

Langue des signes Plan Assurance Qualité

Sommaire :

1 – La charte du projetA

A – Chef de Projet

B – Date de la dernière révision

C – But du projet

D – Objectifs du projet

E – Portée du projet

F – Équipe et ressources du projet

G – Parties prenantes et approbateurs

2 – Le cahier des charges

3 – Plan de ressourcesA – Gestion de projet

B – Calendrier

4 – Plan de tests

5 – Le plan de communication

6 – Gestion des tickets et deliveries

A – Gestion des Tickets

1. Création des tickets

2. Assignations des tickets

3. Branches de fonctionnalité

4. Développement

5. Merge requests

6. Revue de code

7. Intégration continue

8. Suivi des tickets

B – Gestion des deliveries

1. Packaging

2. Pipelines de déploiement

3. Versionning

4. Stockages des deliveries

5. Documentation des versions

6. Déploiement

7. Validation des deliveries

7 – Réunions

1 – La charte du projet

Sign it est un software pour traduire en temps réelle les langue des signes.

A - Chef de projet

Pierre Ginisty

B - Date de la dernière révision

Version	Date	Auteur	Description
1.0	05/02/2024	Pierre	Premier Jet
1.1	08/03/2024	Pierre	Second Jet
1.2	20/10/2024	Pierre	Troisième Jet
2.0	10/01/2025	Pierre	Aboutissement

C - But du projet

Réaliser une software capable de traduire en temps réelle une langue des signes en textuelle et vice versa.

D - Objectifs du projet

Un software compatible avec windows et linux.

Utilisation de MediaPipe pour la détection de gestes, y compris la détection de la main et des mouvements associés.

Utilisation de Unity pour une future modélisation de mains 3D.

E - Portée du projet

Les portées seront nombreuses, et augmenteront avec le temps.

Cependant il est notable que rendre possible une communication en temps réelle de deux personnes signant dans une langue des signes différentes via une application mobile ne pourra être respecté dans les temps du projet.

F - Équipe et ressources du projet

L'équipe est composée de :

Nom	Prénom	Mail
Gaillet	Aïmane	aimane.gaillet@epitech.eu
Loucif	Inasse	inasse.loucif@epitech.eu
Vienne	Killian	killian.vienne@epitech.eu
Ginisty	Pierre	pierre.ginisty@epitech.eu
Rosso	Romain	romain.rosso@epitech.eu
Oukziz	Salma	salma.oukziz@epitech.eu
Tomietto	Vincent	vincent.tomietto@epitech.eu
Hamda	Salsabil	salsabil.hamda@epitech.eu

Les outils seront :

- MediaPipe pour le pôle IA
- Unity pour le pôle AR/VR

le reste des outils sont encore à définir.

G - Parties prenantes et approbateurs

L'équipe de Signit et Epitech représentent les deux parties prenante du projet.

Les livrables du projet et la charte se doivent d'être approuver par les deux parties.

2 – Le cahier des charges

La dernière version (version 1.2.2) de notre cahier des charges est disponible sur gamme au lien suivant :

[Organisation : Cahier des charges SIGNIT | Gamma](#)

3 – Plan de ressources

A – gestion de projet

Notre gestion de projet se fait au travers de la méthode Kanban via un notion.

Lien du notion : <https://www.notion.so/3c27b7c6653f451195aef2431db9127c?v=1c50ec49a4464c40890db5c06f6f5c80&pvs=13>

Langue des signes

The screenshot shows a Kanban board in Notion titled "Langue des signes". The board is organized into three columns: "Pas commencé" (2 items), "En cours" (2 items), and "Terminé" (4 items). Each item is a card with a title and a comment icon. The "Pas commencé" column contains "Effectuer des Tests Utilisateur" and "Établir une Stratégie de Suivi et de Maintenance". The "En cours" column contains "Réaliser un Plan de Qualité" and "Préparer une Stratégie de Lancement". The "Terminé" column contains "Élaborer un Plan de Communication", "Faire de la Recherche", "Développer un Plan de Gestion des Risques", and "Développer une Stratégie de Formation". A "Nouveau" button is visible at the bottom of each column.

