

DÉPARTEMENT INFORMATIQUE – IUT 2 GRENOBLE

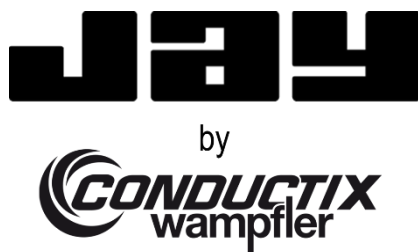


Année Universitaire 2024-2025

MÉMOIRE DE STAGE

CONCEPTION ET DÉVELOPPEMENT D'UN OUTIL GRAPHIQUE DE PERSONNALISATION DE PLASTRON

Jay Electronique



Présenté par

Léa Garaix

Jury

IUT : M. Bertolino

IUT : M. Corset

Société : M. Fournier

Déclaration de respect des droits d'auteurs

Par la présente, je déclare être le seul auteur de ce rapport et assure qu'aucune autre ressource que celles indiquées n'ont été utilisées pour la réalisation de ce travail. Tout emprunt (citation ou référence) littéral ou non à des documents publiés ou inédits est référencé comme tel et tout usage à un outil doté d'IA a été mentionné et sera de ma responsabilité.

Je suis informé qu'en cas de flagrant délit de fraude, les sanctions prévues dans le règlement des études en cas de fraude aux examens par application du décret 92-657 du 13 juillet 1992 peuvent s'appliquer. Elles seront décidées par la commission disciplinaire de l'UGA.

À Grenoble

Le 10 juin 2025



Remerciements

Je tiens à remercier Christian Fournier, mon tuteur, pour son accueil au sein de l'entreprise et son enthousiasme dans le développement du projet.

Je remercie également Valentin Saugnier, développeur, pour les échanges et les conseils.

Merci à Élodie Noël, responsable des ressources humaines, pour les indications sur la vie en entreprise.

Merci à toute l'équipe de la Recherche et Développement, du Support Technique Client, du service Communication et de la Production pour le temps que vous avez consacré à m'expliquer votre travail.

Je remercie Pascal Bertolino pour sa relecture du présent rapport et ses conseils pour la soutenance.

Enfin, merci à mes collègues stagiaires, Péran Amberny, Lucas Fossé et Jade Messaoud pour les discussions, les avis, les conseils, les repas et les rires.

Table des matières

INTRODUCTION.....	6
I. JAY ELECTRONIQUE : RADIOCOMMANDES DE SECURITE	7
I. 1. JAY ELECTRONIQUE, UNE FILIALE DE CONDUCTIX-WAMPFLER	7
Un métier dédié à la sécurité et à la qualité.....	7
Le service Recherche et Développement au sein de Jay Electronique	8
I. 2. PRESENTATION DE LA MISSION ET GESTION DU PROJET	8
Cadrage du projet	8
Objectifs	8
Contraintes techniques	9
Utilisateurs cibles	9
II. ÉTUDE DE L'EXISTANT ET ANALYSE DES BESOINS	10
II. 1. L'EXISTANT CHEZ JAY ELECTRONIQUE : IDIALOG, PARAMETRAGE RADIO	10
II. 2. LE PROCESSUS DE DEFINITION D'UN PLASTRON	11
À quoi ressemble une radiocommande et un plastron ?.....	11
Processus actuel.....	11
Réflexion sur le nouveau processus	12
II. 3. BESOINS FONCTIONNELS	13
Côté client : identification et priorisation des cas d'utilisation	13
Côté administrateur : identification et priorisation des cas d'utilisation.....	14
III. CONCEPTION ET REALISATION	15
III. 1. FAISABILITE DU PROJET	15
III. 2. CONCEPTION DES INTERFACES.....	15
Sélection et application des critères ergonomiques.....	15
III. 3. CHOIX TECHNIQUES	17
Choix des technologies de développement	17
Pratiques de développement.....	18
III. 4. BILAN SUR LES AVANCEES DU PROJET	18
Évolution du projet.....	18
Points techniques et difficultés rencontrées	19
CONCLUSION	20
GLOSSAIRE	21
BIBLIOGRAPHIE-SITOGRAFIE.....	23
ANNEXES	1
Annexe 1 : Organigramme du service Recherche et Développement.....	2
Annexe 2 : Planning du projet.....	3
Annexe 3 : PDF à compléter par le client dans le processus actuel	4
Annexe 4 : Maquettes.....	5

Table des figures

Figure 1. Les solutions de radiocommandes Jay.	7
Figure 2. Exemple de plastron réalisé pour un client, compatible avec la graveuse laser.	9
Figure 3. Interface iDialog pour la configuration du module opérateur.	10
Figure 4. Exemple d'une radiocommande Moka.	11
Figure 5. Processus de commande et définition d'un plastron.	12
Figure 6. Nouveau processus de commande et définition d'un plastron.	12
Figure 7. Tableau de priorisation des cas d'utilisation client.	14
Figure 8. Tableau de priorisation des cas d'utilisation administrateur.	14
Figure 9. Maquette de la page de modification d'un plastron (vue client).	15
Figure 10. Interface de la page de modification d'un plastron.	16

Introduction

Jay Electronique est une entreprise spécialisée dans la fabrication de radiocommandes de sécurité. Ces radiocommandes peuvent servir à piloter une grande variété de machines, et assurent la sécurité des humains et du matériel. Leur avantage concurrentiel tient sur les certifications de leur système d'arrêt d'urgence, qui garantit l'arrêt des machines à distance quoi qu'il advienne. Leur service Recherche et Développement (R&D) conçoit de nouvelles solutions pour les clients. Il développe également certains des logiciels utilisés dans le secteur Production.

Jay Electronique propose une gamme de radiocommandes personnalisables selon les besoins du client, nommée « Pika-Moka » : il choisit les organes (c'est-à-dire les éléments de commande physiques, comme des boutons, des joysticks, des potentiomètres) et leur emplacement, ainsi que les inscriptions à graver pour indiquer les fonctionnalités. Les échanges avec le client se font actuellement par mail, et la personnalisation sur un PDF. Le PDF est repris par les professionnels pour créer un fichier exploitable par le service de Production. Ce processus est long et source d'erreurs.

L'entreprise souhaite donc proposer à ses clients un site web sur lequel réaliser cette personnalisation. Le client pourra personnaliser l'interface physique de la radiocommande, appelée plastron. Sur le même site, le Support Technique Client (STC) pourra récupérer ce qu'a fait le client sous forme de fichier SVG, un format d'image s'appuyant sur le langage XML pour décrire des ensembles de graphiques vectoriels [1]. Ce format assure la qualité du rendu, et est lisible par la graveuse laser qui s'occupe des découpes et des gravures sur les plastrons*.

L'objectif de ce site est double. Il augmentera la rapidité du processus de création des radiocommandes en améliorant le protocole de communication, et il évitera les erreurs commises lors des transmissions des idées de personnalisation. Pour mener à bien ce projet, on m'a confié deux missions : prouver sa faisabilité, et développer un prototype à présenter à la hiérarchie.

Dans ce mémoire, vous trouverez un premier chapitre dédié à la présentation détaillée de Jay Electronique et de la mission qui m'a été confiée. Le second chapitre est consacré à l'analyse du contexte et des besoins de l'entreprise ; le troisième chapitre détaille la conception et la réalisation de l'application. En annexes, vous pourrez consulter l'organigramme du service R&D*, un exemple de PDF envoyé au client lors du processus actuel de configuration du plastron, et les maquettes du projet.

¹ Wikipédia. *Scalable Vector Graphics* [en ligne]. 15 avril 2025.

I. Jay Electronique : radiocommandes de sécurité

I.1. Jay Electronique, une filiale de Conductix-Wampfler

Fondée en 1962, par Aimé Jay, Jay Electronique est installée à Saint-Ismier, proche de Grenoble. Elle a été rachetée en juillet 2019 par Conductix-Wampfler [2], qui est elle-même une filiale du groupe Delachaux [3]. Elle est dirigée par Patrick Berthet (directeur) et elle emploie une soixantaine d'employés. L'entreprise est découpée en plusieurs secteurs : Production, Logistique, Service Après-Vente, Marketing, Support Technique, Finance, Commercial et R&D*.

Un métier dédié à la sécurité et à la qualité

Les radiocommandes Jay permettent d'assurer la sécurité lors de la mobilité des équipements industriels. Elles garantissent une liaison radio fiable, ce qui est un point crucial pour le fonctionnement immédiat de l'arrêt d'urgence. Plusieurs gammes existent, elles sont détaillées dans le synoptique *Radiocommandes Jay* [4] :



Figure 1. Les solutions de radiocommandes Jay.
Conductix-Wampfler, *Synoptique Radiocommandes Jay*, 2020, p.8-9.

Elles sont utilisées dans de nombreux domaines : l'intralogistique, l'automatisation et les véhicules dans l'industrie, le transport de personnes, la construction, ou encore la manutention. Différentes applications sont possibles : flottes d'équipements mobiles dans les entrepôts automatisés, arrêt d'urgence pour les remontées mécaniques, commande de pompes à béton et équipements de projection, et bien d'autres encore.

