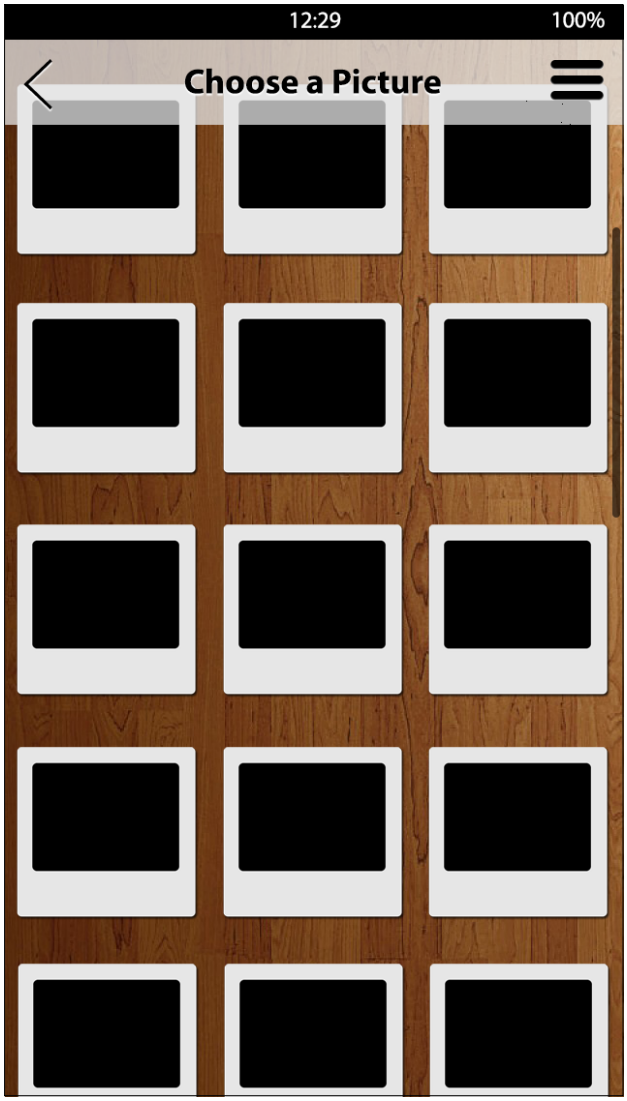


CAHIER D'ANALYSE

Préparé pour : Sebastien Lefevre

Préparé par : Rouillé Jérôme, Loriot Sacha, Haddad Mehdi, Catiau-Tristant Florent

22 avril 2014

Thème du projet : Logiciel de retouche photo pour PDA

SOMMAIRE

I. Objectif du projet

Ce projet a pour but la réalisation d'une application Java destiné à un « PDA ». A l'aide de nos connaissances UML, nous avons réalisé un diagramme d'analyse servant de modèle afin de concevoir le logiciel demandé. Nous avons également produit des diagrammes de séquence « boite noire » afin de comprendre la relation interface / utilisateur. En plus de ça, le design de l'application a été réalisé (via « photoshop ») pour nous procurer un visuel du logiciel.

Suite à cela, nous devrons entamer la phase « codage », la partie la plus chronophage. En effet, nous allons devoir tester, débuguer et optimiser notre code afin de faire fonctionner notre logiciel au mieux.

Lors de la réalisation de ce projet, nos connaissances d'expressions seront mises en jeu. A la suite de cela, nous devrons réaliser un cahier d'analyse, une présentation orale et mettre en oeuvre un travail de groupe organisé et efficace.

II. Description des fonctionnalités

PARTIE RÉGLAGES

- Ecraser ou nouvelle image par défaut
- Méthode : equals() (test d'égalité d'images)
- Type de galerie :
    - Aperçue d'image sous forme de carrés
        - 2 ou 3 colonnes
    - *Coverflow (OPTIONNEL)*

PARTIE RETOUCHE

HEADBAR:

- **Retour :** Annule la dernière action appliquée.
- **Rétablir :** Rétablie une action annulée.
- **Appliquer :** Sauvegarde l'image (ouvre une pop-up "Ecraser ou créer nouveau fichier ?").
- **Annuler :** Annule toutes les modifications de l'image jusqu'à la précédente sauvegarde.
- **Quitter :** Reviens à l'écran d'accueil (pop-up "voulez-vous enregistrer ?").

