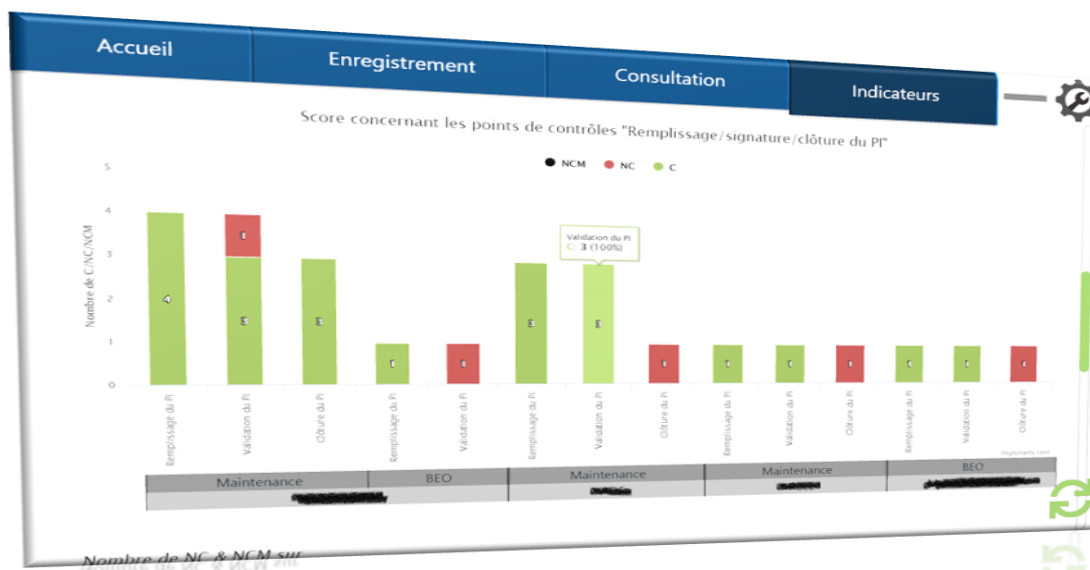


MEMOIRE 2019 - 2020

GESTION DE PROJET : APPLICATION PILOTAGE CHANTIER

CHRISTOPHER MILAZZO



INTRODUCTION

Le projet qui suit a fait l'objet d'un stage d'immersion dans l'entreprise Trigo Qualitaire, dans le cadre de mon titre professionnel de développeur d'application WEB et Mobile de 2018 à 2019 au sein du centre de formation Adrar pôle numérique à Ramonville (31 520).

J'ai été le seul concepteur/développeur sur ce projet, ainsi ma connaissance sur le sujet me permet de poser de manière claire et précise le déroulement et la réflexion que j'ai pu avoir lors de la phase préparatoire et de la gestion de son implémentation.

Table des matières

INTRODUCTION	2
Phase I – DÉFINITION	5
Étape 1 – Le but du projet.....	5
a – Présentation du client	5
b – L'existant	5
c – Recueil du besoin exprimé par le client	5
Étape 2 – Analyse de situation S.W.O.T.....	7
Étape 3 – Définition des objectifs à atteindre S.M.A.R.T.....	7
Étape 4 – Organigramme technique	12
Étape 5 – Organisation du projet.....	13
Étape 6 – Évaluation prévisionnelle des enjeux.....	13
Étape 7 – Les livrables.....	13
Phase II – PLANIFICATION	15
Étape 1 – Déclinaison de la démarche de travail WBS	15
Étape 2 – Matrice des responsabilités	16
Étape 3 – Élaboration du calendrier POINTS DE FONCTIONS, COCOMO, GANTT	18
Envergure du projet – Points de fonctions	18
COCOMO	22
GANTT	23
Étape 4 – Budget prévisionnel	25
Étape 5 – Évaluation des risques - Méthode AMDEC	26
Phase III – RÉALISATION.....	28
1 – Méthodes de travail	28
2 – Le périmètre du projet	28
3 – Le type d'application	28
4 – Les utilisateurs.....	28
5 – Le matériel et le support	29
6 – Contraintes techniques	29
7 – Le choix des technologies	29
8 – Graphisme et ergonomie	31
a – La charte graphique.....	31
b – Wireframe et maquettage	31
c – Les logos et les symboles.....	31
Phase IV – CONCLUSION	32
ANNEXES	33

Phase I – DÉFINITION

Étape 1 – Le but du projet

a – Présentation du client

Le groupe TRIGO a été fondé en 1997. Il est aujourd'hui présent mondialement et compte près de 10000 collaborateurs. Il fournit des solutions opérationnelles de gestion de la qualité, et marginalement, de la gestion de la sécurité des personnes. Son siège est à Nanterre.

TRIGO Qualitaire est une branche qui est spécialisée dans l'aéronautique.

b – L'existant

L'application développée concerne l'une des nombreuses prestations réalisées par Trigo-Qualitaire à Airbus. Celle-ci se nomme "Safety guidance" (Pilotage Chantiers). L'ingénieur sécurité placé sur cette prestation a pour rôle d'auditer les FALs (Final Assembly Line) afin de s'assurer que les entreprises sous-traitantes, travaillant sur les postes dans le domaine des moyens industriels, le font en toute sécurité.

Un audit (appelé également visite), s'effectue en vérifiant plusieurs points de contrôle correspondant chacun à une situation de travail donnée.

Sans l'application, l'ingénieur sécurité note ses visites sur un cahier, puis retourne au bureau afin de les saisir sur son fichier Excel. Puis, il génère de nombreux graphes afin de réaliser des indicateurs (mensuels, trimestriels, semestriels et annuels).

On constate que non seulement il y a une perte de temps, mais également un risque d'erreurs dues à la saisie manuelle.

c – Recueil du besoin exprimé par le client

Le besoin exprimé par Trigo-Qualitaire est non seulement un besoin d'amélioration du quotidien de son salarié (l'ingénieur sécurité) en limitant les tâches répétitives et fastidieuses, mais aussi un besoin de gain de temps afin qu'il puisse délivrer plus de formations préventives aux entreprises auditées.

Concernant le client de Trigo-Qualitaire (Airbus), c'est le besoin d'accès rapide aux données brutes et aux indicateurs sans temps de latence qui a été exprimé. Ce besoin sera comblé par le biais d'un portail WEB qui mettra à disposition d'Airbus une page de consultation de visites munie de multiples filtres, ainsi que d'une page capable de générer les indicateurs en temps réel et de manière entièrement automatisée.

L'idéale serait une application sur tablette tactile qui procurerait praticité, mobilité et ergonomie afin d'assurer sa mission aux quotidiens.

Cette application doit également être portée sur un navigateur WEB (application WEB) afin de mettre à disposition de Airbus les données.

L'application mobile devra respecter certaines contraintes :

- Doit fonctionner sur un matériel masterisé Airbus
- Doit pouvoir continuer de fonctionner même dans le cas d'une perte de connexion avec le serveur (autonome)
- Doit protéger l'accès à l'application par compte utilisateur (même hors-ligne)

Les fonctionnalités attendues :

- L'ingénieur sécurité doit pouvoir créer une nouvelle visite, ou en éditer une déjà existante.
- Une visite sera identifiée par un identifiant numérique unique, une date, un nom, un numéro d'intervention, un lieu d'audit et le nom de l'entreprise auditée.
- Lors d'une visite, l'application doit mettre à disposition une liste de "situations de travail" (ex. travail par point chaud) dans lesquelles se trouvent plusieurs points de contrôle (ex. Port des EPI) qui pourront être cochés comme étant Conforme, Non-Conforme ou Non-conforme-majeure.
- Pour chaque point de contrôle vérifié, l'opérateur doit avoir la possibilité de lui assigner un commentaire.
- En cas de non-conformité, on doit pouvoir assigner une action et une entité responsable de la remise en conformité pour ce point de contrôle ainsi qu'une date limite pour effectuer cette dernière.
- L'ingénieur sécurité doit pouvoir prendre des photos directement via l'application et les ajouter à la visite.
- Il doit également pouvoir sélectionner une photo présente en mémoire sur la tablette et l'associer à la visite.
- L'ingénieur sécurité et Airbus doivent pouvoir consulter les différentes visites.
- La page de consultation des visites doit mettre à disposition divers filtres pour faciliter les recherches.
- L'ingénieur sécurité et Airbus doivent pouvoir consulter les indicateurs, constituer de plusieurs graphiques et tableaux automatiquement générés par l'application en fonction des visites effectuées.

Étape 2 – Analyse de situation S.W.O.T.

STRENGTHS

- Autodidacte.
- Motivé
- Bonne connaissance des technologies du WEB
- Bonnes notions de la programmation depuis l'âge de 14 ans

WEAKNESSES

- Connaissance du SGBD PostgreSQL.
- Expérience professionnelle (junior).
- Aucun appui technique de la part de TRIGO QUALITAIRE du fait que ce ne soit pas une entreprise informatique.

