

Cette page ***/transcription*** permet à l'utilisateur de réaliser la tâche de transcription d'un document (Figure 14).

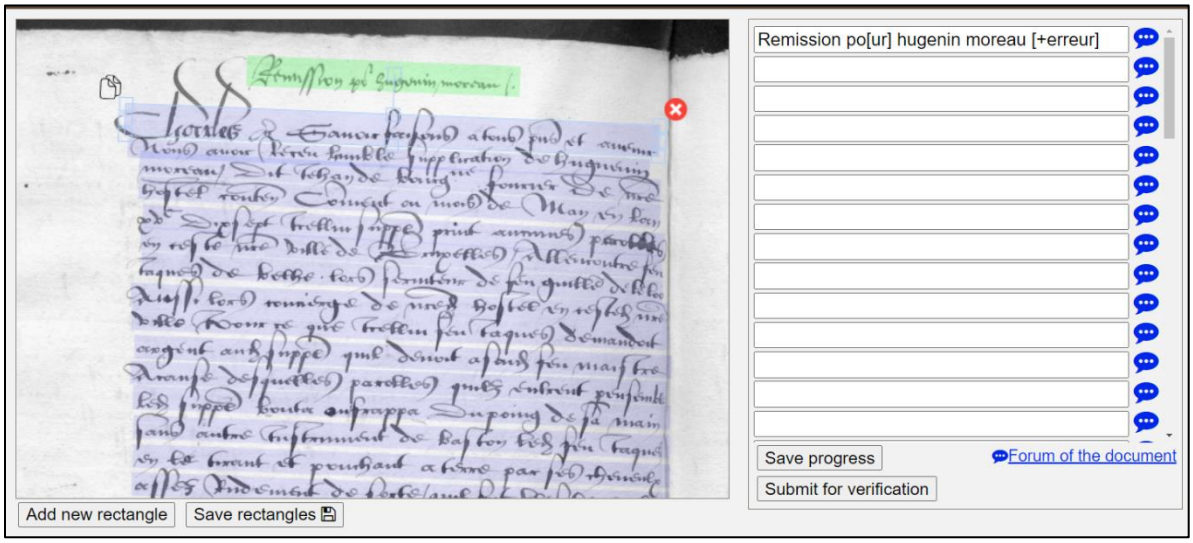


Figure 14. Page de transcription

À gauche de la page se trouve l'image du document à transcrire. Celle-ci est contenue dans un *canvas* HTML. Cela permet à l'utilisateur de zoomer et de dézoomer avec la molette de la souris et de naviguer dans l'image en maintenant le bouton gauche de la souris enfoncé.

L'utilisateur doit superposer un rectangle sur chacune des lignes du document. Pour cela, une fois le rectangle créé, l'utilisateur peut le déplacer, modifier sa hauteur et sa largeur et changer son inclinaison pour correspondre le plus fidèlement à la forme de la ligne de texte. La librairie fabric.js [42] a été utilisée dans le front-end pour permettre de superposer des rectangles sur une image dans un *canvas*.

En bas à gauche, il y a un bouton *add new rectangle* qui permet de créer un rectangle sur l'image.

Lorsqu'on sélectionne un rectangle, deux icônes apparaissent. Celle de droite (⊗) permet tout simplement de supprimer le rectangle. L'icône de gauche (⧉) permet de créer un nouveau rectangle sur base de celui sélectionné. Le nouveau rectangle possède les mêmes propriétés (largeur, hauteur, inclinaison) et apparait en dessous du précédent. Pour la grande majorité des documents, les lignes de texte ont les mêmes propriétés. (L'inclinaison du texte est souvent la même pour toute la page due à la méthode de numérisation). En dupliquant le rectangle de la ligne précédente, on simplifie et réduit le temps requis pour effectuer cette étape de labellisation. Une fois le duplicata créé, il n'y a plus qu'à faire quelques ajustements éventuels.

En bas à gauche, se trouve également un bouton *save rectangles* qui permet de sauvegarder les rectangles et leurs propriétés en base de données. Les rectangles, initialement en rouge, deviennent bleus lorsqu'ils sont sauvegardés.

Sur la droite de la page, il y a une liste de champs de saisie correspondant aux rectangles enregistrés. Quand l'utilisateur clique sur une entrée, le rectangle correspondant devient vert pour permettre à l'utilisateur de facilement retrouver la ligne correspondante qu'il doit transcrire dans ce champ de saisie.

En dessous de la liste des champs, il y a un bouton *save progress* qui permet à l'utilisateur de sauvegarder ses transcriptions. Une fois la sauvegarde effectuée, l'utilisateur est redirigé vers la page du projet et pourra reprendre cette tâche ultérieurement.

Un autre bouton *submit for verification* permet de finaliser la tâche lorsque l'utilisateur l'a terminée.

Pour le développement de cette page, le back-end doit également récupérer les informations envoyées par les utilisateurs lorsqu'ils soumettent ou sauvegardent une transcription. On utilise donc une *class-based view* [43] au lieu d'une simple fonction. Cette classe fournie par Django permet au back-end de traiter différemment les méthodes GET et POST.

Pour les méthodes GET, le back-end retourne la page avec les rectangles et les transcriptions, s'il y en a, qui ont déjà été sauvegardés précédemment.

Pour les méthodes POST, le back-end sauvegarde les transcriptions faites en créant des ***Subtasks.*** S'il s'agit d'une soumission, la ***Task*** de transcription va se clôturer et le ***Document*** va passer en statut *to_verify*.

Lorsque l'utilisateur appuie sur *Save rectangles*, la page envoie une requête POST vers l'API ***/save_rectangles***. Cette API sauvegarde les nouveaux rectangles et met à jour les propriétés des rectangles si elles ont été changées.

L'utilisateur a un mois pour réaliser cette tâche. Passé ce délai, la tâche sera annulée par un script automatique (*cron task*) et le document sera de nouveau disponible à la transcription.

Page de vérification

Dans la page ***/verification*** (Figure 15), le vérificateur doit examiner la transcription réalisée lors de la première étape.

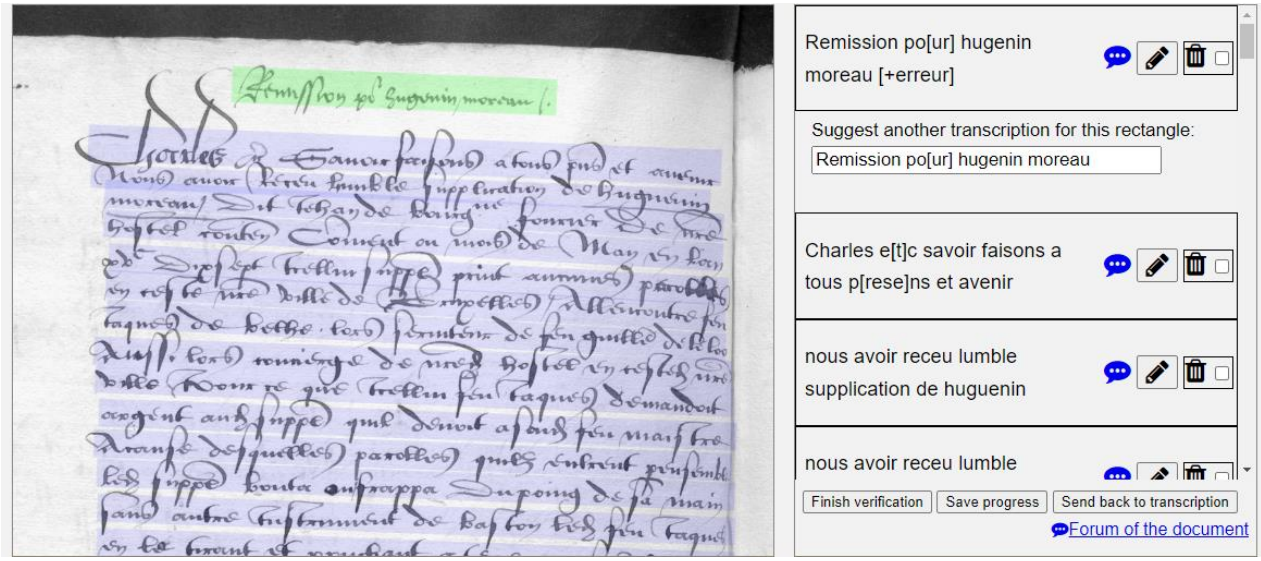


Figure 15. Page de vérification

L'écran est de nouveau séparé en 2 parties. La partie de gauche contient le *canvas* HTML avec l'image du document et les rectangles qui y ont été superposés dans l'étape précédente.

Dans la partie de droite, on retrouve la transcription faite pour chaque rectangle. Pour chacune de ces propositions, l'utilisateur peut effectuer plusieurs actions différentes. Il peut ne rien changer et accepter la proposition ou proposer une modification en cliquant sur le bouton avec un crayon (✎). Un menu va s'ouvrir pour permettre à l'utilisateur d'entrer sa nouvelle proposition. Le vérificateur peut également décider de supprimer le rectangle (🗑) si celui-ci ne correspond pas à une ligne dans le texte. Si l'utilisateur a des questions, il peut créer ou accéder aux différents forums (💬).

À la fin, l'utilisateur peut soit sauvegarder sa progression pour continuer sa tâche plus tard, soit cliquer sur le bouton *finish verification* pour terminer sa tâche.

Comme pour la page de transcription, cette page a besoin de recevoir et d'envoyer des informations, c'est pour cela qu'on utilise aussi une *class-based views* pour l'implémentation séparée des requêtes POST et GET de cette page.

Lorsque le back-end reçoit une requête GET, celui-ci va vérifier si le document est dans l'état *to_verify,* auquel cas il crée une ***Task*** et change le statut du document en *in_verification*. Si l'utilisateur avait déjà commencé la tâche précédemment, le back-end doit vérifier que le document est bien en cours de vérification et que l'utilisateur qui demande l'accès est bien l'auteur de la tâche en cours.

Une fois ces vérifications faites, le back-end va récupérer l'image, les rectangles et les dernières transcriptions de chacun via les ***Subtasks***. Il récupère également la progression sauvegardée le cas échéant. Le *template* HTML et toutes les informations sont ensuite envoyés au front-end.

Lorsque le back-end reçoit une requête POST, le back-end va commencer par vérifier si l'utilisateur sauvegarde ou soumet sa tâche.

S'il s'agit d'une sauvegarde, l'état d'avancement du document ne sera pas changé. L'utilisateur ne peut pas commencer une autre tâche avant d'avoir achevé celle-ci. Seules des ***Subtasks*** sont créées pour sauvegarder les modifications apportées aux rectangles.

Dans le cas où l'utilisateur soumet la tâche finie, les statuts du ***Document*** et de la ***Task*** deviennent *completed.*

Le back-end va ensuite traiter chaque rectangle en fonction de l'action qui a été faite.

- Si une modification a été proposée, une ***Subtask*** va être créée pour sauvegarder cette proposition.
- Si l'utilisateur a demandé de supprimer le rectangle, celui-ci est enlevé de la base de données.
- Si l'utilisateur n'a rien fait pour ce rectangle, le rectangle est considéré comme achevé (le champ *done* du ***Rectangle*** est mis à *true*). Ce rectangle ne pourra plus être modifié dans les étapes suivantes.

S'il y a eu des modifications et que la tâche est terminée, le statut du document va retourner à l'état *to_verify* pour procéder à une nouvelle vérification. À l'inverse, s'il n'y a pas eu de modifications, la transcription du document est terminée.

Une fois la requête traitée, l'utilisateur est redirigé vers la page du projet.

L'utilisateur a une semaine pour réaliser cette tâche. Passé ce délai, la tâche sera annulée par un script automatique (*cron task*) et le document sera de nouveau disponible à la vérification.

Page de consultation

Dans la page **/completed** (Figure 16), il est possible de voir le résultat de la transcription pour un document terminé. Cette page est accessible par tous les utilisateurs.

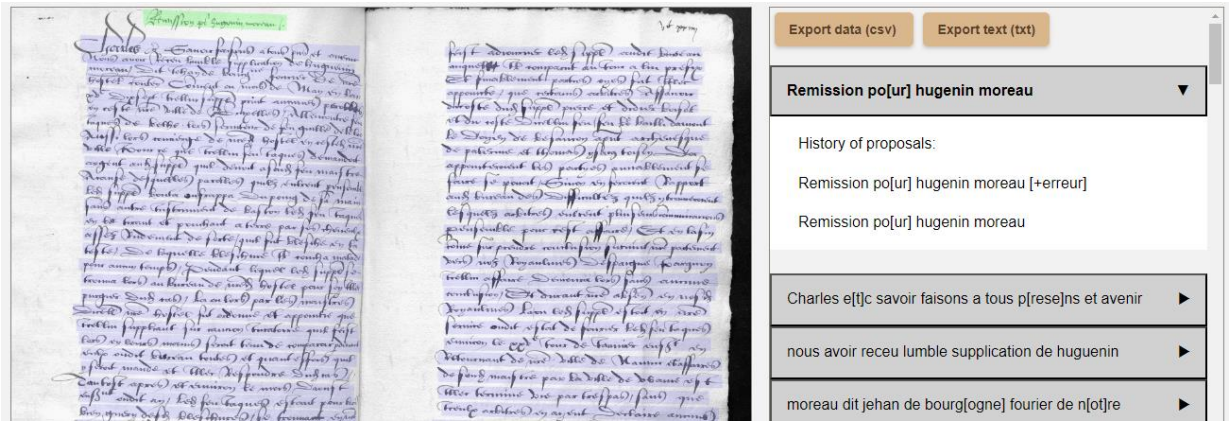


Figure 16. Page de consultation

On retrouve à gauche le *canvas* avec l'image du document et les rectangles associés.

