

Projet synthèse

420-C61-VM

420-C61-IN

Document de conception

Définition technique du projet

Table des matières

Description technique détaillée du projet	1
Introduction	3
Contextualisation	3
Rappel sur le niveau technique attendu du projet	4
Niveau de détail du document de conception	4
Éléments techniques à couvrir	4
1. Infrastructure de développement	4
2. Interface graphique utilisateur	5
3. Données persistantes	5
4. Structures de données	6
5. Patrons de conception	6
6. Développement d'une bibliothèque	7
7. Expression régulière	7
8. Algorithme	7
9. Mathématique	8
10. Conception UML	8
Structure du document à produire	10
Grille de correction	11

Introduction

Le document de conception technique représente l'une des parties les plus importantes du projet. Il suit la phase initiale de définition des objectifs et des spécifications fonctionnelles. Il est conçu pour traduire les besoins et les spécifications en un plan de travail technique détaillé, qui guidera l'équipe de développement à travers les phases d'implémentation. L'accent est mis sur le choix des technologies et outils appropriés, la détermination de l'architecture système, la définition des interfaces utilisateurs ainsi que des interfaces entre les composants logiciels et des constituants techniques du projet comme la conception des algorithmes clés. Ce document s'adresse à toutes les parties prenantes techniques impliquées dans le développement du projet.

L'objectif de ce document est de fournir une feuille de route claire et détaillée pour la réalisation technique du projet. En se basant sur les fondations établies par le document de spécifications fonctionnelles, il vise à éliminer les ambiguïtés, à préciser les responsabilités techniques et à assurer une compréhension commune du projet, de ses constituants et des pratiques à adopter. En définissant rigoureusement l'approche technique, ce document contribue à la cohérence, à l'efficacité et à la qualité du processus de développement, tout en facilitant l'anticipation et la gestion des défis techniques potentiels.

En outre, ce document joue un rôle essentiel dans le processus décisionnel concernant les choix et compromis techniques significatifs du projet. Il sert de cadre pour évaluer différentes options architecturales, sélectionner des technologies et définir des stratégies d'implémentation, tout en tenant compte des contraintes de performance, de sécurité, de maintenabilité et de coût. Une planification et une conceptualisation méticuleuses en amont permettent d'ancrer le processus de développement dans une stratégie cohérente, minimisant ainsi les réorientations et les nécessités de refonte constante qui peuvent survenir en l'absence d'une vision clairement définie dès le départ. Par conséquent, ce document n'est pas seulement un guide pour le développement, mais également un outil crucial pour naviguer dans le paysage complexe des choix techniques, assurant ainsi que le projet reste aligné sur ses objectifs tout en optimisant l'utilisation des ressources disponibles.

Contextualisation

Avant de plonger dans la mise en œuvre technique du projet, il est primordial de marquer une pause stratégique pour articuler clairement les approches et les orientations technologiques prévues. Cette phase préparatoire est fondamentale et, à travers une démarche de réflexion approfondie et de conceptualisation, vise à cristalliser la vision technique qui guidera le projet. Étant donné l'exigence d'atteindre un certain standard technique, une organisation méticuleuse est indispensable pour éviter un développement superficiel et viser la création d'une valeur réelle. Il est donc crucial de trouver un juste milieu entre les résultats immédiats et les bénéfices à long terme tels que la réutilisabilité, la facilité de maintenance, et les potentialités d'évolution du projet.

Bien que les objectifs de développement soient établis au préalable, ils restent flexibles pour s'adapter aux évolutions inhérentes au processus de création. Cette adaptabilité est essentielle pour aligner le produit final avec les exigences évolutives, les orientations stratégiques et les rétroactions, s'inscrivant pleinement dans la philosophie du développement agile. Les premières phases du projet doivent donc être consacrées à la consolidation de ces stratégies et à la préparation du terrain pour les étapes de développement qui suivront.

Le document conçu à cet effet doit offrir une analyse étayée et précise du projet, devenant ainsi le pilier technique sur lequel s'appuiera tout le développement. L'intégration de ce document comme une composante centrale de la méthodologie agile optimisera les chances de succès en fournissant un cadre solide et évolutif pour la réalisation des objectifs définis.