Jay met l'accent sur la qualité de ses processus et de ses produits, ainsi que sur la formation de son personnel. Elle a donc pu obtenir le certificat ISO 9001-2015 [5], la norme de management de la qualité. Jay a aussi des certifications de sécurité sur ses arrêts d'urgence : certificat SIL 2-PLd et SIL 3-PLe selon la gamme. Ces gammes existent aussi en version certifiée ATEX-IECEx (en gris sur l'image ci-dessus), ce qui signifie qu'elles sont utilisables dans des environnements explosifs. Ces certifications doivent être interprétées de la manière suivante :

- SIL* : *Safety Integrity Level*, se traduit par « Niveau d'intégrité de sécurité ». C'est une mesure du niveau de réduction du risque, de 1 (le plus faible) à 4 (le plus élevé) [6].

² Conductix-Wampfler. *Conductix-Wampfler acquiert Jay Electronique* [en ligne]. 4 juillet 2019.

³ Groupe Delachaux. *Nos Entreprises - Delachaux* [en ligne]. 2019.

⁴ Conductix-Wampfler. *Synoptique Radiocommandes Jay* [en ligne]. 2020.

⁵ ISO. *ISO 9001:2015 - Systèmes de management de la qualité — Exigences* [en ligne]. 2021.

⁶ ChatGPT-4o. Réponse à « Explique-moi ce que sont les normes SIL 2-PLd et SIL 3-PLe. », OpenAI, 5 juin 2025.

- PL* : *Performance Level*, se traduit par « Niveau de performance ». C'est une estimation de la capacité d'une fonction de sécurité à fonctionner correctement, de a (le plus faible) à e (le plus élevé). Elle se base sur la durée de vie des composants, la capacité de diagnostic et la résistance aux défaillances communes [7].
- ATEX-IECEX* : ATEX pour ATmosphère EXplosibles, IECEX pour *International Electrotechnical Commission System for Certification to Standards relating for use in Explosive Atmospheres*, ce qui se traduit par « Système de certification de la Commission Électrotechnique Internationale pour les normes relatives aux équipements destinés à être utilisés en atmosphères explosibles ». L'ATEX est spécifique à l'Union européenne, tandis que l'IECEX est un système de certification international. Ces deux normes garantissent que l'équipement n'allumera ni gaz ni poussières combustibles [7].

Le service Recherche et Développement au sein de Jay Electronique

La R&D*, au sein de laquelle j'ai effectué mon stage, est sous la responsabilité de Laurent Damon et compte treize salariés dont les rôles sont variés : développement de solutions radio, développement logiciel, développement logiciel embarqué, homologation et certifications, conception mécanique, industrialisation et méthodes, développement et industrialisation électronique, gestion des données techniques, et administration système et réseau. Mon tuteur de stage est Christian Fournier, chargé du développement logiciel. Vous pouvez consulter l'organigramme de la R&D* en [annexe 1](#).

Ce service se charge de créer, modifier et tester les produits de Jay. De plus, les besoins des clients dépassent parfois le niveau des solutions présentes sur le marché. L'équipe de la R&D* a les capacités de développer de nouveaux produits correspondant à leurs besoins. La R&D* vient également en appui au service de Production. En effet, les outils de Jay Electronique sont des logiciels maison pour la plupart, développés et maintenus par la R&D*.

I. 2. Présentation de la mission et gestion du projet

Cadrage du projet

Le projet est commandité par le service R&D*, à destination du Support Technique Client (STC). Christian Fournier (développeur logiciel) le supervise, et Laurent Damon (responsable R&D*) le valide. Le développement du prototype est réalisé par Léa Garaix (stagiaire). Le développement de l'application finale incluant le tableau de bord, la connexion et la gestion de la base de données sera réalisé par Valentin Saugnier (développeur logiciel).

Objectifs

L'objectif final de l'application est de permettre au STC* d'exporter un fichier SVG* compatible avec la graveuse laser utilisée pour graver les plastrons*. À terme, Jay souhaite ne plus avoir besoin d'utiliser CorelDRAW, un logiciel de dessin vectoriel édité par Corel.

Le projet s'étale sur douze semaines (du 14 avril au 4 juillet 2025). Le planning du projet est détaillé en [annexe 2](#).

⁷ ChatGPT-4o. Réponse à « Explique-moi ce qu'est la norme ATEX-IECEX. », OpenAI, 5 juin 2025.

Contraintes techniques

Le fichier SVG* produit en sortie de l'application doit être lisible par la graveuse laser avec laquelle les plastrons* sont découpés et gravés. Il y a notamment un code couleur à respecter : rouge pour la découpe, noir pour la gravure. Le bleu indique les contours, et le vert les zones dans lesquelles il ne faut pas placer d'éléments, car ils seraient cachés de la vue des utilisateurs de la radiocommande. Le rose est utilisé pour signaler une découpe ayant déjà été réalisée au préalable sur le plastron*, et qui n'a donc pas besoin d'être répétée.

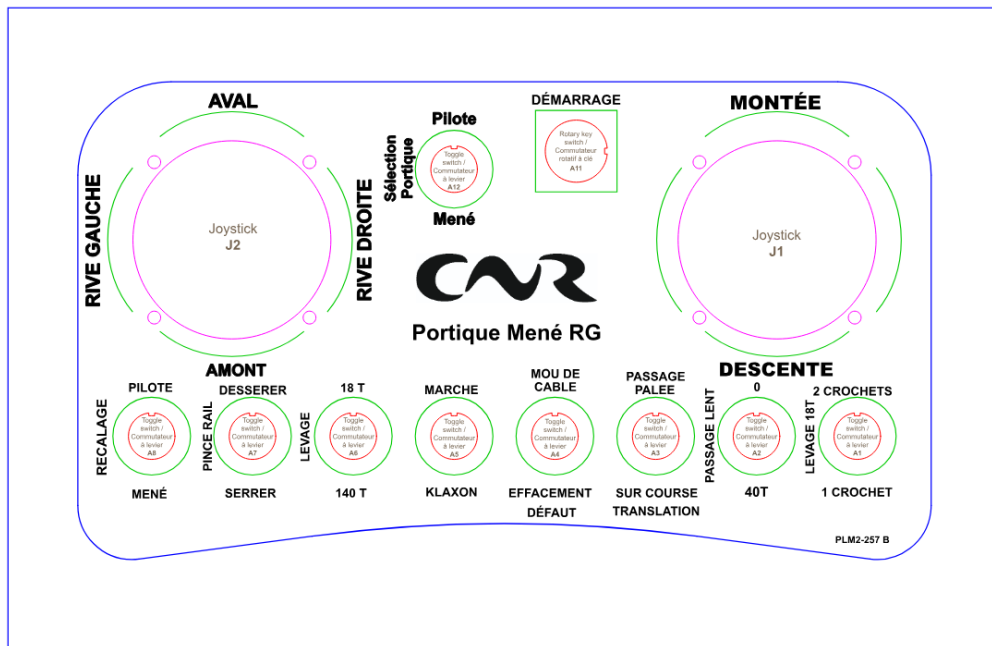


Figure 2. Exemple de plastron réalisé pour un client, compatible avec la graveuse laser.
Jay Electronique, *PLM2-257_B* CNR, 2020.

Les organes* ont des contraintes de placement : les cartes sous le plastron* prennent de la place. Certains organes* ne peuvent être placés qu'à certains endroits. Les clients ont tout de même la possibilité de commander une radiocommande sur laquelle le placement des boutons est défini manuellement, mais cette possibilité n'est pas prise en compte pour le prototype. Ce sera toutefois une fonctionnalité dans l'application finale.

Utilisateurs cibles

Deux utilisateurs cibles ont été identifiés : les opérateurs du STC*, qui auront un rôle administrateur, et les clients. Les opérateurs travaillent tous les jours avec CorelDRAW* et Excel. Ils utilisent le configurateur de plastron* en ligne. Ils sont familiers des outils graphiques, et ont donc des réflexes propres à ces outils. Par exemple, certaines combinaisons de touches déclenchent certaines actions : "entrée" valide une saisie, les flèches déplacent les éléments, Ctrl + C copie, Ctrl + V colle, etc. Il faut tenir compte de cette habitude dans les interactions.

Les clients de Jay ont des niveaux plus variés : certains n'ont jamais utilisé d'outils graphiques, d'autres si. Pour tenir compte de cette différence, les actions de l'application doivent être explicites. Nous devons aussi tenir compte d'handicaps visuels tel que le daltonisme, qui sont courants dans la population et pourrait rendre difficile l'utilisation de l'application selon le choix des couleurs.

