



420-413

Hiver 2025

Projet de session

Développement d'applications pour entreprises

À partir d'un cahier de charge détaillant la demande de l'établissement Marie-Victorien, vous allez développer en équipe une application native ciblant des ordinateurs de bureau qui répondra aux besoins de l'établissement, sachant que :

- La réalisation de ce projet s'échelonne sur plusieurs semaines.
- L'application doit utiliser le langage C# en appliquant les notions de l'orienté objet et les bonnes pratiques de codage.
- Elle doit intégrer l'utilisation d'une base de données relationnelle en utilisant un framework d'ORM (Entity).
- L'application doit utiliser les bibliothèques de la plateforme ciblée afin d'offrir une interface graphique d'apparence native (WPF).
- Finalement, elle doit être empaquetée afin d'être prête au déploiement.

Vous allez travailler sur le projet proposé par votre équipe à la suite de la validation du cahier de charge par l'enseignante. C'est de votre responsabilité de s'assurer d'avoir pris les rendez-vous pour les rencontres nécessaires pour cette fin.

Des exigences plus détaillées et d'autres exigences complémentaires vous seront fournies ultérieurement, à chaque étape de la réalisation de ce projet. L'épreuve finale est divisée en trois volets.

Table des matières

Volet 1 : L'étude conceptuelle (document SEL simplifié).....	2
1. Introduction.....	3
1.1. Objectifs.....	3
1.2. Portée	3
1.3. Définitions, acronymes et abréviations	3
2. Description générale de l'application.....	3
2.1. Vue d'ensemble des fonctions de l'application.....	3
2.2. Description des utilisateurs	3
2.3. Contraintes de l'application.....	3
3. Conception détaillée	3
3.1. Interfaces externes	3
3.2. Les exigences de base données	4
3.3. Stratégie des tests	4
3.4. L'architecture applicative de l'application.....	4
3.5. Les exigences de l'application	4
Volet 2 : La réalisation de l'application (développement)	5
Volet 3 : La présentation et la documentation de l'application.....	7
Rapport final.....	8
1. Présentation du projet :	8
2. La gestion du projet :.....	8
3. Les fonctionnalités de l'application :.....	8
4. Le contrôle de qualité :.....	8
5. La gestion des risques :.....	8
6. La collaboration :.....	8
7. Les difficultés et apprentissages :	8
8. Conclusion :	8
Volet 1 : Conception et architecture	10
1. Liste des livrables :	10
2. Remise et détails :.....	10
3. Critères d'évaluation :.....	10
Volet 2 : Réalisation de l'application	11
1. Liste des livrables :	11
2. Remise et détails :.....	11
3. Critères d'évaluation :.....	11

Volet 3 : Présentation et documentation de l'application	12
4. Liste des livrables :	12
5. Remise et détails :	12
6. Critères d'évaluation :	12

Volet 1 : L'étude conceptuelle (document SEL simplifié)

À la suite de l'étude conceptuelle que vous allez effectuer, vous allez produire un document simplifié de spécifications des exigences d'un logiciel (Adapté de la norme IEEE 830-1993) **SEL**. Les détails sont mentionnés plus bas :

1. Introduction

Dans cette section, il y aura une vue d'ensemble du document.

1.1. Objectifs

Décrire les objectifs du document et son public cible. Ne pas décrire le futur système ou application.

1.2. Portée

Identifier le système par son nom. Il faut donc nommer la future application.

Expliquer brièvement ce que la future application devra faire et ne pas faire.

Décrire le contexte de l'application :

- Commerciale,
- Académique,
- Nouvelle,
- Web,
- Mobile,
- Native,
- Qui bénéficiera de l'application ?
- Quelles sont les retombées (bénéfices) ?

1.3. Définitions, acronymes et abréviations

Donner la définition de chaque terme technique (non conventionnel). De plus décrire les acronymes et abréviations. Il est important de tout mentionner et de ne pas laisser la lecture de document deviner la signification des termes techniques.

Il faudra lister tout document, site web et norme que vous avez utilisés pour réaliser ce document.