TABBAR:

Effets :

- **Flou (Netteté globale) :** Diminue la netteté de l'image.

- **Pixélisation :** Les zones de couleur homogène passe au format d'un pixel. Flou pixelisé.
- **Augmenter netteté, contrastes :** Accroit les nuances, diminue le flou.
- *(OPTIONNEL)* ***Opacité :*** *Augmente la transparence de l'image.*
- *(OPTIONNEL)* ***Filtre couleur :*** *Applique une augmentation de la couleur choisie (ex : image bleutée).*
- *(OPTIONNEL)* ***Cadre :*** *Un cadre sera ajouté à l'image sélectionnée. Cependant, ce cadre ne devra en aucun cas faire obstruction à quelconque partie de l'image choisie. Nous définirons des cadres. (Nous devons nous occuper de l'adaptation du cadre à une image.)*

Sélection :

- **Lasso :** Sélectionne selon une forme non régulière (suis le tracé du pointeur). Possibilité de choisir une sélection interne ou externe à la zone formée.
- *(OPTIONNEL)* ***Baguette magique :*** *Sélection automatique de pixels selon les zones de couleur homogènes.*

Filtres :

- **Sépia :** Colore l'image avec des nuances de bruns, donne un effet vieillis.
- **Black & White :** Enlève les couleurs de l'image, devient noire & blanche.
- **Bande dessinée :** Applique un filtre de style bande dessinée à l'image.
- **Négatif :** Inverse les couleurs d'origine de l'image (par exemple le rouge devient cyan).

Sizing :

- **Redimensionner :** Rognage manuel (barres verticales et horizontales qui délimitent le rognage) ou Rognage automatique (4/3 ou 16/9).
- **Rotation :** Permet de faire une rotation de l'image à 90°,180° ou 270°.
- *(OPTIONNEL)* ***Miroir :*** *Applique une rotation sur un axe vertical pour afficher le reflet de l'image.*

III. L'avancement

Déjà réalisé :

- Diagrammes de séquence boite noire
- Représentation de l'interface graphique
- Enumération des fonctionnalités de l'application
- Diagramme de classe d'analyse

A réaliser :

- Diagramme de classe fonctionnel
- Codage

Outils utilisés lors de la phase de conception en amont :

- Visual Paradigm for UML (Diagrammes)
- Photoshop (Réalisation de l'interface)
- Google drive (mise en commun des documents)

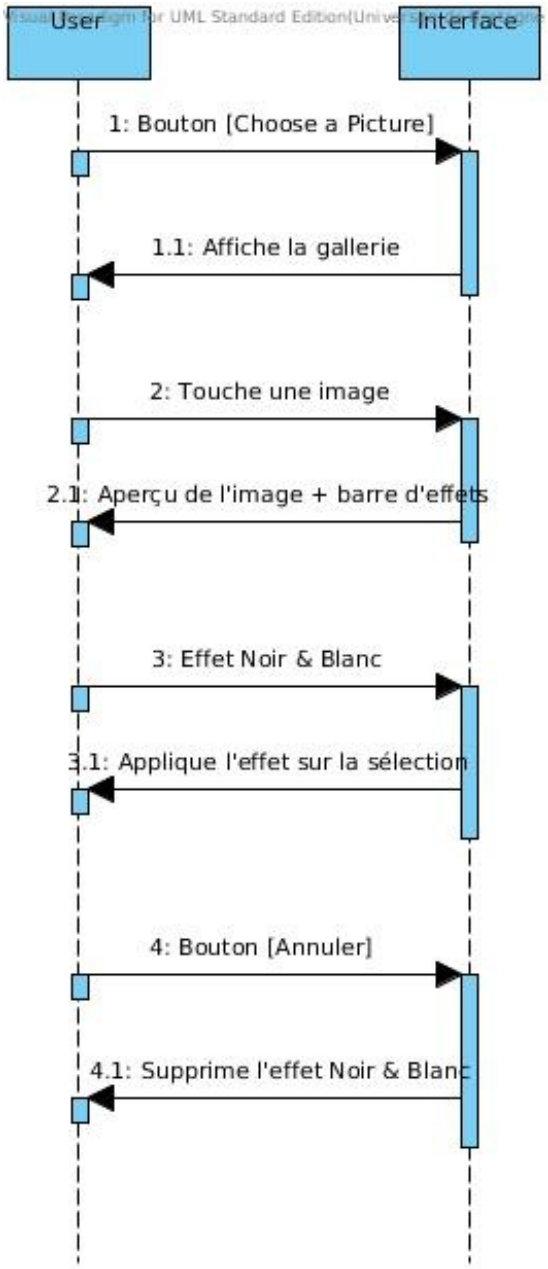
Outils prévus pour la suite du projet :