II. Étude de l'existant et analyse des besoins

Après avoir défini le contexte, la deuxième étape d'un projet est de comprendre les besoins des utilisateurs. Pour commencer, il fallait comprendre le processus actuel de définition des plastrons* suivi par le Support Technique Client (STC)*. Il devenait alors possible de réfléchir à l'intégration de l'application dans ce processus. Quant à la liste des besoins, elle s'est étoffée au fur et à mesure de l'avancement du projet. Ce travail sera utilisé pour le développement de l'application finale, qui a déjà été commencé par Valentin Saugnier (développeur).

II. 1. L'existant chez Jay Electronique : iDialog, paramétrage radio

IDialog est une des applications développées par Jay Electronique. Elle permet notamment aux développeurs embarqués de programmer les cartes des organes*. Cette application est aussi utilisée par le service Production, qui s'appuie sur la section « Module opérateur », dans lequel sont configurés les détails techniques des organes*. L'interface se présente ainsi :

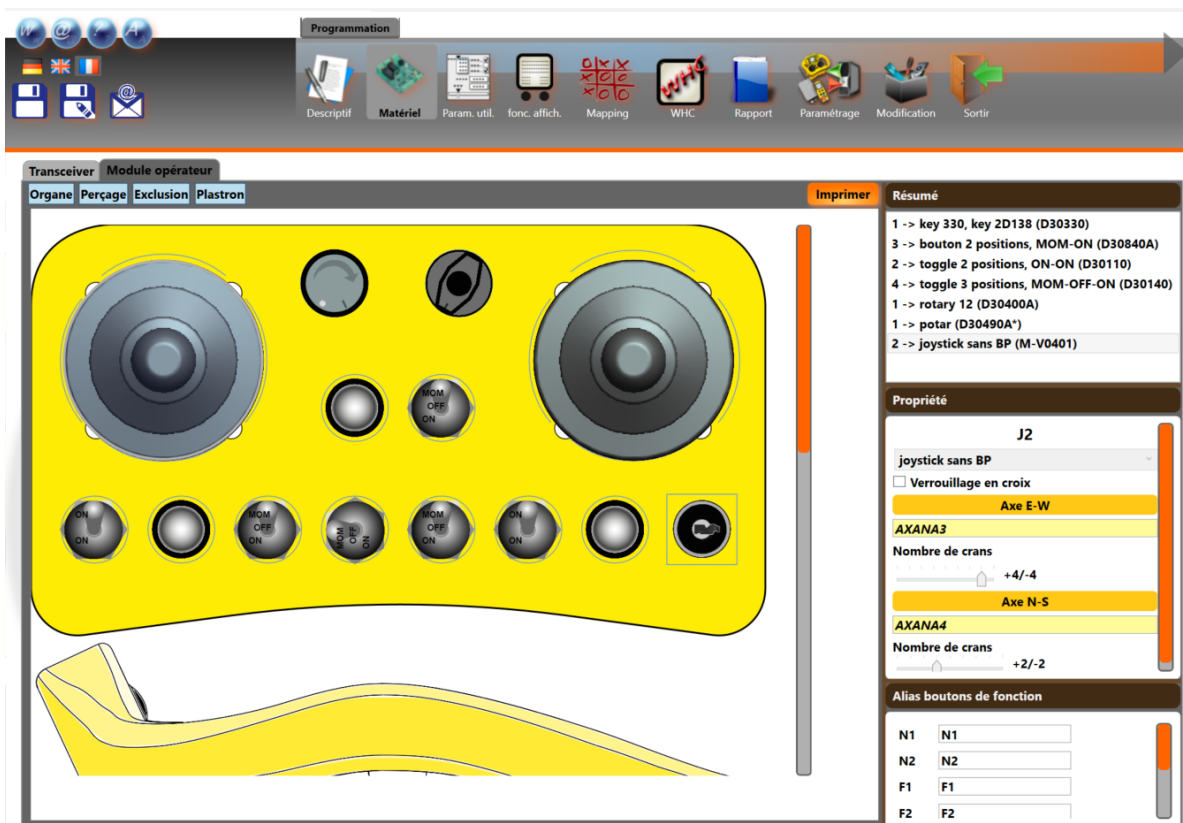


Figure 3. Interface iDialog pour la configuration du module opérateur.
Jay Electronique, *iDialog Version 5.4.0.0 - Module opérateur*, 2025.

En cliquant sur une position, il est possible de choisir un organe* dans le menu à droite, et de paramétrer plusieurs informations (différente selon l'organe* sélectionné) : orientation, nombre de crans ou nombre de positions, et verrouillage. L'interface de la nouvelle application s'inspire grandement de celle-ci.

II. 2. Le processus de définition d'un plastron

À quoi ressemble une radiocommande et un plastron ?

Une fois gravée et assemblée, une radiocommande Pika-Moka* ressemble à ceci :



Figure 4. Exemple d'une radiocommande Moka.
Jay Electronique, PLM3-022 B pour Huwer, 2025.

Pika et Moka désigne deux tailles différentes de radiocommandes, les Pika étant plus petites. En l'occurrence, il s'agit d'une Moka3 puisqu'elle a trois joysticks. Il existe plusieurs types d'organes*. Sur cette radiocommande, on trouve un rotary, qui permet une graduation de vitesse ou de puissance

La référence « PLM3-022 B », en bas à gauche, identifie précisément la radiocommande : PL pour plastron*, M pour Moka, 3 pour trois joysticks, 022 est le numéro identifiant la radiocommande parmi toutes les PLM3, et B indique la version.

Processus actuel

Pour modéliser le processus actuel de commande, j'ai utilisé un *Business Process Model and Notation* (BPMN). C'est un diagramme qui permet de représenter les processus métiers, et de répertorier toutes les activités prenant part à la réalisation d'une tâche. Voici donc le processus actuel :

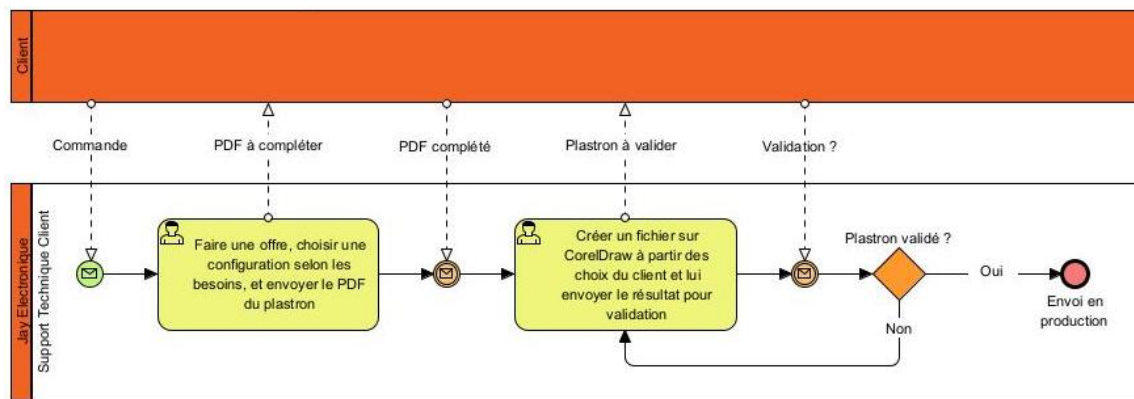


Figure 5. Processus de commande et définition d'un plastron.
Léa Garaix, 2025.

Lorsque le client contacte l'entreprise pour préparer sa commande, le Support Technique Client (STC)* lui propose les radiocommandes les plus adaptées à ses besoins. Si l'offre de Jay comporte des radiocommandes de la gamme Pika-Moka*, l'opérateur décide de la configuration la plus adaptée. Il y a un grand nombre de configurations possibles : elles dépendent du nombre de joysticks et du nombre d'autres organes* nécessaires.

Ensuite, l'opérateur envoie l'offre au client. Il y joint des PDF à compléter avec les spécifications de la radiocommande. Vous pouvez consulter un exemple en [annexe 3](#). Le client peut imprimer le PDF, dessiner, et renvoyer le document scanné, ou bien utilisé un logiciel. Quand le client renvoie le PDF, l'opérateur s'en sert pour créer un fichier lisible par la graveuse laser avec le logiciel CorelDRAW*. Il contacte ensuite le client pour confirmer que le nouveau fichier ne comporte pas d'erreurs. S'il y a des erreurs, l'opérateur doit encore modifier le fichier et demander une validation. Quand le client valide la version, il doit signer un accord. Le fichier peut alors être envoyé vers la production, où le plastron* sera gravé et monté sur la radiocommande.

Ce processus est long et source d'erreurs. Tous les clients ne sont pas français, le report des textes sur le fichier peut donner lieu à des fautes d'orthographe. La nouvelle application doit donc permettre de faciliter le travail et éviter les erreurs.