2. Description générale de l'application

2.1. Vue d'ensemble des fonctions de l'application

Décrire l'ensemble des fonctionnalités de l'application pour donner une vue d'ensemble au lecteur de ce document.

2.2. Description des utilisateurs

Décrire l'ensemble des utilisateurs de l'application en mettant l'accent sur leurs caractéristiques.

2.3. Contraintes de l'application

Décrire les contraintes techniques de l'application. C'est ici qu'il faut mentionner quels seraient les langages de programmation qui seront utilisés pour le développement de l'application.

Il faut aussi lister les limites de l'application, par exemple. Pour bien décrire les contraintes, il vous faut questionner le client.

3. Conception détaillée

3.1. Interfaces externes

Lister, décrire et dessiner les maquettes et/ou les wireframes du GUI de l'application. Ne pas oublier de spécifier les librairies qui seront utilisées pour le développement du GUI et ajouter les liens nécessaires.

3.2. Les exigences de base données

Décrire le type de la BD relationnelle avec son schéma de tables (ERD).

3.3. Stratégie des tests

Décrire la stratégie des tests (celle du code) de l'application (toutes les parties de l'application).

3.4. L'architecture applicative de l'application

Décrire et dessiner l'architecture de l'application en présentant la vue d'ensemble de l'architecture. Il faudra spécifier les liens entre les différentes parties de l'application (à l'aide du diagramme des composants) et les moyens de communications. Il ne faut pas oublier de spécifier les considérations de sécurité et de performance.

3.5. Les exigences de l'application

Dans cette section il faudra décrire toutes les exigences de l'application : les exigences fonctionnelles et non fonctionnelles (définir ce que le système doit faire (fonctionnel) et comment il doit le faire (non fonctionnel, par exemple, performance, sécurité, fiabilité).

Pour la partie des exigences fonctionnelles vous devez fournir un **diagramme de classe** et **au moins 3 use cases** (les plus importants) ainsi que les **diagrammes nécessaires** pour expliquer certaines fonctionnalités (diagramme de séquence, diagramme d'activité, diagramme de flux de données, etc.), selon le besoin et la nature de votre application.

Le formalisme pour les use cases est le suivant :

Id Use case	
Nom use case	
Objectif	
Acteur principal	
Acteur secondaire	
Pré condition	
Scénario	
Post condition	
Description	

Formalisme du use case qui sera utilisé pour votre projet

Id Use case : Le numéro de version.

Nom : Utiliser une tournure à l'infinitif (ex : Se connecter au système).

Objectif : Une description résumée permettant de comprendre l'intention principale du cas d'utilisation. Cette partie est souvent renseignée au début du projet dans la phase de découverte des cas d'utilisation.

Acteurs principaux : Ceux qui vont réaliser le cas d'utilisation (la relation avec le cas d'utilisation est illustrée par le trait liant le cas d'utilisation et l'acteur dans un diagramme de cas d'utilisation).

Acteurs secondaires : Ceux qui ne font que recevoir des informations à l'issue de la réalisation du cas d'utilisation.

Les préconditions : elles décrivent dans quel état doit être le système (l'application) avant que ce cas d'utilisation puisse être déclenché.

Des scénarios : Ces scénarios sont décrits sous la forme d'échanges d'événements entre l'acteur et le système. On distingue le scénario nominal, qui se déroule quand il n'y a pas d'erreur, des scénarios alternatifs qui sont les variantes du scénario nominal et enfin les scénarios d'exception qui décrivent les cas d'erreurs.

Des post conditions : Elles décrivent l'état du système à l'issue des différents scénarios.

La description : est une rubrique optionnelle. Elle contient généralement des spécifications non fonctionnelles (spécifications techniques, . . .). Elle peut éventuellement contenir une description des besoins en termes d'interface graphique.

Volet 2 : La réalisation de l'application (développement)

À la lumière de votre document de conception (SEL) que vous avez rédigé pour le volet 1 de votre projet, vous devez coder l'application en respectant l'ordre de priorité des éléments du carnet de produit et la notion MVP (Minimum Viable Product).