Sur la droite, la transcription finale de chaque ligne est affichée indépendamment. Si on clique sur une de ces transcriptions, un menu déroulant s'ouvre et affiche l'historique des différentes transcriptions faites pour ce rectangle au fil des étapes. Il peut également y avoir un bouton redirigeant vers le forum existant de ce rectangle. Il n'est pas possible de créer un forum car le document est considéré comme terminé. Les documents et les rectangles qui n'ont pas de forum associé n'ont donc pas de bouton pour cela.

Pour générer cette page, le back-end récupère tous les rectangles du document ainsi que toutes les ***subtasks*** de chaque rectangle afin de pouvoir retracer l'historique des transcriptions. Le back-end récupère aussi la liste des forums existants relatifs à ce document.

En haut à droite, deux boutons permettent d'exporter les données sous deux formats différents. Le format *.txt* contient uniquement la transcription finale de tout le document. Le format *.CSV* regroupe, ligne par ligne, la transcription finale ainsi que les informations relatives des rectangles (position, longueur, hauteur, inclinaison). Le fichier *CSV* est plutôt destiné à être lu et utilisé par un programme informatique pour une analyse plus spécifique des données récoltées.

Pour les boutons d'export, la page fait une requête vers l'API de la page **/export**. Pour cette page, le back-end génère le fichier demandé sur base du type et de l'identifiant du document demandé. Il renvoie ensuite directement le fichier au navigateur.

Page d’ajout de documents

La page **/upload** (Figure 17) permet à un *superuser* d’ajouter des documents sur la plateforme. L’utilisateur sélectionne le projet auquel les documents sont assignés. Il dépose ensuite le document d’archives sous forme d’une ou plusieurs images au format jpeg ou png. Ces différentes images sont groupées avec le même *archive_name*. Il est toujours possible de changer l’*archive_name* d’un document via le panneau d’administration. Quand l’utilisateur clique sur le bouton *upload*, les ***Documents*** sont automatiquement créés et prêts pour la première étape de transcription.

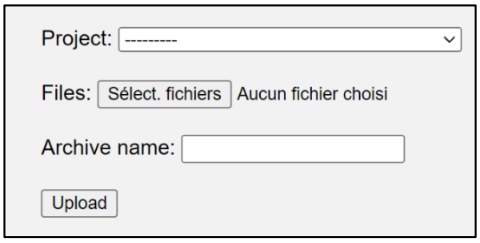


Figure 17. Page d'ajout de documents

Pages annexes

Les pages ***/help*** et ***/about*** sont deux pages permettant aux utilisateurs d’avoir des informations concernant ce site et ce mémoire. Ces pages ont notamment été utilisées comme mode d’emploi pour les utilisateurs lors de la phase de test de ce prototype. Il s’agit de simples pages HTML statiques, accessibles par tous les utilisateurs via les boutons dans l’entête du site. Le contenu de la page d’aide est disponible dans l’Annexe D *: mode d’emploi.*

Page d’administration

Le framework Django intègre de base une page ***/admin*** permettant aux *superusers* d’interagir avec la base de données via une page web. Pour chaque application et chaque modèle de données, il est possible de créer, lire, modifier et supprimer des données. L’interface d’administration de chaque application peut être légèrement personnalisée à l’aide du fichier *admin.py*. Pour chaque modèle, il est possible de définir quels champs doivent être visibles sur cette interface. Il est également possible de définir des *actions* plus complexes sur la base de données. Pour les ***Documents***, une telle action a été créée afin de permettre la duplication d’un document et toutes ses dépendances en un clic depuis cette interface.

C’est dans cette interface qu’il faut faire l’association entre les ***Users*** et les ***Projets*** en créant un ***UserProject*** et gérer les demandes d’accès aux projets.

Hormis les paramètres d’affichage configurés dans le fichier *admin.py*, le front-end et le back-end sont entièrement gérés par le framework lui-même.

7.5 Application de forum

La troisième et dernière application de la plateforme Archio est une application de forum qui permet aux utilisateurs de communiquer entre eux et de recevoir des feed-back sur leur travail. Cette application se base sur une application Django Open Source existante, développée par *Mahmudul Sajib* en 2020 [44].

Cette application possède deux grandes fonctionnalités :

- Un **système d'annonces** permettant aux administrateurs de communiquer des informations générales. Les annonces peuvent être uniquement publiées par un administrateur via le page d'administration. Une annonce est constituée d'un titre, d'un texte et éventuellement d'une image. Les autres utilisateurs peuvent lire ces annonces mais ne peuvent pas y répondre.

- Un **système de forums** permettant à tous les utilisateurs de discuter entre eux. Le forum est composé de différentes discussions, aussi appelées *topics*. Une discussion peut être créée par un utilisateur lors d'une étape de transcription ou de vérification. Chaque topic porte sur un document ou sur un rectangle spécifique. Tous les autres utilisateurs sont libres de participer à la discussion en répondant aux messages. Ils peuvent également donner une appréciation à un message (*like* ou *dislike*).

7.5.1 Base de données

Cette application contient quatre modèles de données Django : trois pour stocker les informations concernant le forum et un pour le système d'annonces.

Description des modèles

Pour chaque modèle présenté ci-dessous, un tableau récapitulatif technique est repris dans l'annexe B.

Announcement

Ce modèle stocke les informations des annonces et contient les champs suivants :

- title : stocke le titre de l'annonce.
- content : stocke le contenu textuel de l'annonce.
- timestamp : représente la date à laquelle l'annonce a été créée.
- thumbnail : contient le chemin vers l'image d'illustration. L'ajout d'une image est optionnel.

Topic

Ce modèle représente une discussion du forum à propos d'un document ou d'un rectangle. Ce modèle possède les champs suivants :

- author : lie le topic avec l'utilisateur qui l'a créé. Si l'utilisateur est supprimé, le topic est également supprimé en cascade.
- title : stocke le titre du topic.
- description : contient le message du topic.
- date_created : représente la date à laquelle le topic a été créé.
- related_document : lie le ***Topic*** avec le document sur lequel il porte. Si le document est supprimé, le T***opic*** est également supprimé en cascade.
- related_rectangle : lie le ***Topic*** avec le rectangle sur lequel il porte. Si le document est supprimé, le ***Topic*** est également supprimé en cascade.

Chaque topic étant associé exclusivement à un document ou à un rectangle, seul un des champs *related_document* et *related_rectangle* peut être défini.

TopicView

Ce modèle établit la relation entre un topic et les utilisateurs qui l'ont consulté afin de pouvoir connaitre le nombre d'utilisateurs qui ont lu la discussion.

Ce modèle contient deux champs :

- user : représente un ***User*** qui a consulté un topic. Si l'utilisateur est supprimé, ses ***TopicViews*** sont également supprimés en cascade.
- user_post : représente le ***Topic*** correspondant. Si le topic est supprimé, les ***TopicViews*** associés sont également supprimés en cascade.

Answer

Une ***Answer*** représente une réponse faite dans une discussion précédemment créée. Ces réponses peuvent être *likées* ou *dislikées* par les autres utilisateurs.

Ce modèle possède les champs suivants :

- user_post : lie la réponse avec le topic correspondant. Si le topic est supprimé, la réponse est également supprimée en cascade.
- user : lie la réponse avec l'utilisateur qui l'a rédigée. Si l'utilisateur est supprimé, la réponse est également supprimée en cascade.
- content : contient le message de la réponse.
- date_created : représente la date à laquelle la réponse a été rédigée.
- upvotes : lie la réponse avec les utilisateurs qui l'ont aimée.
- downvotes : lie la réponse avec les utilisateurs qui ne l'ont pas aimée.

Les champs *upvotes* et d*ownvotes* utilisent le modèle de données *ManyToManyField*. Cela permet de faire le lien entre les ***Users*** et les ***Answers*** en fonction des *likes* et des *dislikes*. Dans ce cas-ci, on ne crée pas de modèles intermédiaires car on n'a pas besoin de stocker d'autres informations dans une relation entre une réponse et un utilisateur.

7.5.2 Pages web

Dans cette section, les différentes pages de l'application de forum vont être présentées, à la fois d'un point de vue de l'interface mais aussi du point de vue du back-end et de la base de données. Une capture d'écran en haute résolution de chaque page est reprise dans l'annexe C.

Page de présentation des annonces

La page de présentation des annonces est une page générique qui affiche les différentes annonces existantes (Figure 18). L'utilisateur peut cliquer sur une annonce pour être redirigé vers la page dédiée. Lorsqu'un utilisateur va sur cette page, le back-end récupère toutes les annonces existantes en base de données et renvoie la page HTML correspondante avec la liste des annonces.

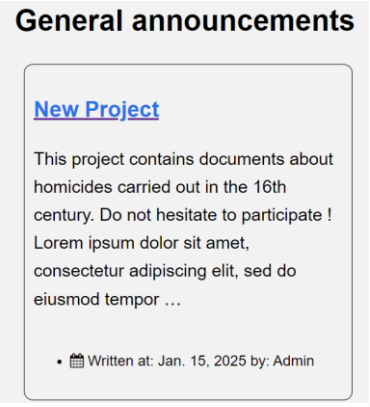


Figure 18. Page de présentation des annonces

Page d’une annonce

La page détaillée d’une annonce est une page générique qui affiche les informations de l’annonce. En plus du titre et du contenu textuel, cette page peut afficher une image le cas échéant.

Page de création d’un topic

Cette page est celle sur laquelle un utilisateur est redirigé lorsqu’il veut créer un topic pour un document ou un rectangle qui n’en a pas encore (Figure 19). Il est uniquement possible d’accéder à cette page depuis les boutons prévus (💬) dans les pages de transcription et de vérification. L’association avec le document ou le rectangle concerné se fait automatiquement. L’utilisateur doit uniquement indiquer le titre et le contenu du message pour créer un topic.

Figure 19. Page de création d’un topic

Page de présentation des topics

Cette page affiche les différents topics existants sous la forme de liste (Figure 20). Pour chaque topic, une prévisualisation du sujet, les nombres de vues et de réponses sont affichés ainsi que le nom de l’auteur et la date de création.

Lorsque l’utilisateur clique sur le topic, il est redirigé vers la page dédiée afin de voir la discussion dans son intégralité.

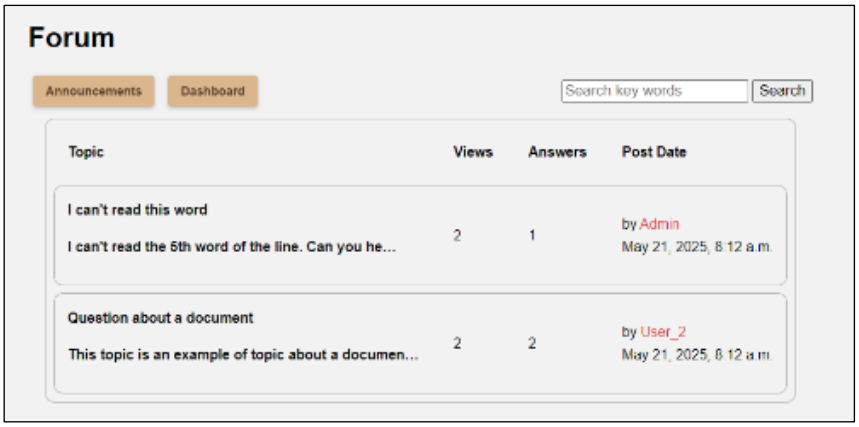


Figure 20. Page de présentation des topics

Page de recherche par mots-clés

En haut de la page de présentation des topics, une barre de recherche est visible. Elle permet aux utilisateurs d'effectuer une recherche à l'aide de mots-clés. La recherche porte uniquement sur le titre et le message initial d'un topic. Le back-end effectue une requête en base de données pour trouver les résultats correspondants. Il redirige ensuite l'utilisateur vers une autre page pour lui afficher les résultats sous une forme similaire à celle présentée précédemment (Figure 20).

Page d'une discussion

La page d'une discussion est divisée en plusieurs parties (Figure 21). La partie supérieure contient le message initial du topic avec le titre, le message et l'image d'illustration. Le système d'affichage du document est le même que celui utilisé dans l'application principale.

La partie inférieure de la page est dédiée aux réponses que chaque utilisateur peut écrire. Chaque réponse peut être *likée* ou *dislikée* par les autres utilisateurs.