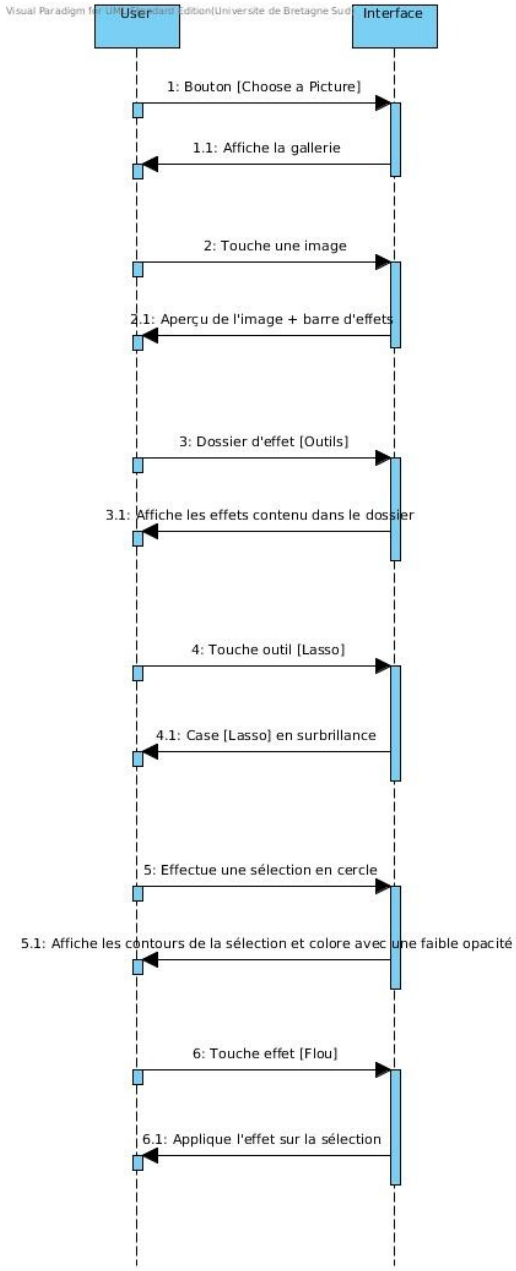
- Langage de programmation : Java
- Eclipse ?

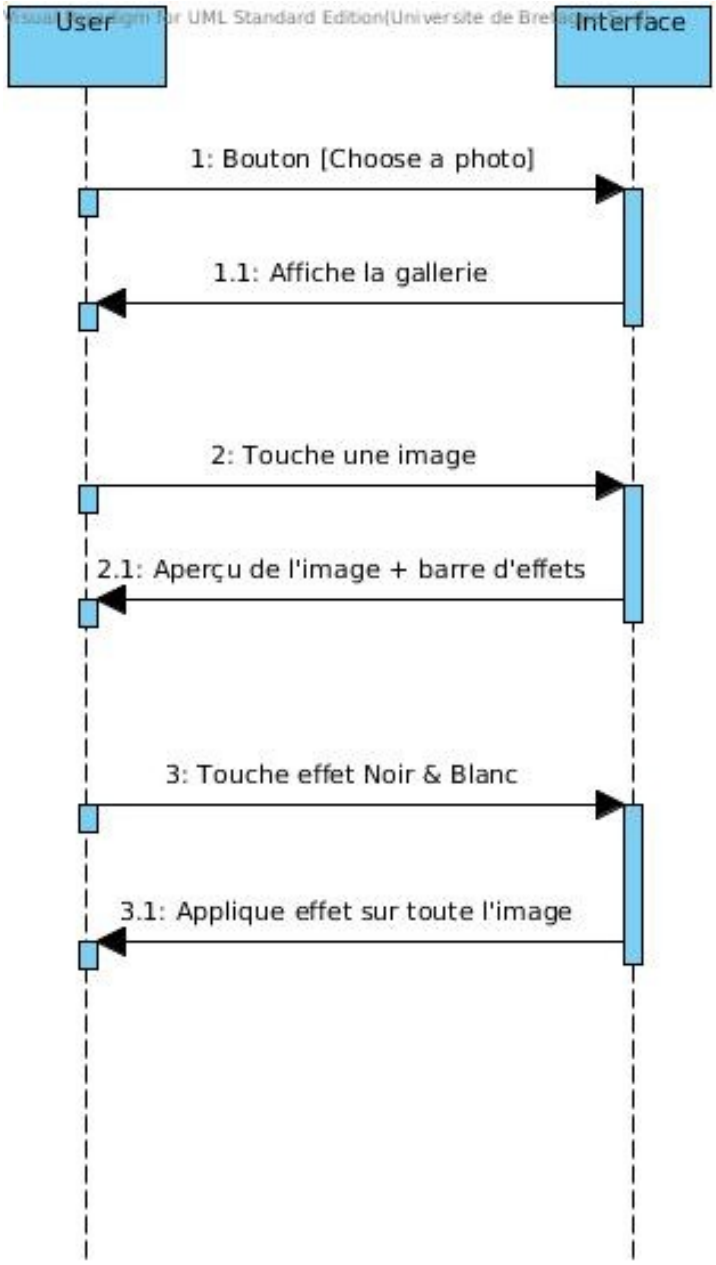
Connaissances développées depuis le début du projet :

