



RAPPORT DE STAGE

Stage effectué du 12 mai au 20 juin 2025

Tuteur de stage : Monsieur BOISSIERE

Enseignant référent : Monsieur DUGAST

Organisme d'accueil : Kiliogene – 26 rue Nicolas Appert, 44100 Nantes

Etablissement scolaire : La Joliverie – BTS Services Informatiques aux Organisations 1SLAM

MERHAND Nora

Remerciement

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont contribué au bon déroulement de mon stage et à la réalisation de ce rapport.

Je remercie tout particulièrement Aurélien BOISSIERE, mon tuteur de stage chez Kiliogene, pour son encadrement, sa disponibilité et ses conseils tout au long de cette expérience. Je remercie également toute l'équipe de Kiliogene Core pour leur accueil chaleureux et leur bienveillance.

Je souhaite également remercier Monsieur DUGAST, mon professeur référent, pour son accompagnement durant cette période.

Enfin, j'adresse un remerciement tout particulier à ma famille et à mes proches pour leur soutien constant.

Table des matières

INTRODUCTION	1
PRESENTATION DE L'ORGANISME D'ACCUEIL.....	1
IDENTIFICATION	1
GESTION DES CONFIGURATIONS	1
CONTEXTE DU STAGE	2
CONDITIONS D'ACCUEIL	2
ENCADREMENT	2
MOYENS MIS A DISPOSITION.....	2
MISSIONS ET ACTIVITES REALISEES	3
PRESENTATION GLOBALE DES MISSIONS	3
ACTIVITES PRINCIPALES	3
OUTILS ET RESSOURCES UTILISES	7
LOGICIELS ET TECHNOLOGIES	7
FORMATIONS PERSONNELLE	8
COMPETENCES DEVELOPPEES	8
BILAN TECHNIQUE	9
REALISATIONS.....	9
ELEMENTS EN COURS OU NON TERMINES.....	9
CONCLUSION	10
ANNEXES.....	11

Introduction

Au cours de ma première année de BTS Services Informatiques aux Organisations (SIO), j'ai eu l'opportunité d'effectuer un stage au sein de l'entreprise Kiliogene, située à Nantes. Ce stage de six semaines a constitué une étape clé de ma formation, en me permettant de mettre en pratique, dans un contexte professionnel, les compétences abordés en formation. Il m'a offert une immersion concrète dans le monde professionnel.

Grâce à cette expérience, j'ai pu approfondir mes compétences techniques notamment en HTML, CSS et JavaScript / TypeScript, mais aussi découvrir de nouvelles technologies spécifiques telles que les Lightning Web Components (LWC). Ce stage a confirmé mon intérêt pour le développement informatique et a renforcé ma motivation à poursuivre dans cette voie, tout en développant une meilleure compréhension du fonctionnement d'une entreprise IT.

Présentation de l'organisme d'accueil

Identification

Kiliogene, nommé Kiliogene Core pour le site de Nantes, est une entreprise de conseil dynamique, spécialisée dans la gestion de la relation client (CRM) via la plateforme Salesforce. Créée à Nantes, la société s'est développée avec des bureaux supplémentaires à Bordeaux, Angers et même à Abidjan en Côte d'Ivoire, témoignant de son rayonnement international. Kiliogene accompagne des clients issus de secteurs variés – du commerce au service en passant par l'industrie – dans leur transformation numérique, en apportant des solutions personnalisées, adaptées à leurs besoins.

L'entreprise se distingue par sa maîtrise de la plateforme Salesforce, couvrant l'intégration, le développement de solutions sur mesure et la maintenance évolutive. Elle joue ainsi un rôle clé dans l'optimisation des processus métiers de ses clients, contribuant à améliorer leur efficacité et leur compétitivité.

Gestion des configurations

La gestion des configurations chez Kiliogene repose sur une méthodologie rigoureuse adaptée à l'environnement Salesforce. Le recours à des systèmes de contrôle de version, notamment Git, constitue un élément central du processus de développement. Cela leur permet de suivre précisément les modifications du code source, d'assurer la traçabilité des évolutions, et de faciliter le travail en équipe, même lorsque plusieurs développeurs interviennent simultanément sur un même projet.

Durant mon stage, j'ai ainsi appris à utiliser Git en l'intégrant à Visual Studio Code, mon principal environnement de développement. J'ai également découvert les

commandes essentielles (commit, push, pull, branch, merge). Cette maîtrise de Git constitue une compétence technique incontournable dans le domaine du développement informatique.

Contexte du stage

Conditions d'accueil

Mon stage s'est déroulé du 12 mai au 20 juin 2025, sur six semaines. Mes horaires étaient de 9h à 17h, avec une pause déjeuner de 12h30 à 13h30. Mon poste de travail était situé dans un open-space partagé avec mon tuteur Aurélien BOISSIERE, un développeur et un autre stagiaire. Cette proximité a facilité les échanges et l'entraide, tout en m'offrant un environnement stimulant et propice à la concentration.

Encadrement

Le rôle de mon tuteur a été primordial tout au long de mon stage. Aurélien BOISSIERE a su m'accompagner avec bienveillance et disponibilité, répondant à mes questions et m'orientant dans la réalisation de mes missions. Son expérience m'a permis d'acquérir des connaissances solides et de progresser rapidement.

L'équipe dans son ensemble a été très accueillante, créant une ambiance de travail agréable et motivante. Un rendez-vous quotidien en visioconférence, appelé "daily", avait lieu chaque jour à midi. Il rassemblait les membres de l'équipe, y compris ceux travaillant à distance ou depuis d'autres villes que Nantes. Ces points réguliers favorisaient la cohésion, permettaient de suivre l'avancée des projets, et donnaient l'occasion à chacun de partager ses avancées, ses blocages ou ses besoins.

Même si j'ai souvent travaillé en autonomie, cette dynamique collective et cette culture d'échange m'ont aidée à m'intégrer facilement et à prendre confiance en mes capacités. L'ambiance générale chez Kiliogene alliait professionnalisme et convivialité, ce qui a largement contribué à rendre ce stage aussi formateur qu'agréable.

Moyens mis à disposition

À mon arrivée, Nicolas CORCELLE, fondateur et PDG de l'entreprise, m'a fourni un ordinateur équipé de deux écrans, d'un clavier et d'un trackpad. J'ai également reçu une tasse et une gourde à l'effigie de l'entreprise, un geste d'accueil que j'ai beaucoup apprécié.

J'ai également bénéficié d'un accès aux outils indispensables à mon travail, notamment Arc, Slack, Notion, Passbolt, ainsi que Visual Studio Code, mon principal environnement de développement. J'ai pu me familiariser progressivement avec l'environnement de travail et les outils internes, ce qui m'a permis de devenir rapidement autonome.

Enfin, la plateforme Trailhead m'a permis de me former de manière autonome sur Salesforce et le développement LWC, en lien direct avec les missions qui m'ont été confiées par la suite.