Votre code devrait être hébergé dans un référentiel (repo) distant sur Github. Vous devez utiliser des branches et le merge se fait à l'aide des pull request (PR) avec approbation (tel qu'expliqué en classe).

Les tâches devraient être créées, détaillées et suivies selon les pratiques montrées en classe.

Vous devez avoir au moins un fichier readme (à la racine de votre solution) dans lequel vous allez décrire les technologies utilisées, comment installer votre application, comment faire le déploiement de votre application et comment est structuré votre code, etc. ainsi que les détails en lien avec votre code et qui méritent d'être mentionnés (pour s'inspirer de ce qu'on peut mettre dans un fichier readme vous pouvez consulter ces liens : [exemple 1](#), [exemple 2](#)).

La version finale à livrer devrait se retrouver sur la branche principale main. C'est cette version qui sera évaluée et testée. Le code devrait compiler et s'exécuter sans aucune intervention de ma part.

Documentation Wiki tel que :

- Documenter les éléments à considérer lors de la revue de code (code review d'un PR)
- DoD (Definition of Done)
- Style code de l'équipe
- Etc.

Le travail devrait être fait d'une façon équitable, les tâches assignées et les commits feront partie des métriques à évaluer.

Vous devez me donner accès à votre backlog en m'invitant sur trello ou Jira avec le nom d'utilisateur : SaraBoumechraz.

Volet 3 : La présentation et la documentation de l'application

Rapport final

1. Présentation du projet :

- Les technologies et les outils utilisés dans votre application.
- Les fonctionnalités et l'état de complétion de chacune.
- Les patrons de conception (design patterns) utilisés dans votre application.
- La méthode de déploiement.

2. La gestion du projet :

- Comment avez-vous géré votre projet et quelle méthodologie agile avez-vous utilisée et quels sont les outils utilisés. Indiquez la durée de vos sprints et comment avez-vous réparti le travail au long des sprints.
- Parlez des actions, améliorations et des solutions implémentées à la fin de chaque sprint.

Note : vous êtes tenus à planifier et finaliser au moins 2 sprints.

3. Les fonctionnalités de l'application :

Parlez des différentes fonctionnalités telles que l'authentification : Expliquez la méthode d'authentification utilisée dans votre application

- *Manuel d'utilisation de l'application* :
Environ une page à 2 pages présentez le manuel d'utilisation de votre application.

Note : utiliser des captures d'écran au besoin.

4. Le contrôle de qualité :

- Un plan de tests assez concis qui aborde les éléments que vous avez découverts dans vos cours avec une suite de tests pour un cas avec différents scénarios de test.
- En plus vous devez indiquer :
 - o Comment gérez-vous vos problèmes de code (bugs détectés).
 - o Les techniques de tests utilisées dans votre projet.

5. La gestion des risques :

Énumérez les risques que vous avez rencontrés lors de la réalisation de votre projet et comment avez-vous procédé pour les remédier.

Note : certaines informations étaient déjà abordées dans votre charte d'équipe.

6. La collaboration :

- Les outils de collaboration utilisés
- La méthode de collaboration et de communication au sein de l'équipe.

Note : certaines informations étaient déjà abordées dans votre charte d'équipe.

7. Les difficultés et apprentissages :

- Parlez des problèmes rencontrés lors de la réalisation de votre projet.
- Parlez des apprentissages et acquis en lien avec la réalisation de ce projet.

8. Conclusion :

Une conclusion en un ou deux paragraphes.

Livrables, critères d'évaluation et détails de remise

Volet 1 : Conception et architecture

1. Liste des livrables :

- Document de l'étude conceptuelle (avec les diagrammes demandés).
- Tableau des liens des différents documents et projets.