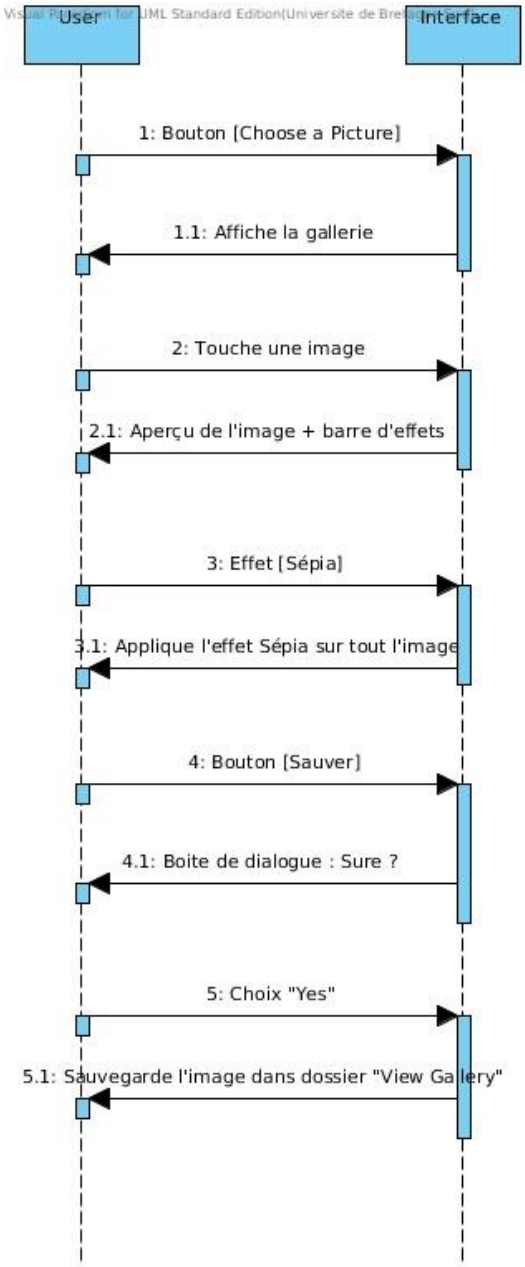
- Gestion de groupe, organisation des tâches
- Respect d'un projet
- Réflexion sur un diagramme de classes d'analyse à partir de peu

IV. Diagramme de séquence boite noire






User
Interface
1: Bouton [Choose a photo]
1.1: Affiche la gallerie
2: Touche une image
2.1: Aperçu de l'image + barre d'effets
3: Touche effet Noir & Blanc
3.1: Applique effet sur toute l'image