OPPORTUNITIES

- Montée en compétence en flèche
- Possibilité de prouver ma valeur
- Grande liberté d'action (autonomie quasi-totale)
- Sujet de stage sérieux et très professionnel pour de gros clients (Très valorisant sur le CV)

THREATS

- Temps pour la conception et le développement de l'application court

Étape 3 – Définition des objectifs à atteindre S.M.A.R.T.

Objectif 1 – Apprentissage autodidacte du Framework Electron

- **Spécifique** : Apprendre à utiliser le Framework
- **Mesurable** : Par expérience, l'apprentissage d'une nouvelle technologie prend environ une bonne semaine.
- **Orienté vers l'Action** : Il faut être capable de développer une application de base avec Electron. Les spécificités viendront par la suite suivant les besoins.
- **Réaliste** : Un développeur est habitué à apprendre de nouveaux langages ou Framework
- **Facteur de Temps et ressources** : Environ une semaine.
- **Enoncé de l'objectif** : *“L'objectif est de se donner environ 1 semaine afin d'être capable de démarrer et de commencer à développer une application simple en utilisant l'écosystème du Framework Electron.”*

Objectif 2 – Apprentissage autodidacte de J2EE (partie backend en JAVA sur serveur TOMCAT)

- **Spécifique** : Apprendre à utiliser l'environnement de travail Java J2EE
 - **Mesurable** : Apprendre une nouvelle techno (1 semaine), apprendre les bases et architecture d'une application de type serveur (2 semaines) et appréhender l'écosystème J2EE (1 semaine)
 - **Orienté vers l'Action** : Appréhender les concepts de servlet (classe contrôleur) servant de point d'entrées pour une application de type serveur. Il y a également la notion de cycle de vie et de "servlet container" à comprendre pour pouvoir produire un code de qualité.
 - **Réaliste** : Un développeur est habitué à apprendre de nouveaux langages ou Framework
 - **Facteur de Temps et ressources** : Environ 1 mois
-
- **Enoncé de l'objectif** : *"L'objectif est d'être capable de développer une application dans le respect des standards de type serveur en utilisant l'environnement J2EE et le serveur de type TOMCAT"*

Objectif 3 – Effectuer toute la partie conception de Base de données (méthode MERISE)

- **Spécifique** : Créer le model de données représentant l'architecture de la BDD de type PostgreSQL puis générer son script SQL.
 - **Mesurable** : Une fois la partie métier bien comprise, la création du modèle de données ne dépasse pas 1 journée. On compte 1 journée supplémentaire pour tester et valider avec le client avec un jeux de données fictif, puis 1 jour pour les éventuelles corrections à apporter.
 - **Orienté vers l'Action** : Il faut être capable de développer une application de base avec Electron. Les spécificités viendront par la suite suivant les besoins.
 - **Réaliste** : Créer un modèle de données n'est pas compliqué et fait partie des compétences d'un développeur.
 - **Facteur de Temps et ressources** : 3 jours
-
- **Enoncé de l'objectif** : *"En 3 jours on doit avoir le modèle de données de la BDD de type PostgreSQL, testée et approuvée par le client. Ainsi que sont script SQL afin de pouvoir la créer et y injecter les données initiales."*

Objectif 4 – Effectuer les mockup et la partie conception de l'application (diagrammes UML)

- **Spécifique** : Réaliser au moins l'itération 1 du diagramme des cas d'utilisation, le diagramme des classes de l'application mobile et WEB ainsi que celui du serveur.
 - **Mesurable** : Il y a beaucoup de choix quant à l'architecture à utiliser pour l'application WEB et mobile. 1 mois et demi semble être un minimum. Pour la partie serveur, l'utilisation du design pattern MVC et Factory est suffisant et facile à mettre en place ; 1 à 2 semaines suffisent.
 - **Orienté vers l'Action** : Il faut connaître les standards MVC et Factory pour pouvoir réaliser des diagrammes de classe en bonne et due forme.
 - **Réaliste** : Même si ce n'est pas forcément sa première fonction, un bon développeur doit être capable d'implémenter différents design pattern pour les retranscrire dans un langage comme UML.
 - **Facteur de Temps et ressources** : Environ 2 mois.
-
- **Enoncé de l'objectif** : *“On se donne 2 mois pour choisir le modèle d'architecture à employer et de poser le tout sur papier (UML).”*

Objectif 5 – Etablir le cahier des charges et le faire valider par le client Trigo Qualitaire

- **Spécifique** : Effectuer le cahier des charges.
 - **Mesurable** : En supposant que le client fasse des retours rapides à nos questions, on est capable de comprendre le besoin métier du client, de budgétiser le tout et de retranscrire l'ensemble par écrit en 1 semaine.
 - **Orienté vers l'Action** : Comprendre le besoin exprimé par le client et le retranscrire par écrit
 - **Réaliste** : Ce n'est pas compliqué de faire un cahier des charges
 - **Facteur de Temps et ressources** : Environ une semaine.
-
- **Enoncé de l'objectif** : *“L'objectif est de finir en 1 semaine afin de comprendre le besoin métier du client et de rédiger le cahier des charges.”*

Objectif 6 - Développer et tester l'application mobile JALON 1

- **Spécifique** : Développer une application
 - **Mesurable** : En priorisant on est capable de gérer le temps de développement d'une application.
 - **Orienté vers l'Action** : Développer une application
 - **Réaliste** : Coder une application fait partie du quotidien du développeur.
 - **Facteur de Temps et ressources** : 3 mois maximum pour atteindre le premier JALON. (1 mois de stage en entreprise + 2 mois à l'école). L'objectif est de finir avant le début de la deuxième période de stage (octobre – novembre).
-
- **Enoncé de l'objectif** : *“Il y a beaucoup de fonctionnalités et peu de temps. Toutefois parmi les fonctionnalités il y en a qui sont prioritaires (obligatoires). En définissant les fonctionnalités obligatoires et en découpant le développement de l'application mobile en 2 JALONS, on doit être capable de gérer efficacement le temps ”*

Objectif 7 - Développer et tester l'application mobile JALON 2

- **Spécifique** : Développer une application
 - **Mesurable** : En priorisant on est capable de gérer le temps de développement d'une application.
 - **Orienté vers l'Action** : Développer une application
 - **Réaliste** : Coder une application fait partie du quotidien du développeur.
 - **Facteur de Temps et ressources** : 3 mois maximum pour atteindre le deuxième JALON. (1 mois et demi de stage en entreprise + 1 mois et demi à l'école). La date limite est début janvier car il faut garder 1 mois pour la rédaction de la documentation technique, et la date de fin de projet est fixée à fin janvier 2019.
-
- **Enoncé de l'objectif** : “Il faut finir de développer les fonctionnalités du deuxième JALON et si possible faire les fonctionnalités optionnelles. Le développement démarre lors du second stage en entreprise (début octobre 2018) et doit finir au maximum 1 mois avant la date de fin de projet fixée à fin janvier 2019.”

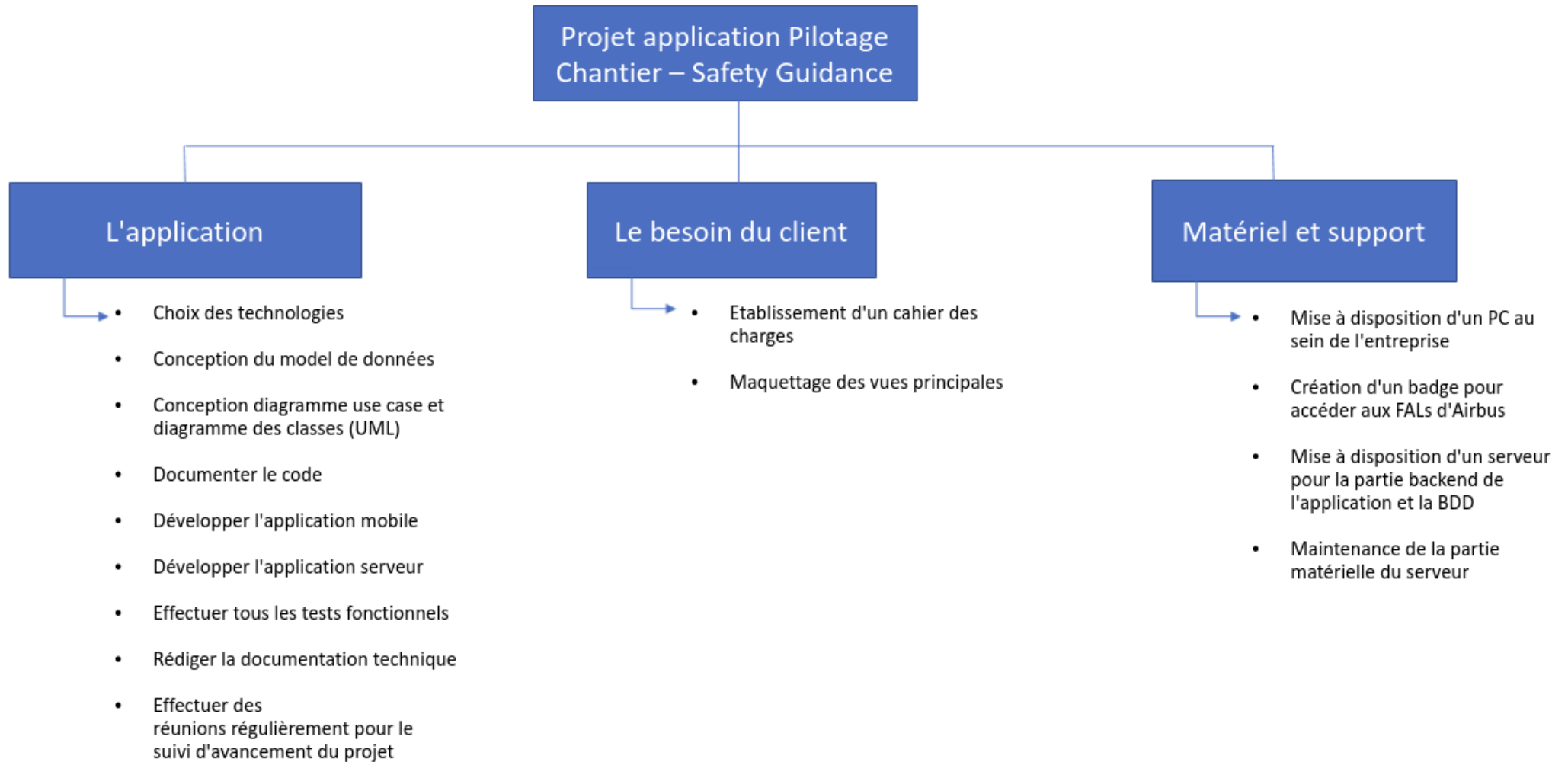
Objectif 8 - Développer la partie BackEnd