Réflexion sur le nouveau processus

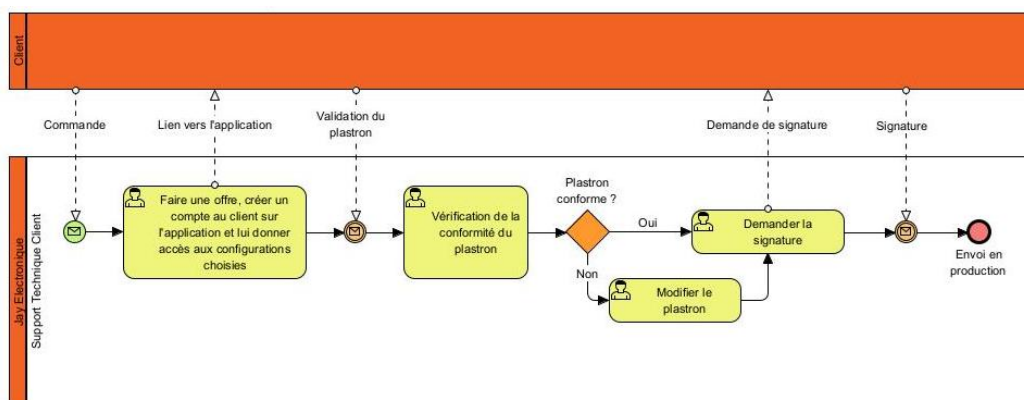


Figure 6. Nouveau processus de commande et définition d'un plastron.
Léa Garaix, 2025.

Dorénavant, lorsque le client reçoit l'offre de Jay Electronique, il lui est demandé de se connecter sur le site. Il s'agira d'un mail automatique, envoyé au client lors de la création de son compte par le STC* : il contiendra un mot de passe aléatoire, à modifier lors de la première connexion.

Le client peut alors modifier le ou les modèles de plastron* contenu dans l'offre. Il peut annuler une demande, et valider les autres. Lorsqu'il a fini, il peut « Demander la validation de la commande ». Le STC* sont prévenus que le client a terminé, et peuvent aller vérifier si tout est conforme. Si c'est le cas, et que les clients ont signé la validation, le plastron* peut partir en production.

II. 3. Besoins fonctionnels

Côté client : identification et priorisation des cas d'utilisation

Ce tableau présente tous les cas d'utilisation de l'application qui ont été identifiés, classés par priorité. On s'intéresse ici aux besoins des clients.

Priorité	Cas d'utilisation	Justification
1	Choisir un organe* pour une position	Le choix des organes* détermine les découpes à faire dans le plastron*. Choisir la position d'un organe* était l'une des étapes les moins évidentes à faire avec le PDF, car le client ne se représentait pas le résultat final.
	Importer une image	Les clients doivent pouvoir indiquer les fonctionnalités des organes* à l'aide d'icônes. Ils peuvent aussi vouloir ajouter leur logo.
	Modifier, déplacer, supprimer, et redimensionner un texte ou une image	Nous devons permettre au client de personnaliser son plastron* au maximum.
2	Choisir une image dans la bibliothèque d'images	Certaines images sont utilisées régulièrement dans les plastrons*, comme les flèches ou la graduation d'un bouton rotatif. Le but est d'éviter que le client perde du temps à chercher des icônes.
	Dissimuler les positions non-utilisées	Le client visualisera mieux l'espace qu'il lui reste sur la radiocommande après avoir choisi des organes*, afin de positionner ses textes et icônes.
	Visualiser le rendu final	Aider le client à comprendre à quoi ressemblera sa radiocommande.

	Annuler ou refaire une action	Permettre au client de tester plusieurs dispositions en ayant en moyen facile de revenir à la précédente.
	Zoomer et dézoomer	Permettre au client de voir de plus près les modifications qu'il fait, afin d'être plus précis, mais également de prendre du recul.
3	Changer la langue de l'application	De nombreux clients ne parlent pas français. Pour le prototype, ce n'est pas la priorité.
	Accéder à son espace entreprise	Le client garde une trace en ligne du travail effectué. Il peut voir les commandes précédentes et reprendre le modèle de l'un des plastrons*. Pour le prototype, ce n'est pas la priorité.
	Exporter le fichier en PDF	Le client garde une trace dans ses serveurs du travail effectué.

Figure 7. Tableau de priorisation des cas d'utilisation client.

Côté administrateur : identification et priorisation des cas d'utilisation

Les administrateurs auront besoin des mêmes fonctionnalités que les clients afin de corriger les plastrons, mais ils auront aussi besoin de fonctionnalités d'administration. Le prototype présenté ici développe peu de ces fonctionnalités, hormis celles utilisées pour tester le fonctionnement.

Priorité	Cas d'utilisation	Justification
1	Exporter le plastron* au format SVG*	Ce fichier sera envoyé à la graveuse laser. Il n'y aura plus besoin d'utiliser une application tierce comme CorelDRAW* pour générer un fichier compatible.
1	Créer un compte pour un client	Il faut pouvoir donner accès au client à tous les plastrons* contenus dans l'offre qui lui est faites.
	Consulter la liste des clients	Trouver un client pour consulter ses plastrons* en cours ou terminés.
	Consulter les commandes d'un client et leur état	Gérer les plastrons* d'un client.
2	Gérer l'attribution de la référence du plastron*	Le client ne doit pas pouvoir la modifier. Le STC* n'aura pas besoin d'utiliser une application tierce pour écrire la référence.

Figure 8. Tableau de priorisation des cas d'utilisation administrateur.

III. Conception et réalisation

III. 1. Faisabilité du projet

La faisabilité du projet a été prouvée rapidement : il est possible d'interagir avec du SVG* sur le web, tout en respectant les contraintes techniques vues en partie I.2. Pour confirmer cette faisabilité, j'ai imprimé un plastron* avec la graveuse laser, à partir d'un fichier SVG* généré par le prototype. À ce moment-là, le premier jet de l'application permettait d'importer une image, d'ajouter un texte et de le modifier, et de déplacer textes et images. Il était aussi possible de placer des organes* dans les zones prédéfinies. Le plastron* ainsi créé pouvait être téléchargé au format SVG* et utilisé par la graveuse laser.

Le premier plastron* imprimé respectait les contraintes. Pour vérifier encore cela, nous en avons imprimé un second avec des lignes permettant de mesurer précisément les dimensions.

III. 2. Conception des interfaces

Sélection et application des critères ergonomiques

Afin d'assurer que l'application soit agréable à utiliser et ne procure pas de frustration au client, nous avons sélectionné plusieurs critères ergonomiques à respecter. Ils sont issus des critères décrits par Bastien et Scapin [8]. Nous nous sommes concentrés sur quatre critères : le guidage, la gestion des erreurs, la signification des codes et dénominations, et l'homogénéité.

Pour y réfléchir, j'ai créé des maquettes à l'aide de l'outil en ligne Figma, un éditeur graphique. Vous pouvez consulter l'ensemble des maquettes du site en [annexe 4](#). Voici la maquette de la page de modification des plastrons* :

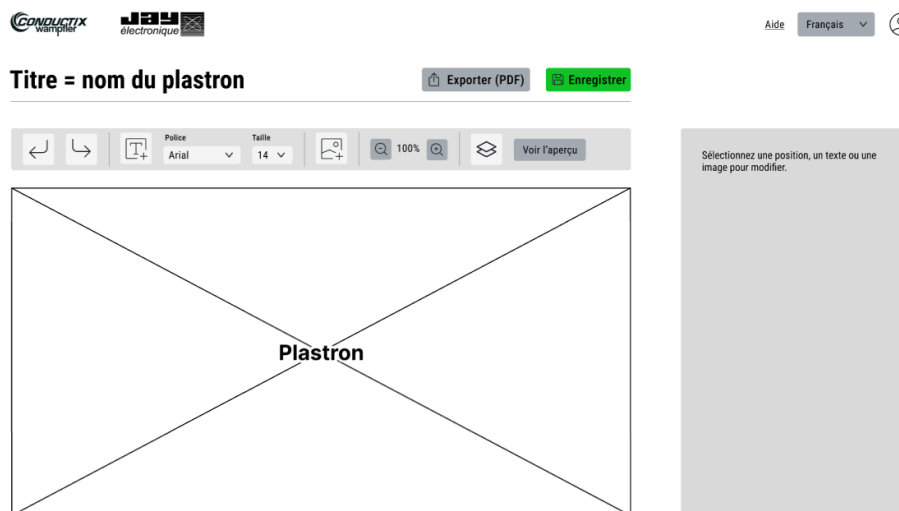


Figure 9. Maquette de la page de modification d'un plastron (vue client).
Léa Garaix, 2025.

⁸ BASTIEN, Christian et SCAPIN, Dominique. *Ergonomic criteria for the evaluation of human-computer interfaces* [en ligne]. Juin 1993.