Rappel sur le niveau technique attendu du projet

Ce projet offre l'opportunité d'appliquer des principes de conception avancée et d'adopter une perspective de plus haut niveau en matière de développement logiciel. L'importance réside dans la qualité de l'architecture mise en place plutôt que dans le nombre de fonctionnalités différentes. L'objectif est de valoriser votre travail afin qu'il soit attrayant pour de futurs employeurs, mettant en avant des qualités telles que la modularité, la réutilisabilité, et la facilité de maintenance, qui sont hautement appréciées dans le domaine professionnel.

Vous êtes encouragés à approfondir les aspects techniques de votre projet en explorant des concepts de conception sophistiqués. Cela comprend une attention particulière à la gestion des données, à l'efficacité des structures de données et des algorithmes, à l'exploration de divers paradigmes de programmation y compris la programmation orientée objet et ses principes clés comme l'encapsulation, l'héritage, et le polymorphisme. L'utilisation de patrons de conception, la mise en œuvre d'architectures logicielles solides, les considérations d'ergonomie logicielle, et les approches de gestion de projet sont également des éléments essentiels. Même si certains projets ne se prêtent pas directement à une structure orientée objet, il est important d'intégrer des principes comparables adaptés au paradigme de programmation choisi.

Gardez en tête que le niveau d'abstraction mis de l'avant dès votre phase de conception sera l'un des points techniques évalués les plus importants.

Niveau de détail du document de conception

Éléments techniques à couvrir

IMPORTANT : Vous devez absolument vous référer à la section *Contraintes de contenu technique* du document *Éléments techniques* pour compléter les informations qui suivent. Vous y trouverez quelques détails supplémentaires dont il faut absolument tenir compte.

1. Infrastructure de développement

Vous devez présenter les éléments techniques fondamentaux associés au développement informatique de votre projet. Les points à couvrir sont :

- Plateforme(s) cible(s) : Windows, Android ou Web
- * Langage(s) de programmation sélectionné(s)
- * Bibliothèque(s) principale(s)

- Environnement(s) de développement intégré utilisé(s)
- Matériel si requis par le projet (autre que les ordinateurs de développement et d'utilisation)
- Divers

* Pour ces éléments, vous devez justifier votre choix par un court texte.

2. Interface graphique utilisateur

Vous devez réaliser toutes les maquettes de vos interfaces graphiques utilisateur.

- **Production des maquettes** : Élaborez les maquettes pour **toutes** vos interfaces graphiques utilisateur. Ces prototypes visuels doivent illustrer avec précision l'agencement et l'apparence des éléments de l'interface, tels que les menus, les boutons, les zones de texte ou multimédias.
- **Objectif des maquettes** : Utilisez les maquettes pour visualiser la structure, le design et l'organisation des composants de vos applications bureau, ou sites web. Elles doivent faciliter la compréhension du fonctionnement attendu et de l'expérience utilisateur avant le démarrage du développement logiciel.
- **Nombre de maquettes** : Déterminez le nombre nécessaire de maquettes en fonction du niveau des interactions avec les utilisateurs, pas la complexité du projet en tant que tel. Certains projets nécessitent plusieurs maquettes pour différentes interfaces (un site web transactionnel par exemple), tandis que d'autres projets sont plus simples et ne requièrent qu'une ou deux (un logiciel d'analyse technique par exemple). Chaque projet est différent à cet effet. Malgré tout, soyez conscient que le développement d'interface utilisateur est relativement simple techniquement, leur mise en place prend un temps substantiel.
- **Clarté et détail** : Assurez-vous que chaque maquette soit claire et détaillée, permettant une compréhension univoque de la conception prévue.
- **Qualité des maquettes** : Vous pouvez produire vos maquettes à l'aide d'un logiciel ou à main levée. Dans le cas où vous utilisez un logiciel, ce dernier doit être de type vectoriel. Si vous faites vos esquisses à main levée, assurez-vous qu'elles soient propres, claires et numérisées avec une résolution adéquate. On vous suggère l'un des logiciels suivant [figma](#) ou [balsamiq](#) (uniquement des suggestions).

3. Données persistantes

Votre projet doit utiliser des données persistantes en entrée et en sortie.