- **Spécifique** : Développer une application J2EE
 - **Mesurable** : En parallèle avec le développement de l'application mobile.
 - **Orienté vers l'Action** : Développer une application de type serveur.
 - **Réaliste** : Coder une application fait partie du quotidien du développeur.
 - **Facteur de Temps et ressources** : En parallèle avec l'application mobile donc 6 mois de développement.
-
- **Enoncé de l'objectif** : *“L'application backend est présente pour répondre aux besoins de l'application mobile. Pour éviter les fausses routes, le développement du serveur se fera en parallèle avec la partie frontend. Pour chaque fonctionnalité de l'application mobile, il sera associé un contrôleur et un service à développer coté serveur.”*

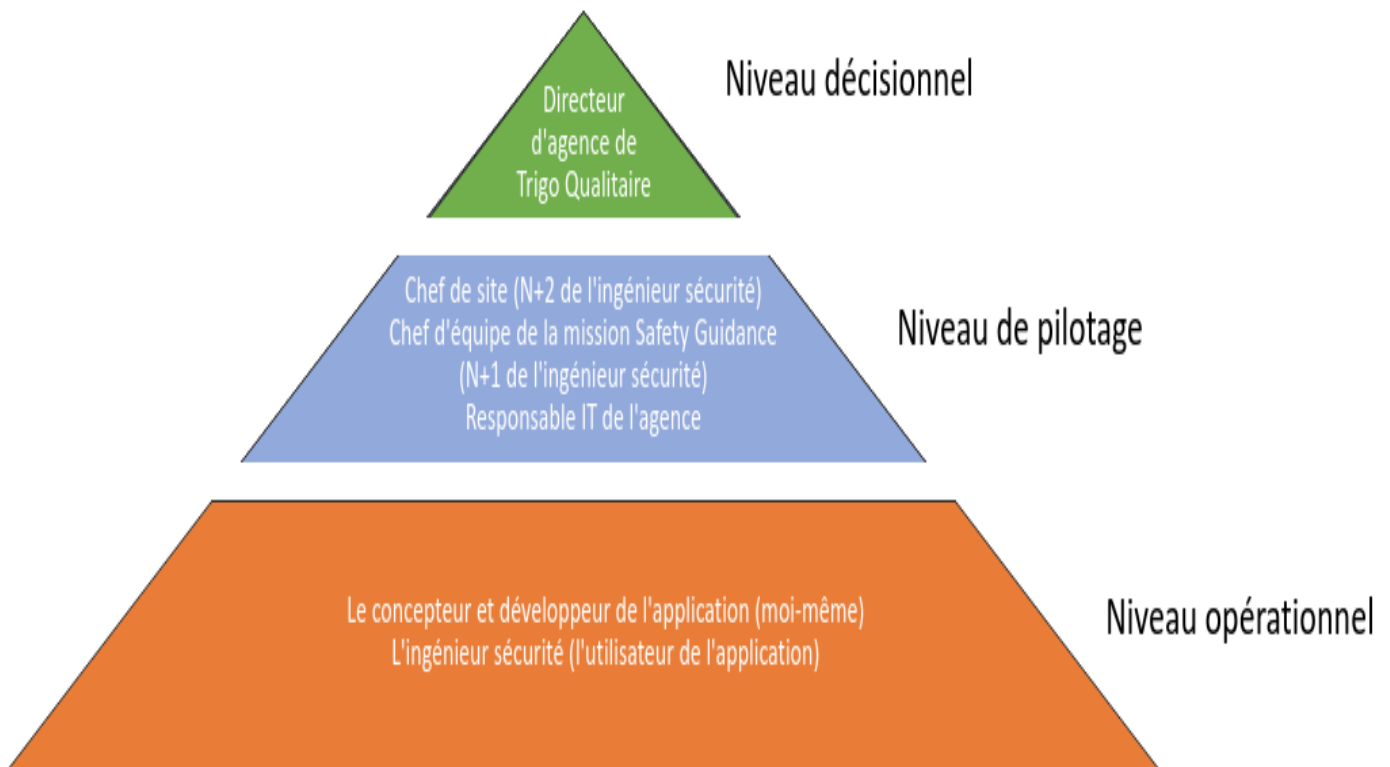
Objectif 9 - Rédiger la documentation technique

- **Spécifique** : Rédiger une documentation technique
 - **Mesurable** : Plus on prend du temps et plus la documentation pourra être précise. 1 mois semble être un délai raisonnable compte tenu de l'envergure de l'application (petite)
 - **Orienté vers l'Action** : Rédiger la documentation technique.
 - **Réaliste** : Toute personne ayant participé au projet est capable de rédiger sa documentation technique
 - **Facteur de Temps et ressources** : 1 mois.
-
- **Enoncé de l'objectif** : *“On se donne 1 mois pour rédiger la documentation technique du projet. Celle-ci devra faire apparaître la doc de l'API REST entre l'application mobile et le serveur, l'architecture globale du projet par le biais de diagrammes UML, le script SQL (MPD) de la BDD, définir les technologies utilisées, le diagramme de déploiement, les méthodes de déploiement et la façon de configurer l'application.”*