B – calendrier

Un premier prototype d'un calendrier a été réalisé sur notion. Le calendrier sera vérifié régulièrement comme étant respecté par le chef de projet : Pierre GINISTY

The screenshot shows a calendar view in Notion titled "Langue des signes". The calendar is organized into three rows: "Pas commencé" (2 items), "En cours" (2 items), and "Terminé" (4 items). Each item is a card with a title and a comment icon. The "Pas commencé" row contains "Effectuer des Tests Utilisateur" and "Établir une Stratégie de Suivi et de Maintenance". The "En cours" row contains "Réaliser un Plan de Qualité" and "Préparer une Stratégie de Lancement". The "Terminé" row contains "Élaborer un Plan de Communication", "Faire de la Recherche", "Développer un Plan de Gestion des Risques", and "Développer une Stratégie de Formation". A "Nouvelle page" button is visible at the bottom of each row. The calendar is displayed for the month of February, with a date picker at the top showing "février" and "6".

4 – Plan de tests

TECHNOS

Selenium :

Framework de test automatisé qui prend en charge le navigateur pour le Web, mais également des applications mobiles via des outils tels que Appium. - Avantages : Universel pour le Web, Android et iOS.

- *Inconvénients : Pour les applications mobiles, il fonctionne souvent en combinaison avec d'autres frameworks comme Appium.*

Appium :

Framework open source pour le test d'applications mobiles qui prend en charge Android et iOS.

- *Avantages : Multiplateforme, prend en charge plusieurs langages de programmation (Java, Python, JavaScript, Ruby, C#).*
- *Inconvénients : La configuration peut être complexe.*

BATTERIE DE TESTS

1. Tests Unitaires :

Vérifier chaque composant individuel de l'application.

- *Tester les fonctions de traduction, les fonctionnalités d'apprentissage, les interactions AR.*

2. Tests d'Intégration :

Vérifier la communication entre les différents modules de l'application.

- *S'assurer que la traduction en temps réel s'intègre correctement avec la réalité augmentée.*

3. Tests de Système :

Évaluer l'application dans son ensemble.

- *Vérifier la compatibilité avec différents dispositifs, évaluer les performances globales.*

4. Tests de Performance :

Évaluer la réactivité et la vitesse de l'application.

- *Tester la traduction en temps réel dans des conditions de charge élevée, évaluer la gestion des ressources.*

5. Tests d'Interface Utilisateur (UI) et d'Expérience Utilisateur (UX) :

Assurer une interface utilisateur intuitive et accessible.

- *Vérifier la facilité d'utilisation des fonctionnalités AR, tester l'accessibilité pour les utilisateurs avec des besoins particuliers.*

6. Tests de Sécurité et de Confidentialité :

Protéger les données utilisateur et garantir la sécurité de l'application.

- *Tester la robustesse contre les attaques, assurer la conformité aux normes de protection des données.*

7. Tests de Compatibilité :

Vérifier le bon fonctionnement sur différentes plateformes et dispositifs.

- Tester sur différents navigateurs, systèmes d'exploitation, résolutions d'écran.

8. Tests d'Accessibilité :

Assurer l'inclusion pour tous les utilisateurs, y compris ceux avec des besoins particuliers. - Tester la compatibilité avec des lecteurs d'écran, évaluer la navigation pour les utilisateurs malvoyants.

9. Tests de Charge et de Stress :

Vérifier la stabilité de l'application sous des conditions extrêmes.

- Tester la traduction en temps réel avec un grand nombre d'utilisateurs simultanés.

10. Tests de Mises à Jour et de Rétrocompatibilité :

Vérifier que les mises à jour n'altèrent pas la stabilité de l'application.

- Tester la rétrocompatibilité avec les versions précédentes de l'application.

11. Tests de Communication :

Vérifier la qualité des interactions avec les utilisateurs.

- Tester les fonctionnalités de communication en temps réel, les notifications.

12. Tests de Scénarios Utilisateur Réels :

Évaluer l'application dans des situations proches de l'utilisation réelle.

- Simuler des scénarios d'utilisation quotidienne, tester la réaction de l'application à des situations imprévues.