- **Sélection des technologies** : Choisissez les technologies adaptées pour le stockage des données persistantes. Prenez en compte les spécificités de votre projet pour faire ce choix. Vous devez documenter les technologies retenues et justifier votre choix.
- **Conception de la structure des données par la création d'un schéma technique** : Élaborez un schéma technique qui présente de façon détaillée les données persistantes, leur organisation et leurs interrelations.
- **Personnalisation et clarté du schéma** : Si vous utilisez une base de données relationnelle, la création d'un schéma relationnel va de soi. Toutefois, ce n'est pas tous les modèles de données qui possèdent

un diagramme standardisé. Dans ce cas, vous pouvez créer votre propre style de schéma, à condition qu'il soit clair, intuitif et, surtout, bien expliqué par une légende.

- **Détail des informations** : Fournissez des détails précis sur les types de données, leur regroupement (comme une table) et leurs relations (comme une clé étrangère). Assurez-vous que ces informations soient suffisamment claires pour permettre une compréhension approfondie de la gestion des données dans le système. La lecture du diagramme devrait rendre facile la compréhension des données qui transigent.

4. Structures de données

Vous devez faire la démonstration d'un niveau de compétence minimum concernant l'utilisation des structures de données. Les éléments suivants sont à considérer :

- **Sélection des structures de données** : Identifiez quatre structures de données distinctes qui seront utilisées dans votre projet. Assurez-vous que chaque structure est bien adaptée aux besoins spécifiques du projet.
- **Documentation et justification** : Pour chaque structure de données sélectionnée
 - Documentez le contexte spécifique dans lequel elle sera appliquée.
 - Bien que les structures de données puissent être utilisées à divers endroits dans le projet, décrivez une instance d'utilisation particulièrement pertinente de son usage. Cette description doit mettre en évidence l'importance et l'efficacité de la structure de données dans le contexte donné.
 - Justifiez clairement pourquoi elle a été choisie afin de répondre au contexte d'utilisation.
- **Choix de la structure à développer** : Déterminez laquelle des quatre structures de données sera développée de manière approfondie dans le cadre du projet.
 - Ce choix doit être réfléchi et basé sur les exigences du projet et les capacités du langage de programmation utilisé.
 - Assurez-vous que le langage soit capable de supporter efficacement l'implémentation de la structure de données choisie pour un développement approfondi et réaliste dans le cadre du cours.

5. Patrons de conception

Vous devez exploiter les patrons de conception afin d'augmenter la qualité de votre architecture.

- **Sélection des patrons de conception** : Choisissez au moins trois patrons de conception différents à utiliser dans votre projet. Assurez-vous que chaque patron est adapté aux besoins et aux défis spécifiques du projet.
- **Documentation et justification** : Pour chaque patron de conception sélectionné :
 - Quelle problématique spécifique tentez-vous d'adresser en l'utilisant? Expliquez concrètement son usage dans votre projet. Cette explication doit démontrer clairement la pertinence du patron dans la perspective de l'amélioration de l'architecture de votre programme.
 - Justifiez le choix du patron de conception en expliquant pourquoi il représente une solution à la problématique identifiée. Autrement dit, expliquez en quoi ce patron de conception est

pertinent dans le cadre de votre projet en discutant des avantages qu'il apporte lorsqu'on le compare à des implémentations alternatives.

- **Choix du patron à développer** : Déterminez lequel des trois patrons de conception sera développé entièrement par vous afin d'en explorer tous les détails.
 - Décrivez quel type d'implémentation vous envisagez et quelle sera son interface de programmation.
 - N'hésitez pas à faire référence à aux ressources que vous avez utilisées.

6. Développement d'une bibliothèque

Afin de développer votre capacité à reconnaître des parties génériques et à séparer les dépendances d'un projet, vous devez procéder à la conceptualisation d'un module générique sous la forme d'une bibliothèque universelle.