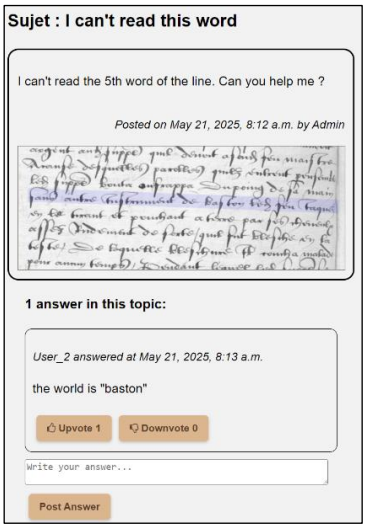


Figure 21. Page d'une discussion

Page de profil

Chaque utilisateur peut aller sur sa page de profil (Figure 22), en cliquant sur le bouton *Dashboard* dans la page de présentation des topics. Comme pour la page principale du forum, cette page va afficher les différents topics existants mais ici les topics sont filtrés. Seuls les topics créés par l'utilisateur sont affichés dans le premier tableau. Le second tableau indique les topics dans lesquels l'utilisateur a écrit une réponse. Cela permet à l'utilisateur de trouver plus rapidement des topics pour lesquels il a déjà manifesté un intérêt. Pour afficher la page, le back-end va simplement récupérer les topics qui correspondent aux filtres.

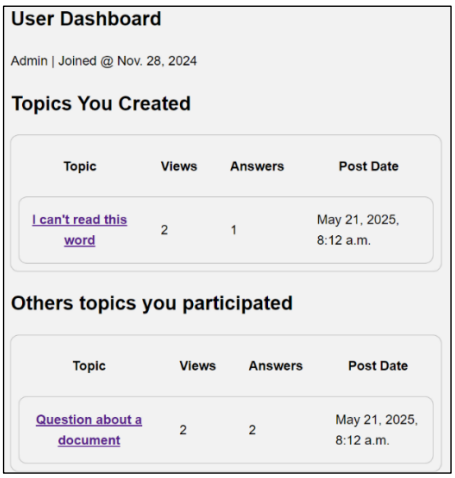


Figure 22. Page de profil

7.6 Déploiement

Pour réaliser des tests avec les archivistes, la plateforme Archio a été déployée sur un serveur web hébergé par l'UCLouvain (http://tfe-sidgwick.info.ucl.ac.be/). Cette application a été déployée dans une machine virtuelle en Linux (Ubuntu) et dans un environnement virtuel Python 3.10 avec les dépendances requises. Les librairies Django 5.1.3 et Pillow 11.1.0 sont requises pour ce projet. Ce déploiement a été fait avec la commande *runserver*. Cette commande permet un déploiement rapide pour des tests mais cette approche ne convient pas pour un système en production.

Pour la mise en production, Django permet d'utiliser différents serveurs web. Pour WSGI (Web Server Gateway Interface), il est notamment possible d'utiliser des serveurs tels qu'Apache, Gunicorn ou uWSGI. Pour l'interface ASGI (*Asynchronous Server Gateway Interface*), les serveurs de production recommandés par Django sont Daphne, Hypercorn et Uvicorn [45].

8 Analyses de la plateforme

Après le développement de la plateforme, deux analyses ont été réalisées : une analyse pratique de l'interface, réalisée par des archivistes, et une analyse de sécurité.

8.1 Tests et retours d'utilisateurs

Il a été demandé à deux archivistes de tester la plateforme Archio (http://tfe-sidgwick.info.ucl.ac.be/) en réalisant les étapes de transcription d'un document afin de valider son bon fonctionnement.

8.1.1 Premier retour d'utilisateur

Le premier test a été réalisé par Dr Gert Gielis (archiviste responsable du projet PARDONS). Il en ressort que les fonctionnalités sont conformes aux attentes. L'ajout et le déplacement des rectangles sur une image sont fluides. La rapidité du processus est également grandement améliorée par rapport au processus actuel.

Lors de l'utilisation de ce prototype, certains axes d'amélioration ont été proposés par Dr Gielis. Les améliorations apportées à la plateforme en réponse à ce retour d'utilisateur vont être expliquées ci-dessous.

La navigation dans l'image

Les touches du clavier utilisées pour se déplacer dans l'image étaient peu intuitives pour des utilisateurs peu familiers avec l'informatique. Il fallait presser différentes touches si on souhaitait déplacer l'image ou si on voulait déplacer un rectangle. Les raccourcis clavier ont donc été simplifiés et rendus plus intuitifs. Le zoom par défaut appliqué à l'image a été modifié afin que l'image puisse directement s'adapter à l'écran indépendamment de la taille de l'écran et de l'image.

L'organisation des documents dans la page d'un projet

Un document d'archives peut être constitué de plusieurs images (pages). Pour rendre l'interface plus claire, la présentation de la page d'un projet a été améliorée pour regrouper toutes les images appartenant à la même archive. Cette amélioration a également impliqué une modification au niveau de la page d'ajout de document. Il est désormais possible d'ajouter de multiples images en même temps. Ces images partagent alors les mêmes métadonnées (nom d'archive et projet).

L’export dans un format adapté

Initialement, seul l’export au format *CSV* avait été prévu. Suite à ce retour, l’export au format texte a été ajouté. Le code a également été adapté pour faciliter l’ajout d’autres formats dans de futurs développements.

Certaines propositions d’amélioration étaient trop spécifiques au projet PARDONS. Elles n’ont donc pas été implémentées afin de garder l’aspect général de cette plateforme.

Reconnaissance automatique des lignes de texte.

Il serait possible d’améliorer le processus et de le rendre un peu plus rapide en automatisant la labellisation des lignes. Cette amélioration n’a pas été réalisée car elle mobilise des notions de traitement d’images et de reconnaissance de texte qui sortent du cadre de ce mémoire. De plus, il est déjà possible de dupliquer facilement les rectangles d’une ligne à l’autre. La labellisation est déjà efficace et réalisable rapidement.

L’ajout de métadonnées

Il a été demandé s’il était possible d’ajouter des métadonnées supplémentaires directement sur la plateforme.

Si on veut ajouter des métadonnées relatives à un projet, il faudra ajouter des champs dans le modèle ***Project***. Ces informations pourront être affichées dans la page du projet si nécessaire. Actuellement, il est seulement possible d’indiquer des métadonnées dans la description du projet.

S’il s’agit de métadonnées portant sur un document d’archives, il est possible de rajouter des champs texte dans la page d’ajout de documents. Ces métadonnées seront stockées pour chaque image dans le modèle ***Document.*** Il sera ensuite possible d’afficher ces métadonnées dans chacune des pages où un document est affiché (prévisualisation, transcription, vérification, forum, etc).

Cette amélioration n’a pas été implémentée car les métadonnées sont différentes pour chaque projet. Il a été décidé de garder une approche généraliste dans le développement de cette plateforme. De plus, la plupart des métadonnées n’ont pas d’impact sur le processus de transcription ni sur sa qualité.

8.1.2 Second retour d'utilisateur

Un second test a été réalisé par Dr Klaas Van Gelder (archiviste de l'État et professeur adjoint à l'Université Libre de Bruxelles). Cet archiviste n'a pas été précédemment impliqué dans le projet, ni pendant de la création du cahier des charges ni lors du développement. Il en ressort que la création des rectangles est très simple et que la plateforme est très facile d'utilisation. On peut également remarquer que l'interface est intuitive pour une personne extérieure au projet et que la page de tutoriels explique correctement le processus de transcription. Comme lors du premier test, il a été remonté que la reconnaissance automatique des lignes serait une fonctionnalité intéressante. Pour finir, Dr Van Gelder a également manifesté son intérêt et celui de bénévoles pour des logiciels faciles d'utilisation tels que cette plateforme.

8.2 Analyse de sécurité

Dans cette analyse de sécurité, nous allons identifier les grands risques d'attaque, leur vraisemblance, leur gravité et les mesures mises en place pour les éviter.

8.2.1 Les risques d'attaque

Attaques communes à tous les sites web

De nombreuses attaques sont communes à tous les sites web. Django propose déjà de nombreuses protections contre celles-ci [46].

Les requêtes en base de données effectuées par Django sont protégées des injections SQL (« *L'injection SQL est un type d'attaque où un utilisateur malveillant est capable d'exécuter du code SQL arbitraire sur une base de données.*» [46]). Dans ce projet, les requêtes passent par l'ORM (Object-Relational Mapping) qui garantit une protection contre ces attaques.

Django permet également d'utiliser le protocole HTTPS qui encrypte les données pour empêcher un utilisateur malveillant d'intercepter ces données sur le réseau. Cette mesure de protection doit être activée sur le serveur lors de la mise en production du site.

Des protections sont également intégrées par défaut dans Django pour protéger contre les attaques de types « *Cross-site scripting* », « *Cross-site request forgery* » et « *clickjacking* ». Ces attaques visent à altérer une page web pour tromper l'utilisateur ou pour exécuter du code malveillant.

Attaques via les données encodées par les utilisateurs

Les nombreux champs textuels que le site contient peuvent être le point d'entrée de différentes attaques. Cependant, les nombreuses protections rendent les attaques de ce type peu probables. Pour les attaques qui visent la base de données, des protections sont déjà présentes au niveau de l'ORM comme expliqué précédemment. De plus, lorsque le formulaire envoyé par l'utilisateur est converti en *formulaire Django,* différentes étapes de vérification sont systématiquement réalisées. Il faut cependant garder le réflexe de toujours vérifier les données envoyées par un utilisateur.

Attaques à l'aide des images

Des attaques pourraient être réalisées en utilisant les images ajoutées sur la plateforme. Il est possible de concevoir une image qui cache du code HTML. Lorsque l'image est affichée par le site, le code malveillant est interprété, ce qui peut conduire à différents risques. Il est peu probable qu'une attaque de ce type soit réalisée, car seuls les gestionnaires de projet (des personnes de confiance) peuvent ajouter des images sur la plateforme. De plus, les images sont encapsulées dans un *canvas* et chargées par la librairie *fabric.js*. Ces différentes couches rendent l'attaque moins probable. Il n'existe pas de mesures de protection au niveau de Django pour ce genre d'attaque. Certaines librairies existent (*Pillow, ExifRead*,…) pour analyser les métadonnées des images mais ces approches ne sont pas infaillibles.

Il est également possible de faire des attaques de *DoS* (*Deny of service*) et de *Buffer Overflow*. Ces attaques visent à surcharger le serveur pour le rendre indisponible. Si trop d'images sont envoyées simultanément ou si la taille des images est trop grande, il est possible que le serveur soit surchargé. De nouveau, cela est peu probable car seuls les gestionnaires peuvent ajouter des images.

Attaques sur les mots de passe

Des attaques, telles que le *brute force* ou une *attaque par dictionnaire,* peuvent également être réalisées afin de trouver le mot de passe d'un utilisateur. Django propose de nombreuses mesures de protection pour cela. Il propose de nombreux *validateurs* qui obligent les utilisateurs à choisir un mot de passe robuste (beaucoup de caractères, pas de mot de passe courant, ...). Il est également possible de créer ses propres *validateurs*. Il est donc peu probable qu'une attaque de ce type réussisse sur cette plateforme. De plus, les mots de passe ne sont pas stockés en clair dans la base de données et de nombreux algorithmes de cryptographie sont proposés pour *hasher* les mots de passe.

8.2.2 La gravité des attaques

Dans l'éventualité où une des attaques présentées ci-dessus est réalisée et qu'elle fonctionne, il est important d'estimer la gravité des conséquences. De manière générale, la gravité d'une attaque sur cette plateforme est assez faible car ce site n'est pas critique et n'est pas une cible intéressante.

Une attaque qui donne accès à toutes les données de la plateforme aurait une gravité très faible. En effet, cette plateforme ne contient pas de données sensibles ou confidentielles. Les archives ne sont pas confidentielles, les comptes des utilisateurs ne contiennent aucune information personnelle et les messages du forum ne sont pas critiques. Il n'y a donc pas de risque du point de vue du RGPD [47]. L'attaque pourrait supprimer des données, mais un système de sauvegarde (backup) peut réduire ce risque et être mis en place facilement.

De même, si l'attaque permet d'avoir accès au compte d'un gestionnaire de projet, le seul risque serait que l'attaquant supprime ou modifie des données. Ici aussi, un système de sauvegarde permet de pallier ce risque.

En revanche, si l'attaque permet d'obtenir un accès privilégié au serveur, aux autres services et bases de données présents sur ce même serveur, la gravité est nettement plus élevée. Cela dépend du niveau de sécurité implémenté au niveau du serveur et de la sensibilité des autres services présents sur la même machine.

Pour finir, une attaque peut également viser les utilisateurs de ce site web. Dans ce cas, la gravité dépend du type d'attaque réalisée sur leur ordinateur.

En résumé, la plateforme propose un bon niveau de sécurité par rapport aux faibles risques d'attaques et à leurs conséquences.