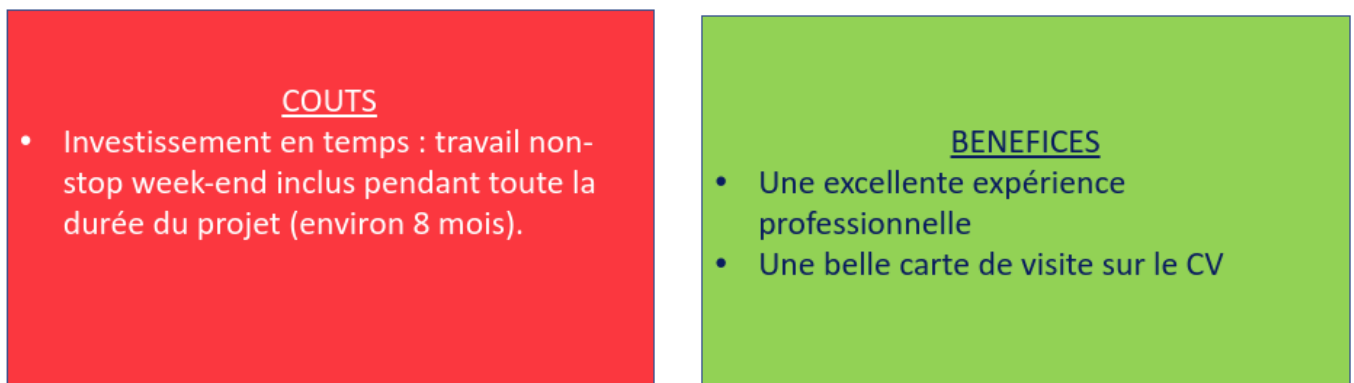
Étape 4 – Organigramme technique



Étape 5 – Organisation du projet



Étape 6 – Évaluation prévisionnelle des enjeux

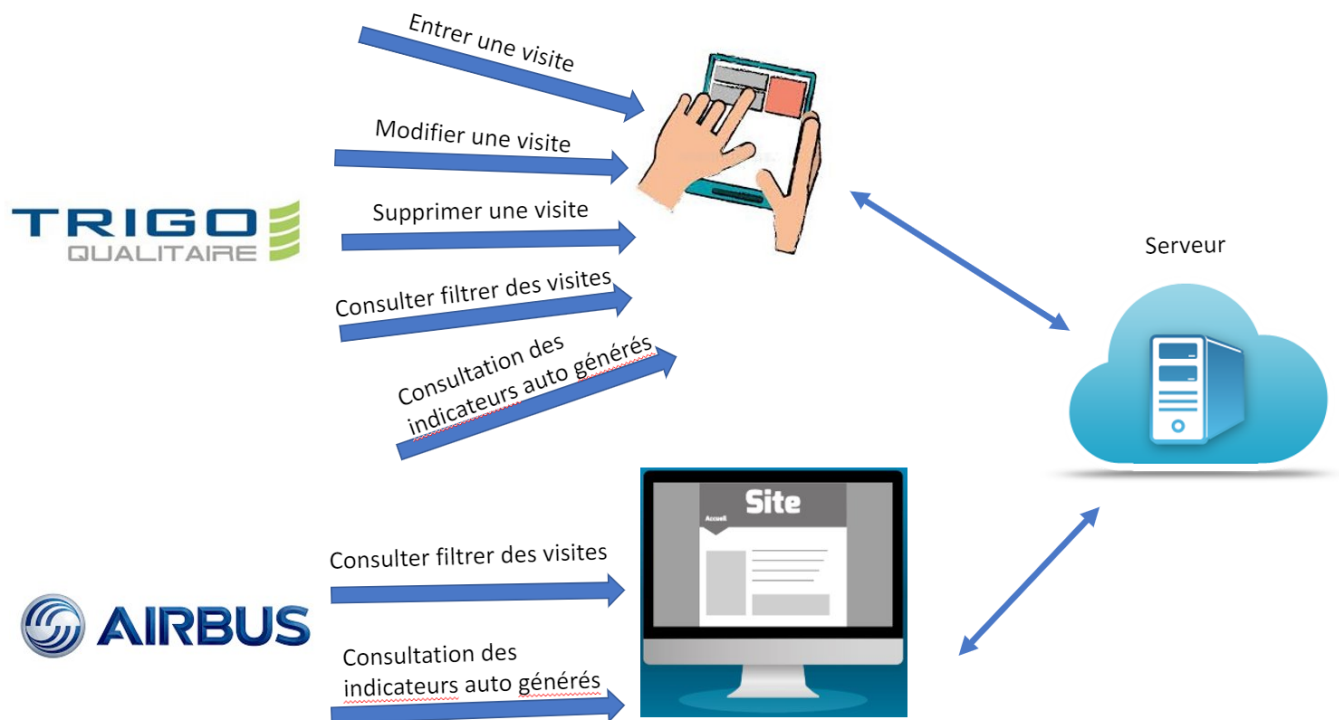


Étape 7 – Les livrables

Les livrables prévus sont :

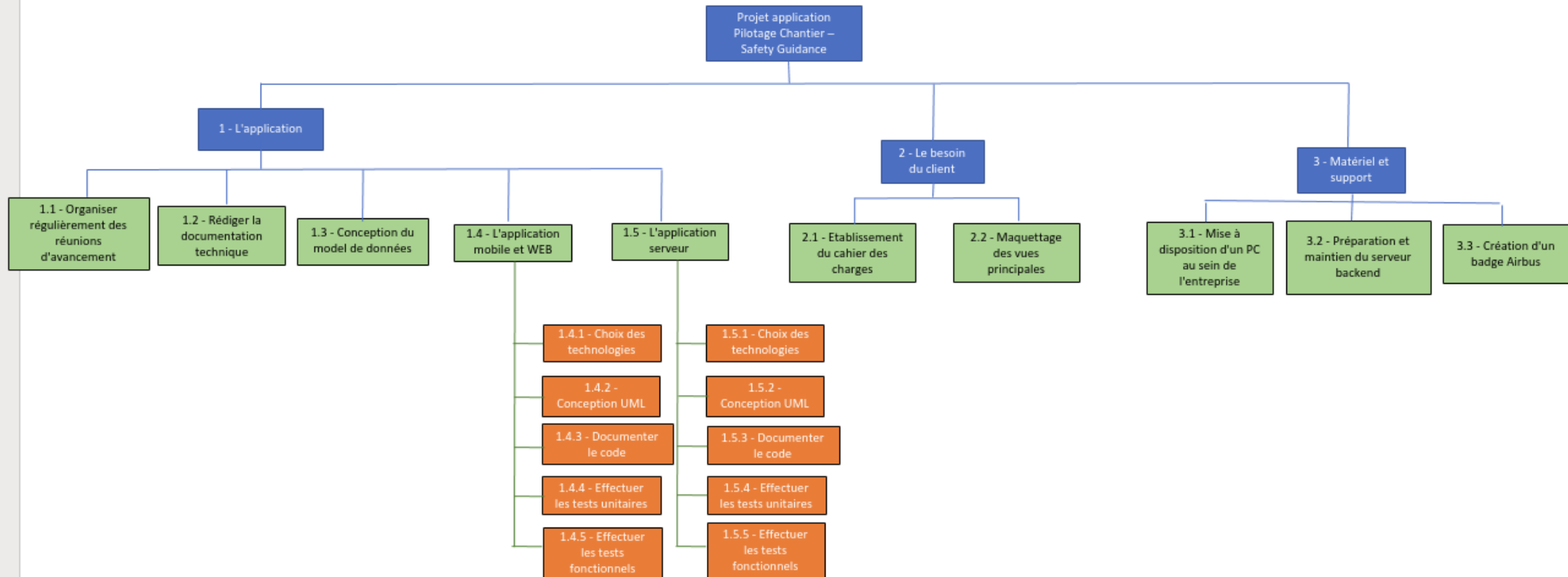
- Les diagrammes UML et MERISE concernant la conception
- La documentation technique permettant le maintien, l'évolution et le déploiement de l'application
- L'application qui sera constituée de 3 parties (voir schéma ci-dessous) :
 - Une version mobile de l'application pour tablette tactile.
 - Une version site internet hébergée sur un serveur distant.

- L'application serveur (backend) qui permettra la gestion et la mise à disposition de la banque de données.



Phase II – PLANIFICATION

Étape 1 – Déclinaison de la démarche de travail WBS



Étape 2 – Matrice des responsabilités

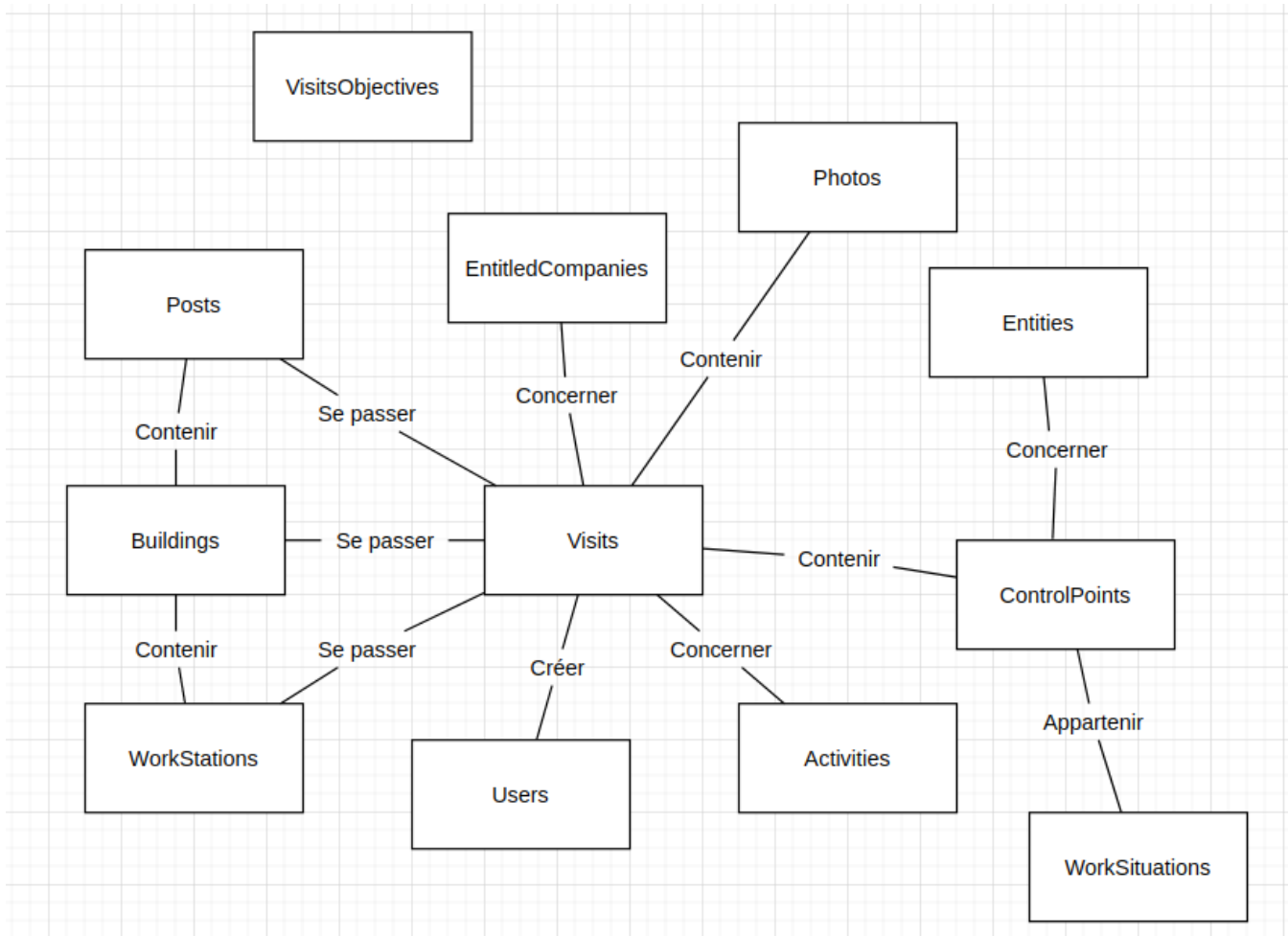
	Développeur	Ingénieur sécurité	Responsable de site	Responsable IT Agence Toulouse	Service IT Siege social Trigo Qualitaire
1.1 - Organiser régulièrement des réunions d'avancement	Prépare et anime la réunion	Participe	Organise et planifie la réunion d'avancement	Participe	
1.2 - Rédiger la documentation technique	Rédige			Utilise	
1.3 - Conception du modèle de données	Produit				
1.4.1 - Choix des technologies pour l'application WEB	Choisi les technologies en fonctions des besoins et du support physique disponible		Valide les choix	Valide les choix	
1.4.2 - Conception UML application WEB	Produit et propose des solutions architecturales			Valide les propositions	
1.4.3 - Documenter le code de l'application WEB	Produit		Valide le contenu dans les grandes lignes	Valide d'un œil plus technique	
1.4.4 - Effectuer les tests unitaires de l'application WEB	Effectue tous les tests avant mise en production				
1.4.5 - Effectuer les tests fonctionnels de l'application WEB		Effectue les tests fonctionnels sur le terrain en situation réelle			
1.5.1 - Choix des technologies pour le serveur	Produit et propose des solutions		Valide les choix	Valide les choix	
1.5.2 - Conception UML pour le serveur	Produit et propose des solutions architecturales			Valide les propositions	

