

Cahier des charges

Objectifs

L'objectif d'UMLight est de proposer une solution pour faciliter la gestion de projet.

L'utilisateur devra pouvoir exporter son travail au format PDF.

Cette application sera une application locale pour Windows codée dans le langage informatique de Microsoft, le C#.

Chaque projet aura un seul et unique chef de projet.
Les rôles (chef de projet et collaborateur) seront définis par projet.

Génération

Cette solution sera un pseudo langage permettant de générer des modèles UML. Les autres productions qu'UML seront :

- MCD
- Schéma entité-association
- Scripts SQL pour la création de la base de données
- Diagramme Gant
- Temps et coût estimé pour la réalisation du projet

Toutes ces solutions seront générées à partir du modèle UML, et non à partir du pseudo langage en lui-même.

Interface

L'interface utilisateur sera une interface graphique réalisé à l'aide du framework .NET dans sa version 4.8.

Cette interface devra obligatoirement respecter la charte graphique défini pour l'application UMLight.

Au démarrage de l'application un écran de connexion apparaîtra. Un bouton devra permettre de modifier les paramètres de connexion à la base de données distante.

Après l'écran d'identification apparaîtra un écran listant les projets auxquels l'utilisateur participe, lui permettant des les charger ou de les supprimer s'il en est le chef de projet. Cet écran devra également permettre de créer un nouveau projet.

L'interface principale d'un projet sera divisée en 2 pour le chef de projet, horizontalement ou verticalement au choix de l'utilisateur. Ce dernier devra également pouvoir modifier à sa guise la taille de ces parties.

- La première de ces parties sera une console permettant d'exécuter le pseudo langage.

- La deuxième de ces parties sera un aperçu du travail effectué via la partie console. Cette partie aura donc plusieurs onglets :

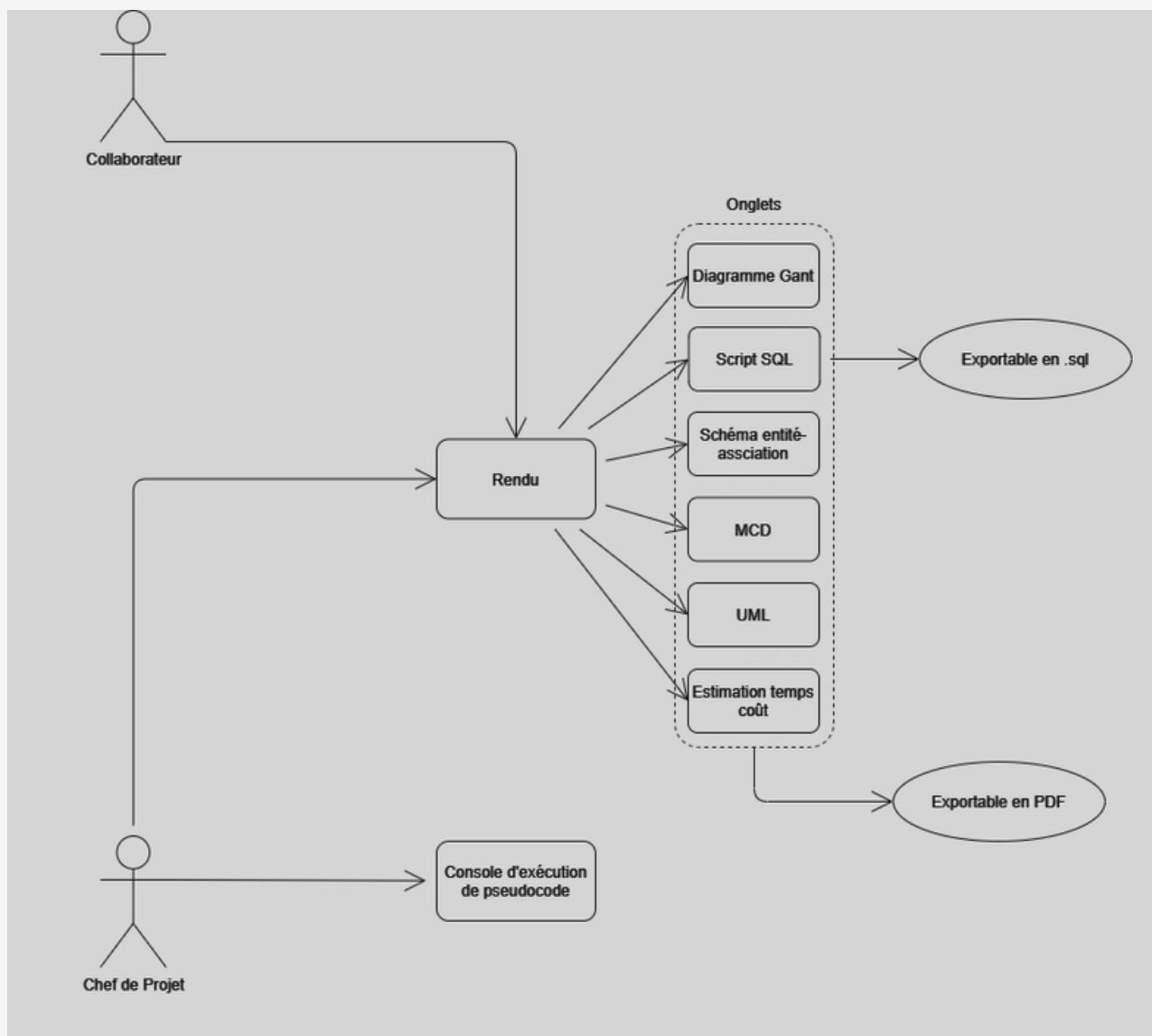
- o UML
- o MCD
- o Schéma entité-association
- o Script SQL
- o Diagramme Gant
- o Estimation temps et coûts.