Missions et activités réalisées

Présentation globale des missions

Au cours de mon stage chez Kiliogene, ma principale mission a été l'amélioration et la création de composants LWC (Lightning Web Component), une technologie liée à la plateforme Salesforce. J'ai été amenée à intervenir à la fois sur des composants existants, en les améliorant selon les besoins de l'entreprise, et sur la création de composants personnalisés, en partant d'un cahier des charges et avec des indications complémentaires de mon tuteur. Ce travail m'a permis de mobiliser des compétences en développement front-end (JavaScript / TypeScript, HTML, CSS) tout en découvrant l'environnement Salesforce et les spécificités du framework LWC.

Activités principales

J'ai eu l'occasion de travailler sur quatre composants différents : une *dual-picklist*, une *timeline*, une *checkbox* et une *tree grid*. Ces composants ont chacun posé des problématiques spécifiques, allant de l'ajout de fonctionnalités à l'adaptation sur différents supports. Leur développement m'a offert l'opportunité de structurer mon code, de gérer des cas d'usage variés, et de mieux appréhender les bonnes pratiques professionnelle attendues dans le métier de développeuse.

J'ai également veillé à ce que chaque composant soit réutilisable et adaptable, c'est-à-dire générique. Les données utilisées ne devaient pas être codées en dur, mais transmises sous forme paramétrique, afin de permettre une intégration souple dans différents contextes. Chaque composant devait être utilisable à la fois dans un Flow Salesforce et dans un autre composant LWC. Bien que je me sois principalement servi des Flows pour tester mes composants, il s'agit d'outils visuels permettant d'automatiser des processus métiers sans code, en enchaînant des actions telles que l'affichage d'écrans, la mise à jour de données ou l'envoi d'e-mails. Pour certains composants, j'ai également développé des CPE (Custom Property Editors), qui m'ont permis d'ajouter des listes déroulantes dans l'interface de configuration du Flow, afin de sélectionner des valeurs prédéfinies (voir *Annexe 1*).

Pour chaque composant, j'ai également dû réaliser des tests unitaires à l'aide du framework Jest, couramment utilisé dans l'écosystème JavaScript. Ces tests avaient pour objectif de valider le bon fonctionnement des différentes fonctionnalités en simulant des comportements utilisateurs et des cas d'usage variés. L'utilisation de Jest m'a permis d'automatiser ces vérifications, de repérer plus rapidement les erreurs

potentielles, et de garantir la fiabilité du code dans la durée. Ce travail m’a sensibilisée à l’importance des tests dans le cycle de développement (voir *Annexe 2*).

◆ Dual-picklist

Le premier composant sur lequel j’ai travaillé était une dual-picklist, un composant existant mais qui ne répondait pas entièrement aux besoins de l’entreprise. En effet, ce composant permet de sélectionner plusieurs éléments parmi deux listes, mais son utilisation devenait difficile en présence d’un grand nombre d’éléments présentant des similitudes.

La demande de l’entreprise portait donc sur l’ajout d’une barre de recherche pour filtrer dynamiquement le contenu de chaque liste. J’ai intégré cette fonctionnalité en TypeScript et HTML. Cette amélioration a significativement amélioré l’ergonomie du composant (voir *Annexe 3*). J’ai également proposé d’ajouter un message d’erreur s’affichant lorsqu’aucune correspondance n’était trouvée : cette suggestion a été examinée puis validée par mon tuteur, qui a jugé cette indication plus claire pour les utilisateurs (voir *Annexe 4*). Par ailleurs, mon tuteur m’a demandé de prendre en compte une recherche en multi-mots floue. C’est-à-dire que l’utilisateur peut taper plusieurs mots, même dans un ordre différent, et le filtre doit tout de même être capable de retrouver les résultats pertinents. Pour cela, j’ai mis en place un algorithme de comparaison flexible permettant d’identifier les correspondances partielles et de les afficher dynamiquement.

Un autre point de réflexion a concerné le nombre de barres de recherche à intégrer : une par liste, ou une seule pour l’ensemble. Mon tuteur penchait initialement pour deux barres distinctes, afin de permettre une recherche séparée dans chaque liste. De mon côté, je voyais davantage une seule barre de recherche, appliquée globalement. J’ai donc commencé par implémenter cette version, que mon tuteur a finalement jugée tout à fait pertinente malgré sa première idée. Pour trancher, nous avons consulté d’autres membres de l’équipe, dont le retour a confirmé que le filtrage unique facilitait la comparaison et la sélection. Ce choix a donc été validé collectivement, et j’ai pu finaliser le développement de la dual-picklist

◆ Timeline

Le deuxième composant sur lequel j’ai travaillé était une timeline (ligne du temps), que j’ai développée à partir d’un cahier des charges précis et de maquettes réalisées en amont. À première vue, la structure de ce composant semblait relativement simple : il s’agissait d’afficher une succession d’événements le long d’une ligne. Toutefois, la véritable complexité est apparue lorsque j’ai dû intégrer la gestion des différentes orientations. En effet, la timeline devait pouvoir s’afficher sous quatre formes : verticale (voir *Annexe 5*), horizontale (voir *Annexe 6*), verticale inversée (voir *Annexe 7*) et horizontale inversée (voir *Annexe 8*).

Cette exigence impliquait d'adapter dynamiquement la disposition des éléments, leur alignement, ainsi que la direction de lecture. J'ai notamment dû créer une flèche indiquant le sens de progression de la timeline. Le positionnement précis de cette flèche s'est avéré complexe, car il devait s'ajuster automatiquement à l'orientation choisie, tout en conservant une cohérence esthétique et fonctionnelle.

D'un point de vue fonctionnel, la timeline devait aussi limiter l'affichage initial à trois événements, avec la possibilité pour l'utilisateur de révéler les événements suivants en cliquant sur un bouton (« Afficher plus », voir *Annexe 9*). En développant cette fonctionnalité, j'ai rencontré un cas particulier lié à l'affichage horizontal. Lorsque le nombre d'événements dépassait la largeur disponible du composant, il devenait nécessaire d'introduire une fonction de défilement horizontal (scroll) pour garantir la visibilité de tous les éléments.

Cependant, cela a généré un problème d'affichage : en orientation horizontale classique, les icônes positionnées sur la ligne centrale apparaissaient coupées en deux (voir *Annexe 10*), alors que ce dysfonctionnement ne survenait pas en orientation horizontale inversée. Pour résoudre cela, j'ai ajusté la position des éléments dans leur conteneur afin de les rendre de nouveau visibles dans leur intégralité. Ce correctif a toutefois révélé une nouvelle difficulté : la ligne de la timeline ne s'étendait pas au-delà de la largeur visible de l'écran, ce qui cassait l'alignement graphique.

J'ai donc créé un main-container englobant tous les éléments, et appliqué une propriété CSS 'overflow-x: auto' à ce conteneur, permettant d'activer le défilement horizontal tout en maintenant la continuité de la ligne. Cette solution a permis de conserver l'esthétique attendue du composant tout en assurant sa compatibilité avec les différents cas d'usage.