	Développeur	Ingénieur sécurité	Responsable de site	Responsable IT Agence Toulouse	Service IT Siege social Trigo Qualitaire
1.5.3 - Documenter le code pour le serveur	Produit		Valide le contenu dans les grandes lignes	Valide d'un œil plus technique	
1.5.4 - Effectuer les tests unitaires pour le serveur	Effectue tous les tests avant mise en production				
1.5.5 - Effectuer les tests fonctionnels pour le serveur		Effectue les tests fonctionnels sur le terrain en situation réelle			
2.1 - Etablissement du cahier des charges	Rédige le cahier des charges en fonction des besoins et des contraintes exprimées par le client	Exprime le besoin métier	Exprime les objectifs globaux de l'application	Enonce les contraintes technologiques , physiques et légales que doit respecter l'application	
2.2 - Maquettage des vues principales	Réalise et propose des interfaces	Valide les propositions		Valide la charte graphique	
3.1 - Mise à disposition d'un PC au sein de l'entreprise			Attribue du matériel informatique pour le bon déroulement de la mission du développeur		
3.2 - Préparation et maintien du serveur backend				Effectue la demande au sein du service IT du siège	Prépare et ouvre les accès à distance vers le serveur physique. Maintient les accès et le bon fonctionnement du serveur
3.3 - Création d'un badge Airbus			Demande de badge à Airbus		

Étape 3 – Élaboration du calendrier POINTS DE FONCTIONS, COCOMO, GANTT

Envergure du projet – Points de fonctions

Ci-dessous, le modèle de domaine simplifié qui nous permettra de dénombrer les différents groupes de données :



Identification des groupes de données internes et externes :

Groupe de données externes (GDE)	Groupe de données internes (GDI)
Users (faible) Activities (faible) ControlPoints (faible) WorkSituations (faible) Entities (faible)	Posts (faible) Buildings (faible) WorkStations (faible) VisitsObjectives (faible) EntitledCompanies (faible) Photos (moyenne) Visits (élevée)

Dénombrement des entrées (ENT)

- 3 pages permettant d'éditer une visite. 1 à faible complexité et 2 à complexité moyenne
- 1 écran d'ajout d'éléments (bâtiment, FAL, entreprise, etc...). Complexité faible
- 1 page permettant de prendre des photos ou d'attacher une photo à une visite. Complexité moyenne

Dénombrement des sorties (SORT)

- 1 écran d'accueil affichant la liste des visites non synchronisées avec le serveur + une liste d'actions à date de péremption proche ou dépassée + actions non définies. Complexité moyenne
- 1 écran pour voir l'état de la mise à jour, la version de l'application, la date de dernière synchronisation et l'état de la liaison entre l'application et le serveur. Complexité faible

Dénombrement des interrogations (INT)

- 1 écran permettant de demander la génération des graphiques et tableaux. Complexité faible
- 1 écran de consultation de son profil utilisateur, complexité faible.
- 1 écran pour changer son mot de passe, complexité faible.
- 1 écran permettant de charger une liste de visites en fonction de filtres, complexité faible
- 1 écran permettant de consulter le contenu d'une visite en détail, complexité faible

APPLICATION MOBILE :

Entité	Complexité : Nb de composants	Poids	Nb de Pts fonction
GDI	Faible : 5	7	35
	Moyen : 1	10	10
	Elevé : 1	15	15
GDE	Faible : 5	5	25
	Moyen : 0	7	0
	Elevé : 0	10	0
ENT	Faible : 2	3	6
	Moyen : 3	4	12
	Elevé : 0	6	0
SORT	Faible : 1	4	4
	Moyen : 1	5	5
	Elevé : 0	7	0
INT	Faible : 5	3	15
	Moyen : 0	4	0
	Elevé : 0	6	0
PFB	96		

Facteur d'ajustement	Valeur
F1 - Télécommunication	3
F2 – Traitement distribué	1
F3 – Performance requise	3
F4 – Configuration saturée	1
F5 – Taux de transaction	2
F6 – Saisie interactive	3
F7 – Efficacité utilisateur	4
F8 – m-a-j interactive	3
F9 – Traitement complexe	2
F10 – Réutilisabilité	4
F11 – Facilité d'installation	5
F12 – Facilité d'exploitation	5
F13 – Plusieurs sites	1
F14 – Configurable	2
Total Di (DiT)	39

Facteur d'ajustement $FA = 0.65 + (0.01 * DiT) = 1.04$

$PFA = FA * PFB = 1.04 * 96 = 99.84$

$ISL = 118.7 * PFA - 6490 = 118.7 * 99.84 - 6490 = 5361$

CONCLUSION : Il faut environ 5 361 lignes de code pour l'application mobile.

SERVEUR :

Identification des groupes de données internes et externes :

Groupe de données externes (GDE)	Groupe de données internes (GDI)
Aucun, la BDD fait partie du système	Users (faible) Activities (faible) ControlPoints (faible) WorkSituations (faible) Entities (faible) Posts (faible) Buildings (faible) WorkStations (faible) VisitsObjectives (faible) EntitledCompanies (faible) Photos (moyenne) Visits (elevée)

Dénombrement des entrées (ENT)

- Pas d'entrée utilisateur

Dénombrement des sorties (SORT)

- Pas d'IHM, donc pas de sortie

Dénombrement des interrogations (INT)

- Service de génération de 22 graphiques et tableaux (14 à complexité moyenne et 8 complexité élevée)
- Enregistrement d'une visite. Complexité élevée
- Sauvegarde d'une photo. Complexité moyenne
- Sauvegarde d'un élément (FAL, bâtiment, entreprise, etc.). Complexité faible
- Changement de mot de passe. Complexité faible.
- Récupération de la liste des visites en fonction de filtres. Complexité moyenne
- Récupération des photos d'une visite. Complexité faible.
- Récupération des informations de pense bête (action à date limite proche, actions non définies, etc.). Complexité faible
- Récupération d'une mise à jour. Complexité faible
- Connexion d'un utilisateur en session. Complexité faible.

