PROJET TEST DRIVE AND DEVELOPPEMENT

THÈME: Gestionnaire de PAE (programme annuel d'étude)

github: https://github.com/benjvdb9/PAE-manager.git

But: le client de l'application pourra consulter des informations, gérer des informations, générer des informations liées au PAE

Notre application aura 2 point de vue:

1. Point de vue professeur
2. Point de vue étudiant

1. Point de vue professeur

a. Statut de professeur:

- Le professeur est responsable d'une ou plusieurs unités
- Le professeur peut être chef de departement

b. Fonction du professeur:

Le professeur doit pouvoir faire des modifications

- Encodage de notes liées à son ou ses unités d'enseignement
- Produire des résumés et fiches à titre informatif pour son/ses unités
- Modifier le nombre de crédits lié à son/ses unités
- Modifier le nombre d'heures lié à son/ses unités

Le professeur doit pouvoir approuver la demande d'un étudiant à son unité :

- Faire une validation de l'étudiant
- Consulter l'historique des PAE de l'étudiant (avec pour but de vérifier si l'étudiant à des prérequis pour son unité)

2. Point de vue étudiant

a. L'étudiant doit être capable de consulter:

- Être capable de voir son nombre de crédits validés
- Voir son nombre de crédits annuels.

b. L'étudiant doit être capable de créer son programme:

- Vérifier la validation des unités auxquelles il veut accéder
- Choisir les unités (officiels et officieux)

3. Diagramme de packages.

Après l'analyse préliminaire de notre projet pour une application permettant la gestion de PAE, nous pouvons regrouper les besoins en différents package et donc déjà réalisation un diagramme de package.
À note:
Un package représente un ensemble de fonctionnalités et de besoins ayant de liens ou d'une même famille.
Un diagramme de package permet de décomposer le système en catégories ou parties plus facilement observables, appelés « packages ». Cela permet également d'indiquer les acteurs qui interviennent dans chacun des packages.
Ce diagramme peut déjà nous aider à concevoir et faire le découpage de l'interface de l'application par bloc de fonction.

Les différents packages:

a. Point de vue professeur:

Package "Gestion d'une unité" contient:

- Encodage de notes liées

- Rédiger un résumé
- Rédiger une fiche
- Modifier le nombre de crédits
- Modifier le nombre d'heures

Package "Gestion d'une demande" contient:

- Valider ou non un étudiant
- Consulter l'historique d'un étudiant

b. Point de vue étudiant:

Package "Consultation" contient**:**

- Consulter le nombre de crédits validés
- Consulter le nombre de crédits annuels

Package "gestion d'un programme" contient:

- Verifier la validation des unités
- Choisir des unités avec option officiels ou officieux