Peut importe l'onglet, un bouton « Exporter » devra permettre d'exporter le rendu observer au format PDF.

Les script SQL devront également pouvoir être exportés au format « .sql ».

L'utilisateur devra pouvoir ajouter, modifier ou supprimer des collaborateurs à un projet, s'il en est le chef de projet, à partir d'un bouton « Collaborateurs ». Ce dernier devra se trouver sur l'interface principale du projet et devra être accessible depuis le menu « Projet ».

L'interface d'un projet est résumé sur le Use Case suivant :



Lorsque l'utilisateur est simplement collaborateur, celui-ci n'a pas de partie console, il ne peut que consulter les rendus et les exporter.

L'utilisateur devra pouvoir sauvegarder son travail ainsi que charger les autres projets dont il est membre. Il devra également pouvoir créer de nouveau projet, et supprimer ceux dont il est chef de projet à partir du menu « Projets ».

L'utilisateur devra pouvoir modifier ses informations personnelles à partir du menu « Utilisateur », tel que son identifiant et son mot de passe.

Ce menu permettra également de lister tous les utilisateurs de l'application, ainsi que d'ajouter et supprimer des collaborateurs aux projets dont l'utilisateur est chef de projet.

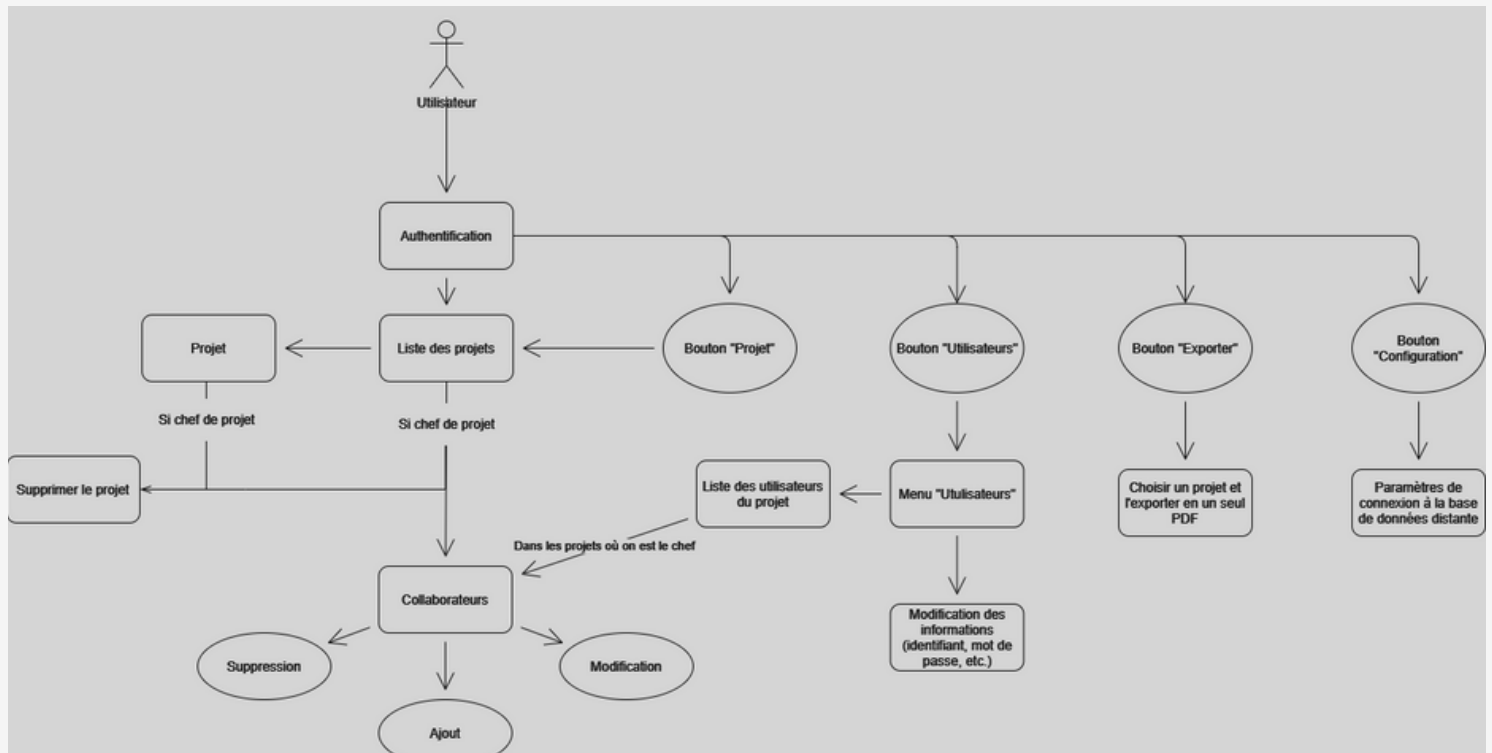
Les menus devront être accessible dans une barre horizontale en haut de la fenêtre. Ceux-ci sont :

- Projets
- Utilisateur
- Exporter
- Configuration

