**Projet de programmation**

**CargoFlow**

Martet Antoine

[antoine.martet@eduvaud.ch](mailto:antoine.martet@eduvaud.ch)

Clot Ian

[ianalain.clot@eduvaud.ch](mailto:ianalain.clot@eduvaud.ch)



SI-CA2a

30 décembre 2023

**Table des matières**

[1. Introduction 3](#_Toc1)

[1.1. Cadre, description et motivation 3](#_Toc2)

[1.2. Organisation 3](#_Toc3)

[1.3. Objectifs 3](#_Toc4)

[1.4. Planification 4](#_Toc5)

[2. Analyse 4](#_Toc6)

[2.1. Use cases et scénarios 4](#_Toc7)

[2.2. Maquettes 4](#_Toc8)

[2.3. MCD 4](#_Toc9)

[3. Implémentation 4](#_Toc10)

[3.1. Choix techniques 5](#_Toc11)

[3.1.1. Base de données 5](#_Toc12)

[3.1.2. Documentation 5](#_Toc13)

[3.1.3. Développement 5](#_Toc14)

[3.2. Conventions 5](#_Toc15)

[3.2.1. Commentaires 5](#_Toc16)

[3.2.2. Conventions de nommage 5](#_Toc17)

[3.3. MLD 6](#_Toc18)

[3.4. Fonctionnement 6](#_Toc19)

[4. Tests 6](#_Toc20)

[4.1. Tests effectués 6](#_Toc21)

[4.2. Erreurs restantes 6](#_Toc22)

[5. Conclusions 6](#_Toc23)

[5.1. Conslusion de groupe 6](#_Toc24)

[5.2. Conclusions individuelles 6](#_Toc25)

[6. Annexes 6](#_Toc26)

[6.1. Sources - Bibliographie 6](#_Toc27)

[6.2. Table des abbréviations 6](#_Toc28)

[6.3. Table des illustrations 7](#_Toc29)

# Introduction

## Cadre, description et motivation

CargoFlow est un projet de programmation qui a pour objectif de nous apprendre à gérer un projet de développement tout en nous apprenant à gérer un projet de sa planification à sa concrétisation. CargoFlow est un logiciel permettant la gestion de la logistique dans une entreprise de logistique. Il permet de gérer le stock, les client.es, les employé.es, les transporteurs, les entrepôts ainsi que les livraisons.

Ce projet respecte les conditions du module et nous intéresse dans la mesure où il est relativement utile et concret : on pourrait être amené, un jour, à travailler sur un projet similaire en entreprise.

Ce projet est développé en C# et se base sur des Windows Form. Il inclut également un serveur local MySQL hébergeant la base de donnée contenant les informations nécessaires à la gestion du stock par le logiciel.

Le logiciel est prévu pour être utilisé localement sur un ordinateur individuel mais rien n’empêcherait d’héberger la base de données sur un serveur en ligne.

## Organisation

Ce projet a été divisé en 3 grosses parties. Tout d’abord, nous avons commencé par planifier les différentes étapes de notre projet et nous sommes penchés sur l’établissement des uses cases. Concernant la planification des tâches, nous avons utilisé iceScrum. Pour ce qui en est des uses cases et des scénarios, il nous a semblé judicieux de l’effectuer sur Excel. Nous avons durant cette même période mis en place le repository Github, afin d’archiver notre progression et de travailler facilement sur les mêmes fichiers.

Dans un second temps, nous avons cherché à modéliser la BDD qui serait utilisée dans notre logiciel ainsi que l’apparence du logiciel. Nous avons ainsi un MCD puis un MLD et réalisé des maquettes.

La troisième et dernière étape était l’écriture du code. Cependant, de nombreuses recherches ont dues être effectuées avant l’écriture de C#, car nous avions besoin d’outils que nous n’avions pas vu jusqu’ici, en particuliers le DataGridView, une classe liée aux WindowsForm permettant de créer dynamiquement des tableaux. En parallèle des recherches et de l’écriture du code, nous avons pu transformer le MLD en requête SQL, pour ensuite intégrer notre base de données dans notre logiciel.

Durant ces 3 parties, nous avons documenté chacun de notre côté, les éléments nous semblant pertinents, afin de simplifier l’écriture de la documentation.

## Objectifs

Notre page de login doit interagir avec la base de données, afin d’avoir accès à l’adresse email et au mot de passe des personnes ayant accès à notre logiciel. Si les informations d’identification sont correctes, la page d’accueil de notre logiciel doit s’afficher. Sur le haut de la page d’accueil du logiciel, des onglets permettent de choisir quelles sont les informations que l’on souhaite afficher. Lors du chargement de l’onglet sélectionné, le logiciel interagit avec la base de données pour afficher les données nécessaires.