8.2.3 Détection de vulnérabilités

Pour détecter des failles de sécurité sur la plateforme, il est possible d'utiliser un scanner de vulnérabilités. Ces programmes sont conçus pour détecter automatiquement les failles de sécurité d'une cible, un site web dans ce cas-ci.

Pour tester cette plateforme, **Nuclei** [48] a été utilisé. Ce scanner de vulnérabilités Open Source permet de tester un site web à l'aide de *templates*. Chaque *template* représente une vulnérabilité, avec son type, sa criticité et la manière de la détecter. Le programme va réaliser un certain nombre de requêtes sur la cible pour pouvoir couvrir toutes les vulnérabilités à tester.

Lors du test, 7695 *templates* ont été testés via 1626 requêtes. Ce test a permis d'identifier une seule vulnérabilité, modérée : certaines informations de *debug* étaient affichées lorsqu'on accédait à une page inexistante (la page d'erreur 404) (Figure 23). Cette vulnérabilité est due au paramètre *debug* inhérent à l'environnement de développement. Il faudra le changer lors de la mise en production de cette plateforme.

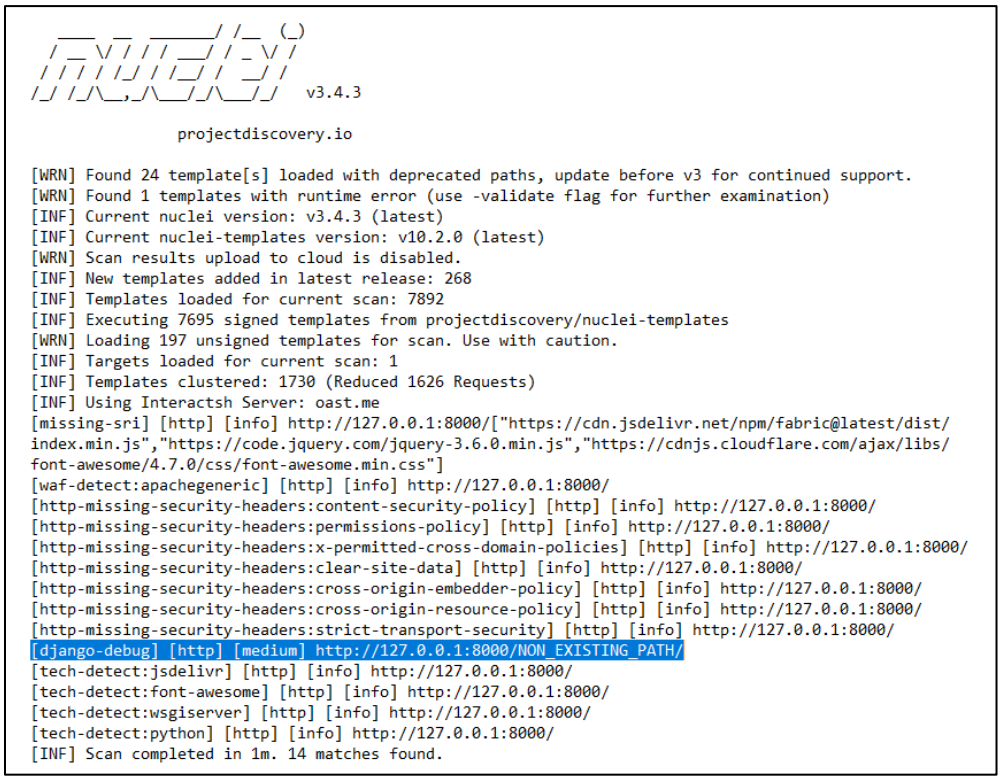


Figure 23. Rapport du scan de vulnérabilités de Nuclei

9 Recommandations

Dans ce chapitre, différentes améliorations réalisables dans de futurs développements vont être présentées. Dans un premier temps, les améliorations vont viser à enrichir le projet existant. Ensuite, des idées d'évolutions qui vont au-delà du cahier des charges initialement défini vont être présentées.

9.1 Améliorations de la plateforme Archio

Des améliorations peuvent être faites au niveau de l'interface pour les gestionnaires de projet. Dans l'implémentation actuelle, l'accent a surtout été mis sur l'expérience des bénévoles pour la rendre simple et rapide. Pour le moment, les gestionnaires doivent réaliser la majorité de leurs actions sur la page d'administration du site global, ce qui n'est pas optimal. Pour améliorer cela, on pourrait créer une page d'administration de projet. Dans cette page uniquement accessible aux gestionnaires du projet, il serait possible de modifier les métadonnées du projet (titre, description, etc.), gérer les documents (ajouter, supprimer des images, etc.) ainsi que gérer les utilisateurs (candidature et suppression de membres).

Le système de candidature peut également être plus étoffé. Pour le moment, il s'agit d'un simple bouton. Il serait possible d'implémenter un système de candidature qui demande un texte de motivation, ou un système plus complexe qui teste les capacités de transcription du bénévole avant d'accepter sa candidature.

Un système de notification pourrait être mis en place pour indiquer aux utilisateurs qu'il y a de nouveaux messages dans le forum ou de nouvelles annonces. Un système de rappel par mail pourrait aussi être mis en place pour rappeler aux bénévoles de finaliser leur tâche en cours.

Il est toujours possible d'améliorer l'interface et l'expérience utilisateur. L'interface pourrait être proposée en plusieurs langues et s'adapter aux différentes tailles d'écran. On pourrait également implémenter des confirmations pour éviter que l'utilisateur ne quitte accidentellement une page sans sauvegarder ou soumettre une tâche.

Les performances peuvent toujours être améliorées. On peut réduire le temps de réponse du serveur en optimisant les requêtes en base de données et en utilisant un système de base de données plus performant (tel que *Redis*). Bien que cela soit améliorable, les performances de cette plateforme ne posent aucun problème pour son utilisation. L'optimisation des performances n'est pas critique dans ce projet.

On peut également optimiser la quantité de stockage requise pour l'hébergement des images. Si les images sont déjà hébergées sur un autre serveur, il serait possible de directement récupérer ces images sans les stocker sur le serveur du site web. Cette amélioration dépend de l'endroit où sont hébergées les images.

9.2 Perspective d'évolution d'Archio avec l'intelligence artificielle

Cette perspective d'évolution fait écho à une réflexion initiée par Dr Gielis lors de la phase de test.

Avec les progrès fulgurants de l'intelligence artificielle dans le domaine de la reconnaissance de texte, les besoins dans le domaine de la transcription peuvent rapidement évoluer.

Actuellement, les intelligences artificielles de HTR (Handwritten text recognition) font de grands progrès. Cependant, il semble que ces intelligences artificielles ont du mal à réaliser la transcription des noms propres (nom de personnes, d'institutions, de lieux, ...). Il serait donc possible de modifier l'approche de cette plateforme pour résoudre ce problème plus spécifique. L'idée serait de ne plus transcrire l'intégralité du texte ligne par ligne, mais uniquement les noms propres. Cela permettrait de constituer un corpus de noms propres que l'intelligence artificielle pourrait utiliser pour apprendre et combler ce problème.

Au niveau du développement, grâce à une conception flexible et polyvalente, cette modification fondamentale n'impliquerait presque aucune modification. Si on utilise des rectangles plus petits qui ne représentent plus une ligne mais un simple mot, il est possible de garder les mêmes processus de labellisation, transcription et vérification (Figure 24). Quelques modifications sur l'interface et sur les formats d'export devraient être réalisées, mais aucun changement structurel ne serait à prévoir.

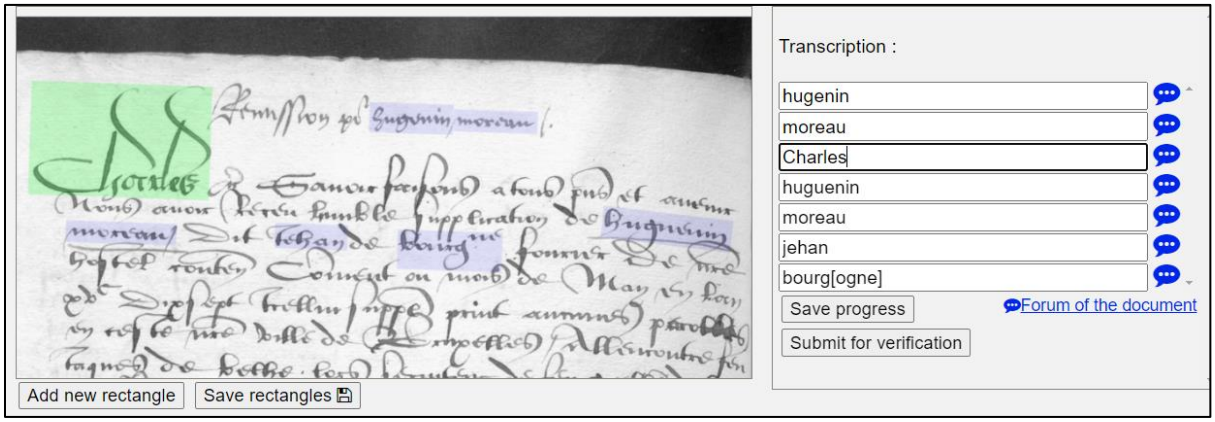


Figure 24. Exemple de transcription limitée aux noms propres

À plus long terme, les intelligences artificielles pourront réaliser la transcription dans son intégralité. À ce moment-là, les utilisateurs devront uniquement réaliser les tâches de vérifications pour contrôler la qualité des transcriptions faite par l'intelligence artificielle. La plateforme pourrait être modifiée pour permettre d'importer des images avec leur transcription déjà réalisée afin de commencer directement par la tâche de vérification.

10 Conclusion

L'objectif de ce mémoire était de développer une plateforme permettant à des bénévoles de transcrire et de labelliser des documents anciens à l'aide du crowdsourcing.

Dans un premier temps, l'étude du contexte a permis de mieux comprendre les besoins et les limitations de la transcription de documents. Cela a également permis de comprendre que les transcriptions sont actuellement réalisées manuellement et de manière archaïque. L'étude du crowdsourcing a permis de déterminer qu'il s'agissait, dans ce mémoire, d'un crowdsourcing altruiste qui vise à produire un contenu. Cela a également permis de confirmer ses avantages et de mettre en lumière les risques relatifs à cette pratique.

Ensuite, le cahier des charges a formalisé les fonctionnalités attendues en se basant sur les besoins pratiques exprimés dans le projet PARDONS. La solution recherchée est un site web qui rend la transcription de documents et sa gestion plus simple et mieux organisée. La solution devait idéalement être Open Source et avec un faible coût de maintenance.

Après avoir envisagé un grand nombre de solutions de crowdsourcing existantes, aucune ne répondait aux défis lancés par la transcription d'archives et il n'était pas possible de partir d'un projet existant pour réaliser ce travail. Dans ce mémoire, une plateforme dédiée à la transcription de documents a donc été développée afin de répondre aux besoins spécifiques des Archives.

Avant d'effectuer le développement, l'analyse fonctionnelle a permis d'identifier les différents rôles des utilisateurs (transcripteur, vérificateur et gestionnaire). Elle a également permis de définir le fonctionnement de la phase de transcription ainsi que le mécanisme de vérification qui garantit la qualité du travail.

Le développement de la plateforme a ensuite été réalisé. L'implémentation, la structure de la base de données et les choix de développement ont été faits en adoptant une approche globale et généraliste. Cette plateforme vise à améliorer la transcription de tout type de document, indépendamment du sujet, de la langue et de la mise en page. Le code source de ce projet est Open Source et disponible sous licence MIT sur GitHub (https://github.com/JeremySidgwick/thesis).

Grâce à cette plateforme, la transcription de documents s'avère plus pratique pour les bénévoles. Pour les administrateurs, la gestion est plus simple, moins chronophage et plus automatisée. La vérification n'est plus exclusivement réalisée par les gestionnaires mais également par des bénévoles experts. Grâce au forum, les bénévoles peuvent aussi compter sur l'aide d'une communauté.

Deux archivistes ont ensuite testé la plateforme et ont confirmé son bon fonctionnement, sa facilité d'utilisation ainsi que son utilité.

Une analyse de sécurité a également été réalisée pour identifier d'éventuelles vulnérabilités sur la plateforme et leurs impacts. Cette analyse a confirmé que la plateforme est sécurisée et qu'elle ne compromet pas de données sensibles.