Entité	Complexité : Nb de composants	Poids	Nb de Pts fonction
GDI	Faible : 10 Moyen : 1 Elevé : 1	7 10 15	70 10 15
GDE	Faible : 0 Moyen : 0 Elevé : 0	5 7 10	0 0 0
ENT	Faible : 0 Moyen : 0 Elevé : 0	3 4 6	0 0 0
SORT	Faible : 0 Moyen : 0 Elevé : 0	4 5 7	0 0 0
INT	Faible : 6 Moyen : 16 Elevé : 9	3 4 6	18 64 54
PFB	231		

Facteur d'ajustement	Valeur
F1 - Télécommunication	2
F2 – Traitement distribué	1
F3 – Performance requise	3
F4 – Configuration saturée	1
F5 – Taux de transaction	2
F6 – Saisie interactive	1
F7 – Efficacité utilisateur	1
F8 – m-a-j interactive	2
F9 – Traitement complexe	4
F10 – Réutilisabilité	1
F11 – Facilité d'installation	5
F12 – Facilité d'exploitation	5
F13 – Plusieurs sites	1
F14 – Configurable	4
Total Di (DiT)	33

Facteur d'ajustement $FA = 0.65 + (0.01 * DiT) = 0.98$

$PFA = FA * PFB = 0.98 * 231 = 226.38$

$ISL = 118.7 * PFA - 6490 = 118.7 * 226.38 - 6490 = 20381.3$

CONCLUSION : Il faut environ 20 381 lignes de code pour l'application serveur.

COCOMO

Définition de la complexité du projet :

Grâce à la méthode "points de fonctions" on a pu estimer que le projet global ferait environ 25 742 lignes de code (5361 + 20381) soit 25 Kisl.

De plus, il n'y aura qu'un seul développeur, qui connaît les technologies qui vont être employées pour le projet. Au vu de ces éléments, on peut affirmer que le projet sera de type **organique**.

COCOMO simplifié :

Charge de travail (C) exprimé en homme mois (HM). $C = a (Kisl)^b$		
Type de projet	Charge de travail (C)	Durée en mois (TDEV)
Organique (notre cas)	$C = 3.2(Kisl)^{1.05}$	$TDEV = 2.5 (C)^{0.38}$
Semi-détaché	$C = 3(Kisl)^{1.12}$	$TDEV = 2.5 (C)^{0.35}$
Embarqué	$C = 2.8(Kisl)^{1.20}$	$TDEV = 2.5 (C)^{0.32}$

Calcul de la charge brute :

$C = a (Kisl)^b = 3.2 (25)^{1.05} = 94 \text{ HM}$

COCOMO Détaillé :

Les facteurs d'ajustement :

Facteurs de productivité	Evaluation					
	Très bas	Bas	Nominal	Haut	Très haut	Extrêmement haut
Attributs du produit						
RELY	.75	.88	1.00	1.15	1.40	
DATA		.94	1.00	1.08	1.16	
CPLX	.70	.85	1.00	1.15	1.30	1.65
Attributs de l'ordinateur						
TIME			1.00	1.11	1.30	1.66
STOR			1.00	1.06	1.21	1.56
VIRT		0.87	1.00	1.15	1.30	
TURN		0.87	1.00	1.07	1.15	
Attributs du personnel						
ACAP	1.46	1.19	1.00	.86	.71	
AEXP	1.29	1.13	1.00	.91	.82	
PCAP	1.42	1.17	1.00	.86	.70	
VEXP	1.21	1.10	1.00	.90		
LEXP	1.14	1.07	1.00	.95		
Attributs du projet						
MODP	1.24	1.10	1.00	.91	.82	
TOOL	1.24	1.10	1.00	.91	.83	
SCED	1.23	1.08	1.00	1.04	1.10	

Calcul de la charge nette (C'):

$$C' = C * \text{fact1} * \text{fact2} * \dots * \text{fact15} = 94 * 1.4 * 0.94 * 0.71 * 1.17 * 0.90 * 0.95 * 0.82 * 0.83 = 59.79 \text{ HM}$$

$$\text{TDEV} = 2.5 (C')^{0.38} = 2.5 (59.79)^{0.38} = 11.8 \text{ mois}$$

$$\text{MOE} = C'/\text{TDEV} = 59.79/11.8 = 5 \text{ Développeurs}$$

Conclusion : La taille idéale de l'équipe serait de 5 développeurs et le projet aurait une durée normale d'environ 11 mois et demi.

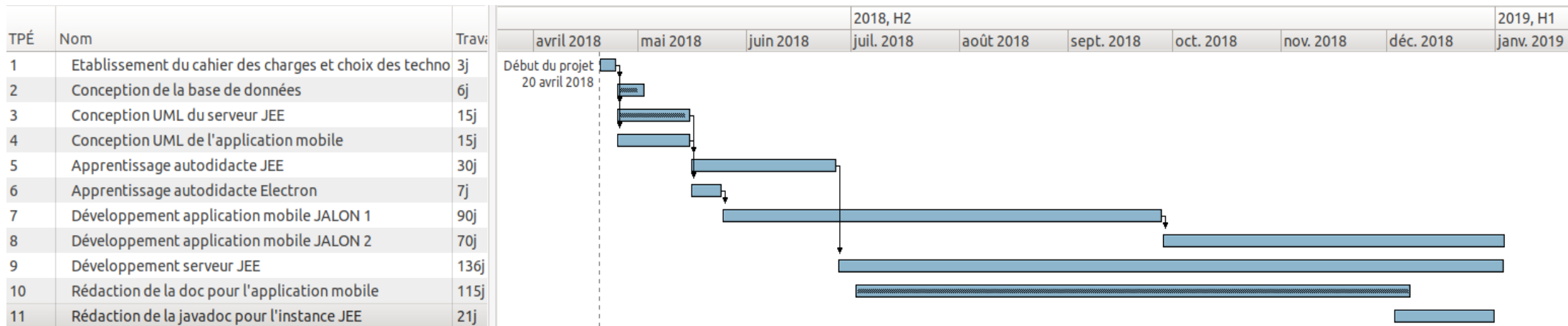
GANTT

Sur la page suivante, se trouve un diagramme de Gantt regroupant les principales étapes de conception et de développement de l'application. Le planning a été décidé en commun accord avec mon tuteur de stage comme suit :

Le premier Jalon de l'application devait rassembler les fonctionnalités principales et devait fonctionner dans un premier temps hors ligne, donc pas de serveur.

Le deuxième Jalon consistait à finir l'ensemble des fonctionnalités attendues et à avoir un serveur permettant à l'application de pouvoir sauvegarder les données en ligne.

Il était convenu de faire avec mon tuteur de stage une réunion hebdomadaire afin d'étudier l'avancement du projet et de lever les points bloquants.



Étape 4 – Budget prévisionnel

Salaire développeur débutant : 22 500 euros brut annuel soit, 1875 euros brute par mois soit, 93.75 euros brut par jour.

Salaire d'un ingénieur concepteur débutant : 49 000 euros brute annuel soit, 4083 euros brut par mois soit, 204 euros brut par jour.

Pour un projet de cette taille, un seul développeur débutant et un seul concepteur suffiraient. Le concepteur s'occuperait de toute la partie conception et de la rédaction de la documentation technique, tandis que le développeur s'occuperait du code et des tests unitaires.

Ci-dessous le coût total qu'aurait coûté le projet s'il avait été payant :

N°	Nom	Prix unitaire (HT)	Quantité (jour)	Prix (HT)
1	Conception de la BDD	204	6	1224
2	Conception UML du serveur JEE	204	15	3060
3	Conception UML de l'application mobile	204	15	3060
4	Développement de l'application mobile + tests	93.75	160	15000
5	Développement du serveur JEE + tests	93.75	136	12750
6	Rédaction de la documentation technique	204	5	1020
			TOTAL (HT)	36114
			TVA (20%)	7222.8
			TOTAL (TTC)	43336.8

Enfin, pour pouvoir générer du bénéfice, on pourrait décider d'appliquer 30% de plus sur les prix unitaires.

Ainsi un développeur débutant coûterait 121.87 euros brut par jour de travail et un concepteur 262.2 euros brut par jour de travail :

N°	Nom	Prix unitaire (HT)	Quantité (jour)	Prix (HT)
1	Conception de la BDD	262.2	6	1573
2	Conception UML du serveur JEE	262.2	15	3933
3	Conception UML de l'application mobile	262.2	15	3933
4	Développement de l'application mobile + tests	121.87	160	19499,2
5	Développement du serveur JEE + tests	121.87	136	16574,32
6	Rédaction de la documentation technique	262.2	5	1311
			TOTAL (HT)	46823.52
			TVA (20%)	9364,704
			TOTAL (TTC)	56188.22

Étape 5 – Évaluation des risques - Méthode AMDEC

LA POLITIQUE DE GESTION DES RISQUES :

NOTE : Trigo Qualitaire ne souhaitant pas que j'utilise le document officiel de leur politique de gestion des risques, les tableaux ci-dessous ont été refait de manière fictive afin de pouvoir appliquer la méthode AMDEC.

NOTE : Trigo Qualitaire n'utilise pas la notion de déterminabilité dans leur gestion des risques.