Description	Lien	Remarque
Document de l'étude conceptuelle		
Collaboration et équipe : Teams		
Repo : Github		
Projet des maquettes et prototypes : Figma		
Backlog :Trello, Jira, Miro. Notion ou équivalent		
Diagrammes : draw.io, miro ou autres		
Wiki : Github, Miro, Notion ou autres		
Dossier partagé pour divers documents		

2. Remise et détails :

- Vous devez fournir le **document SEL simplifié** dans un fichier Word portant le nom de votre équipe ainsi que le document contenant le tableau des différentes ressources utilisées dans la réalisation de cette application.
- Le document de conception doit avoir une page de garde avec le nom de l'équipe et des membres. Une table des matières doit suivre la page de garde du document.
- Les supports additionnels peuvent être inclus, comme des liens vers les diagrammes et les schémas sur un outil en ligne ou les diagrammes dans des fichiers séparés pour faciliter la consultation (si l'image dans le document ne permet pas de voir certains détails clairement). Cela ne signifie pas de ne pas les inclure comme images dans votre document Word.
- Vous devez faire attention à la présentation et l'uniformité de vos documents. Sinon une pénalité allant jusqu'à 10% de la note du travail peut être appliquée.
- La remise se fait sur Léa à l'heure indiquée sur la boîte de remise.

3. Critères d'évaluation :

- La présence de tous les points mentionnés dans l'énoncé.
- Le format et la mise en page des documents.
- Le respect des consignes du format et du fond des fichiers.
- Le niveau de détail et l'analyse conceptuelle et architecturale.
- La pertinence de la conception et du choix de l'architecture.

Volet 2 : Réalisation de l'application

1. Liste des livrables :

- Le référentiel (repository) Github incluant : le code, le readme, le wiki et les scripts utilisés.
- L'exécutable.
- Une démonstration en classe de l'application finale précédée par une présentation de 5 à 10 minutes de l'application.

2. Remise et détails :

- La remise se fait sur Léa à l'heure indiquée sur la boîte de remise.
- La remise inclus un document contenant le lien vers votre repo (vous pouvez rajouter des détails sur vos livrables dans ce document si nécessaire) et l'exécutable le tout dans un fichier compressé.

3. Critères d'évaluation :

- Le respect des consignes et la présence de tous les points mentionnés dans l'énoncé.
- Le format et la mise en page des différents fichiers et le code.
- La qualité du code et le respect des bonnes pratiques.
- La structure du code.
- Le code compile et s'exécute correctement sans aucune intervention.
- L'exécutable.
- Le wiki.
- Le fichier readme.
- L'utilisation des branches.
- La fusion (merge) des branches à l'aide des pull request.
- La revue de code.
- La distribution de la charge de travail.
- La présence des fonctionnalités principales.
- La bonne priorisation des éléments de backlog.

Volet 3 : Présentation et documentation de l'application

4. Liste des livrables :

- Le rapport final
- Les maquettes finales.
- Les comptes rendus des rétrospectives (au moins 2 rétrospectives) incluant les 3 points suivants : ce qui s'est bien passé pendant le sprint, ce qui s'est mal passé et les points à améliorer avec les actions prises s'il en a. (environ ½ à 1 page).
- Des annexes peuvent être ajoutées au besoin après la conclusion.

5. Remise et détails :

- Le document doit avoir une page de garde avec le nom de l'équipe et des membres. Une table des matières doit suivre la page de garde du document.
- Les supports additionnels peuvent être inclus, comme des liens vers les diagrammes et les schémas sur un outil en ligne ou les diagrammes dans des fichiers séparés pour faciliter la consultation (si l'image dans le document ne permet pas de voir certains détails clairement). Cela ne signifie pas de ne pas les inclure comme images dans votre document Word.
- Vous devez faire attention à la présentation et l'uniformité de vos documents. Sinon une pénalité allant jusqu'à 10% de la note du travail peut être appliquée.
- La remise se fait sur Léa à l'heure indiquée sur la boîte de remise.

6. Critères d'évaluation :

- La présence de tous les points mentionnés dans l'énoncé.
- Le format et la mise en page des documents.
- Le respect des consignes du format et du fond des fichiers.
- La qualité du contenu des sections et le niveau d'analyse.