Et une fois développé, cela ressemble à ça :

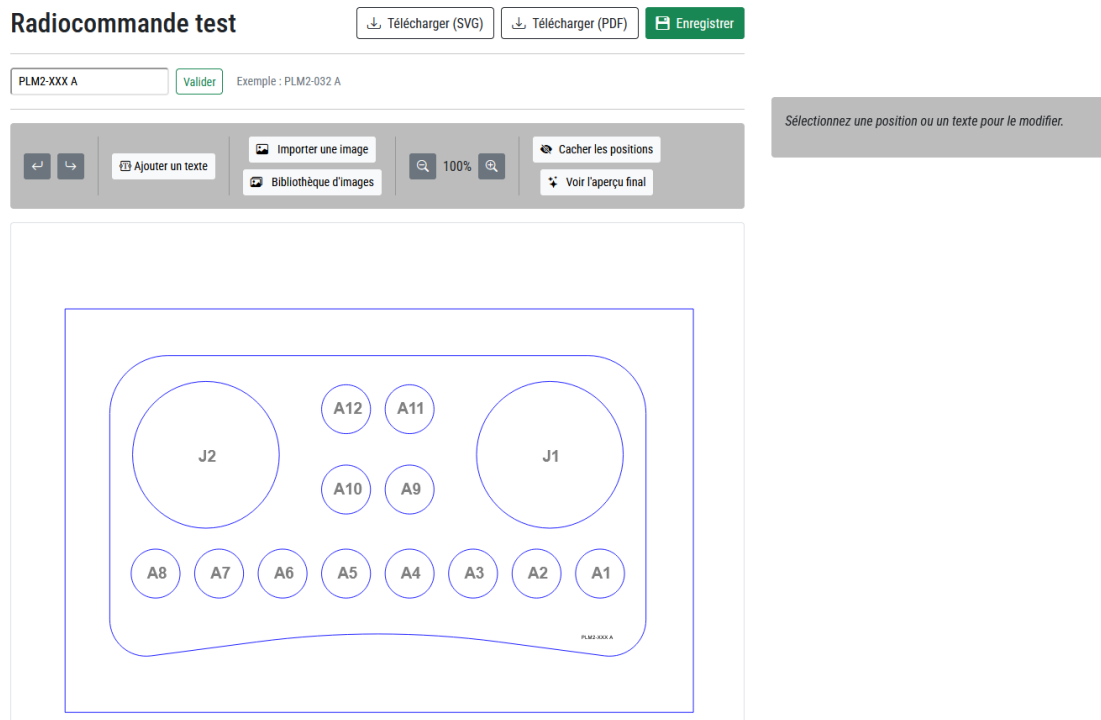


Figure 10. Interface de la page de modification d'un plastron.
Léa Garaix, 2025.

Le **guidage** consiste à accompagner l'utilisateur dans ses interactions avec l'application. Pour assurer cela, nous avons incité l'utilisateur à effectuer certaines actions en lui fournissant des indices : quand aucun organe*, texte ou image n'est sélectionné, le menu de droite affiche un message indiquant à l'utilisateur ce qu'il peut faire pour travailler. Ce n'est pas visible sur la figure 8, mais le curseur change de forme au survol d'une zone cliquable ou déplaçable, pour indiquer à l'utilisateur qu'il peut interagir avec l'objet. Lors de l'insertion d'un texte, le texte par défaut est « Cliquez deux fois pour modifier » afin de lui montrer comment interagir avec cet objet. En plus de l'avoir incité à faire ces actions, nous avons groupé les boutons d'export et d'enregistrement dans la même zone, car ils servent tous deux à effectuer une sauvegarde du travail. Enfin, toutes les modifications faisables sur un élément doivent être faites à partir du menu de modification sur la droite.

La **gestion des erreurs** vise à aider l'utilisateur à percevoir les erreurs. Nous souhaitons notamment l'en protéger. Nous avons donc proposé une liste prédéterminée d'organes* pour une position – ces contraintes doivent forcément être respectées pour la fabrication du plastron*. De plus, l'utilisateur sera averti s'il tente de quitter la page sans sauvegarder : cela reste à ajouter dans l'application.

La **signifiante des codes et des dénominations** se retrouve dans le sens des éléments de l'interface pour l'utilisateur : il faut éviter le vocabulaire technique et respecter les conventions du web. Pour que l'interface soit facilement compréhensible par l'utilisateur, nous avons évité d'utiliser des termes techniques. Les conventions du web ont été appliquées, dans les comportements, les placements et les couleurs : le profil est accessible en haut à droite, un clic

sur les logos ramène à la page d'accueil, le vert signifie « Enregistrer » (action positive) et le rouge « Supprimer » (action négative, dangereuse).

L'**homogénéité** d'une interface est réussite lorsque les choix de conception sont conservés : la localisation des composants est stable, et les choix de libellés, les alignements, les couleurs ou encore les messages sont cohérents entre eux. Dans notre application, elle est garantie par les présentations stables : les menus et les informations sont toujours à droite, et les actions des boutons sont illustrées par des icônes.

III. 3. Choix techniques

La partie modification du plastron* se fait au travers d'un fichier SVG* intégré à la page. La mise en page et les icônes sont gérées à l'aide de la bibliothèque Bootstrap. Le projet était versionné avec Git, sur la plateforme de Jay Electronique.

Choix des technologies de développement

Les projets de Jay sont réalisés avec .NET, une plateforme de Microsoft fournissant une architecture et un support pour le développement d'applications Windows, Linux, web... Un projet .NET* contient du code source (en C# chez Jay Electronique), un fichier décrivant la structure du projet, des dépendances, et des configurations de compilation et de déploiement [9]. J'ai utilisé cette plateforme afin que mon code soit facilement maintenable par les développeurs de Jay. Plus précisément, mon projet est développé avec ASP.NET, un framework côté serveur utilisé pour construire des applications web : il gère le backend, c'est-à-dire l'accès aux données, les API, ou encore la connexion [10].

Pour le frontend, nous avons sélectionné React, une librairie JavaScript dédiée à la réalisation d'interfaces utilisateur dynamiques. Elle peut notamment mettre à jour le Document Object Model (DOM), représentant l'arborescence d'une page web, sans recharger la page : cela est très utile pour gérer une zone de modification d'image, comme cela est prévu dans le projet. De plus, React* permet de créer des composants pouvant être combinés pour assembler des pages web [11]. Ces composants ont une logique de réaction à certains événements, qui peut être décrite dans les fichiers des composants. React* offre la possibilité d'intégrer des composants prédéfinis au projet en ajoutant des dépendances. Par exemple, pour gérer le déplacement, le redimensionnement et la rotation des textes, j'ai pu utiliser le composant React* Moveable. Celui-ci s'appuie sur Moveable, une bibliothèque JavaScript.

Pour gérer les interactions entre React* et ASP.NET*, j'ai utilisé Webpack, un outil de *bundling* JavaScript (*bundling* signifie « regroupement » en anglais). Il compile les fichiers .jsx utilisés pour React* dans un fichier appelé « bundle ». Au passage, il les optimise à l'aide de Babel, un compilateur JavaScript. Ce bundle est chargé dans le layout ASP.NET*. Le client reçoit la page HTML et le bundle React*. React* est alors chargé dans le navigateur, il affiche l'interface et il appelle les API ASP.NET* via HTTP.

⁹ ChatGPT-4o. Réponse à « Qu'est-ce qu'un projet .NET ? Quand est-il utilisé ? », OpenAI, 7 juin 2025.

¹⁰ ChatGPT-4o. Réponse à « À quoi sert ASP.NET ? », OpenAI, 7 juin 2025.

¹¹ Meta Platforms. *React* [en ligne]. 2025.

Pour la mise en forme, j'ai utilisé Bootstrap, un framework HTML, CSS et JavaScript facilitant la création d'applications responsive. Il suffit d'installer le package (node module) dans le projet, et pour styliser les éléments HTML, il faut indiquer la classe correspondant à un certain style Bootstrap. De plus, Bootstrap offre un service appelé Bootstrap Icons, qui permet d'intégrer facilement des icônes.

Pour respecter les contraintes de la graveuse laser, l'idéal est de partir d'un fichier SVG* dans lequel les spécifications sont déjà renseignées, et de laisser l'utilisateur modifier ce SVG* de manière contrôlée. Nous avons donc utilisé des modèles, des fichiers prédéfinis contenant tous les éléments nécessaires à une intégration dans le code, ainsi qu'à une lecture par la graveuse. La structure est réfléchi : un groupe pour les aperçus des positions (affichés en bleu), un groupe pour les découpes (vert et rouge), un groupe pour les textes et un groupe pour les images. Ces groupes et les objets qu'ils contiennent ont des identifiants et des classes, qui permettent si besoin de les récupérer, de gérer leur style. Au chargement du fichier SVG* sur la page, React* ajoute des écouteurs d'événements aux éléments appropriés : le but est de repérer lorsque l'utilisateur clique sur un élément, pour réagir de manière appropriée.

Pratiques de développement

Jay utilise l'architecture Modèle-Vue-Contrôleur (MVC) dans ses applications. Cette architecture divise le code en trois parties : le dossier Contrôleur gère les échanges avec l'utilisateur, le dossier Modèle contient la logique métier, et le dossier Vue gère l'affichage. Le projet suit donc également cette logique de structuration.