Echelle de probabilité :

Niveau	Probabilité
1	Quasiment nulle : évènement jamais survenu
2	Rare : évènement déjà survenu dans moins de – de 10% des projets similaires
3	Fréquent : évènement déjà survenu au cours de plusieurs projets similaires (entre 10% et 80 %)
4	Quasi systématique : évènement survenu dans plus de 80% des projets similaires

Echelle de gravité :

Niveau	Conséquences		
	Délais	Coûts	Qualité
1 Insignifiantes	Pas de retard notable	Pas de conséquence financière notable	Pas de conséquence sur l'application
2 Faible	Retard du projet inférieur à 2 semaines	Coûts supplémentaires ou perte < 1000 euros	Conséquences sur les fonctions optionnelles de l'application
3 Importantes	Retard du projet entre 2 semaines et 2 mois	Coûts supplémentaires ou perte entre 1000 et 10000 euros	NA
4 Catastrophiques	Retard du projet supérieur à 2 mois	Coûts supplémentaires ou perte supérieure à 10000 euros	Conséquences sur les fonctions essentielles de l'application

Tableau de cotation de la criticité :

	Niveau de probabilité				
		1	2	3	4
Niveau de gravité	1	1	2	3	4
	2	2	4	6	8
	3	3	6	9	12
	4	4	8	12	16

Ligne de conduite :

Gravité entre 1 et 3 : criticité faible -> action facultative

Gravité entre 4 et 8 : criticité moyenne -> action à mettre en place impérativement

Gravité entre 9 et 16 : criticité élevée -> action prioritaire, à mettre en place impérativement

L'ANALYSE DE RISQUE :

ENGAGEMENT OU RISQUES CONTRACTUELS						
Risques et Facteur de risques	Probabilité	Gravité	Criticité	Impact potentiel	Nature du risque	Action de réduction du risque
<u>Facteur de risque</u> : Beaucoup de fonctionnalités différentes demandées par le client, dans un délai court <u>Risque</u> : Travail désorganisé, fonctionnalités essentielles mal identifiées et pas toutes réalisées	2	4	8	Prestation non renouvelée (perdue lors du prochain appel d'offre)	Qualité	Etablissement de jalons pour prioriser les fonctionnalités essentielles.
<u>Facteur de risque</u> : Absence d'accompagnement au changement <u>Risque</u> : Rejet de l'application par les utilisateurs	3	1	3	Application moins utilisée, donc moins efficace	Qualité	Ingénieur HSE formé par le développeur ; client (Airbus) formé à son tour par l'ingénieur HSE.
RESSOURCE						
Risques et Facteur de risques	Probabilité	Gravité	Criticité	Impacts	Nature du risque	Action de réduction du risque
<u>Facteur de risque</u> : Développeur débutant <u>Risque</u> : Ne pas livrer l'application dans les délais.	2	3	6	Prestation non renouvelée (perdue lors du prochain appel d'offre)	Délai	Suivre des formations en ligne concernant les technologies employées pour le projet.
<u>Facteur de risque</u> : Développeur débutant <u>Risque</u> : application qui perd des données ou qui génère des indicateurs erronés	2	4	8	Mauvaise gestion du pilotage des chantier. Insatisfaction client, risque d'accident accrue.	Qualité	Effectuer des tests fonctionnels et unitaires poussés. Faire valider régulièrement les résultats par l'ingénieur HSE.
<u>Facteur de risque</u> : Maladie ou indisponibilité du développeur. <u>Risque</u> : Ne pas livrer l'application dans les délais.	2	2	4	Prestation non renouvelée (perdue lors du prochain appel d'offre)	Délai	Etablissement de jalon pour prioriser les fonctionnalités les plus essentielles.
PRODUIT ET TECHNOLOGIE						
Risques et Facteur de risques	Probabilité	Gravité	criticité	Impacts	Nature du risque	Action de réduction du risque
<u>Facteur de risque</u> : Code mal ou peu documenté <u>Risque</u> : Maintenance et évolution de l'application difficile et longue	3	3	9	Perte de temps et d'agent lors de la maintenance et des évolutions. Démotivant pour les développeurs.	Délai Coût	Respect des standards et des bonnes pratiques. Prévoir du temps pour la rédaction de la documentation technique.

Phase III – RÉALISATION

1 – Méthodes de travail

Trigo Qualitaire n'étant pas une entreprise informatique, je n'avais pas de chef de projet à proprement parler. Mon tuteur était là pour vérifier que l'avancement de l'application correspondait bien au besoin. Le responsable IT quant à lui assurait uniquement le lien entre le service IT du siège et moi-même, afin de répondre à mes besoins liés à l'hébergement et au déploiement de la partie backend de l'application.

Ainsi j'étais libre sur la façon de travailler : J'ai utilisé GIT comme gestionnaire de version ainsi que l'hébergeur GitHub pour la sauvegarde en ligne du projet.

Compte tenu de la contrainte temps j'ai décidé de mettre en place des mini-sprints (méthodes agiles) pour chaque fonctionnalité majeure. Cela m'a permis de me concentrer sur les fonctionnalités les plus importantes et incontournables dès le début, et de garder les fonctionnalités facultatives pour la fin en bonus si du temps était disponible.

Toujours à cause du peu de temps, de backup et d'expérience que j'avais, je n'ai pas pris le temps de mettre en place des tests unitaires pour tous les modules ni de pipelines d'intégration continue.

2 – Le périmètre du projet

Il est prévu que l'application soit testée et utilisée sur la mission Pilotage Chantiers à Toulouse. Mais il n'est pas exclu qu'elle puisse être aussi utilisée plus tard sur une autre prestation du groupe Trigo.

C'est donc dans cette hypothèse et pour adopter de bonnes habitudes que j'ai commenté et documenté mon code source entièrement en anglais.

3 – Le type d'application

D'une part, l'application devra être compilée sous deux formes, une sous la forme d'application mobile (tablette tactile) et l'autre sous la forme d'une application WEB (destinée aux navigateur internet).

D'autre part, c'est une application de type client-serveur, il y aura donc une application de type "serveur" (backend) qui assurera la mise à disposition des données persistantes et rendra un certain nombre de services.

4 – Les utilisateurs

Il y a en tout deux utilisateurs principaux pour l'application : l'ingénieur sécurité qui sera l'opérateur et qui alimentera l'application de données, et Airbus qui consultera les données renseignées par l'opérateur, ce qui lui permettra de "piloter" (superviser) ses sous-traitants.

5 – Le matériel et le support

Tel que déjà mentionné dans la phase I, étant donné la nature mobile du poste qu'occupe l'opérateur, l'interface permettant d'entrer de nouvelles données doit fonctionner sur une tablette tactile HP Elite masterisée Airbus tournant sous Windows 10.

Concernant le portail d'accès, ce sera une application web (site internet) classique fonctionnant sur n'importe quel navigateur web. L'opérateur aura également accès à cette application web qui lui fournira strictement la même interface que l'application mobile, alors que pour le superviseur seules les pages permettant de consulter les données et les indicateurs seront disponibles.

6 – Contraintes techniques

Le seul besoin technique de l'application est un serveur mis à disposition et entretenu par Trigo Qualitaire, sur lequel la base de données PostgreSQL et l'instance JEE tourneront.

Etant donné que l'application est de petite envergure, aucun Framework n'a été utilisé.

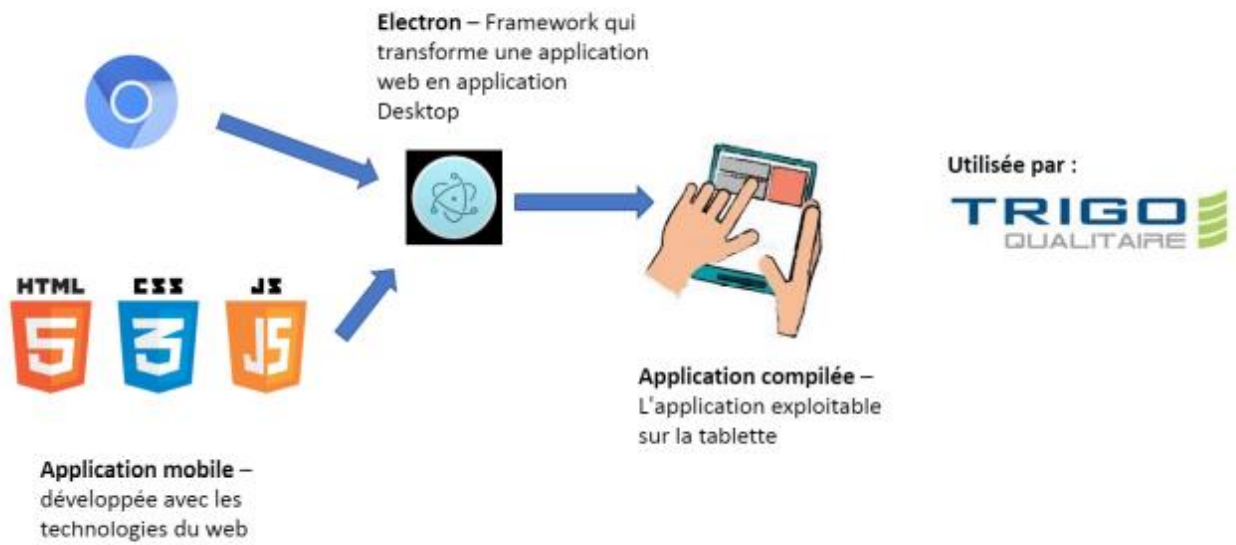
Bootstrap a été utilisé mais uniquement pour le design des formulaires, des boutons, des barres de progression et pour la grille de positionnement responsive.