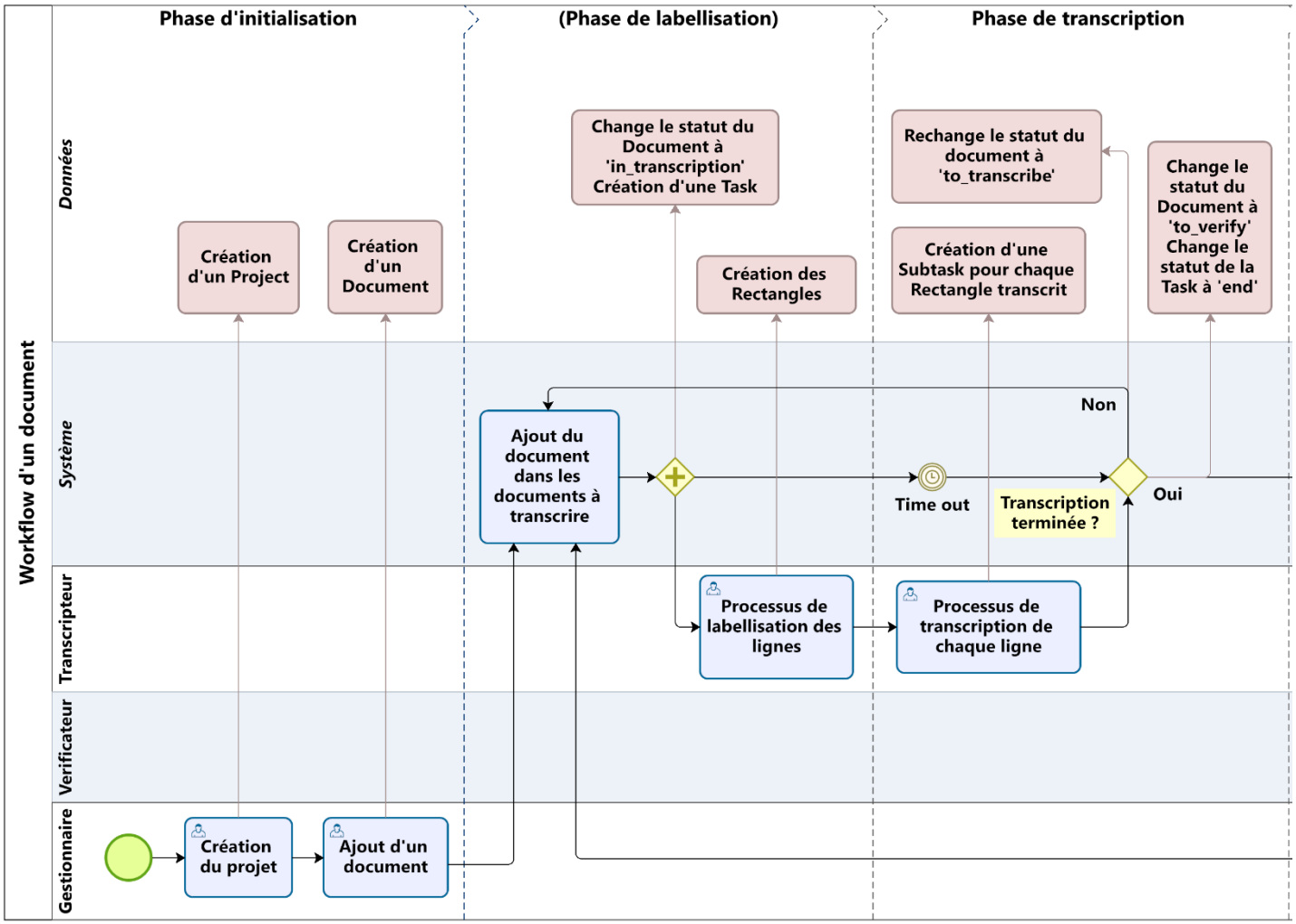
Pour finir, une série d'améliorations, telles que l'amélioration de l'interface et la détection automatique des lignes, ont été présentées et pourront faire l'objet d'un développement futur.

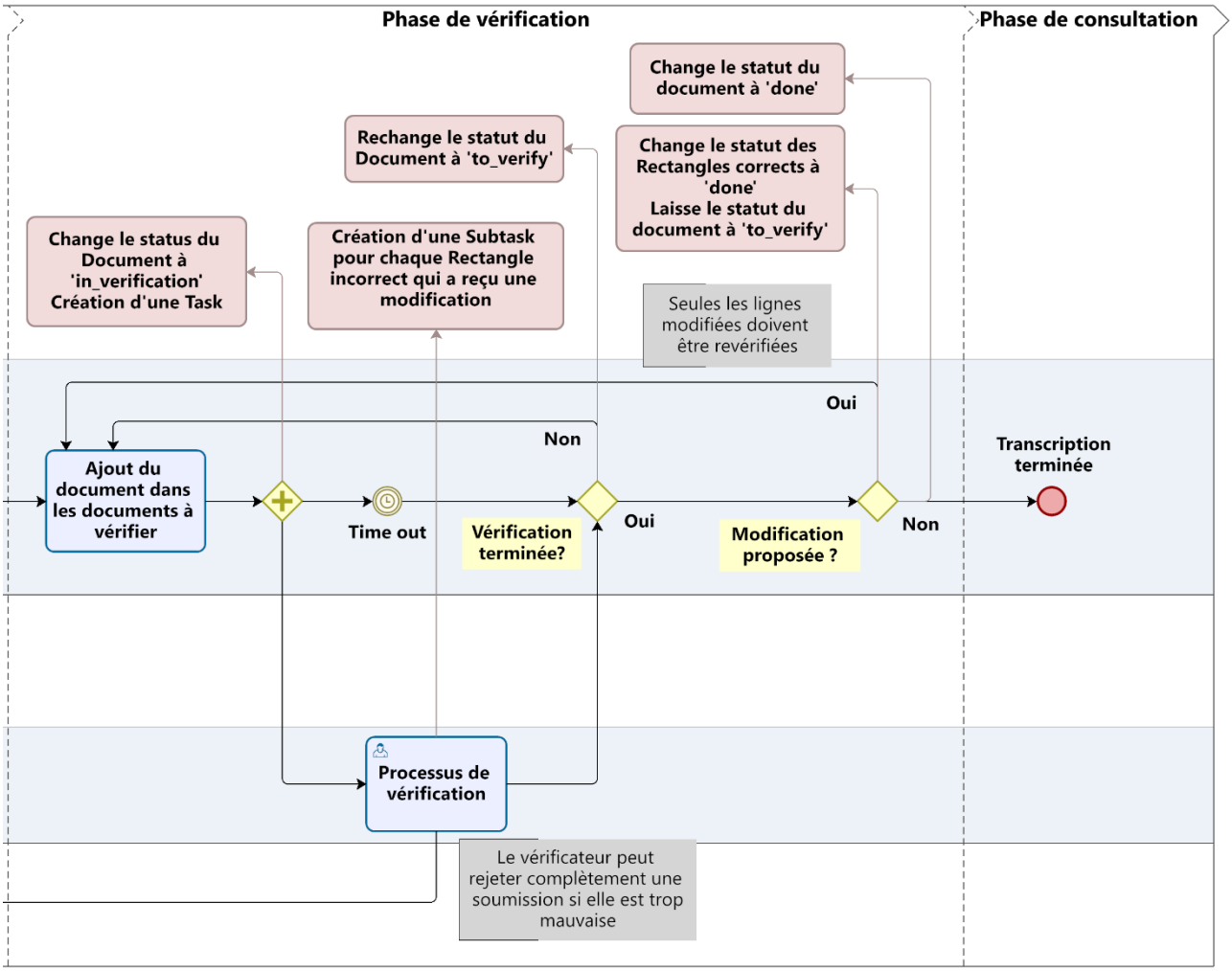
Des perspectives d'évolution plus globales, dues aux progrès récents de l'intelligence artificielle, ont également été apportées. Ces évolutions impliqueraient peu de changements pour la plateforme, ce qui met en lumière sa polyvalence et sa flexibilité.

En conclusion, ce mémoire a permis de développer une plateforme de transcription accessible et polyvalente qui répond aux besoins actuels et qui est capable d'évoluer pour répondre aux futures attentes des Archives.

11 Annexes

A. Workflow d'un document




Phase de vérification
Phase de consultation
Change le statut du document à 'done'
Rechange le statut du Document à 'to_verify'
Change le statut des Rectangles corrects à 'done'
Laisse le statut du document à 'to_verify'
Change le status du Document à 'in_verification' Création d'une Task
Création d'une Subtask pour chaque Rectangle incorrect qui a reçu une modification
Seules les lignes modifiées doivent être revérifiées
Oui
Non
Ajout du document dans les documents à vérifier
Time out
Vérification terminée?
Oui
Modification proposée ?
Non
Transcription terminée
Processus de vérification
Le vérificateur peut rejeter complètement une soumission si elle est trop mauvaise

B. Récapitulatif technique des tables de base de données

1. Application d'authentification

User

Champ	Type	Propriétés
username	CharField	Maximum 150 caractères
password	Hash	Algorithme : PBKDF2, hash SHA256
is_superuser	BooleanField	Valeur par défaut : False
date_joined	DatetimeField	
last_login	DatetimeField	
completed_tasks	IntegerField	Valeur par défaut : 0

2. Application principale

Project

Champ	Type	Propriétés
name	CharField	Maximum 100 caractères
description	TextField	

Document

Champ	Type	Propriétés
image	ImageField	
project	ForeignKey de Project	Suppression en cascade
status	CharField	Valeur par défaut : "*to_transcribe*". Liste limitée de valeurs
name	CharField	Optionnel, maximum 200 caractères.
archive_name	CharField	

Task

Champ	Type	Propriétés
document	ForeignKey de Document	Suppression en cascade
user	ForeignKey de User	Suppression en cascade
started	DateTimeField	Horodatage automatique
ended	DateTimeField	
status	CharField	Liste de valeurs limitée
type	CharField	Valeur par défaut : "transcription". Liste de valeurs limitée

Rectangle

Champ	Type	Propriétés
document	ForeignKey de Document	Suppression en cascade
x	IntegerField	
y	IntegerField	
width	IntegerField	
height	IntegerField	
angle	FloatField	Valeur par défaut : 0
done	BooleanField	Valeur par défaut : False

Subtask

Champ	Type	Propriétés
rectangle	ForeignKey de Document	Suppression en cascade
task	ForeignKey de Task	Suppression en cascade
text	TextField	
time	DateField	Horodatage automatique
status	CharField	Maximum 100 caractères

UserProject

Champ	Type	Propriétés
user	ForeignKey de User	Suppression en cascade
projet	ForeignKey de Projet	Suppression en cascade
role	CharField	Maximum 100 caractères
last_participation	DateField	
participation_amount	IntegerField	Valeur par défaut : 0

3. Application du forum

Announcement

Champ	Type	Propriétés
title	CharField	Maximum 100 caractères
content	TextField	Valeur par défaut : Null
timestamp	DateTimeField	Horodatage automatique
thumbnail	ImageField	Valeur par défaut : Null

Topic

Champ	Type	Propriétés
author	ForeignKey de User	Suppression en cascade
title	CharField	Maximum 200 caractères
description	TextField	Maximum 500 caractères
date_created	DateTimeField	Horodatage automatique
related_document	ForeignKey de Document	Suppression en cascade Valeur par défaut : Null
related_rectangle	ForeignKey de Rectangle	Suppression en cascade Valeur par défaut : Null

TopicView

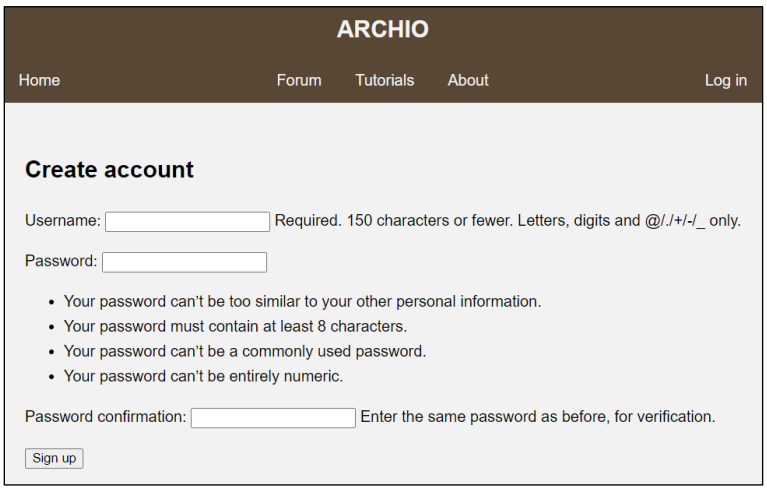
Champ	Type	Propriétés
user	ForeignKey de User	Suppression en cascade
user_post	ForeignKey de Topic	Suppression en cascade

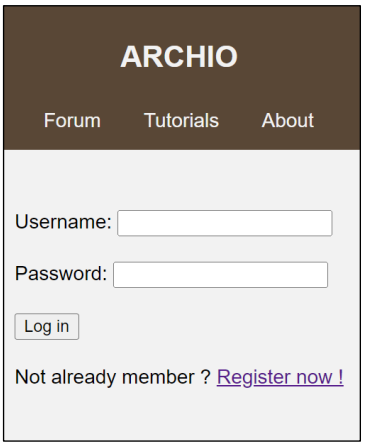
Answer

Champ	Type	Propriétés
user_post	ForeignKey de Topic	Suppression en cascade
user	ForeignKey de User	Suppression en cascade
content	TextField	Maximum 500 caractères
date_created	DateTimeField	Horodatage automatique
upvotes	ManyToManyField	
downvotes	ManyToManyField	