◆ Nullable checkbox

Ce troisième composant m'a été confiée suite à une réunion de mon tuteur. Cette mission consistait en la modification d'une checkbox afin qu'elle puisse gérer un booléen nullable (c'est-à-dire pouvant être true, false, null ou undefined). Dans les cas où la valeur était null ou undefined, la checkbox devait automatiquement devenir désactivée (disabled). De plus, lorsque la checkbox était désactivée, son label devait être affiché dans une teinte grisée afin de signaler visuellement son inactivité à l'utilisateur. J'ai également ajouté une propriété personnalisable permettant de positionner le label à gauche ou à droite de la checkbox selon les préférences d'affichage (voir *Annexe 11*).

◆ Tree grid

Enfin, le composant le plus complexe auquel j'ai été confrontée fut une tree grid, un tableau hiérarchique permettant d'afficher des données imbriquées. Le composant existant n'était pas adapté à une utilisation sur mobile, ce qui en limitait fortement l'usage. Ma mission a donc été de le rendre 'responsive'. Pour cela, j'ai dû reproduire intégralement le composant, d'abord en version desktop, puis en version mobile. Ce

travail a impliqué la mise en place d'une récursivité pour gérer l'affichage des lignes enfants.

Sur mobile, l'affichage classique sous forme de tableau n'était ni lisible ni ergonomique. Après échange avec mon tuteur, nous avons convenu de n'afficher que la première colonne sur téléphone, puis de réfléchir à une solution adaptée pour le reste des données. J'ai donc réalisé plusieurs maquettes sur Figma (voir *Annexe 12, 13 & 14*) afin d'explorer différentes possibilités d'affichage. Ma première idée était d'afficher une fenêtre modale afin de visualiser les données des autres colonnes (voir *Annexe 13*). Ma deuxième idée, partagé par mon tuteur, a été la mise en place d'une fonctionnalité de type 'accordéon', permettant ainsi d'afficher, à la demande, l'ensemble des informations associées à une ligne, en dessous de celle-ci (voir *Annexe 14*). Pour m'éclairer sur la meilleure solution à adopter, j'ai consulté certains membres de l'entreprise, notamment ceux plus orientés vers l'expérience utilisateur. Il a été décidé que la mise en place de la fonctionnalité "accordéon" répondait au mieux aux besoins. J'ai donc appliqué cette option pour le composant Tree grid sur mobile.

Le développement de ce composant m'a confrontée à plusieurs défis technique, notamment concernant la gestion des paramètres d'entrée au format JSON. Ce format m'était totalement inconnu avant ce stage, j'ai donc dû apprendre à m'en servir. J'ai également dû faire face à des problèmes d'affichage liés aux différences entre les formats ordinateur et mobile, ce qui m'a obligée à adapter le composant pour qu'il offre une expérience optimale sur chaque type d'écran. J'ai ainsi conditionné le comportement du composant en fonction de la taille de celui-ci, avec des affichages et fonctionnalités distinctes. Cette approche m'a permis de conserver la version tableau sur ordinateur (voir *Annexe 15*), tout en n'affichant que la première colonne de celui-ci sur téléphone et d'y insérer la fonctionnalité 'accordéon', améliorant ainsi significativement l'ergonomie et la lisibilité sur mobile (voir *Annexe 16*).

Une fois l'affichage stabilisé, j'ai travaillé à l'ajout de fonctionnalités avancées afin d'atteindre le niveau de l'ancien composant. J'ai notamment détecté automatiquement les colonnes de type 'url' pour transformer les valeurs concernées en liens cliquables. J'ai également implémenté un système de sélection par checkbox : l'utilisateur peut cocher une ligne, mais cette sélection ne s'applique qu'à la ligne en elle-même, sans affecter ses enfants (voir *Annexe 17*). Sur ordinateur, ce système fonctionnait parfaitement, mais sur mobile, l'ajout de checkbox alourdissait visuellement l'interface. Après échange avec mon tuteur, nous avons convenu d'une interaction plus naturelle : la sélection d'une ligne s'effectue par un appui long. Pour indiquer qu'une ligne est sélectionnée, j'ai choisi d'en griser le fond, ce qui respecte les standards d'accessibilité (voir *Annexe 18*).

En prenant exemple sur la version originale du composant Tree Grid, j'ai ajouté une autre checkbox à côté du nom de la première colonne, permettant de sélectionner l'ensemble des lignes du tableau, y compris les enfants non dépliés. Une fois cette fonctionnalité ajoutée, j'ai demandé l'avis de mon tuteur. Il m'a dit que c'était bien, mais

qu'il faudrait montrer que toutes les lignes sont sélectionnées : soit en les ouvrant toutes, afin que l'utilisateur puisse visualiser ce qu'il a sélectionné, soit en ne sélectionnant que les lignes visibles. Il m'a alors conseillé de demander à notre designeuse, afin qu'elle me donne son avis sur la meilleure solution à adopter. Elle m'a dit que le mieux serait d'ouvrir toutes les lignes, et que même s'il y en a beaucoup, cela ne poserait pas de problème majeur, étant donné que cette fonctionnalité est uniquement disponible sur ordinateur. J'ai donc ajusté mon composant en suivant ces consignes (voir *Annexe 19*).

Ce composant m'a permis de mettre en œuvre de nombreuses compétences techniques, mais aussi de réfléchir à l'ergonomie, à l'adaptabilité du code, et à la conception orientée utilisateur dans un cadre professionnel exigeant.

L'ensemble des missions réalisées durant ce stage m'a permis d'explorer concrètement les différentes facettes du développement front-end dans un environnement professionnel exigeant. Travailler sur des composants LWC réutilisables, adaptables et robustes m'a permis d'approfondir mes connaissances techniques tout en découvrant les standards attendus dans le cadre d'un projet d'entreprise.

Chaque composant m'a confrontée à des problématiques variées, parfois inattendues, qui m'ont poussée à chercher, tester, échouer, recommencer, et finalement trouver des solutions efficaces, souvent en collaboration avec mon tuteur ou d'autres membres de l'équipe. Ces échanges ont été particulièrement enrichissants sur les aspects liés à l'ergonomie et à l'expérience utilisateur, notamment dans le cadre du développement de la tree grid, qui m'a demandé rigueur, créativité et persévérance.

Cette expérience m'a permis de gagner en autonomie, d'améliorer ma capacité à prendre des décisions techniques argumentées, et d'acquérir une méthodologie de travail rigoureuse que je pourrai réinvestir dans mes futurs projets.

Outils et ressources utilisés

Logiciels et technologies

Durant mon stage, j'ai utilisé plusieurs outils professionnels :

- ◆ Visual Studio Code pour le développement des composants LWC
- ◆ Arc pour les recherches et la consultation des documentations
- ◆ Slack pour la communication interne
- ◆ Notion pour le suivi des missions (tickets)
- ◆ Passbolt pour la gestion des mots de passe
- ◆ Figma pour la création de maquettes
- ◆ Salesforce Developer Console et Lightning App Builder pour le test et l'intégration des composants

Les langages utilisés étaient principalement TypeScript / JavaScript, HTML et CSS, dans le cadre du développement LWC, avec des notions de JSON pour le paramétrage.