7 – Le choix des technologies

Pour l'application mobile, aucune technologie n'a été imposée. Mon choix s'est donc porté sur les technologies du web. En effet elles permettent de mettre en place simplement des interfaces graphiques esthétiques, et offrent la possibilité de factoriser le code source de l'application mobile et du portail client. Cela divise par 2 le travail.

En revanche, pour le serveur, Trigo Qualitaire m'a contraint à utiliser une base de données PostgreSQL et une instance Java EE. Le choix de Trigo Qualitaire est motivé par la volonté d'uniformiser ce projet avec un autre projet déjà en cours de développement.

Moteur web Chromium –
utilisé par Electron



Portail client – développée avec les technologies du web



8 – Graphisme et ergonomie

a – La charte graphique

Aucune exigence n'a pour le moment été définie. Mais cela sera fait par la suite si le client le demande. Comme je lui ai expliqué, il n'y aura aucune difficulté à modifier les couleurs et l'organisation des éléments d'une page par le biais des feuilles de style CSS. Néanmoins, j'ai pris le soin de récupérer les codes couleurs utilisés sur le site web du groupe Trigo (<https://www.trigo-group.com/>) afin de pouvoir proposer quelque chose de pertinent dès le début.

b – Wireframe et maquettage

Les interfaces ont été réalisées et pensées intégralement par moi-même avec cependant l'aide et les conseils précieux de mes professeurs à l'Adrar que je remercie encore. Elles ont par la suite été vérifiées, testées et approuvées par l'ingénieur sécurité de Trigo Qualitaire.

En **annexe** se trouve quelques mockup réalisés à l'aide de l'application WEB (<https://www.draw.io>) qui m'ont permis d'appréhender et de faire confirmer le besoin métier auprès du client.

c – Les logos et les symboles

Les logos et symboles composant les interfaces de l'application sont soit récupérés sur le net et libre de droit, soit générés par mes soins à l'aide du logiciel libre **InkScape**.

Concernant les logos représentant les "situations de travail", ils ont été fournis par Trigo Qualitaire et sont paramétrables plus tard à convenance.

Phase IV – CONCLUSION

Le projet s'est bien déroulé. Toutes les fonctionnalités attendues ont pu être implémentées, ainsi que celles qui étaient optionnelles. L'application tourne actuellement sur Airbus et permet à l'ingénieur sécurité de Trigo Qualitaire d'effectuer son travail plus rapidement et plus efficacement. Ainsi il peut se concentrer d'avantages sur les missions de formation et de prévention.

Une documentation technique d'une cinquantaine de pages a été rédigée afin de permettre l'évolution et le maintien de l'application par une personne autre que moi.

Ce projet a été très formateur : j'ai pu appréhender le cycle de vie complet d'un projet informatique, j'ai appris à développer en utilisant des langages modernes et des techniques tels que les patrons de conception MVC, Factory et DAO.

Je suis fier de ce que j'ai accompli. Cette application constitue l'une de mes meilleures expériences professionnelles dans le domaine du développement.

Ci-dessous quelques captures d'écran de l'application mobile (version 2.0). A présent elle est en version 2.8 et les interfaces ont été un peu améliorées.

The first screenshot shows the 'Enregistrement' (Registration) screen. It features a top navigation bar with 'Accueil', 'Enregistrement', 'Consultation', and 'Paramétrage'. Below the bar, there are fields for 'LIEUX' (Location) including 'FAL', 'A319', 'Bâtiment', 'B21', 'Poste', 'P13', 'Activité', and 'Maintenance'. To the right, there are 'AUTRES INFOS' (Other Info) fields for 'Désignation de la visite', 'Entreprise titulaire', 'Choisir une entreprise', 'Entreprise réalisant les travaux', 'Numéro de PI', and 'Type de PI'. A 'Suivant' (Next) button is at the bottom right.

The second screenshot shows the 'Consultation' screen, titled 'Points de contrôle pour : Travail en coactivité'. It displays a table with columns: 'Point de contrôle', 'C', 'NC', 'NCM', 'Sous situation', 'Entité responsable', 'Commentaire', and 'Action'. The table lists several control points such as 'Signature du DAI', 'Respect du balisage / des affichages liés à des risques de travail en coactivité', 'Signature du PI', 'Balisage des zones à risques', and 'Balisage des zones à risques hors d'usage'. At the bottom, there are 'Annuler' (Cancel) and 'Valider' (Validate) buttons.

The third screenshot shows the 'Paramétrage' (Configuration) screen. It features a top navigation bar with 'Accueil', 'Enregistrement', 'Consultation', and 'Paramétrage'. Below the bar, there are icons for various work environments: 'Generalités', 'Travail en environnement à risques', 'Ambiance de travail', 'Travail en hauteur', 'Travaux électriques', 'Manutention', 'Travail en coactivité', and 'Manutention manuelle'. At the bottom, there are 'Précédent' (Previous) and 'Fin de visite' (End of visit) buttons.

ANNEXES

La page de connexion :

Page de login classique qui offrira également la possibilité de récupérer un mot de passe oublié.



The image shows a login page mockup with a light gray grid background. At the top right, there is a red button labeled "Quitter". In the center, the text "LOGO TRIGO QUALITAIRE" is displayed in large, bold, black capital letters. Below this, there are two input fields. The first is labeled "Identifiant" and contains the text "Line 1". The second is labeled "Mot de passe" and contains seven asterisks "*****". At the bottom left, there is a blue button labeled "Se connecter". To the right of this button is a blue hyperlink labeled "Mot de passe oublié ?".

La page d'accueil :

Page principale listant toutes les visites en cours et/ou non synchronisées avec la base de données en ligne.

[illegible]

La page d'enregistrement 1/3 :

Première page d'enregistrement d'une visite. Elle permet de renseigner les informations générales et d'identification de la visite.

Accueil	Enregistrement	Consultation	Indicateur
---------	----------------	--------------	------------

LIEUX	AUTRES INFOS
FAL A330	Désignation de la visite Ma visite
Bâtiment C03	Entreprise titulaire A330
Poste P35A	Entreprise réalisant les travaux Ma visite
Activité Maintenance	Numéro de PI Ma visite
Type de chantier <input type="radio"/> 1 <input type="radio"/> 2	Type de PI <input type="radio"/> digital <input type="radio"/> papier

Suivant

La page d'enregistrement 2/3 :

Deuxième page d'enregistrement de la visite. Met à disposition toutes les situations de travail pouvant être rencontrées sur une FAL.

De plus un petit bouton d'appareil photo en bas à gauche ouvrira un panneau latéral depuis lequel l'opérateur pourra ajouter une image depuis la mémoire de la tablette ou prendre directement des photos avec la caméra de la tablette.

Accueil Enregistrement Consultation Indicateur

Titre de la visite : Ma visite

Situation de travail Situation de travail Situation de travail

Situation de travail Situation de travail Situation de travail

<- précédent Fin de visite

La page d'enregistrement 3/3 :

Troisième page d'enregistrement de la visite. Celle-ci apparaît lorsque l'on clique sur une situation de travail.

Elle met à disposition tous les points de contrôle associés à la situation de travail sélectionnée.

Un point de contrôle peut être coché "C" (conforme), "NC" (non conforme), "NCM" (non-conformité majeure) ou bien laissé décoché si l'opérateur n'a pas vérifié ce point de contrôle.

L'opérateur peut ajouter un commentaire à un point de contrôle et sélectionner l'entité responsable.

Lors d'une non-conformité ou d'une non-conformité majeure, l'opérateur peut définir une action corrective par le biais d'un panneau latéral en cliquant sur le bouton présent dans la colonne "action".

Accueil

Enregistrement

Consultation

Indicateur

Points de contrôle pour : situation de travail

Point de contrôle	C	NC	NCM	Entité responsable	Commentaire	Action
Travail par point chaud	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	EE	<input type="text"/>	
XXXXXXXXXX	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	EE	<input type="text"/>	
XXXXXXXXXX	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	EE	<input type="text"/>	
XXXXXXXXXX	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	EE	<input type="text"/>	
XXXXXXXXXX	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	EE	<input type="text"/>	
XXXXXXXXXX	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	EE	<input type="text"/>	
XXXXXXXXXX	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	EE	<input type="text"/>	
XXXXXXXXXX	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	EE	<input type="text"/>	
XXXXXXXXXX	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	EE	<input type="text"/>	

<- précédent

Fin de visite

La page des indicateurs :

Cette page permet d'afficher les nombreux graphiques et tableaux générés de manière dynamique et automatisée par rapport aux informations de visites entrées en base de données.