C. Captures d'écran des pages web

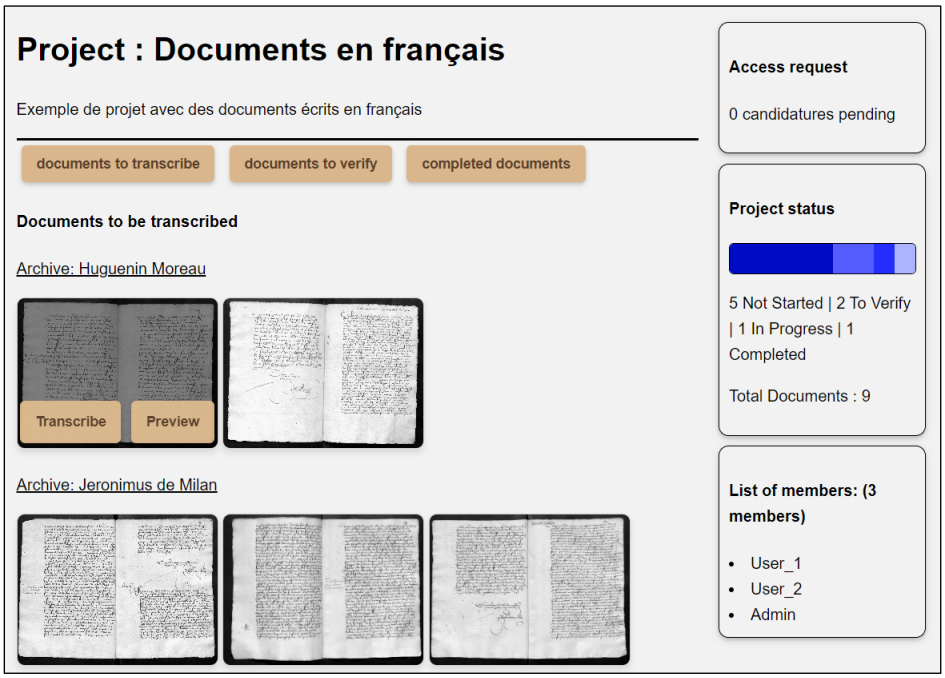
Page de connexion et de création de compte