L'onglet exporter permettra simplement d'exporter toutes les visualisations d'un projet (UML, MCD, schéma entité-association, diagramme Gant et estimation temps et coût) dans un seul PDF.

Le menu "Configuration" permettra de modifier les paramètres de connexion à la base de données distante.

L'interface global de l'application et les interactions entre les menus est résumé par le Use Case suivant :



Stockage

Toutes les données, les projets et comptes utilisateurs, seront enregistrées sur un serveur distant. Les paramètres de connexion à ce serveur seront enregistrés dans un fichier de configuration.

Les utilisateurs

Plusieurs données seront nécessaires pour les utilisateurs :

- Nom
- Prénom
- Salaire à la journée
- Mot de passe

Chaque utilisateur pour être chef de projet ou collaborateur sur chaque projet. Il pourra créer ses projets ainsi que modifier ou supprimer les projets dont il est le chef de projet.

Diagramme de Gant

Pour permettre la génération d'un diagramme de Gant, la réalisation d'une classe ne pourra pas concerner plusieurs tâches.

Par défaut la réalisation d'une classe correspondra à une tâche. L'utilisateur aura la possibilité de mettre la réalisation de plusieurs classes dans une même tâche ainsi que de créer des tâches sans classes (tel que la réalisation d'une charte).

Pseudo langage

Le pseudo langage devra permettre de créer un modèle UML, et devra donc gérer à ce titre :

- Les classes
- Les classes abstraites
- L'héritage
- Les relations

La syntaxe de ce pseudo langage est défini dans un document annexe.