- **Sélection d'un module technique réutilisable** : Effectuer une analyse approfondie du projet afin d'identifier un module technique pouvant être avantageusement réutilisé dans un cadre élargi à même ce projet ou pour d'autres initiatives futures.
- **Justification** : Justifier de manière explicite le choix de ce module, en précisant spécifiquement son usage avec les besoins du projet, mais aussi comment il pourrait être réutilisé dans d'autres contextes. Autrement dit, bien que le module puisse trouver son utilité en divers points du projet, il est nécessaire de détailler un cas d'utilisation particulièrement significatif à même le projet, mais aussi dans un contexte générique, mettant en exergue la pertinence du module.
- **Implémentation** : Détaillez en quoi votre implémentation nécessitera une transformation générique et s'éloignera d'une implémentation spécifique. Soyez technique, précis et concis.

7. Expression régulière

L'utilisation des expressions régulières est une compétence essentielle en informatique.

- Identifiez au moins un usage pertinent de l'utilisation des expressions régulières lié à votre projet.
- Pour cet usage, indiquez quelles informations vous tentez de valider et/ou d'extraire.
- Déterminez le motif (le « *pattern* ») requis pour votre expression régulière. Validez-le avec un site comme [Regex101](#) en utilisant quelques phrases tests. Faites des captures d'écran compactes démontrant vos tests.

8. Algorithme

Vous devez démontrer votre capacité à identifier et réaliser le développement d'au moins un algorithme.

L'objectif principal est de sélectionner et d'implémenter un algorithme qui répond de manière efficace et pertinente à la problématique posée. Il est crucial de comprendre que la complexité d'un algorithme n'est pas toujours synonyme de performance ou d'adéquation à la tâche à accomplir. Ainsi, votre choix ne devrait pas reposer sur la recherche de la complexité, mais plutôt sur la pertinence de ce dernier à répondre aux spécificités et aux exigences des requis.

Présentez d'abord la problématique que vous souhaitez adresser. Présentez l'algorithme sélectionné et, si applicable, citez vos sources et références. Justifiez en quoi cet algorithme s'avère approprié pour adresser le

problème en question. Si vous vous inspirez d'une référence qui présente l'algorithme choisi, indiquez s'il est nécessaire d'adapter l'implémentation présentée à votre projet, pourquoi et quelles sont les considérations. Finalement, décrivez, en vos mots, le raisonnement derrière l'algorithme.

À titre de référence, vous pouvez consulter ces articles ([français](#) et [anglais](#)).

9. Mathématique

On vous demande de mettre en lumière l'application des mathématiques dans la résolution d'un défi spécifique lié à votre projet. L'enjeu principal est d'identifier et d'exploiter judicieusement les théories ou les méthodes mathématiques permettant d'aborder efficacement le problème identifié. Il est important de souligner que le niveau des mathématiques attendu est celui de vos études collégiales.

Introduisez d'abord la question ou le problème auquel vous faites face dans votre projet. Présentez ensuite les méthodes ou principes que vous utiliserez pour y répondre, en veillant à mentionner vos références si vous en avez. On vous demande d'explicitement indiquer vos outils mathématiques. Quelles sont les limites de votre approche.

Si l'application directe des concepts mathématiques nécessite des ajustements pour s'intégrer dans votre projet, décrivez comment vous comptez les adapter ou les personnaliser.

Ce travail est une occasion de démontrer non seulement votre compréhension des mathématiques, mais aussi votre capacité à les appliquer de manière concrète dans des contextes informatiques réels.

10. Conception UML

Il est requis de vous que vous élaboriez une conception UML approfondie de votre projet, afin de démontrer une compréhension et une analyse poussées de celui-ci. Cette conception doit se traduire par des diagrammes précis et détaillés, reflétant une étude minutieuse des constituants et allant largement au-delà d'une approche superficielle.

Les diagrammes exigés pour cette tâche se limitent à ceux des cas d'utilisation et des classes :