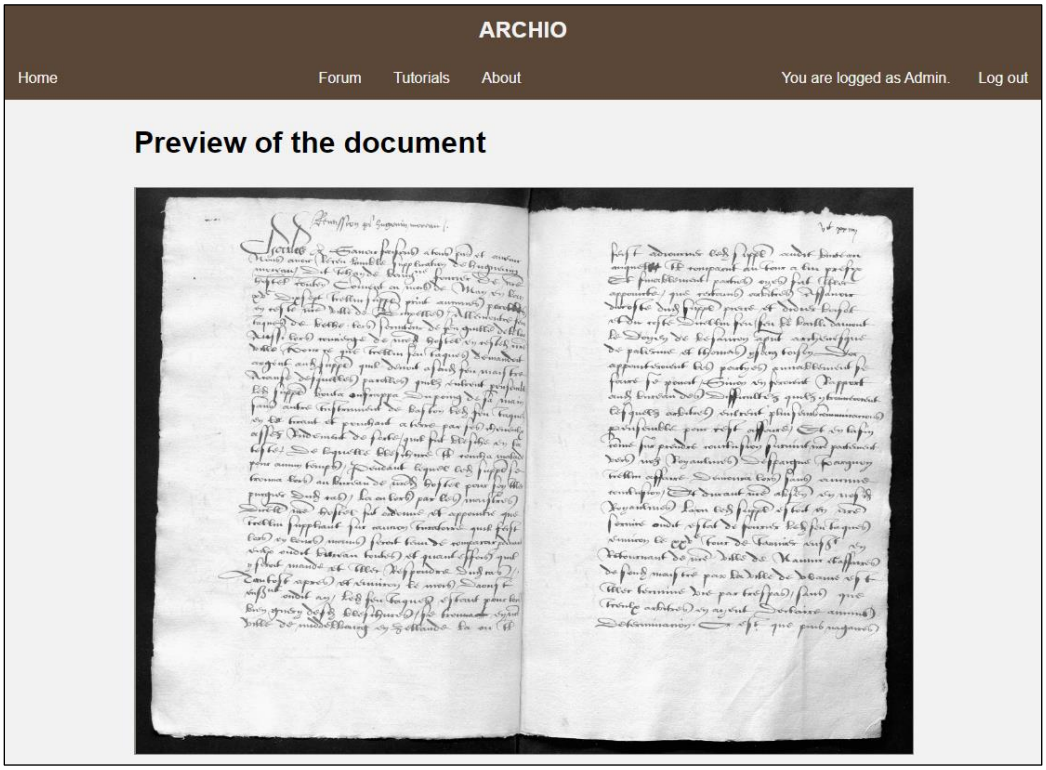


Page d'accueil

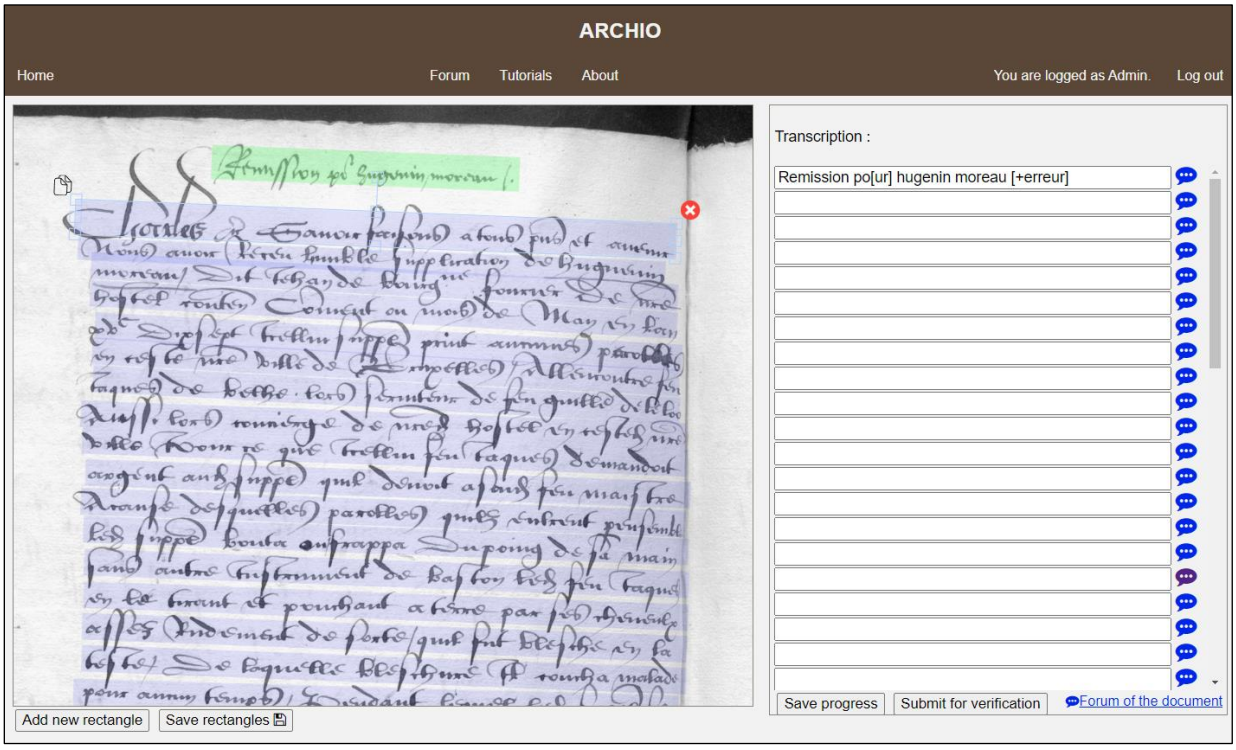
Page d'un projet



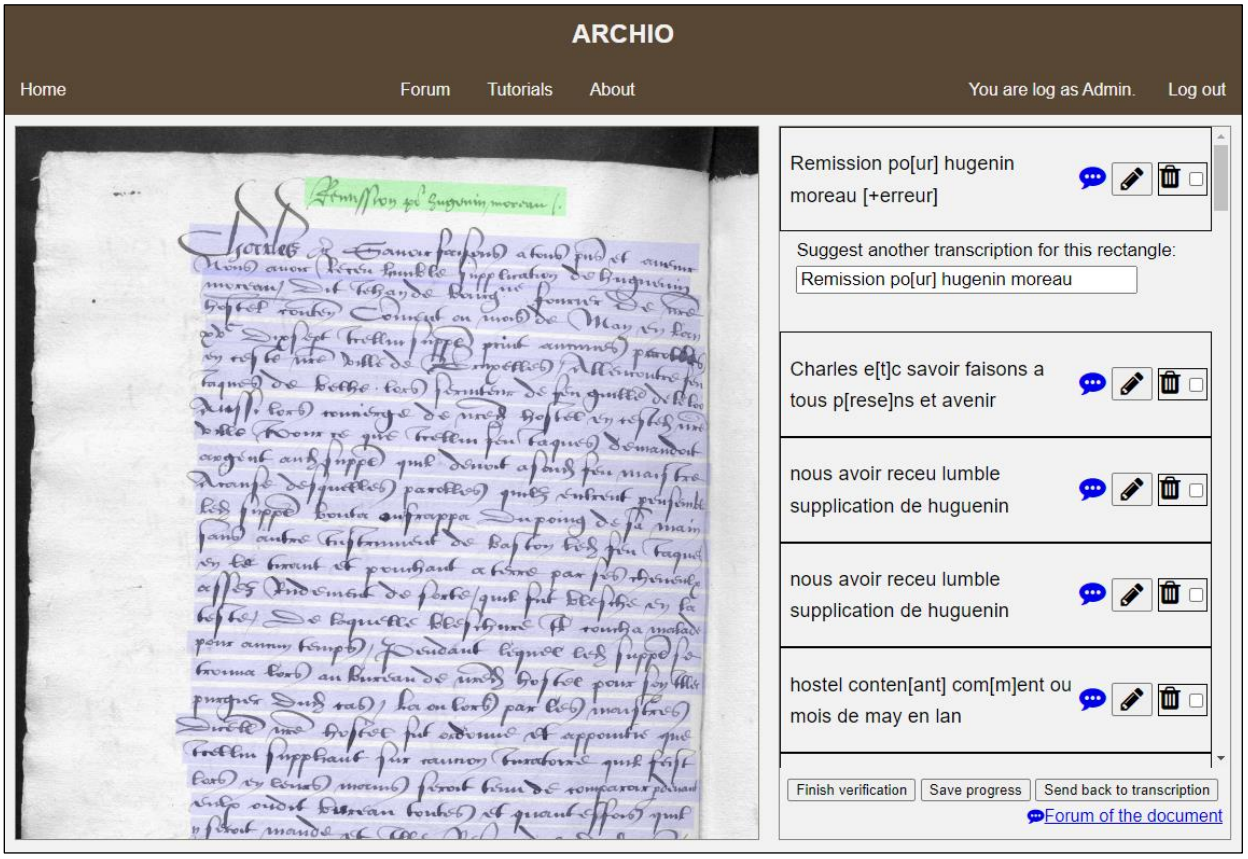
Page de visualisation



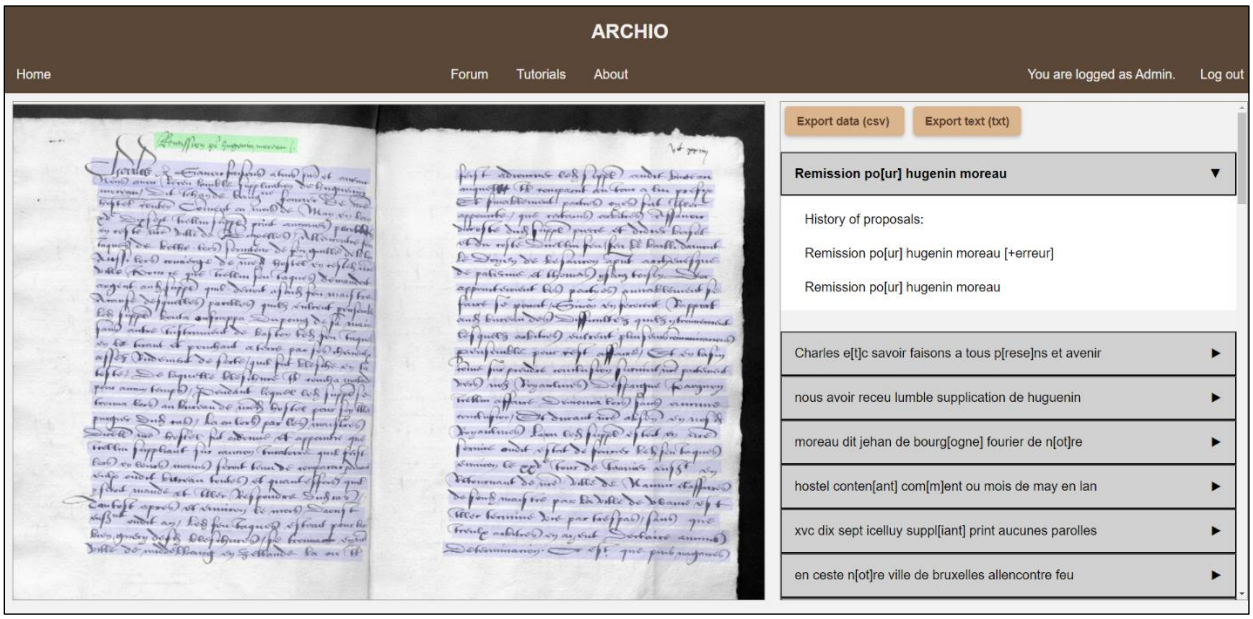
Page de transcription



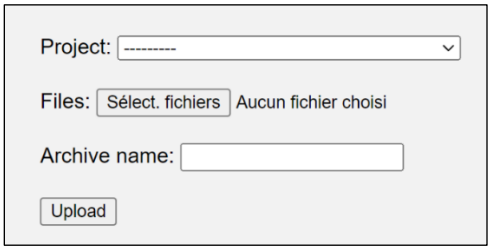
Page de vérification



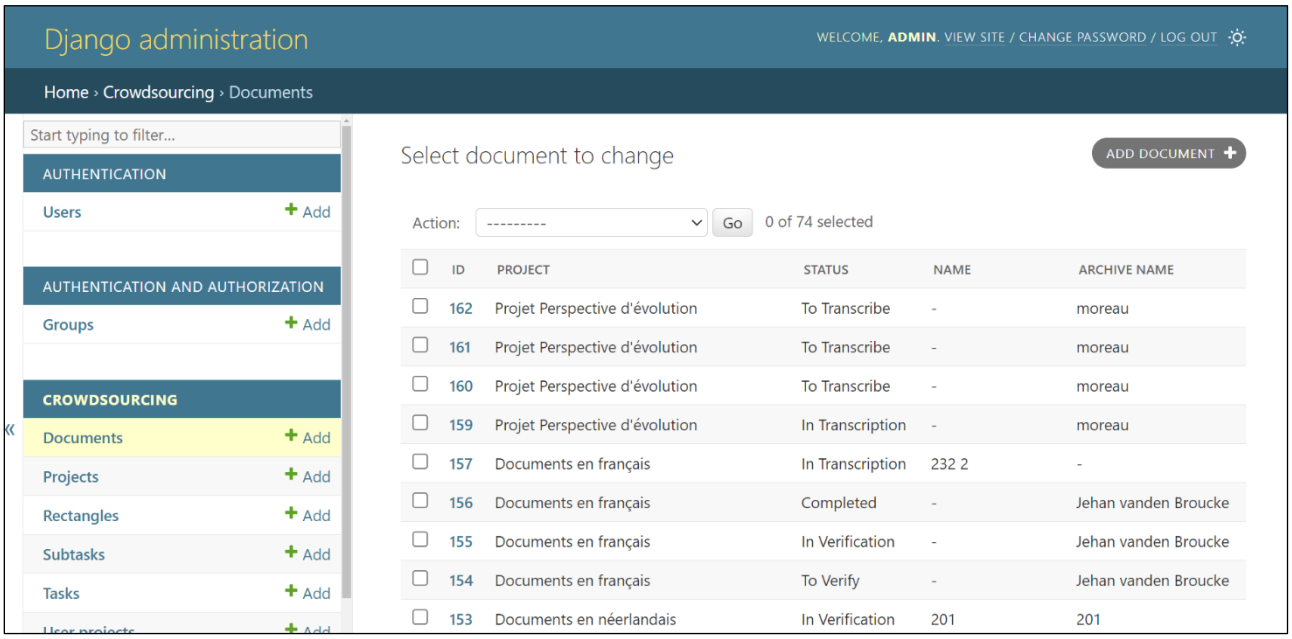
Page de consultation



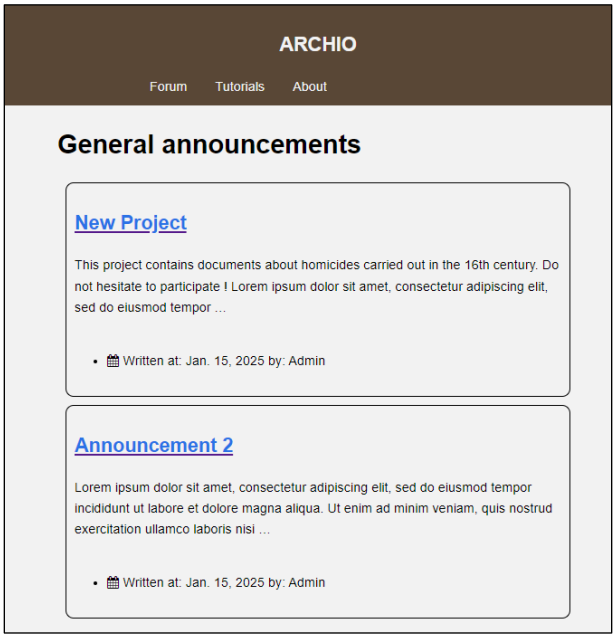
Page d’ajout de documents



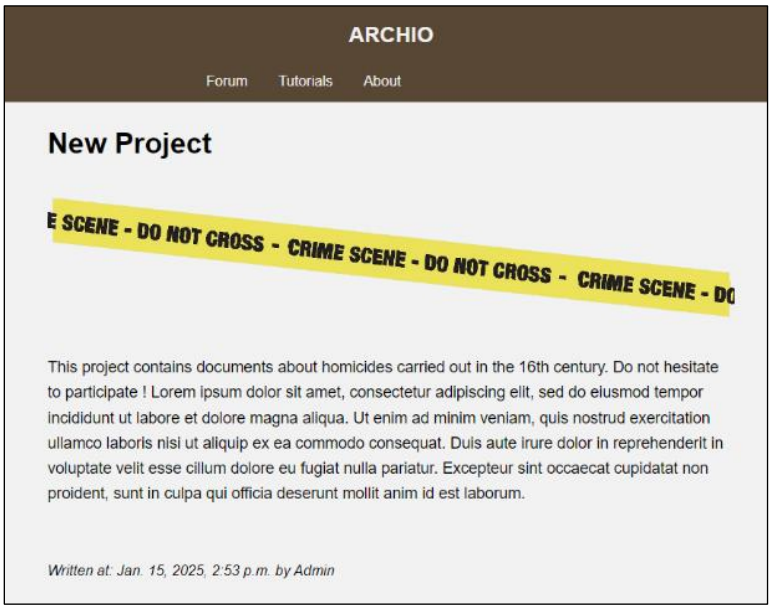
Page d’administration



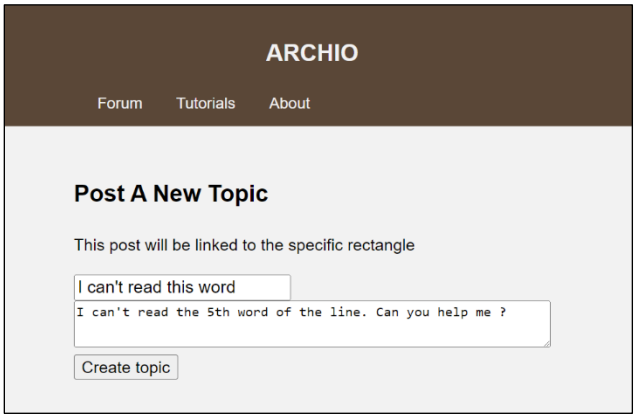
Page de présentation des annonces



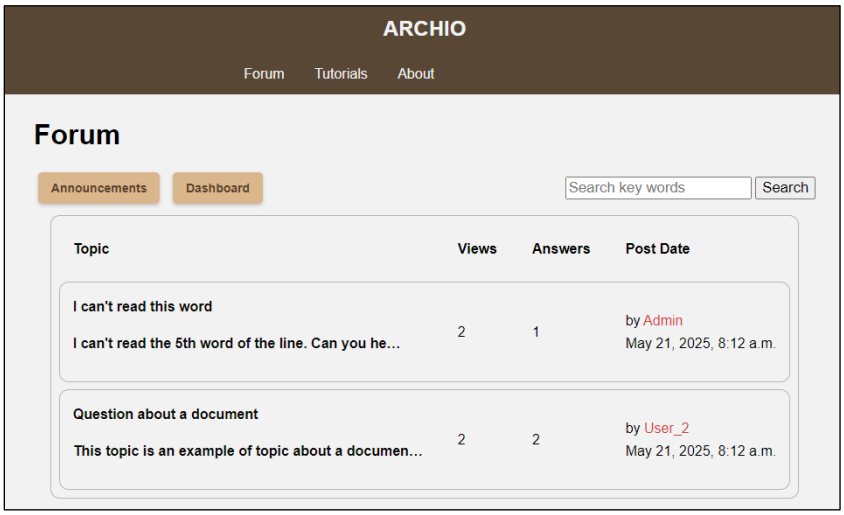
Page d'une annonce



Page de création d’un topic

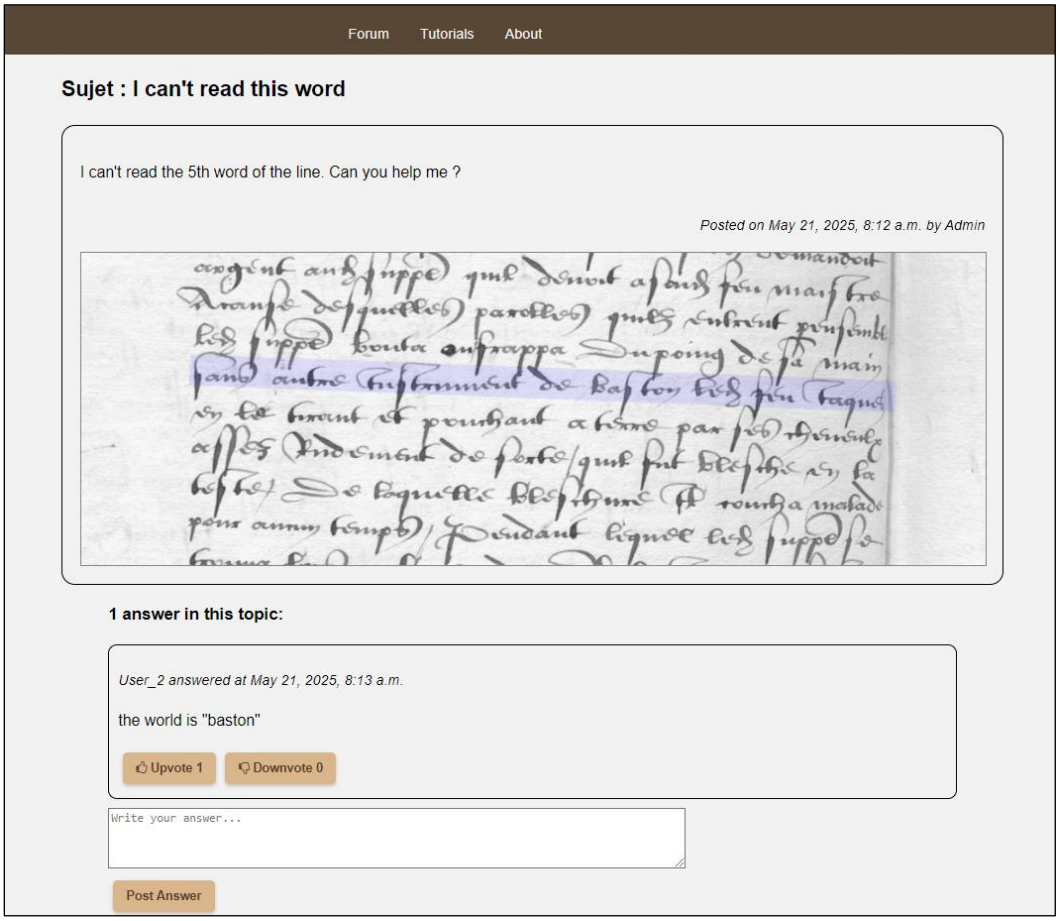


Page de présentation des topics

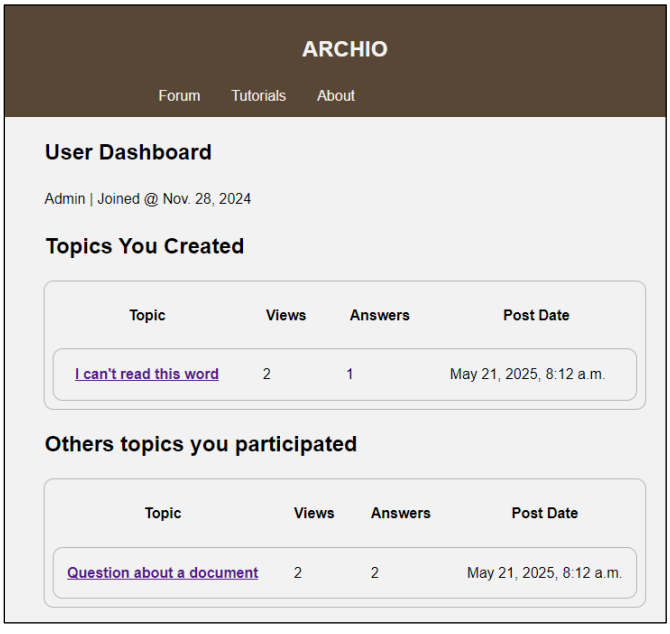


Page de recherche par mots-clés

Page d’une discussion



Page de profil



D. Mode d’emploi

Cette annexe reprend les informations données sur la page de tutoriels du site web.