Formations personnelle

Une part importante de mon autonomie s'est appuyée sur l'autoformation. J'ai suivi plusieurs modules de Trailhead, la plateforme d'apprentissage de Salesforce, pour mieux comprendre l'architecture de la plateforme et les bonnes pratiques du développement LWC.

Sur les recommandations de mon tuteur, j'ai également visionné plusieurs vidéos de la chaîne [CodeAesthetic](#) sur YouTube. Ces ressources m'ont permis de repenser ma manière d'écrire du code, notamment en mettant l'accent sur la lisibilité et la clarté. Par exemple, j'ai appris qu'il était souvent inutile, voire contre-productif, de commenter son code : un code bien écrit doit se comprendre sans avoir besoin d'explications supplémentaires. J'ai aussi découvert des pratiques de structuration plus efficaces, comme éviter les imbrications excessives (limitées à trois niveaux maximum), ou encore inverser les conditions pour réduire l'usage des blocs « if...else » et ainsi rendre le code plus linéaire et facile à suivre.

Enfin, lorsque je me suis retrouvée confrontée à un problème d'affichage CSS lors de la création du composant Timeline, mon tuteur m'a proposé une mini-formation. J'avais en effet du mal à positionner une pointe de flèche au bout de la ligne du temps pour indiquer son orientation. J'avais essayé plusieurs solutions, comme l'utilisation des pseudo-éléments `::before` et `::after` en CSS, ou encore l'intégration d'un SVG, mais aucune ne fonctionnaient comme je l'espérais.

Face à cette difficulté, j'ai demandé de l'aide à mon tuteur, qui m'a alors recommandé une formation ludique appelée [Flexbox Froggy](#). Ce jeu pédagogique apprend à maîtriser la flexbox en CSS en plaçant des grenouilles sur des nénuphars, à travers différents niveaux de difficulté. Contre toute attente, ce petit jeu m'a vraiment aidée à comprendre le fonctionnement de la flexbox, et m'a permis de trouver une solution efficace à mon problème.

Grâce à cette formation, j'ai pu repositionner la pointe de la flèche exactement comme je le souhaitais, en utilisant la flexbox pour gérer l'agencement des éléments du composant Timeline.

Compétences développées

Ce stage m'a permis de développer plusieurs compétences clés, aussi bien techniques que transversales :

- ◆ Développement front-end en HTML, CSS, JavaScript / TypeScript

- ◆ Utilisation du framework LWC dans un contexte professionnel
- ◆ Utilisation des outils Git et Visual Studio Code pour la gestion de versions
- ◆ Gestion de données paramétriques en JSON
- ◆ Travail en autonomie
- ◆ Prise d'initiatives

Toutes ces compétences m'ont permis de gagner en rigueur, en efficacité et en confiance dans mes capacités à m'adapter à un environnement de travail exigeant et structuré. Ce stage a représenté une véritable mise en situation professionnelle, en m'offrant l'opportunité d'appliquer mes connaissances tout en découvrant les pratiques du développement en entreprise.

Bilan technique

Réalisations

J'ai pu livrer des composants fonctionnels, documentés, testés et prêts à être intégrés dans des projets concrets de l'entreprise. Chaque composant respectait les standards du framework LWC ainsi que les consignes techniques et méthodologiques définies par mon tuteur. Mes réalisations ont été examinées, testées et validées par ce dernier. Certains de ces composants pourront d'ailleurs être utilisés dans des projets clients ou internes à l'entreprise.

Éléments en cours ou non terminés

Bien que tous les composants que j'ai réalisés soient fonctionnels, je n'ai pas eu le temps de rendre la tree grid entièrement performante, avec l'ensemble des fonctionnalités qui m'avaient été confiées. Mon tuteur souhaitait notamment qu'un effet 'sticky' soit appliqué sur mobile : les lignes ouvertes devaient rester visibles en haut de l'écran lors d'un défilement vers le bas, jusqu'à ce que la dernière ligne enfant sorte à son tour de l'écran. Or, la propriété CSS 'position: sticky;' n'est pas prise en charge sur l'application mobile. Mon tuteur m'a donc proposé de reproduire ce comportement manuellement, en détectant la sortie des éléments de l'écran. Faute de temps, je n'ai pas pu finaliser cette fonctionnalité.

Il avait également été envisagé que je travaille davantage sur Figma, dans l'éventualité où j'aurais terminé tous mes composants avant la fin du stage. Cela n'a finalement pas été le cas, mais si je les avais terminés, j'aurais réalisé des maquettes des pages d'interface Salesforce pour les objets les plus courants, comme les contacts ou les comptes. L'objectif aurait été de visualiser les interfaces utilisées par les clients, afin d'y intégrer, si besoin, des composants personnalisés.

Conclusion

Ce stage chez Kiliogene a été une expérience extrêmement enrichissante, tant sur le plan technique que sur le plan humain. Il m'a permis de confronter mes acquis scolaires aux attentes du monde professionnel et de confirmer mon intérêt pour le développement informatique. J'ai particulièrement apprécié la confiance qui m'a été accordée, l'ambiance de travail bienveillante, ainsi que l'autonomie dont j'ai pu faire preuve.

Je ressors de cette expérience avec des bases solides en développement web, une meilleure compréhension des environnements de travail collaboratifs, et une motivation renforcée pour poursuivre dans cette voie. Ce stage a joué un rôle important dans la validation de mon projet professionnel. Je suis reconnaissante à toute l'équipe de Kiliogene pour leur accueil et leur accompagnement tout au long de ces six semaines.

Annexes

← kgn_timeline ↗

* API Name ⓘ

kgn_timeline

Labels

["Création", "Validation", "Signature", "Publi"]

Dates

["2025-06-02", "2025-06-15", "2025-07-2"]

Orientation

Horizontal ▼

Vertical

✓ Horizontal

Vertical Reverse

Horizontal Reverse

← kgn_nullableCheckbox ↗

* API Name ⓘ

kgn_nullableCheckbox

Label

Checkbox label

Label Direction

Right ▼

✓ Right

Left

Annexe 1 – CPE des composants Timeline et Nullable Checkbox

```
force-app > main > default > lwc > kgn_timeline > __tests__ > JS kgn_timeline.test.js > describe('c-kgn-timeline') callback
1  import { createElement } from '@lwc/engine-dom';
2  import Kgn_timeline from 'c/kgn_timeline';
3
4  describe('c-kgn-timeline', () => {
5    afterEach(() => {
6      while (document.body.firstChild) {
7        document.body.removeChild(document.body.firstChild);
8      }
9    });
10
11    Run Test | Debug Test
12    it('affiche les 3 événements les plus récents par défaut', async () => {
13      const element = createElement('c-kgn-timeline', {
14        is: Kgn_timeline
15      });
16
17      element.labels = JSON.stringify(['A', 'B', 'C', 'D']);
18      element.dates = JSON.stringify(['2024-01-01', '2024-02-01', '2024-03-01', '2024-04-01']);
19
20      document.body.appendChild(element);
21      await Promise.resolve();
22
23      const items = element.shadowRoot.querySelectorAll('.timeline-item');
24      expect(items.length).toBe(3);
25
26      expect(items[0].textContent).toContain('D');
27    });
28  });
```

Annexe 2 – Exemple de test unitaire du composant Timeline

Search

Filter options...