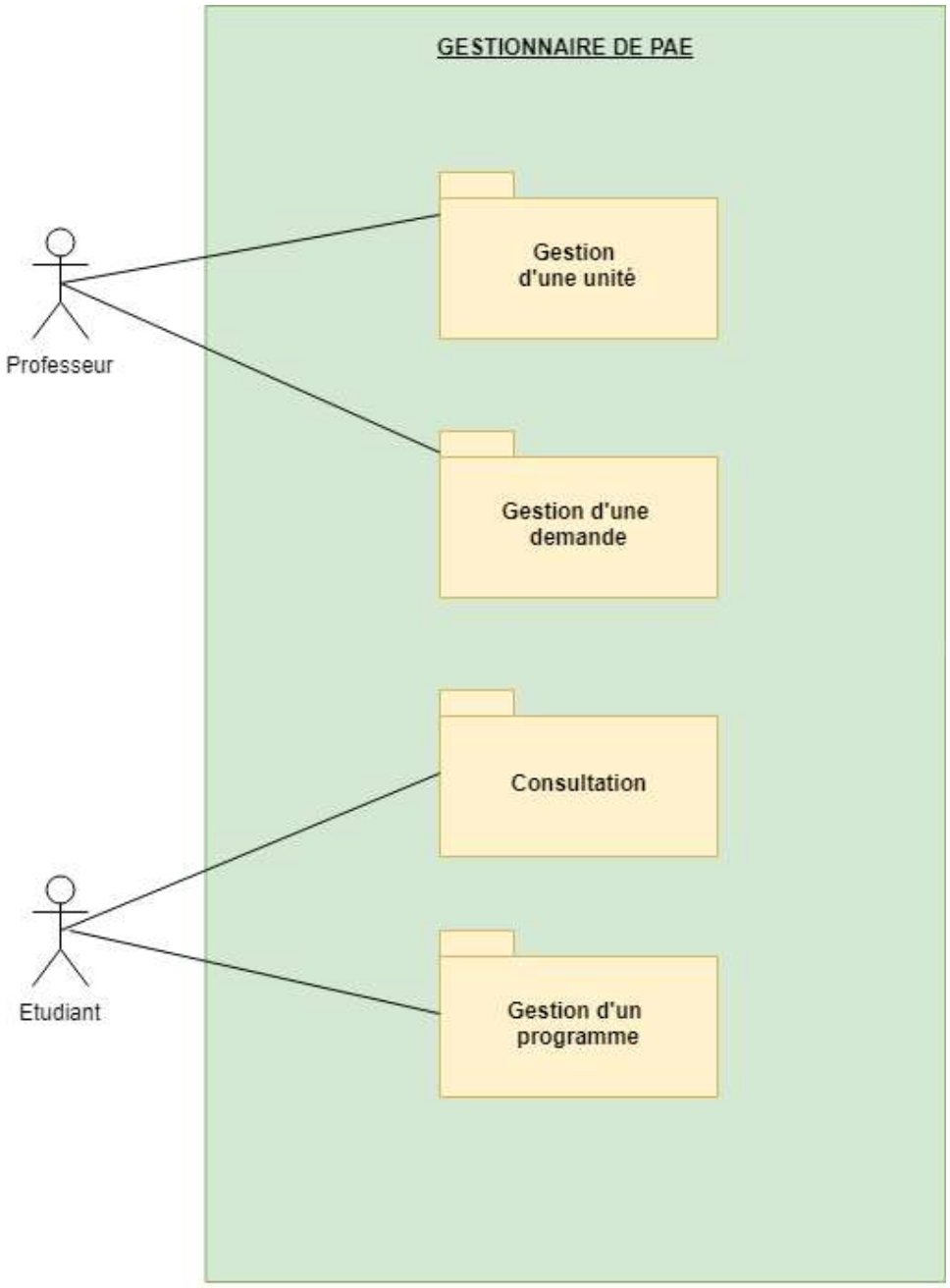


Diagramme de package

4. Diagramme de cas d'utilisation:

A partir de notre de nos analyse précédente, nous pouvons réaliser notre diagramme de cas d'utilisation.

Diagramme de cas d'utilisation: représente les fonctionnalités (ou dit cas d'utilisation) nécessaires aux utilisateurs. On peut faire un diagramme de cas d'utilisation pour le logiciel entier ou pour chaque package.