13. Tests de Retour en Arrière (Rollback) :

Vérifier la capacité à revenir à une version antérieure en cas de problème majeur.

- Tester le processus de rollback après une mise à jour défectueuse.

14. Tests de Documentation :

Vérifier la cohérence et l'utilité de la documentation associée à l'application.

- Tester les tutoriels, les guides d'utilisation, etc.

15. Tests de Traduction et de Localisation :

Vérifier la qualité des traductions et la localisation de l'application.

- Tester sur différentes langues et s'assurer que les éléments de l'interface utilisateur sont correctement traduits.

16. Tests de Communication avec les Dispositifs Matériels :

Tester l'intégration avec les caméras, capteurs, dispositifs AR, etc.

- S'assurer que l'application utilise correctement les fonctionnalités matérielles.

17. Tests de Coexistence avec d'autres Applications :

Vérifier que l'application fonctionne correctement en même temps que d'autres applications.

- Tester la coexistence avec d'autres applications AR ou de communication.

18. Tests de Récupération après Panne (Recovery) :

Vérifier la capacité de l'application à récupérer après un incident.

- Tester la restauration des données après une coupure de connexion.

19. Tests de Performance en Conditions Réelles :

Tester l'application dans des conditions réelles d'utilisation.

- Tester la traduction en situation de faible luminosité, des mouvements rapides, etc.

20. Tests de Compatibilité avec les Standards du Langage des Signes :

Vérifier la précision de la traduction par rapport aux normes du langage des signes.

- Comparer la traduction avec des experts en langage des signes.

Documentation des Tests :

• Documentez soigneusement chaque scénario de test, les résultats attendus, les résultats réels, et les problèmes identifiés.

5 – Le plan de communication

Introduction : Le projet de traducteur de langage des signes français vers le langage humain vise à faciliter la communication inclusive en permettant aux personnes sourdes de communiquer plus facilement avec les autres. Cette introduction met en lumière l'importance de la communication inclusive dans la société et souligne le rôle crucial que la technologie peut jouer dans ce domaine.

Objectifs de communication :

Sensibiliser le public à l'importance de la communication inclusive pour les personnes sourdes : Ce premier objectif vise à sensibiliser le grand public sur les défis auxquels sont confrontées les personnes sourdes dans leur quotidien en matière de communication, et à promouvoir une meilleure compréhension de leurs besoins.

Promouvoir l'utilisation de la technologie pour faciliter la communication interculturelle : Ce deuxième objectif met en avant le rôle de la technologie, en particulier le traducteur de langage des signes, dans la création de ponts entre les cultures et les langues, favorisant ainsi une communication plus fluide et inclusive.

Publics Cibles :

Personnes sourdes ou malentendantes : Ces individus sont les principaux bénéficiaires du traducteur de langage des signes. Ils sont au cœur de l'effort de sensibilisation et de promotion de la technologie.

Professionnels de la santé : Les médecins, infirmiers et autres professionnels de la santé sont des cibles importantes car ils peuvent bénéficier directement de l'utilisation du traducteur pour améliorer leurs interactions avec les patients sourds.

Éducateurs et enseignants : Les éducateurs et enseignants sont également des publics clés, car ils peuvent utiliser la technologie pour faciliter la communication avec les élèves sourds et favoriser un environnement d'apprentissage inclusif.

Grand public : Sensibiliser le grand public est essentiel pour favoriser l'inclusion et encourager une meilleure compréhension des besoins des personnes sourdes.

Messages Clés :

Pour les personnes sourdes : "Une communication fluide et inclusive grâce à notre traducteur de langage des signes." Ce message met en avant les avantages directs du traducteur pour les utilisateurs sourds, en soulignant son rôle dans l'amélioration de leur communication.

Pour les professionnels de la santé : "Améliorez la qualité des interactions avec

vos patients sourds grâce à notre technologie innovante." Ce message cible les professionnels de la santé en mettant en avant les bénéfices professionnels de l'utilisation du traducteur dans un contexte médical.