Select Options

Available

Pomme
Banane
Cerise
Fraise
Kiwi

Selected

Selected values are: []

Annexe 3 – Composant Dual-Picklist

Search

z

Aucune option ne correspond à votre recherche.

Select Options

Available

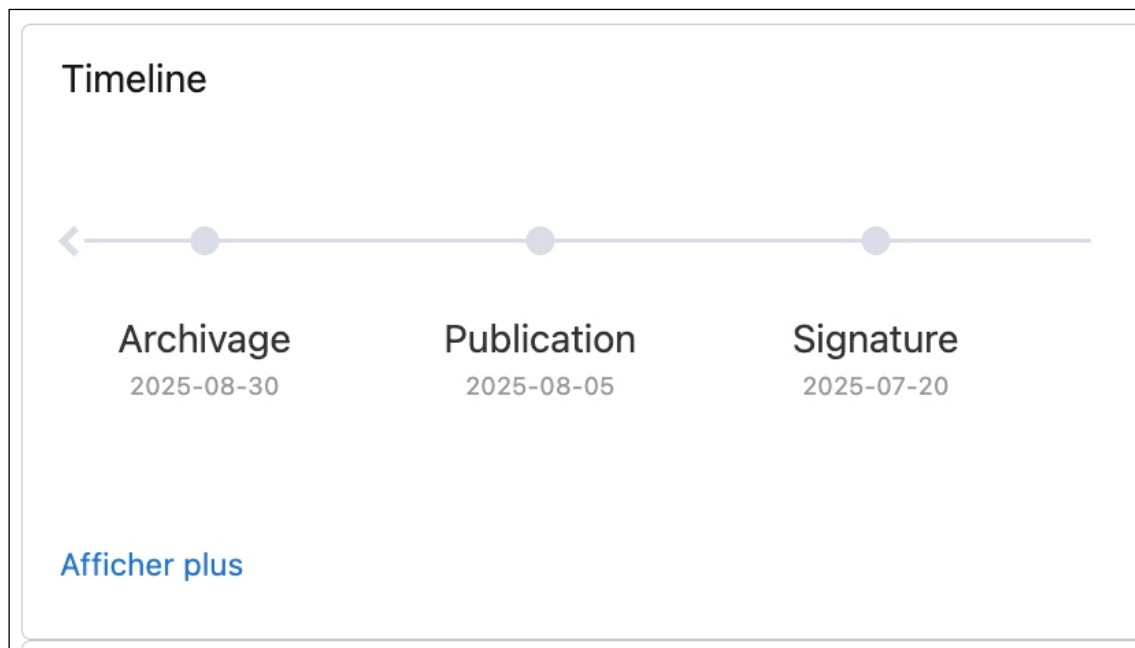
Selected

Selected values are: ["FR001","FR002"]

Annexe 4 – Message d'erreur du composant Dual-Picklist lorsqu'aucune correspondance n'est trouvée



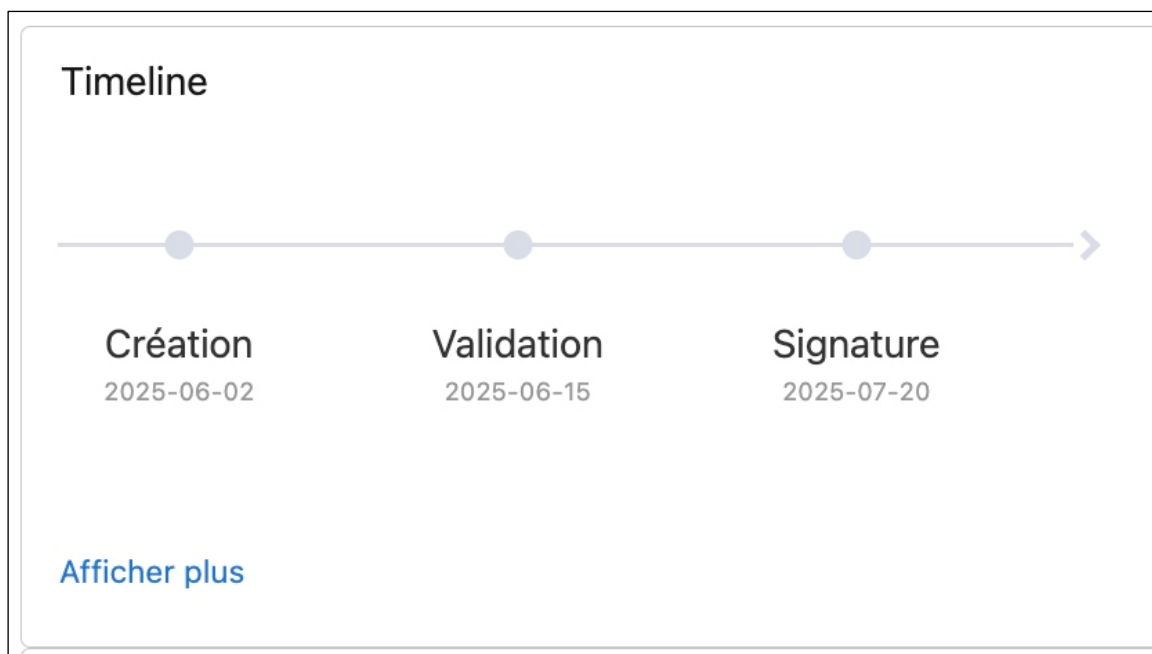
Annexe 5 – Composant Timeline en vue verticale



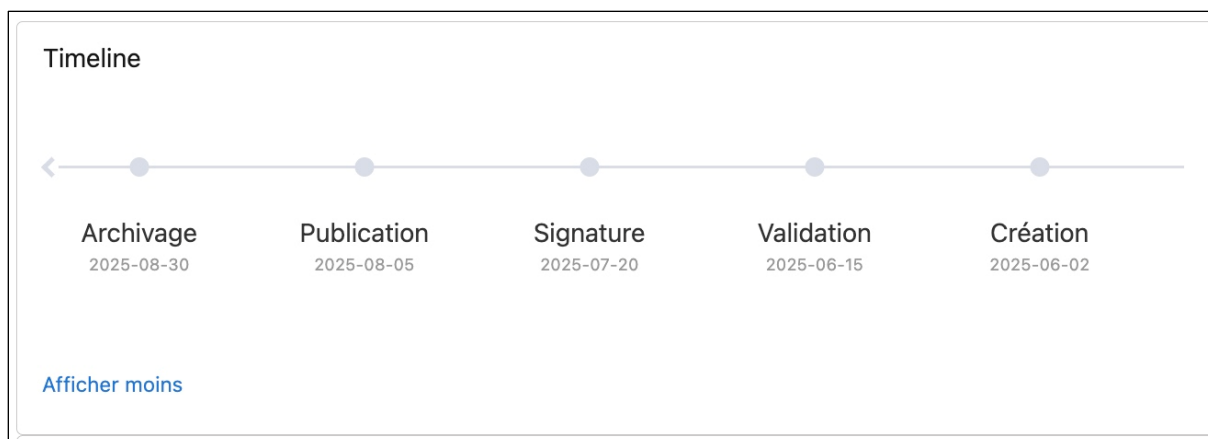
Annexe 6 – Composant Timeline en vue horizontale



