

HENRY  
MIRICA  
VAN DE WALLE

20.02.2020



---

# LOGICIEL DE FACTURATION

---

Cahier de charges



# EPHEC

AVEC LE SOUTIEN DE :

X. Dubruille - V. Van Den Schriek – L. Van Dormael

# Table des matières

<b>1. Présentation du projet</b>	<b>2</b>
a. Contexte	2
b. Objectifs	2
c. Description de l'existant	2
d. Critères d'acceptabilité du produit	2
<b>2. Expression des besoins</b>	<b>3</b>
a. Besoins fonctionnels	3
b. Besoins non fonctionnels	3
<b>3. Contraintes</b>	<b>3</b>
a. Coûts	3
b. Délais	3
c. Autres contraintes	3
<b>4. Déroulement du projet</b>	<b>4</b>
a. Planification	4
b. Plan d'assurance qualité	4

## Liens

Trello : <https://trello.com/b/TEBaZ5gH/projet-facturation>

Github : <https://github.com/Const-Babar-Tortue/>

### 1. Présentation du projet

#### a. Contexte

Dans le cadre de notre cours de développement web nous avons dû développer une application web qui répondra à la demande de notre client TM Net. Cette petite entreprise de maintenance a besoin d'une solution afin de gérer sa comptabilité d'une manière rapide et efficace. Nous allons nous baser sur le fichier Excel utilisé par l'entreprise pour le moment afin de nous assurer que tous les calculs et toute la législation soit respectée.

#### b. Objectifs

Le but du projet est de créer une application qui sera capable, grâce aux informations reçues de la base de données principale de l'entreprise, de créer un pattern afin d'anticiper les pannes des clients. Ce système sera utilisé pour éviter les encombrements durant l'hiver, car la plupart des retards s'annoncent lorsque l'agenda est le plus chargé. Pendant cette période-là les commandes inattendues sont inacceptables et une organisation presque militaire est nécessaire pour pouvoir répondre aux demandes de tous les clients. Afin de diminuer les conséquences financières ainsi que les soucis organisationnels, l'application montre bien sa nécessité dans le cadre de l'entreprise.

Du point de vue technique, l'application sera constituée de 3 parties : DB, Web et virtualisation.

La base de données choisie est SQLite car il est cross platform, facile à utiliser grâce aux nombreux forums et à sa popularité, mais aussi open source.

Python sera utilisé pour le côté web et la gestion de données sortante de la base de données. En étant extrêmement polyvalent, facile à utiliser et à comprendre, il est le choix idéal pour notre projet.

La virtualisation de nos services sera faite avec Docker afin de nous assurer que si un service tombera, les autres ne seront pas affectés. De plus la mise en place des services est beaucoup plus rapide

#### c. Description de l'existant

En ce moment l'entreprise fonctionne avec un fichier Excel qui regroupe tout l'historique comptables, mais aussi les formules relatives aux divers calculs. Lors de la création de ce fichier, l'aide d'un comptable autorisé a été fourni afin d'assurer la véracité des calculs. Le problème de cette solution est qu'elle n'est pas extrêmement facile d'utilisation, elle n'est pas en ligne et les fonctionnalités sont limitées. Le nouveau projet utilisera aussi les calculs, et l'idée de ce fichier Excel en l'améliorant du point de vue visuel et du point de vue de son utilisation.

#### d. Critères d'acceptabilité du produit

La première condition pour que le projet soit accepté pour la mise en production, la plateforme doit être plus facile à utiliser que l'ancien système.

La deuxième condition est la création des nouvelles fonctionnalités utiles.

Le client préfère une application web plus basique du point de vue du design, mais qui fonctionne très bien.

## 2. Expression des besoins

### a. Besoins fonctionnels

Les fonctions primaires / obligatoires sont :

- Connexion à la db pour pouvoir extraire les informations nécessaires
- Encoder des factures
- Envoyer par mail, un fichier pdf avec la facture générée
- Pouvoir imprimer la facture
- Gérer le paiement des factures (si le client a payé ou pas – bouton « oui / non » )
- L'application sera facilement modifiable car elle sera fournie avec une documentation qui expliquerait l'utilité et l'utilisation de chaque module
- Possibilité d'ajouter et d'effacer des utilisateurs
- Pouvoir générer un fichier csv avec les clients qui ont eu des factures de +250 Euros entre certaines dates.

### b. Besoins non fonctionnels

- La DBJ ne sera jamais impactée/modifiée par le logiciel
- Les données vont être stockées dans une base de données afin que l'application puisse les traiter
- L'application utilisera Python 3 et SQLite et Docker
- L'application va toujours prendre en compte les nouvelles données
- L'application adaptera ses calculs en fonction des nouvelles données
- Chaque module sera testé avant d'être mis dans l'ensemble du programme

## 3. Contraintes

### a. Coûts

Suite à notre analyse, les coûts engendrés vont être de maximum 20 euros par un pour un VPS.

### b. Délais

Le délai final pour la réalisation de l'application sera le mois de juin 2020, lors de la fin du projet.

### c. Autres contraintes

En tenant compte que l'entreprise TM Net respecte les règles RGPD, l'application doit être parfaitement sécurisée dans le transfert des données avec la base de données principale mais aussi avec la base de données propre. De plus, toute information qui se trouve dans les DB doit rester confidentielle.

## 4. Déroulement du projet

### a. Planification

Toutes les communications seront faites via la plateforme **Teams** qui centralisera toutes nos ressources et toutes nos plateformes.

**Trello** sera utilisé afin d'avoir une meilleure vision de notre avancement du travail de chacun. Chaque tâche sera conçue afin qu'un des membres puisse la réaliser seul.

Notre rendement sera aussi calculé en fonction de notre nombre d'heures travaillée et du résultat obtenu. Pour comptabiliser nos heures nous utilisons **Clockify**.

Afin de partager nos codes et d'avoir un versioning de notre travail, nous utiliseront **GitHub**.

### b. Plan d'assurance qualité

Pour assurer la qualité et le bon fonctionnement, chaque fonctionnalité sera vérifiée par des tests unitaires.