Lorsqu’on se trouve sur un de ces onglets, par exemple le transporteur, la liste des éléments que le client a exigée s’affiche dans le DataGridView. A droite de cette liste, quatre boutons permettent d’effectuer les opérations CRUD. Le bouton Détail permet d’afficher toutes les caractéristiques du transporteur se trouvant dans la BDD. Le bouton Ajouter permet d’ajouter un nouveau transporteur dans la BDD. Le bouton Modifier permet de modifier un transporteur existant déjà dans la BDD. Le bouton Supprimer permet de supprimer un transporteur de la BDD.

## Planification

Nous avons décidé de diviser le temps que nous avions à disposition, soit du 6 novembre au 19 janvier, en 5 sprint. Le détail de nos activités durant cette période se trouve sur [iceScrum](https://icescrum.cpnv.ch/p/CARGOFLOW/#/planning). Sur demande de M. Ithurbide, nous avons aussi effectué un diagramme de Gantt, afin de schématiser lors de la première semaine du module, le déroulement approximatif de notre projet sur l’ensemble du trimestre.

Concernant la division du temps à notre disposition, nous avons décidé d’effectuer 5 sprints. Aux 4 premiers sprints ont été attribuées deux semaines chacun. Concernant la dernière semaine à disposition, nous avons préféré créer un 5e sprint plutôt que d’allonger le 4e d’une semaine. La raison de cette décision a été que cette dernière semaine était assez différente des autres, dans le sens où elle était prévue essentiellement pour la fin de l’écriture de la documentation et de la préparation de la présentation. L’essentiel de l’implémentation était déjà effectué.

# Analyse

## Use cases et scénarios

Comme indiqué plus haut, nous avons détaillé les uses cases sur un document Excel. Il se trouve sur Github dans la rubrique Documentation sous le nom CargoFlow\_Gantt\_UseCases.xlsx.

## Maquettes

Nous avons conçu nos maquettes avec Balsamiq. Chaque onglet de notre logiciel a été créé sur une nouvelle fenêtre, afin de bien distinguer les différences entre les pages malgré leur similitude. Le fichier de maquettes se trouve dans le dossier Documentation sous le nom maquette\_cargoFlow.bmpr dans le branche main.

## MCD

Le MCD se trouve sur Github dans le dossier Documentation sous le nom CargoFlow\_MCD\_v3.drawio dans la branche main.

# Implémentation

## Choix techniques

### Base de données

Pour faire fonctionner notre BDD, nous avons utilisé deux logiciels. Le logiciel jouant le rôle de serveur est MySQL server et celui jouant le rôle du client est HeidiSQL. Concernant les données, elles sont pour la plupart modifiables dans CargoFlow mais nous avons également jugé pertinent de créer manuellement un ensemble d’enregistrements afin de mieux travailler par la suite et pour pouvoir présenter le logiciel au client. Il a fallu s’assurer que les données aient une certaine cohérence, tant au niveau des champs qu’au niveau des clés étrangères. La BDD est hébergée localement sur l’ordinateur de la personne utilisant le logiciel.

### Documentation

Tout au long du module, nous avons pris des notes sur les éléments devant figurer dans la documentation de notre projet. Nous avons par la suite effectué une mise en commun de nos notes, afin de les inclure dans cette documentation. Ce document représente le document principal de notre documentation. Il fait référence à d’autres documents qui, pour des questions de lisibilité, se trouve dans des documents annexes : MCD, MLD, fichiers de requêtes SQL d’insertion de données, maquettes.

### Développement

Notre outil principal pour le développement a été le logiciel Visual Studio. Il nous a permis d’écrire le code, de créer les WinForms, de gérer les erreurs de compilation et de build. En bref, il a été le centre de notre projet. De plus, nous avons utilisé git et Github pour se transmettre et stocker les fichiers.

## Conventions

### Commentaires

Le code contient des commentaires. Ceux-ci sont écrits au-dessus de la ligne ou des lignes de code concernées.

### Conventions de nommage

Les conventions sont les mêmes que celles utilisées habituellement en C# au CPNV :

Classe : PascalCase

Objet : camelCase

Méthode publique d’une classe : PascalCase

Attribut privé d’une classe : camelCase

## MLD

Le MLD se trouve sur Github dans le dossier Documentation sous le nom CargoFlow\_MLD\_v3.mwb dans la branche main.

## Fonctionnement

Le code de CargoFlow repose sur des Windows Forms et des classes métier personnalisées.

Il existe :

* 1 Form pour le login. Celui est un particulier dans le

# Tests

## Tests effectués

## Erreurs restantes

# Conclusions

## Conclusion de groupe

Ce projet s’est dans l’ensemble bien déroulé. Nous avons néanmoins été confronté à un certain nombre de problèmes.

## Conclusions individuelles

Ian :

Antoine :

# Annexes

## Sources - Bibliographie

A REMPLIR - ANTOINE

## Table des abbréviations

BDD : Base de données

MCD : modèle conceptuel de données

MLD : modèle logique de données

WinForms : Windows Froms

DB : Database

CRUD : create, read, update, delete

## Table des illustrations

**No table of figures entries found.**