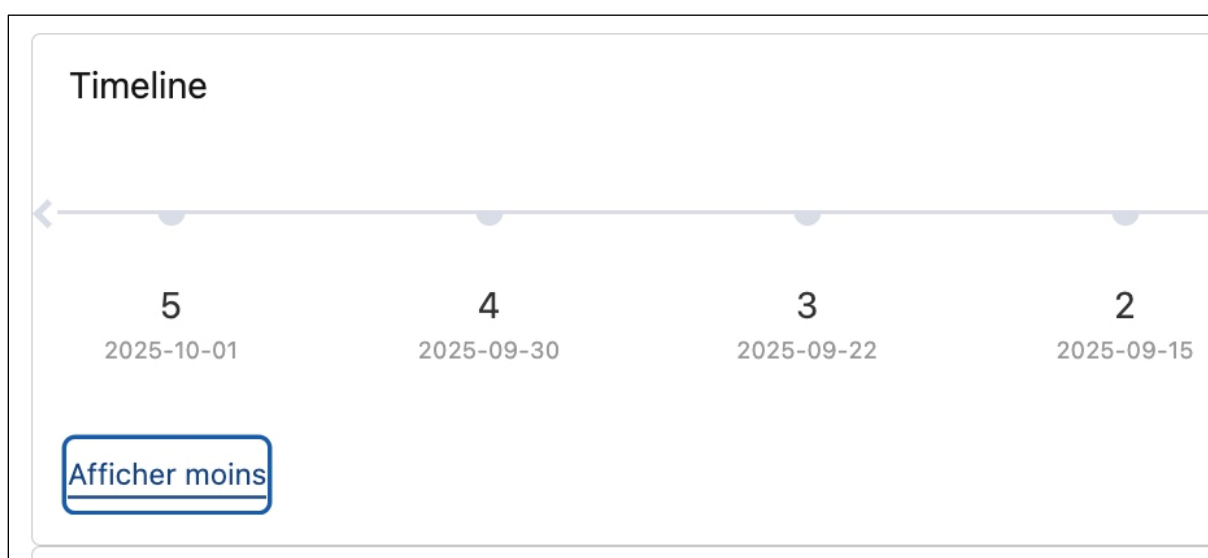
Annexe 7 – Composant Timeline en vue verticale inversée



Annexe 8 – Composant Timeline en vue horizontale inversée



Annexe 9 – Composant Timeline en vue horizontale lorsque tous les évènements sont affichés



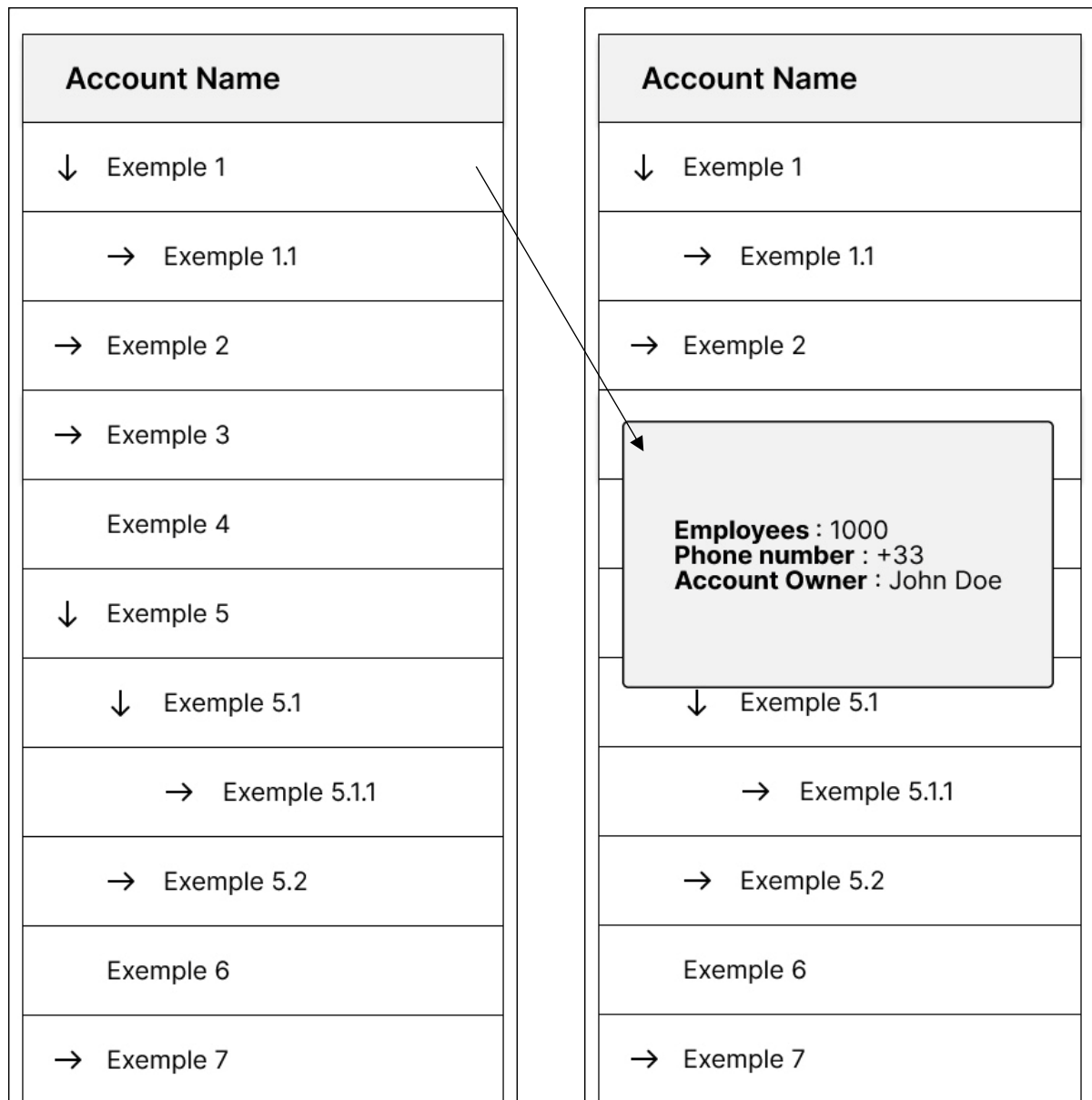
Annexe 10 – Composant Timeline avec les icones coupées en deux