Pour le grand public : "Découvrez comment notre traducteur de langage des signes favorise l'inclusion et la diversité." Ce message vise à sensibiliser le grand public sur le rôle de la technologie dans la promotion de l'inclusion et de la diversité dans la société.

Canaux de Communication :

Médias sociaux : Utilisation de LinkedIn pour partager des actualités, des vidéos explicatives et des témoignages sur le projet.

Site Web dédié au projet : Le site web sera une plateforme centrale pour présenter le projet, fournir des informations détaillées et recueillir des retours d'utilisateur.

Événements de sensibilisation : Organisation de conférences, d'ateliers dans les écoles et les centres communautaires pour sensibiliser le public aux enjeux de la communication inclusive et présenter le traducteur de langage des signes.

Partenariats avec des associations de personnes sourdes et des organisations communautaires : Collaborations avec des organisations ayant une expertise dans le domaine de la surdité pour étendre la portée du projet et bénéficier de leur soutien dans la sensibilisation et la promotion de la technologie.

Communiqués de presse et articles dans les médias locaux : Diffusion d'informations sur le projet et ses avancées à travers des communiqués de presse et des articles dans les médias locaux pour toucher un public plus large.

Calendrier de Communication :

Février : Lancement du site web et des pages sur les réseaux sociaux pour présenter le projet au grand public.

Mars : Participation à des événements de sensibilisation et publication d'articles dans les médias locaux pour accroître la visibilité du projet.

Avril : Développement de partenariats avec des associations de personnes sourdes pour renforcer les efforts de sensibilisation et de promotion.

Mai : Organisation d'ateliers dans les écoles et les centres communautaires pour présenter le traducteur de langage des signes et recueillir des retours d'utilisateur.

Juin : Évaluation des retombées médiatiques et de l'engagement sur les médias sociaux pour mesurer l'impact des efforts de communication.

Évaluation :

Nombre de visiteurs sur le site web et les réseaux sociaux : Mesure de l'engagement en ligne avec le projet.

Engagement sur les publications des médias sociaux (likes, partages,

commentaires) : Indicateur de l'interaction et de l'intérêt du public pour le contenu partagé.

Retombées médiatiques (nombre d'articles publiés, audiences touchées) : Évaluation de la couverture médiatique et de la portée du projet.

Retours d'information des publics cibles lors des événements de sensibilisation et des ateliers : Collecte de retours d'utilisateur pour évaluer l'efficacité du traducteur de langage des signes et identifier les possibilités d'amélioration.

Ajustements et Optimisations :

Analyse des résultats et ajustements du plan de communication en fonction des retours d'information et des performances obtenues. Cette analyse comprendra une évaluation approfondie des données quantitatives et qualitatives, ainsi que des ajustements stratégiques pour renforcer les points forts et corriger les lacunes identifiées dans le plan de communication.

6 – Gestion des Tickets et Deliveries

Pour assurer une gestion efficace des tâches et des fonctionnalités, nous utilisons Jira pour transformer les tâches en tickets et suivre leur progression. Chaque ticket représente une tâche ou une fonctionnalité à implémenter, tester ou valider.

Notre jira est disponible ici :

<https://signitproject.atlassian.net/jira/software/projects/SIG/boards/1>

A – Gestion des tickets

1. Création de Tickets :

- Les tâches à réaliser sont créées sous forme de tickets dans GitHub Project.
- Chaque ticket doit contenir une description claire et concise de la tâche à accomplir, les critères d'acceptation, le temps estimé et les dépendances éventuelles.

2. Assignment des Tickets :

- Le Scrum Master assigne un développeur à chaque ticket, ainsi qu'un reviewer qui sera responsable de la validation du travail.
- L'assignation se fait en fonction des compétences et de la charge de travail des développeurs.

3. Branches de Fonctionnalité :

- Pour chaque ticket, une branche de fonctionnalité est créée à partir de la branche principale (master).
- La convention de nommage pour les branches est la suivante : `feature/<nom_du_ticket>`.

4. Développement :

- Le développeur travaille sur la branche de fonctionnalité et implémente les modifications nécessaires.
- Les commits doivent être fréquents et bien documentés, en suivant la convention de messages de commit.