Pour assurer la maintenabilité (notamment en termes de lisibilité), j'ai fait attention à écrire régulièrement des commentaires descriptifs : au moins un pour chaque nouvelle fonction, puis un à chaque étape de la fonction. Il est ainsi possible de comprendre l'utilité d'une fonction uniquement en consultant les commentaires. Cet aspect du projet est crucial, car mon code est destiné à être utilisé comme base ou exemple pour un futur logiciel : les développeurs qui le reprendront doivent pouvoir comprendre l'utilité de chaque partie.

Les fichiers des composants sont structurés en plusieurs parties : d'abord, déclaration des constantes et des variables globales ; ensuite, gestion des événements ; puis fonctions de calcul ; et enfin, rendu du composant. Cette organisation, reprise dans tous les composants et délimitée par des lignes de commentaires, permet de retrouver facilement les différentes parties d'un fichier.

III. 4. Bilan sur les avancées du projet

Évolution du projet

Le projet est passé par plusieurs phases. Au début, ma mission était seulement de prouver la faisabilité et de faire un prototype. La première version du prototype (réalisée en quatre semaines) a prouvé que le projet était viable.

Par la suite, j'ai étudié les besoins des utilisateurs, et j'ai conçu des maquettes représentant l'intégralité du site. Cela a été l'occasion de mieux discuter avec le STC*, et de définir un

nouveau processus de création des plastrons*. J'ai présenté le prototype, expliqué les objectifs, et j'ai pris notes de leurs idées de fonctionnalités et de leurs retours.

J'ai ensuite effectué une revue de code sur mon projet : j'ai progressé en React* au fur et à mesure, donc j'ai pu remarquer que mes premières briques de code étaient désorganisées. Elles ne respectaient pas les principes de React*. J'ai donc réorganisé mes fichiers en créant plus de composants, et j'ai utilisé le principe de « hook » (hameçon en anglais) pour gérer les interactions et les effets d'un élément sur un autre.

Enfin, j'ai ajouté des fonctionnalités répondant aux besoins des utilisateurs. Par exemple, la suppression des textes et des images n'était pas encore possible. Pour permettre à l'utilisateur de redimensionner et de faire tourner les textes et les images, j'ai utilisé le composant React Moveable*. Ce composant gère également l'affichage de lignes directrices pour aider l'utilisateur à placer ses textes et images : les aligner les uns par rapport aux autres, les centrer dans le plastron*, ou ne pas dépasser les bords.

Afin qu'il soit plus facile pour l'utilisateur d'illustrer son plastron*, j'ai intégré une bibliothèque d'icônes. Les images sont affichées à l'aide d'une API qui récupère tous les noms des fichiers du dossier dans lequel sont rangées les images. Cela permet au code React* d'appeler toutes les images du dossier.

Environ six semaines après le début du projet, Valentin Saugnier, un deuxième développeur logiciel, a rejoint Jay. C'est lui qui sera chargé de développer le site au complet. Son langage de prédilection étant Angular, c'est celui-ci qui sera utilisé pour la version finale. La réflexion sur les besoins et les processus, ainsi que les maquettes et les briques de code que j'ai développé seront utilisés comme support lors du développement.

Points techniques et difficultés rencontrées

L'architecture .NET* et le C# ne m'étaient pas familiers. Il a fallu que je comprenne leur fonctionnement pour lancer le projet.

De même, React* comporte une logique nouvelle pour moi, et j'ai rencontré quelques difficultés pour la prendre en main. L'apprentissage s'est fait au cours du projet, et c'est avec l'expérience que j'ai pu me corriger petit à petit. Par exemple, je n'ai vraiment compris la notion de « hooks » qu'après avoir développé une grande partie du projet, et les hooks que j'avais déjà intégrés sans les comprendre n'étaient pas forcément pertinents. Les dernières semaines du projet étaient agréables, car je comprenais mieux ce que je faisais et je pouvais réellement faire des choix logiques et justifiés.

Une autre difficulté fut de gérer l'intégration du SVG* dans le HTML. Le SVG* est importé après le premier chargement du DOM*. C'est un élément extérieur au React*. React* doit donc l'utiliser comme tel, et n'a pas d'effet direct sur lui, ce qui nous empêche d'utiliser toutes les fonctionnalités de React*.

Ainsi, après ces tests approfondis, React* n'est pas vraiment le langage le plus approprié pour l'application. Il n'a pas été utilisé au maximum de ses capacités. Le développeur ayant rejoint l'équipe après le début du stage utilisera plutôt Angular pour la version finale.

Conclusion

Le projet de création d'un configurateur de plastron* graphique est sur la bonne voie. Le prototype réalisé au cours de mon stage prouve sa faisabilité : il est possible d'interagir avec du SVG sur le web, tout en respectant des contraintes techniques.

L'étude des besoins, la définition des processus et les maquettes présentées dans ce rapport sont d'ores et déjà utilisées par Jay Electronique pour développer la version finale. Cependant, cette version sera développée avec Angular plutôt qu'en React, car React a montré certaines limites lors des interactions avec le SVG.

Pendant ce stage, j'ai mobilisé mes compétences en gestion de projet, en conception et en développement. Grâce aux nombreuses réunions de présentation des avancées du projet, je suis devenue plus à l'aise à l'oral et j'ai amélioré mes capacités de communication.

Pour mon parcours professionnel, ce stage était une vraie opportunité. J'ai pu aborder la partie des projets qui m'intéressent le plus : la réflexion en amont autour des besoins des utilisateurs, les choix ergonomiques. Avoir abordé un framework JS est aussi un plus : j'ai progressé en React au fur et à mesure, et j'ai compris certains des principes fondamentaux de React.

Glossaire

ASP.NET : framework côté serveur utilisé pour construire des applications web : il gère le backend, c'est-à-dire l'accès aux données, les API, ou encore la connexion.

ATEX-IECEEx : certification de sécurité ATEX-IECEEx pour une utilisation dans des environnements explosifs. ATEX pour ATmosphère EXplosibles, IECEEx pour *International Electrotechnical Commission System for Certification to Standards relating for use in Explosive Atmospheres*, ce qui se traduit par « Système de certification de la Commission Électrotechnique Internationale pour les normes relatives aux équipements destinés à être utilisés en atmosphères explosibles ». L'ATEX est spécifique à l'Union européenne, tandis que l'IECEEx est un système de certification international. Ces deux normes garantissent que l'équipement n'allumera ni gaz ni poussières combustibles.

CorelDRAW : logiciel de dessin vectoriel édité par Corel.

DOM : Document Object Model, arborescence d'une page web.

ISO 9001-2015 : norme de management de la qualité.

MVC : Modèle-Vue-Contrôleur. Cette architecture divise le code en trois parties : le dossier Contrôleur gère les échanges avec l'utilisateur, le dossier Modèle contient la logique métier, et le dossier Vue gère l'affichage.

.NET : plateforme de Microsoft fournissant une architecture et un support pour le développement d'applications Windows, Linux, web, et autres. Un projet .NET contient du code source (en C# chez Jay), un fichier décrivant la structure du projet, des dépendances, et des configurations de compilation et de déploiement.

Organe : éléments de commande physiques, tels que des boutons, des joysticks ou des potentiomètres.

Pika-Moka : gamme de radiocommandes dont les organes et les inscriptions sur le plastron sont personnalisables.

Plastron : interface physique de la radiocommande, sur laquelle sont positionnés les organes et les inscriptions.

React : librairie JavaScript dédiée à la réalisation d'interfaces utilisateur dynamiques, dont le principe repose sur l'utilisation de composants pour construire les interfaces.

React Moveable : composant React s'appuyant sur Moveable (une bibliothèque JavaScript). Il gère le déplacement, le redimensionnement et la rotation des textes et images.

R&D : Recherche et Développement. Service chez Jay Electronique chargé de concevoir de nouvelles solutions pour les clients et de développer les logiciels du secteur Production.

SIL 2-PLd et SIL 3-PLe : certifications de sécurité sur les arrêts d'urgence. SIL (*Safety Integrity Level*), se traduit par « Niveau d'intégrité de sécurité » est une mesure du niveau de réduction du risque, de 1 (le plus faible) à 4 (le plus élevé). PL (*Performance Level*), se traduit par « Niveau de performance », est une estimation de la capacité d'une fonction de sécurité à fonctionner correctement, de a (le plus faible) à e (le plus élevé). Elle se base sur la durée de vie des composants, la capacité de diagnostic et la résistance aux défaillances communes.

STC : Support Technique Client.

SVG : Scalable Vector Graphics. Format d'image s'appuyant sur le langage XML pour décrire des ensembles de graphiques vectoriels.

Webpack : outil de *bundling* JavaScript (*bundling* signifie « regroupement » en anglais). Il compile les fichiers .jsx utilisés pour React* dans un fichier appelé « bundle ». Il optimise ces fichiers avec Babel, un compilateur JavaScript.