<input type="checkbox"/>	Checkbox label
Checkbox label	<input checked="" type="checkbox"/>

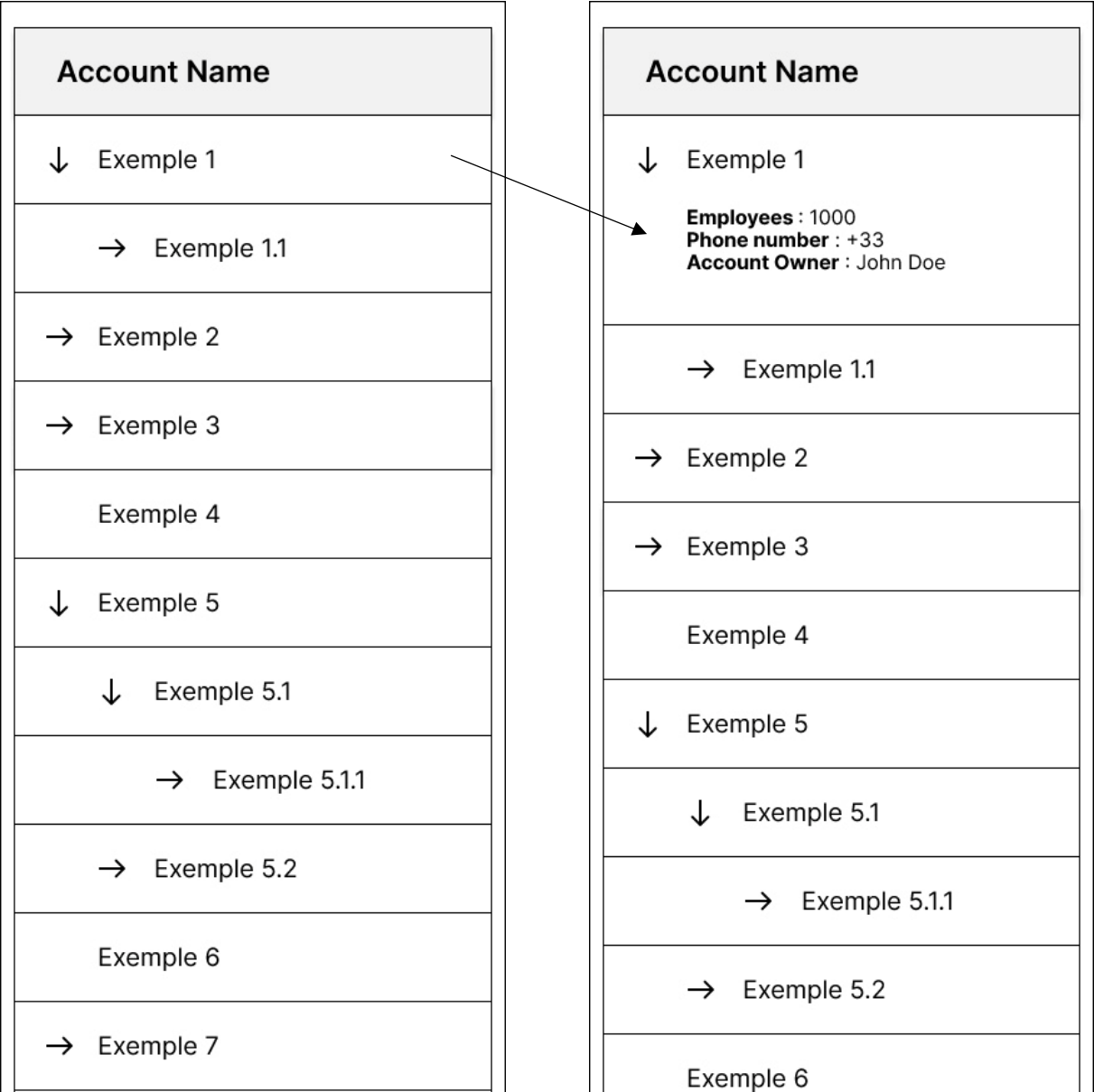
Annexe 11 – Composant Nullable Checkbox (2 versions)

Account Name ↓	Employees ↓	Phone Number ↓	Account Owner ↓
↓ Exemple 1	1000	+33	John Doe
→ Exemple 1.1	2000	+33	John Doe
→ Exemple 2	3000	+33	John Doe
→ Exemple 3	4000	+33	John Doe
Exemple 4	5000	+33	John Doe

Annexe 12 – Maquette Figma version ordinateur



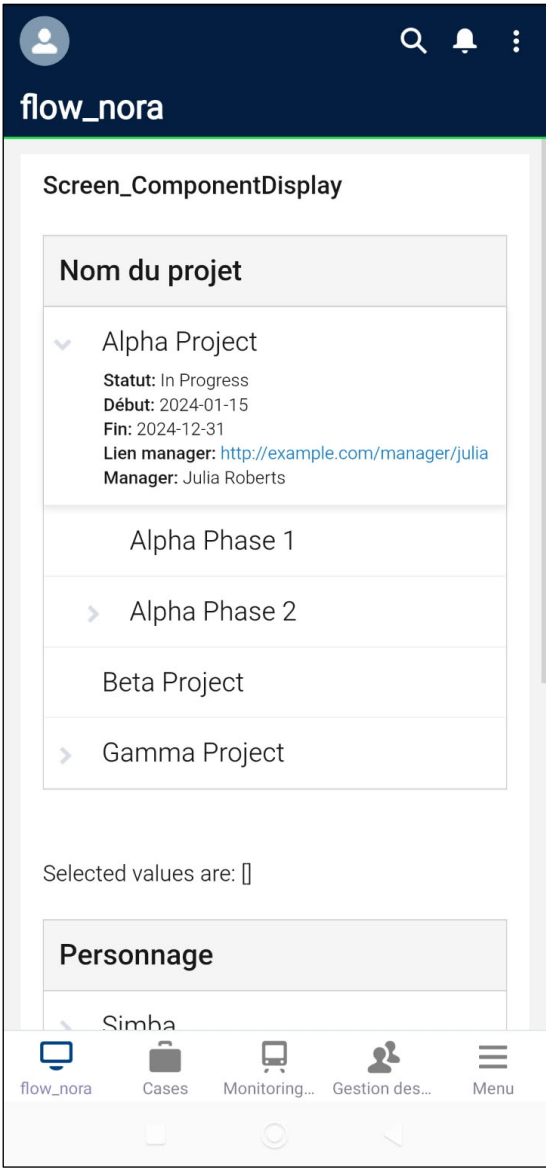
Annexe 13 – Maquettes Figma version mobile pour le test de la fenêtre modale



Annexe 14 – Maquettes Figma version mobile pour le test de la fonctionnalité « accordéon »