5. Merge Requests :

- Une fois le développement terminé, une merge request (MR) est créée pour intégrer la branche de fonctionnalité dans la branche principale.
- La MR doit inclure une description détaillée des modifications apportées, ainsi que les tests effectués.

6. Revue de Code :

- Le reviewer assigné vérifie le code, s'assure qu'il répond aux critères d'acceptation et respecte les standards de qualité.

- Le reviewer peut approuver la MR, demander des modifications ou rejeter la MR en fonction de la qualité du code.

7. Intégration Continue :

- Des pipelines CI/CD sont configurés pour s'exécuter à chaque push sur une branche de fonctionnalité.
- Les pipelines installent, testent et buildent le projet automatiquement.
- Si tous les tests passent et que le build est réussi, la MR peut être fusionnée dans la branche principale.

8. Suivi des Tickets :

- Une fois la MR fusionnée, le ticket est fermé et marqué comme complété dans GitHub Project.
- Le Scrum Master suit la progression des tickets pour assurer le respect des délais et des objectifs du sprint.

Outils utilisés:

- **GitHub Project** pour la gestion des tickets et la planification des sprints.
- **GitHub Issues** pour la création et le suivi des tickets.
- **GitHub Actions** pour les pipelines CI/CD.

B – Gestion des deliveries

Les deliveries (livraisons) représentent les versions packagées de notre application, prêtes à être déployées ou distribuées. Une gestion efficace des deliveries garantit que chaque version de l'application est stable, bien testée et documentée.

1. Packaging :

- Lorsqu'une merge request est fusionnée dans la branche principale, une nouvelle version du projet est packagée.
- Le packaging inclut la compilation du code, la génération des fichiers binaires et la création des archives de distribution.

2. Pipelines de Déploiement :

- Des pipelines CI/CD spécifiques sont configurés pour s'exécuter à chaque merge dans la branche principale.
- Ces pipelines réalisent les étapes suivantes :
 - **Compilation** : Compiler le code source.
 - **Tests** : Exécuter tous les tests unitaires et d'intégration pour s'assurer que le build est stable.

- **Packaging** : Créer les artefacts nécessaires pour la distribution (APK pour Android, IPA pour iOS, etc.).

3. Versioning :

- Chaque livraison est associée à une version unique, suivant les conventions de versioning sémantique (MAJOR.MINOR.PATCH).
- Le numéro de version est mis à jour en fonction des changements apportés :
 - **MAJOR** : Changements incompatibles avec les versions précédentes.
 - **MINOR** : Ajouts de fonctionnalités compatibles avec les versions précédentes.
 - **PATCH** : Corrections de bugs et améliorations mineures.

4. Stockage des Deliveries :

- Les artefacts générés par le pipeline de packaging sont enregistrés dans un stockage sécurisé.
- GitHub Releases est utilisé pour stocker et distribuer les versions packagées de l'application.

5. Documentation des Versions :

- Chaque livraison est accompagnée de notes de version (release notes) détaillant les nouvelles fonctionnalités, les améliorations et les corrections de bugs.
- Les notes de version sont mises à jour dans GitHub Releases.

6. Déploiement :

- Les versions packagées sont déployées sur les plateformes cibles (App Store, Google Play, etc.).
- Les déploiements peuvent être automatiques ou manuels, en fonction de la stratégie de déploiement définie par l'équipe.

7. Validation des Deliveries :

- Avant le déploiement, chaque version packagée est validée par l'équipe QA.
- Des tests de régression et des tests de performance sont exécutés pour garantir la qualité de la livraison.

Outils utilisés:

- **GitHub Actions** pour l'automatisation des pipelines CI/CD.
- **GitHub Releases** pour le stockage et la distribution des versions packagées.
- **Systèmes de stockage sécurisé** pour les artefacts de build.
- **App Store Connect** et **Google Play Console** pour le déploiement des applications mobiles.

7 – Les minutes de réunion

Tous les minutes réunions sont disponibles sur notre github :

<https://github.com/ESP-SignIt/ESP-SignIt/wiki/Rapport-de-R%C3%A9union>

Et sont tenus par le scribe de l'équipe : Killian Vienne.