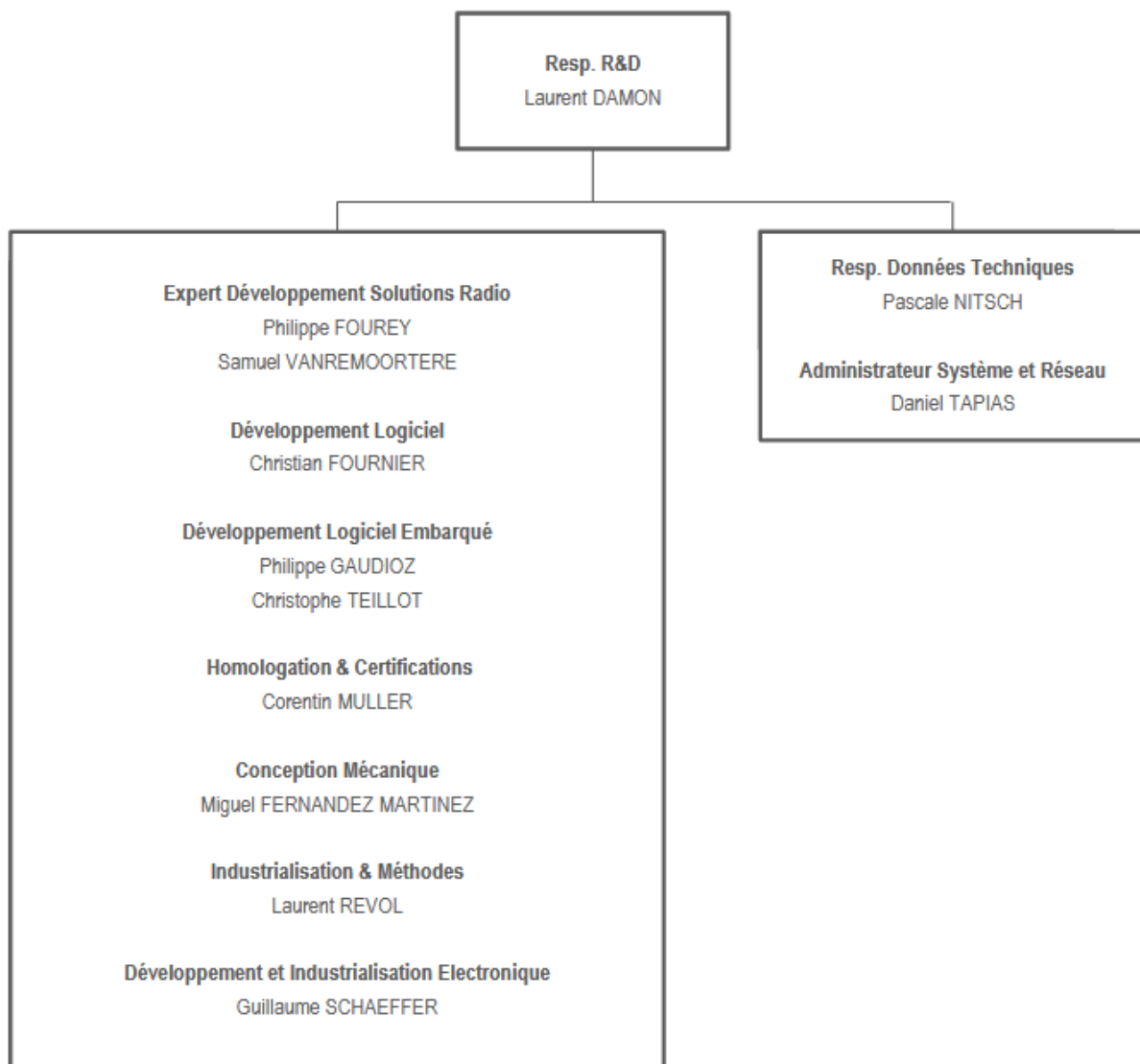
Bibliographie-sitographie

- [1] Wikipédia. *Scalable Vector Graphics – Wikipédia* [en ligne]. 15 avril 2025. Disponible à l'adresse : https://fr.wikipedia.org/wiki/Scalable_Vector_Graphics [consulté le 6 juin 2025].
- [2] Conductix-Wampfler. *News: Conductix-Wampfler acquiert Jay Electronique | France* [en ligne]. 4 juillet 2019. Disponible à l'adresse : <https://www.conductix.fr/fr/nouvelles/conductix-wampfler-acquiert-jay-electronique> [consulté le 8 juin 2025].
- [3] Groupe Delachaux. *Nos Entreprises – Delachaux* [en ligne]. 2019. Disponible à l'adresse : <https://delachaux.com/fr/le-groupe/nos-entreprises/> [consulté le 10 juin 2025].
- [4] Conductix-Wampfler. *Synoptique Radiocommandes Jay* [en ligne]. 2020. Disponible à l'adresse : <https://www.jay-electronique.com/sites/default/files/Documentations%20commerciales/KAT0500-0001-FR%20Synoptique%20Radiocommandes%20Jay.pdf> [consulté le 10 juin 2025].
- [5] ISO. *ISO 9001:2015 - Systèmes de management de la qualité — Exigences* [en ligne]. 2021. Disponible à l'adresse : <https://www.iso.org/fr/standard/62085.html> [consulté le 10 juin 2025].
- [6] ChatGPT-4o. Réponse à « Explique-moi ce que sont les normes SIL 2-PLd et SIL 3-PLe. », OpenAI, 5 juin 2025.
- [7] ChatGPT-4o. Réponse à « Explique-moi ce qu'est la norme ATEX-IECEx. », OpenAI, 5 juin 2025.
- [8] BASTIEN, Christian et SCAPIN, Dominique. *Ergonomic criteria for the evaluation of human-computer interfaces* [en ligne]. Rapport Technique RT-0156, INRIA, Juin 1993. Disponible à l'adresse : <http://hal.inria.fr/inria-00070012/en/> [consulté le 10 juin 2025].
- [9] ChatGPT-4o. Réponse à « Qu'est-ce qu'un projet .NET ? Quand est-il utilisé ? », OpenAI, 7 juin 2025.
- [10] ChatGPT-4o. Réponse à « À quoi sert ASP.NET ? », OpenAI, 7 juin 2025.
- [11] Meta Platforms. *React* [en ligne]. 2025. Disponible à l'adresse : <https://fr.react.dev/> [consulté le 10 juin 2025].

Annexes

Annexe 1 : Organigramme du service Recherche et Développement	2
Annexe 2 : Planning du proje.....	3
Annexe 3 : PDF à compléter par le client dans le processus actuel	4
Annexe 4 : Maquettes.....	5

Annexe 1 : Organigramme du service Recherche et Développement




Annexe 2 : Planning du projet


Semaine 1	<ul style="list-style-type: none"> État de l'existant, analyse concurrentielle, étude des processus courants Étude technique, choix des technologies, test des technologies : interagir avec un SVG*, le faire réagir aux clics, l'exporter
Semaine 2	<ul style="list-style-type: none"> Tests de faisabilité avancés : prouver qu'il est possible d'interagir avec un SVG* sur le web, de choisir des fonctionnalités et d'exporter ce fichier Faire une démonstration avec un fichier ressemblant le plus possible aux fichiers utilisés par la graveuse
Semaines 3 et 4	<ul style="list-style-type: none"> Possibilité de lire le fichier exporté dans l'outil CorelDRAW* Faire un test de découpe avec la machine à partir d'un fichier exporté depuis le site
Semaine 5	<ul style="list-style-type: none"> Proposer des maquettes pour les interfaces du site Réflexion sur le nouveau processus (avec le STC*) : comment gérer la création d'un compte, la connexion, la validation du plastron*, à quelles données peut-il accéder... Faire un organigramme de l'application représentant le parcours entre les pages Trouver un outil pour gérer la dimension et le déplacement des textes et des images
Semaine 6	<ul style="list-style-type: none"> Réparation de bugs Réorganisation des fichiers du projet : meilleure compréhension du framework donc revue du propre code pour le nettoyer et réorganiser les composants Développement de fonctionnalités : ajout de guidelines pour aider au placement des éléments, suppression des éléments, annulation des actions
Semaines 7 et 8	<ul style="list-style-type: none"> Développement de fonctionnalités : bibliothèque d'icônes, zoom dans la zone de modification, ajout de la référence du plastron* Présentation de l'application au STC* Rédaction du mémoire
Semaine 9	<ul style="list-style-type: none"> Rédaction finale du mémoire et préparation de la soutenance Ajout d'un masque pour afficher/cacher certains éléments dans la zone de modification Tests de gravure
Semaine 10	<ul style="list-style-type: none"> Ajout d'un aperçu concret du plastron* Apporter des corrections à l'application selon les retours du STC*
Semaine 11	<ul style="list-style-type: none"> Créer les fichiers de données pour la création de plastrons* (XML, SVG*)
Semaine 12	<ul style="list-style-type: none"> Vérifier le respect des contraintes techniques Documentation du projet pour les futurs développeurs