- **Diagramme des cas d'utilisation** : Ce diagramme doit recenser exhaustivement tous les cas d'usage, les acteurs impliqués ainsi que les systèmes concernés par votre projet. Un niveau de réflexion avancé est attendu pour dégager les abstractions pertinentes. Vous devez faire usage des éléments de notation appropriés tels que les cas d'usage, les systèmes, les acteurs, les associations, l'héritage, l'abstraction et les notes. Sachez que d'autres concepts liés à ce diagramme existent, mais ne sont pas requis.
- **Diagramme de classes** : La réalisation d'un diagramme de classes exhaustif est cruciale et doit mettre en évidence chaque classe de votre projet, leur construction ainsi que la nature de leurs interactions les unes avec les autres. Cet exercice de modélisation est fondamental, car il permet de décomposer la complexité du projet en une modélisation informatique pertinente, facilitant ainsi la compréhension, l'implémentation des classes elles-mêmes et de leurs relations. Une attention particulière doit être portée à la définition précise de leurs attributs, opérations et sous-types en précisant certaines caractéristiques (abstraction, typage, masquage et élément statique) tout en spécifiant clairement les relations existantes entre les classes (généralisations, agrégations,

compositions et dépendances). Il est impératif d'utiliser ces éléments de notation de manière judicieuse pour illustrer clairement les dépendances et les hiérarchies au sein de votre projet. Cet effort de conception n'est pas seulement une formalité, mais une étape essentielle pour garantir que la structure sous-jacente de votre application soit robuste, flexible, facile à maintenir et extensible.

Ces diagrammes constituent des outils cruciaux pour la visualisation, la spécification, la construction et la documentation des aspects de votre projet. Il est donc essentiel de les aborder avec sérieux et précision, en veillant à ce qu'ils offrent une vision claire et complète des mécanismes et des structures de votre projet.

Structure du document à produire

Le document de conception vise à présenter l'ensemble des efforts réalisés pour mettre en forme les nombreux aspects techniques demandés. Ce document est constitué d'abord de schémas et de diagrammes (maquettes, UML, données persistantes) avec des descriptions présentant les autres éléments techniques spécifiés précédemment dans ce document.

Voici la structure de votre document.

1. Page titre
 - Le numéro et le titre du cours
 - Les prénom, nom et matricule de tous les étudiants de l'équipe
 - Le nom du projet
 - Sous le nom du projet, un texte bref contextualisant ou décrivant le projet (de 3 à 6 mots)
 - Le nom du document intitulé : *Document de conception*
2. Introduction
 - Rappel du projet :
 - Présentation sommaire du projet (1 à 3 phrases).
 - Présentation:
 - Brièvement, quel est l'objectif de ce document.
3. Éléments de conception (les 10 éléments présentés antérieurement)
 - 1. Infrastructure de développement *½ page environ*
texte descriptif
 - 2. Interface graphique utilisateur *nombre de page variable*
les schémas de toutes les maquettes avec un titre pour chacune d'entre elle – aucun texte
 - 3. Données persistantes *nombre de page variable*
texte présentant les technologies utilisées
diagramme présentant le schéma des données tel qu'indiqué
 - 4. Structures de données *½ page environ*
texte descriptif
 - 5. Patrons de conception *½ page environ*
texte descriptif
 - 6. Développement d'une bibliothèque *½ page environ*
texte descriptif
 - 7. Expression régulière *1 page environ*
texte descriptif avec captures d'écran
 - 8. Algorithme *½ page environ*
texte descriptif
 - 9. Mathématique *½ page environ*
texte descriptif
 - 10. Conception UML *nombre de page variable*
diagrammes des cas d'utilisation
diagrammes de classes
4. Annexe (optionnelle selon les projets)
 - Toutes vos références.
 - Toute autre élément de conception ou documentation que vous jugez pertinentes.

Grille de correction

-	Page titre	2 points
○	Contenu adéquat.	1 points
○	Qualité de la présentation.	1 points
-	Introduction	2 points
○	Concision et précision du rappel.	1 points
○	Pertinence de la présentation du document.	1 points
-	Éléments de conception	92 points
○	1. Infrastructure de développement	6 points
○	2. Interface graphique utilisateur	12 points
○	3. Données persistantes	10 points
○	4. Structures de données	6 points
○	5. Patrons de conception	6 points
○	6. Développement d'une bibliothèque	6 points
○	7. Expression régulière	4 points
○	8. Algorithme	6 points
○	9. Mathématique	6 points
○	10. Conception UML (diagramme des cas d'utilisation)	6 points
○	10. Conception UML (diagramme de classe)	24 points
-	Qualité générale du document	4 points
○	Cohérence, structure et qualité de la rédaction	1 points
○	Utilise un niveau langage approprié	1 point
○	Soins de la présentation et qualité des images insérées	2 points