Comment naviguer dans l’image

Pour naviguer dans l'image, on peut zoomer et dézoomer à l'aide de la molette de la souris. Pour se déplacer : il faut maintenir le bouton gauche de la souris enfoncé à l'extérieur d'un rectangle.

Comment réaliser la transcription

Une vidéo montrant les différentes étapes est disponible à l’adresse suivante : https://www.youtube.com/aq-J-yl0LEo

Voici les étapes de la transcription :

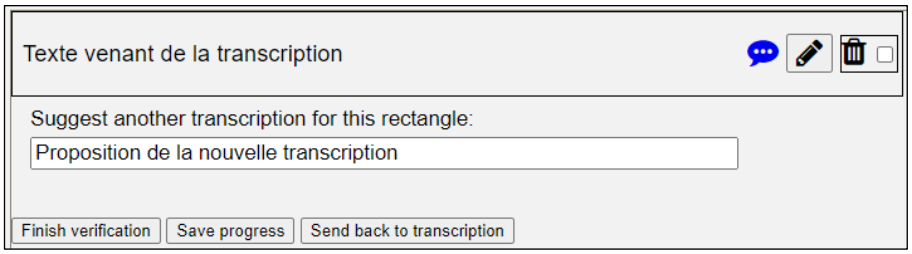
1. Créer un premier rectangle en cliquant sur le bouton *add new rectangle* en bas à gauche.
2. Ajuster sa position, sa taille et son inclinaison pour correspondre au mieux à la ligne du texte.
3. Une fois le premier rectangle correctement positionné, dupliquer celui-ci en cliquant sur l’icône en haut à gauche du rectangle.
4. Ajuster le rectangle et répéter le processus pour chaque ligne.
5. Une fois les lignes labellisées, cliquer sur le bouton *save rectangles* en bas à gauche de la page.
6. Transcrire chaque ligne dans la zone de texte correspondante dans la partie droite de la page.
7. Cliquer sur le bouton *save progress* pour sauvegarder l’avancement ou sur *submit to verification* pour terminer la transcription.

Comment réaliser la vérification

Le vérificateur peut faire quatre actions pour chaque rectangle :

- **Ne rien faire** : cela signifie qu'il accepte la transcription. Ce rectangle sera validé et ne pourra plus être modifié.
- **Supprimer le rectangle** : en cliquant sur la case de la poubelle, on peut supprimer définitivement le rectangle s'il ne correspond pas à une ligne.
- **Modifier le texte** : en cliquant sur le crayon, le vérificateur peut proposer une autre transcription. Cette étape impliquera une étape de vérification supplémentaire.
- **Accéder au forum** : en cliquant sur l'icône bleue, l'utilisateur peut accéder ou créer un forum lié à ce rectangle ou document.

Le vérificateur peut enregistrer sa progression, terminer la vérification pour achever la tâche ou renvoyer le document à l'étape de transcription si le travail produit n'est pas satisfaisant.



Bibliographie

[1] « Chiffres-clés - Archives de l'État en Belgique ». Consulté le: 22 mai 2025. [En ligne]. Disponible sur: https://www.arch.be/index.php?l=fr&m=l-institution&r=presentation&sr=chiffres-cles

[2] « Que conservons-nous ? - Archives de l'État en Belgique ». Consulté le: 22 avril 2025. [En ligne]. Disponible sur: https://www.arch.be/index.php?l=fr&m=l-institution&r=que-conservons-nous#1

[3] « PARDONS ». Consulté le: 22 avril 2025. [En ligne]. Disponible sur: http://pardons.eu/

[4] « Be a volunteer – PARDONS ». Consulté le: 22 avril 2025. [En ligne]. Disponible sur: https://pardons.eu/contribute/be-a-volunteer/

[5] « externalisation ouverte ». Consulté le: 14 avril 2025. [En ligne]. Disponible sur: https://vitrinelinguistique.oqlf.gouv.qc.ca/fiche-gdt/fiche/45436/externalisation-ouverte

[6] « The Rise of Crowdsourcing | WIRED ». Consulté le: 14 avril 2025. [En ligne]. Disponible sur: https://archive.is/20230612120846/https://www.wired.com/2006/06/crowds/

[7] T. Burger-Helmchen et J. Pénin, « Crowdsourcing : définition, enjeux, typologie », *Manag. Avenir*, vol. 41, n° 1, p. 254-269, mai 2011, doi: 10.3917/mav.041.0254.

[8] « Great Internet Mersenne Prime Search - PrimeNet ». Consulté le: 18 avril 2025. [En ligne]. Disponible sur: https://www.mersenne.org/

[9] M. Sabou, K. Bontcheva, L. Derczynski, et A. Scharl, « Corpus Annotation through Crowdsourcing: Towards Best Practice Guidelines », présenté à International Conference on Language Resources and Evaluation, mai 2014. Consulté le: 26 mai 2025. [En ligne]. Disponible sur: https://www.semanticscholar.org/paper/Corpus-Annotation-through-Crowdsourcing%3A-Towards-Sabou-Bontcheva/2435b48cc30e30f3849b9670f263b501ba9c0024#extracted

[10] G. P. Pisano et R. Verganti, « Which Kind of Collaboration Is Right for You? », *Harvard Business Review*. Consulté le: 27 mai 2025. [En ligne]. Disponible sur: https://hbr.org/2008/12/which-kind-of-collaboration-is-right-for-you

[11] « Amazon Mechanical Turk ». Consulté le: 23 avril 2025. [En ligne]. Disponible sur: https://www.mturk.com/

[12] Microworkers, « Templates », Microworkers - work & earn or offer a micro job. Consulté le: 26 avril 2025. [En ligne]. Disponible sur: https://www.microworkers.com/

[13] « Home - Wazoku Crowd ». Consulté le: 25 avril 2025. [En ligne]. Disponible sur: https://www.wazokucrowd.com/

[14] « Managed Service », clickworker.com. Consulté le: 23 avril 2025. [En ligne]. Disponible sur: https://www.clickworker.com/managed-service/

[15] « DataAnnotation | Your New Remote Job ». Consulté le: 23 avril 2025. [En ligne]. Disponible sur: https://www.dataannotation.tech/

[16] « Transcribathon – Transcribe history ». Consulté le: 28 avril 2025. [En ligne]. Disponible sur: https://transcribathon.eu/

[17] « DIGIVOL | Home ». Consulté le: 28 avril 2025. [En ligne]. Disponible sur: https://volunteer.ala.org.au/

[18] « Home | Smithsonian Digital Volunteers ». Consulté le: 28 avril 2025. [En ligne]. Disponible sur: https://transcription.si.edu/

[19] « e-manuscripta ». Consulté le: 25 avril 2025. [En ligne]. Disponible sur: https://www.e-manuscripta.ch/

[20] « FromThePage ». Consulté le: 25 avril 2025. [En ligne]. Disponible sur: https://www.fromthepage.com/landing
[21] « DoeDat | Page d'accueil ». Consulté le: 25 avril 2025. [En ligne]. Disponible sur: https://www.doedat.be/
[22] « Zooniverse ». Consulté le: 25 avril 2025. [En ligne]. Disponible sur: https://www.zooniverse.org/
[23] « Transkribus - Unlocking the past with AI ». Consulté le: 25 avril 2025. [En ligne]. Disponible sur: https://www.transkribus.org/
[24] *AI-Collaboratory/Scribe*. (25 juillet 2024). HTML. Advanced Information Collaboratory. Consulté le: 25 avril 2025. [En ligne]. Disponible sur: https://github.com/AI-Collaboratory/Scribe
[25] *openknowledgebe/W4P*. (26 mai 2024). PHP. Open Knowledge Belgium. Consulté le: 25 avril 2025. [En ligne]. Disponible sur: https://github.com/openknowledgebe/W4P
[26] *Scifabric/pybossa*. (14 mars 2025). Python. Scifabric. Consulté le: 24 mars 2025. [En ligne]. Disponible sur: https://github.com/Scifabric/pybossa
[27] *LibraryOfCongress/concordia*. (24 avril 2025). Python. Library of Congress. Consulté le: 25 avril 2025. [En ligne]. Disponible sur: https://github.com/LibraryOfCongress/concordia
[28] S. Bose, « The Most Popular Programming Languages of 2025: A Comprehensive Overview », Cyber Security News. Consulté le: 24 mars 2025. [En ligne]. Disponible sur: https://cybersecuritynews.com/the-most-popular-programming-languages-of-2025/
[29] « Quickstart — Flask Documentation (3.1.x) ». Consulté le: 25 mars 2025. [En ligne]. Disponible sur: https://flask.palletsprojects.com/en/stable/quickstart/
[30] « Quart documentation — Quart 0.20.0 documentation ». Consulté le: 25 mars 2025. [En ligne]. Disponible sur: https://quart.palletsprojects.com/en/latest/
[31] « Django overview », Django Project. Consulté le: 25 mars 2025. [En ligne]. Disponible sur: https://www.djangoproject.com/start/overview/
[32] J. Swacha et A. Kulpa, « Evolution of Popularity and Multiaspectual Comparison of Widely Used Web Development Frameworks », *Electronics*, vol. 12, n° 17, Art. n° 17, janv. 2023, doi: 10.3390/electronics12173563.
[33] « Top Django Websites: Real-World Examples & Insights ». Consulté le: 9 avril 2025. [En ligne]. Disponible sur: https://www.stxnext.com/blog/django-websites-examples
[34] « Applications | Django documentation », Django Project. Consulté le: 25 mars 2025. [En ligne]. Disponible sur: https://docs.djangoproject.com/fr/5.1/ref/applications/
[35] « Django Packages : Reusable apps, sites and tools directory for Django », Django Packages. Consulté le: 25 mars 2025. [En ligne]. Disponible sur: https://djangopackages.org/
[36] A. G. III, « Basics of Django: Model-View-Template (MVT) Architecture », Medium. Consulté le: 25 mars 2025. [En ligne]. Disponible sur: https://angelogentileiii.medium.com/basics-of-django-model-view-template-mvt-architecture-8585aecffbf6
[37] « ORM Django ». Consulté le: 25 mars 2025. [En ligne]. Disponible sur: https://python.doctor/page-django-orm-apprendre-base-donnees-database-queryset-modeles
[38] « Security in Django | Django documentation », Django Project. Consulté le: 9 avril 2025. [En ligne]. Disponible sur: https://docs.djangoproject.com/en/5.1/topics/security/
[39] J. C. S. Perez, « Django Project Structure: A Comprehensive Guide », Django Unleashed. Consulté le: 8 avril 2025. [En ligne]. Disponible sur: https://medium.com/django-unleashed/django-project-structure-a-comprehensive-guide-4b2ddbf2b6b8

[40] « Allez plus loin avec le framework Django ». Consulté le: 25 mars 2025. [En ligne]. Disponible sur: https://openclassrooms.com/fr/courses/7192426-allez-plus-loin-avec-le-framework-django
[41] « Password management in Django | Django documentation », Django Project. Consulté le: 25 mars 2025. [En ligne]. Disponible sur: https://docs.djangoproject.com/en/5.1/topics/auth/passwords/
[42] « Fabric.js Javascript Library ». Consulté le: 10 avril 2025. [En ligne]. Disponible sur: https://fabricjs.com/
[43] « Class-based views | Django documentation », Django Project. Consulté le: 10 avril 2025. [En ligne]. Disponible sur: https://docs.djangoproject.com/en/5.1/topics/class-based-views/
[44] M. Sajib, *mahmud-sajib/Django-Forum*. (6 février 2025). HTML. Consulté le: 8 avril 2025. [En ligne]. Disponible sur: https://github.com/mahmud-sajib/Django-Forum
[45] « Comment déployer Django | Django documentation », Django Project. Consulté le: 6 mai 2025. [En ligne]. Disponible sur: https://docs.djangoproject.com/fr/5.2/howto/deployment/
[46] « La sécurité dans Django | Django documentation », Django Project. Consulté le: 7 mai 2025. [En ligne]. Disponible sur: https://docs.djangoproject.com/fr/5.2/topics/security/
[47] « Règlement général sur la protection des données (RGPD) | EUR-Lex ». Consulté le: 9 mai 2025. [En ligne]. Disponible sur: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=legissum:310401_2
[48] « Nuclei Overview », ProjectDiscovery Documentation. Consulté le: 9 mai 2025. [En ligne]. Disponible sur: https://docs.projectdiscovery.io/tools/nuclei/overview

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**