<input type="checkbox"/>	Nom du projet	Statut	Début	Fin	Lien manager	Manager
<input type="checkbox"/>	▼ Alpha Project	In Progress	2024-01-15	2024-12-31	http://example.com/manager/julia	Julia Roberts
<input type="checkbox"/>	Alpha Phase 1	Completed	2024-01-15	2024-06-30	http://example.com/manager/julia	Julia Roberts
<input type="checkbox"/>	➤ Alpha Phase 2	In Progress	2024-07-01	2024-12-31	http://example.com/manager/mark	Mark Twain
<input type="checkbox"/>	Beta Project	Planned	2024-03-01	2024-11-30	http://example.com/manager/emily	Emily Clark
<input type="checkbox"/>	➤ Gamma Project	Completed	2023-05-15	2024-01-15	http://example.com/manager/john	John Doe

Annexe 15 – Composant Tree Grid version ordinateur

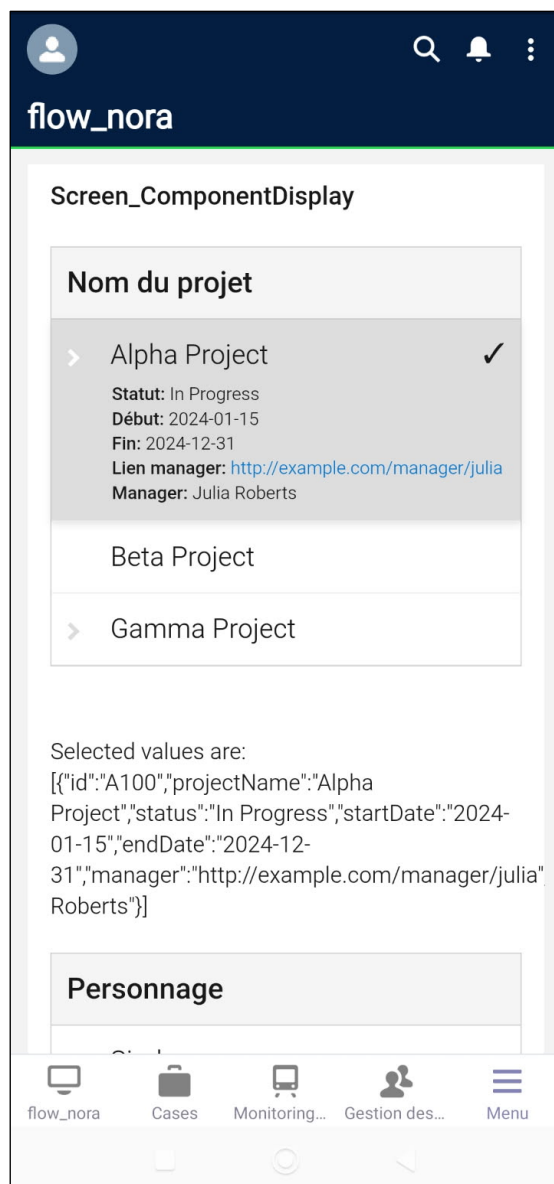


Annexe 16 – Composant Tree Grid version mobile

<input type="checkbox"/>	Nom du projet	Statut	Début	Fin	Lien manager	Manager
<input checked="" type="checkbox"/>	> Alpha Project	In Progress	2024-01-15	2024-12-31	http://example.com/manager/julia	Julia Roberts
<input type="checkbox"/>	Beta Project	Planned	2024-03-01	2024-11-30	http://example.com/manager/emily	Emily Clark
<input type="checkbox"/>	> Gamma Project	Completed	2023-05-15	2024-01-15	http://example.com/manager/john	John Doe

Selected values are: [{"id":"A100","projectName":"Alpha Project","status":"In Progress","startDate":"2024-01-15","endDate":"2024-12-31","manager":"http://example.com/manager/julia","managerName":"Julia Roberts"}]

Annexe 17 – Composant Tree Grid version ordinateur avec la fonctionnalité de sélection de ligne



Annexe 18 – Composant Tree Grid version mobile avec la fonctionnalité de sélection de lignes

<input checked="" type="checkbox"/>	Nom du projet	Statut	Début	Fin	Lien manager	Manager
<input checked="" type="checkbox"/>	▼ Alpha Project	In Progress	2024-01-15	2024-12-31	http://example.com/manager/julia	Julia Roberts
<input checked="" type="checkbox"/>	Alpha Phase 1	Completed	2024-01-15	2024-06-30	http://example.com/manager/julia	Julia Roberts
<input checked="" type="checkbox"/>	▼ Alpha Phase 2	In Progress	2024-07-01	2024-12-31	http://example.com/manager/mark	Mark Twain
<input checked="" type="checkbox"/>	Alpha Subphase 2A	Planned	2024-09-01	2024-10-31	http://example.com/manager/mark	Mark Twain
<input checked="" type="checkbox"/>	Beta Project	Planned	2024-03-01	2024-11-30	http://example.com/manager/emily	Emily Clark
<input checked="" type="checkbox"/>	▼ Gamma Project	Completed	2023-05-15	2024-01-15	http://example.com/manager/john	John Doe
<input checked="" type="checkbox"/>	Gamma Phase 1	Completed	2023-05-15	2023-09-30	http://example.com/manager/john	John Doe
<input checked="" type="checkbox"/>	Gamma Phase 2	Completed	2023-10-01	2024-01-15	http://example.com/manager/anna	Anna Smith

Selected values are: [{"id":"A100","projectName":"Alpha Project","status":"In Progress","startDate":"2024-01-15","endDate":"2024-12-31","manager":"http://example.com/manager/julia","managerName":"Julia Roberts"}, {"id":"A100-1","projectName":"Alpha Phase 1","status":"Completed","startDate":"2024-01-15","endDate":"2024-06-30","manager":"http://example.com/manager/julia","managerName":"Julia Roberts"}, {"id":"A100-2","projectName":"Alpha Phase 2","status":"In Progress","startDate":"2024-07-01","endDate":"2024-12-31","manager":"http://example.com/manager/mark","managerName":"Mark Twain"}, {"id":"A100-2-A","projectName":"Alpha Subphase 2A","status":"Planned","startDate":"2024-09-01","endDate":"2024-10-31","manager":"http://example.com/manager/mark","managerName":"Mark Twain"}, {"id":"B200","projectName":"Beta Project","status":"Planned","startDate":"2024-03-01","endDate":"2024-11-30","manager":"http://example.com/manager/emily","managerName":"Emily Clark"}, {"id":"C300","projectName":"Gamma Project","status":"Completed","startDate":"2023-05-15","endDate":"2024-01-15","manager":"http://example.com/manager/john","managerName":"John Doe"}, {"id":"C300-1","projectName":"Gamma Phase 1","status":"Completed","startDate":"2023-05-15","endDate":"2023-09-30","manager":"http://example.com/manager/john","managerName":"John Doe"}, {"id":"C300-2","projectName":"Gamma Phase 2","status":"Completed","startDate":"2023-10-01","endDate":"2024-01-15","manager":"http://example.com/manager/anna","managerName":"Anna Smith"}]

Annexe 19 – Composant Tree Grid version ordinateur avec la fonctionnalité de sélection total