



Consignes projet fil rouge CDA 2025



TABLE DES MATIERES

<i>Section 1 Généralités.....</i>	<i>3</i>
1.1 But du document.....	3
1.2 Constitution de l'équipe.....	3
1.3 Les livrables du projet.....	3
<i>Section 2 Analyse du cahier des charges.....</i>	<i>6</i>
2.1 Comprendre les besoins du client.....	6
<i>Section 3 La gestion de projet.....</i>	<i>7</i>
<i>Section 4 Spécification fonctionnelle.....</i>	<i>8</i>
<i>Section 5 Spécification technique ou conception.....</i>	<i>9</i>
<i>Section 6 Démarche DEVOPS.....</i>	<i>10</i>



Section 1 Généralités

1.1 But du document

Ce document contient des consignes pour le projet fil rouge qui sera réalisé au fil des semaines par les étudiants de la promotion CDA 2025.

Il accompagne le cahier des charges d'un projet fictif demandé par un client fictif « Minot'Or ».

Le but de ce projet fil rouge est de proposer une réalisation couvrant l'ensemble des items requis de la formation CDA.

Ce qui explique son volume important.

Ce qui suit propose une démarche à suivre pour être efficace dans la réalisation des applications, en accord avec le calendrier des formations Imie.

1.2 Constitution de l'équipe

Ce projet est le même pour l'ensemble de la promotion.

Des groupes de 3 personnes constituent les équipes.

Nous demandons à ce que le travail soit homogène et bien réparti dans chaque équipe.

Nous proposons – à chaque équipe de décider - qu'il y ait un leader, ou chef de projet, pendant une période de 3 à 4 mois. Chacun devrait être à un moment donné leader. C'est lui qui coordonne et met à jour le planning et qui est en interface avec le client (le client est représenté par le formateur)

Vous devez vous organiser pour travail en équipe mais à distance (Discord, GitLab ...)

1.3 Les livrables du projet

Vous êtes dans une formation CDA, Concepteur Développeur d'Applications.

La documentation que vous allez produire est capitale, elle va prouver – ou non – que vous êtes capables d'analyser un cahier des charges et de spécifier puis concevoir des applications.

La documentation contient un formalisme attendu : maquettage, diagrammes use case, planning, conception base de données, autres diagrammes UML

Dans chaque équipe chaque membre écrit son propre document de l'ensemble du projet (pour évaluation individuelle du jury CDA, dans un vrai projet il n'y a qu'un seul document par projet).

C'est un document de qualité professionnelle.



40 à 60 pages pour le corps du document, 40 pages max pour les annexes.

Je vous propose de suivre le plan type suivant fourni dans le document RCV2 de votre formation (page 6) :

- la liste des compétences CDA mises en oeuvre dans le cadre du projet ;
- le résumé du cahier des charges ou l'expression des besoins du projet ;
- la gestion de projet (planning et suivi, environnement humain, objectifs de qualité) ;
- les contraintes du projet et livrables attendus ;
- les spécifications fonctionnelles du projet élaboré par le candidat :
 - les maquettes et enchaînement des maquettes, le zoning ;
 - le diagramme du comportement des fonctionnalités de type cas d'utilisations ;
 - le diagramme du détail des cas d'utilisations les plus significatifs de type diagramme de séquence.
- les spécifications techniques du projet, élaborées par le candidat, y compris pour la sécurité
 - l'architecture logicielle du projet ;
 - le modèle entités-associations MCD de la base de données ;
 - le script de création ou de modification de la base de données ;

;

- les réalisations du candidat comportant les extraits de code les plus significatifs, les arguments qui ont conduit à ses choix, y compris pour la sécurité
- la présentation d'éléments de sécurité de l'application ;
- la présentation du plan de tests ;
- la présentation d'un jeu d'essai élaboré par le candidat de la fonctionnalité la plus représentative (données en entrée, données attendues, données obtenues) et analyse des écarts éventuels ;
- la description de la veille, effectuée par le candidat durant le projet, sur les vulnérabilités de sécurité, description des vulnérabilités éventuellement trouvées et des failles potentiellement corrigées.

Des annexes peuvent exister.

Vous exposerez lors de la présentation orale certains de ces éléments. Une présentation power point est à prévoir.

Vous allez dans le groupe être responsable chacun d'une partie différente. Dans le chapitre gestion de projet, vous présenterez l'équipe et la contribution de chacun.