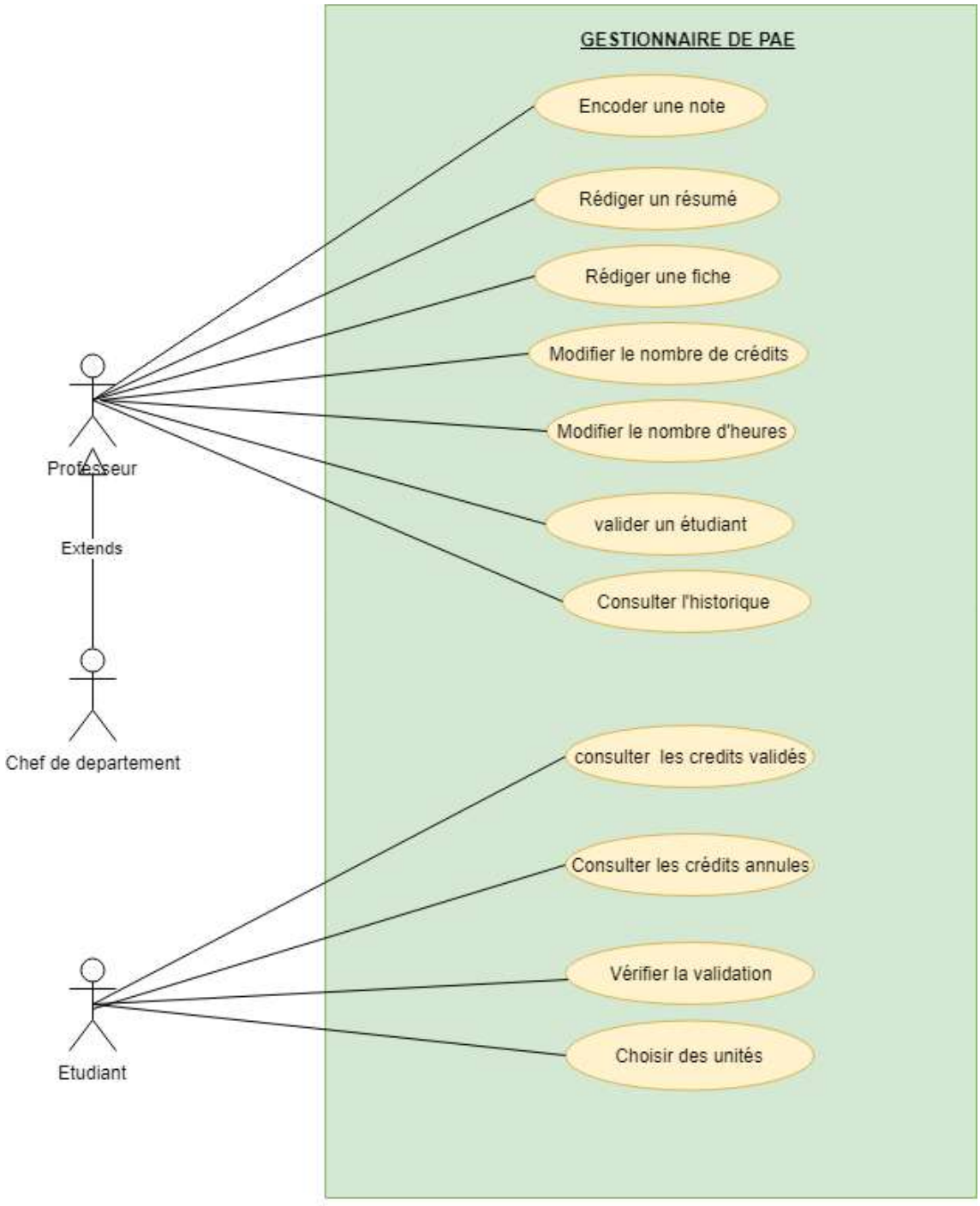
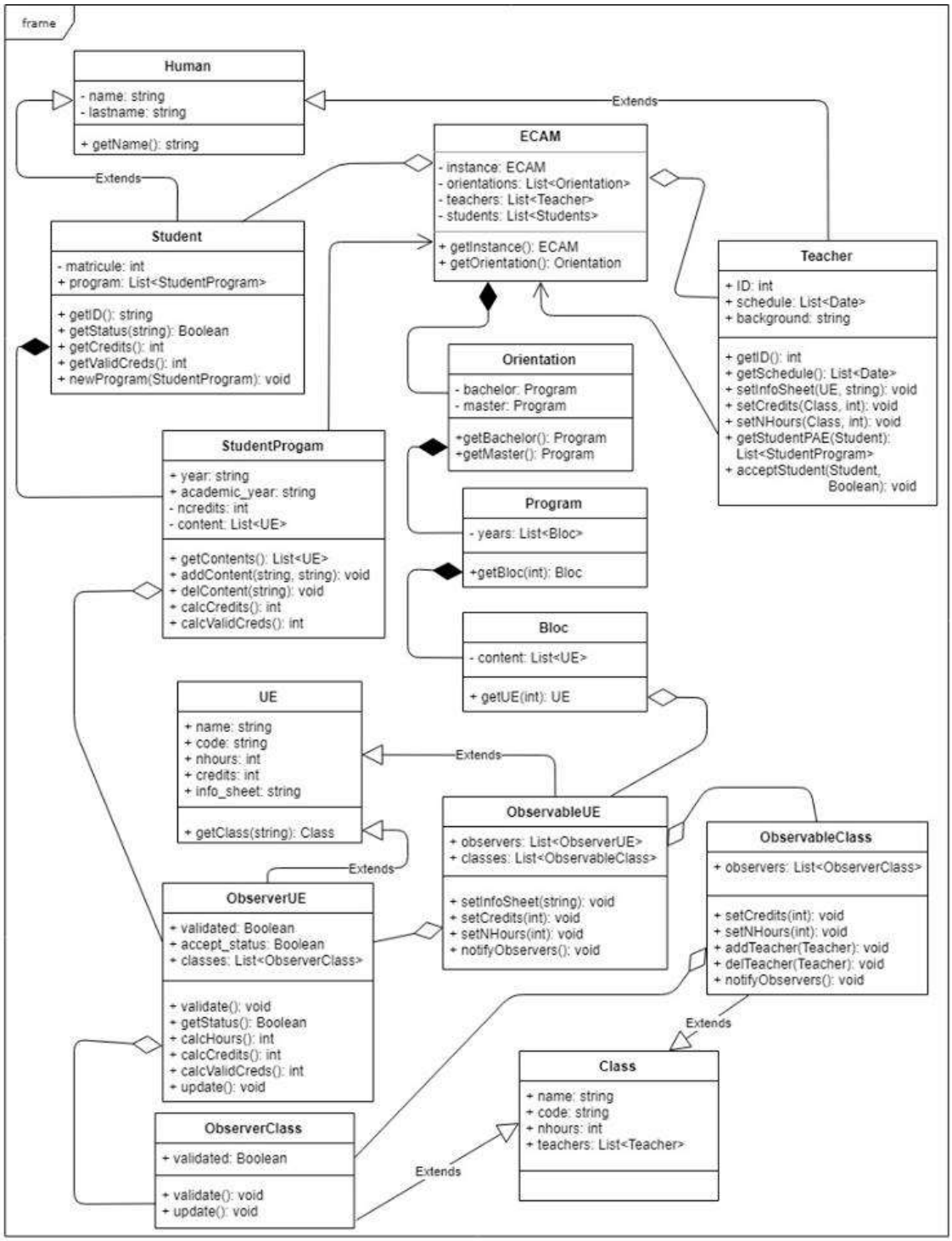


Diagramme de cas d'utilisation

5. Diagramme de classe

Afin de représenter nos objets et leurs différentes relations, nous avons réalisé le diagramme de classe ci-dessus.



6. Livrables à venir

1. Un diagramme d'activité haut niveau qui donne le fonctionnement global de l'application

2. Des diagrammes de séquence pour montrer l'enchaînement des actions pour les parties un peu plus abstraites comme l'implémentation du design pattern observateur (notamment pour les classes observerUE, ObserverbaleUE, Observeclass, Observableclass)

3. La réalisation de nos métriques

4. Et bien sûr, l'application des exigences et des remarques de notre chère prof Combéfis.

Nos métriques:

Densité des commentaires: nous allons commenter un maximum de ligne de code. Notamment

Couverture de code: nous effectuerons des tests unitaires pour les différentes entités et les nos différentes fonctionnalités.

Couplage: nous allons essayer d'avoir un couplage minimum