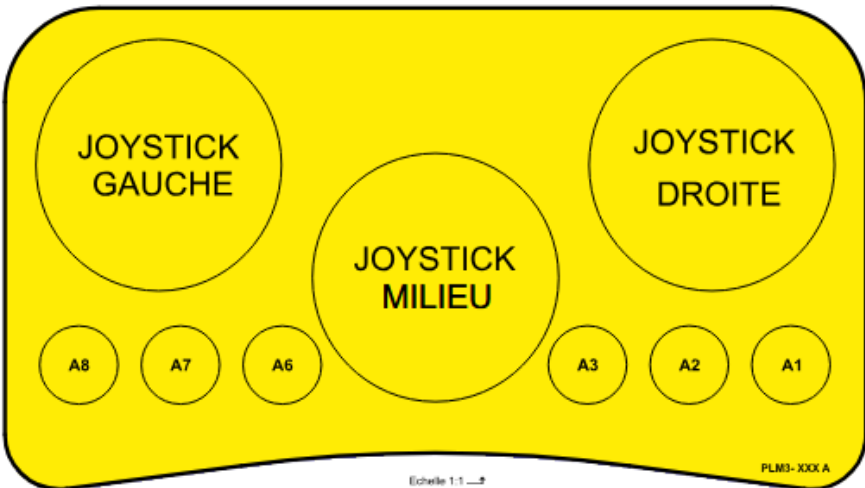
Annexe 3 : PDF à compléter par le client dans le processus actuel

Le client coche les organes pour chacune des positions, dessinent leurs icônes et textes.

Projet à compléter







Mettre vos textes/icônes sur le Plastron.
(Merci de nous joindre vos logos/images pour une meilleure qualité.)

* inscription uniquement en noir sur fond jaune.


Echelle 1:1 →


PLM3-XXX A

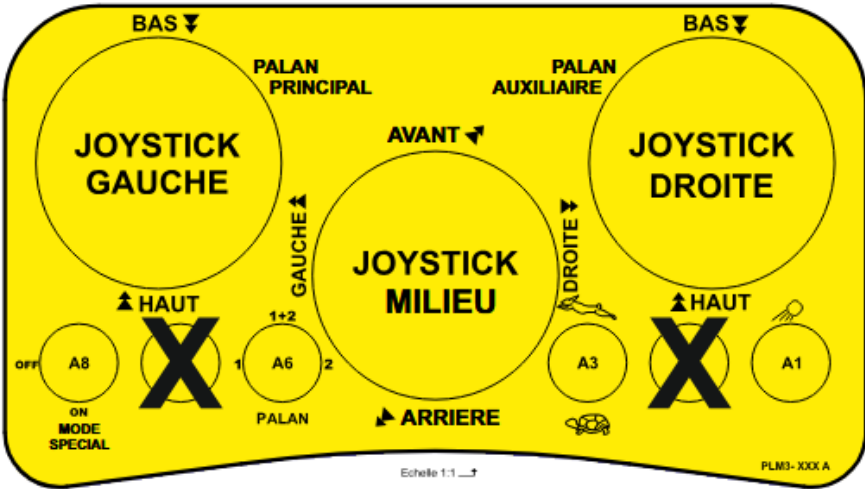
COMMUTATEURS		POSITION SUR LE PLASTRON					
		A1	A2	A3	A6	A7	A8
TYPE DE COMMUTATEUR	Bascule 3 états (fixe/fixe/fixe)						
	Bascule 3 états (fixe/fixe/retour au centre)						
	Bascule 3 états (retour au centre/fixe/retour)						
	Bascule 2 états (fixe/fixe)						
	Bascule 2 états (fixe/retour au centre)						
	Bouton poussoir 2 états						
	Commutateur à clé 2 états						

joysticks	GAUCHE		MILIEU		DROITE	
	Axe E O	Axe N S	Axe E O	Axe N S	Axe E O	Axe N S
Analogique						
CRANS	4/ 4					
	3/ 3					
	2/ 2					
	1/ 1					
Non utilisé						
Verrouillage en croix (Oui/Non)						

EXEMPLE DE FACADE







Echelle 1:1 →



PLM3-XXX A


COMMUTATEURS		POSITION SUR LE PLASTRON					
		A1	A2	A3	A6	A7	A8
TYPE DE COMMUTATEUR	Bascule 3 états (fixe/fixe/fixe)				X		
	Bascule 3 états (fixe/fixe/retour au centre)						
	Bascule 3 états (retour au centre/fixe/retour)						
	Bascule 2 états (fixe/fixe)	X		X			
	Bascule 2 états (fixe/retour au centre)						
	Bouton poussoir 2 états						X
	Commutateur à clé 2 états						

joysticks	GAUCHE		MILIEU		DROITE	
	Axe E O	Axe N S	Axe E O	Axe N S	Axe E O	Axe N S
Analogique				X		
CRANS	4/ 4					
	3/ 3					
	2/ 2			X		
	1/ 1		X			X
Non utilisé	X				X	
Verrouillage en croix (Oui/Non)	NON		OUI		NON	

Annexe 4 : Maquettes

Page de connexion :



[Aide](#) Français ▼ 

Connexion

Email

Mot de passe



[Mot de passe oublié ?](#)


[Se connecter](#)

Nous contacter

Formulaire de contact ?
Adresse mail et téléphone ?
Site public de Conductix

Page d'accueil (non connecté) :



[Aide](#) Français ▼ 

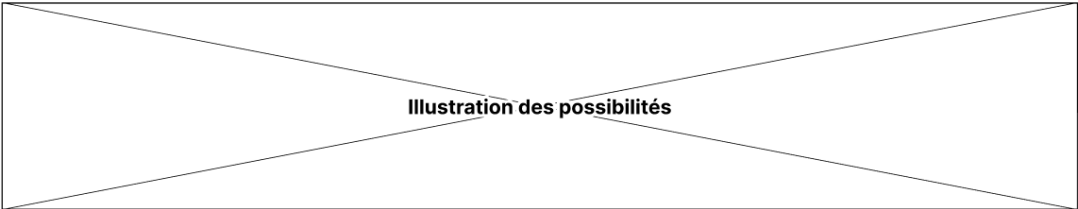
Bienvenue sur NOM LOGICIEL.

Paragraphe introductif, description des possibilités...
Comment obtenir un compte ?
Peu être un peu long.



[Se connecter](#) [Demander la création d'un compte](#)

Nous contacter


Formulaire de contact ?
Adresse mail et téléphone ?
Site public de Conductix et de JAY



Page d'accueil (connecté) :



AideFrançais



Bienvenue NOM ENTREPRISE.

Commande n°...

Valider la commande

Plastron

Plastron

Statut : En cours

Plastron

Plastron

Statut : Abandonné

Plastron

Modifier

Statut : En cours

Plastron

Plastron

Statut : Validé

Commande n°...

Commande validée

Plastron

Modèle

Statut : Validé

Plastron

Plastron

Statut : Validé

Nous contacter

Formulaire de contact ?
Adresse mail et téléphone ?
Site public de Conductix

Résumé :

Jay Electronique est spécialisée dans la fabrication de radiocommandes de sécurité, notamment pour l'industrie. L'une de leur gamme de radiocommandes est personnalisable par les clients, mais le processus de personnalisation actuel n'est pas efficace. Pour remédier à cela, le service Recherche et Développement s'est penché sur la création d'un configurateur graphique de plastron de radiocommande, accessible en ligne par les clients. On m'a confié deux missions : prouver la faisabilité de ce projet, et développer un prototype. La première phase fut consacrée à l'analyse de l'existant, à l'étude des besoins des utilisateurs et à la compréhension des processus. J'ai développé un prototype en React et C#, où la partie graphique est gérée en SVG. Grâce à ce prototype, la faisabilité a bien été confirmée : la première version permet déjà de créer un plastron exploitable par la graveuse laser.

Mots-clés :

Configurateur, graveuse laser, industrie, personnalisation, plastron, prototype, radiocommande, React, sécurité, SVG.

Abstract:

Jay Electronique, a manufacturer specializing in secure remote controls, offers a range of customized products to its customers. However, the current customization process is inefficient, relying on email and phone exchanges, which often lead to costly errors. To enhance this process, the Research and Development team aims to develop an online faceplate configurator. I was entrusted with two missions: to prove feasibility and to develop a prototype. The project began with an analysis of Jay Electronique's existing tools, user needs, and processes. Following this analysis, I designed and developed a functional prototype using React and C#. The graphical part is managed in SVG, and the prototype's output is compatible with the laser engraver. The success of this prototype confirms its feasibility and allows Jay Electronique to proceed with the development of the final version. This project represents a significant step forward in improving the customization and process for Jay Electronique's customers.

Keywords:

Configurator, customization, faceplate, industry, laser engraver, prototype, React, remote control, security, SVG.