Vous serez capable d'expliquer l'ensemble du projet, y compris ce que vous n'avez pas réalisé.

Je vous propose de commencer à réaliser en commun le plan type et de vous répartir des rédactions même très incomplètes.

Y placer les mots A CONFIRMER , A DEFINIR, A VERIFIER.... dans un texte en cours de rédaction dans lequel il y a des hypothèses à vérifier ensuite. Il est infiniment mieux de mettre qqe chose plutôt que de laisser un paragraphe vide jusqu'au dernier moment.



Il est IMPOSSIBLE de rédiger correctement un document du 1^{er} coup, et dans l'ordre de lecture. Rédigez ce que vous savez faire, dans l'ordre que vous souhaitez.



Section 2 Analyse du cahier des charges

2.1 Comprendre les besoins du client

Tout cahier des charges est imprécis et incomplet.

Le rédacteur décrit son point de vue par rapport à son métier, pas par rapport au vôtre.

C'est à vous de l'interpréter et de réfléchir à une solution qui corresponde à son besoin.

Relever les fonctionnalités attendues.

La démarche est de penser : quand cela sera fini voilà ce qu'il y aura et voilà à quoi cela ressemblera.

Ne pas penser à « comment cela va être fait » car beaucoup trop tôt.

Tout ceci va contribuer à imaginer et rédiger les spécifications fonctionnelles du projet.



Section 3 La gestion de projet

Il faut décrire la façon dont vous allez organiser le déroulement du projet.

Choix de la méthode de gestion de projet et pourquoi cette méthode. Comparez la à d'autres méthodes.

La répartition de l'équipe dans le projet. Qui fera quoi.

Un planning prévisionnel (un 1^{er} planning gantt) montrant les étapes et les noms.



Section 4 Spécification fonctionnelle

Cette partie peut être montrée au client pour obtenir son accord.

Il n'y a donc pas de description technique de comment va être faite l'application.

On s'intéresse ici à répondre aux questions

- Qui vont être les utilisateurs de l'application
- Pour quoi faire

La spécification doit contenir au minimum :

- des diagrammes UML use case. Ce diagramme doit être commenté. Exploiter ce lien : <https://www.cybermedian.com/fr/use-case-diagram-vs-use-case-specification/>
- des diagrammes UML de séquence s'il est nécessaire d'expliquer une chronologie entre uses cases
- une/des maquettes des interfaces graphiques, par exemple avec Figma, commencer par un zoning. Il y a 3 interfaces graphiques : front Web, mobile, desktop

Etant donné la complexité des échanges de données dans ce projet des diagrammes de flux genre JFLUX peuvent aider à la compréhension de gestion des données.



Section 5 Spécification technique ou conception

Grâce à la spécification fonctionnelle, on sait maintenant ce qu'il y a à réaliser.

On s'intéresse ici à concevoir COMMENT on va le réaliser.

Il faut montrer un schéma d'architecture de votre système, serveurs Web ou API, clients ...

Ce schéma est un dessin non normalisé (pas UML à priori)

Le schéma doit être commenté pour y voir vos choix de conception d'architecture.

C'est dans cette partie que vous allez décrire :

- La conception de la base de données relationnelle avec Merise
- La conception de la base de données NOSQL
- Les diagrammes de classes par exemple de la couche métier
- Si une API Web est prévue, la décrire par ses chemins et usages (REST). Un tableau décrit les routes (endpoints), méthodes GET, POST,DELETE ...
- diagramme de composants, de déploiement ...
- traitement de la sécurité

...

Ce document et cette partie **doivent vous être utile** (même indispensable) pour réaliser ensuite en autonomie les parties du projet. Y mettre donc tout ce qui lève des ambiguïtés.



Section 6 Démarche DEVOPS

Vous pouvez utiliser la démarche DevOps sur l'ensemble des composants de l'application (Tests Unitaires automatiques, Intégration continue, déploiement continu) ... mais sera volumineux vis à vis du temps dont vous disposez.

Plutôt se concentrer sur une partie et aller le plus loin possible. Si vous définissez une API Web, cette partie sera la plus facile et utile à gérer en DevOps. Vous disposerez alors en déploiement d'une API disponible pour les autres applications ...

Indiquez dans le document votre démarche et avec quels supports vous le faites (Docker, GitLab ...)