User
Interface
1: Bouton [Choose a Picture]
1.1: Affiche la gallerie
2: Touche une image
2.1: Aperçu de l'image + barre d'effets
3: Effet [Sépia]
3.1: Applique l'effet Sépia sur tout l'image
4: Bouton [Sauver]
4.1: Boite de dialogue : Sure ?
5: Choix "Yes"
5.1: Sauvegarde l'image dans dossier "View Gallery"

V. Diagramme de classe d'analyse

RETRO-CONCEPTION DU PDA :

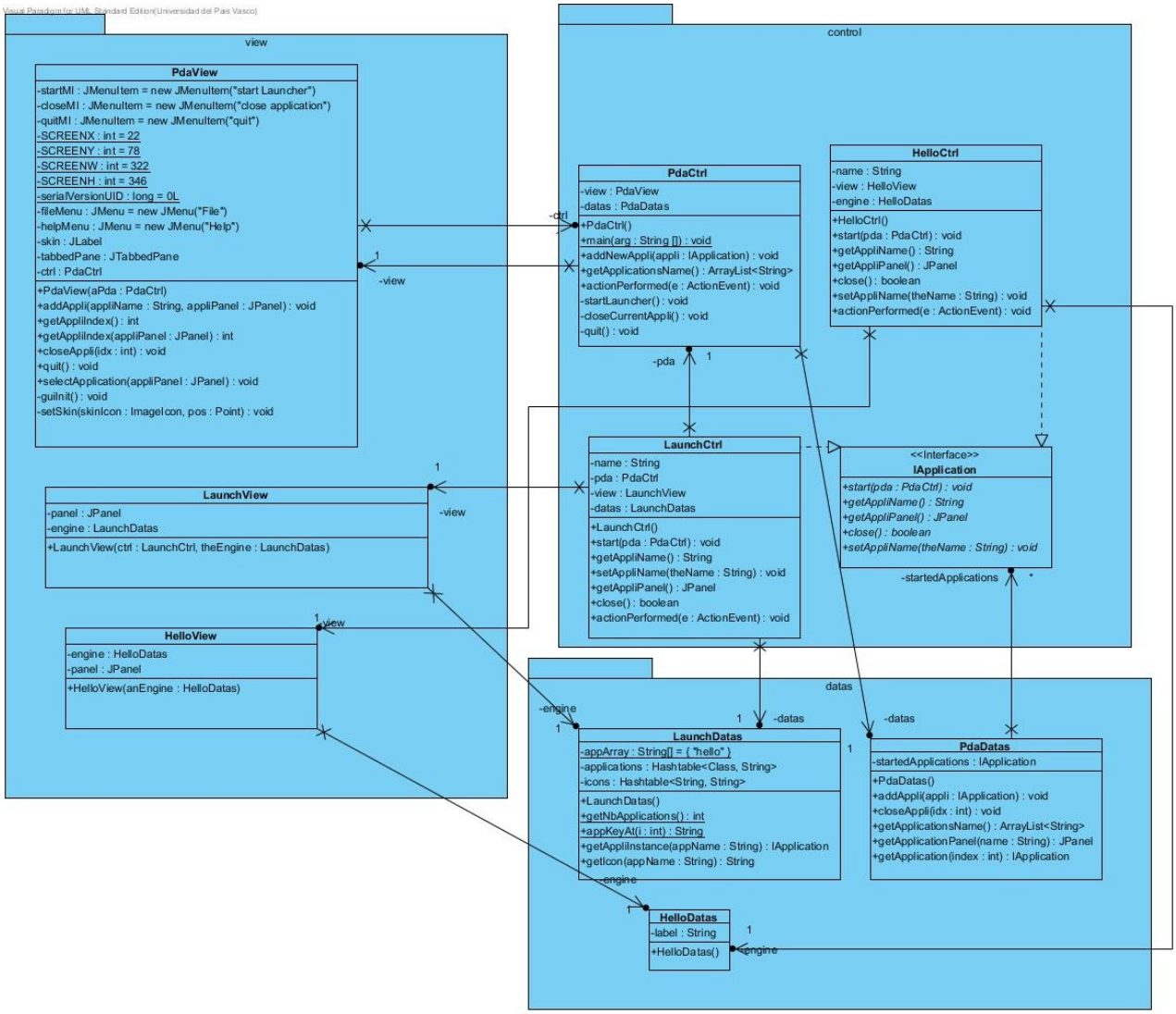
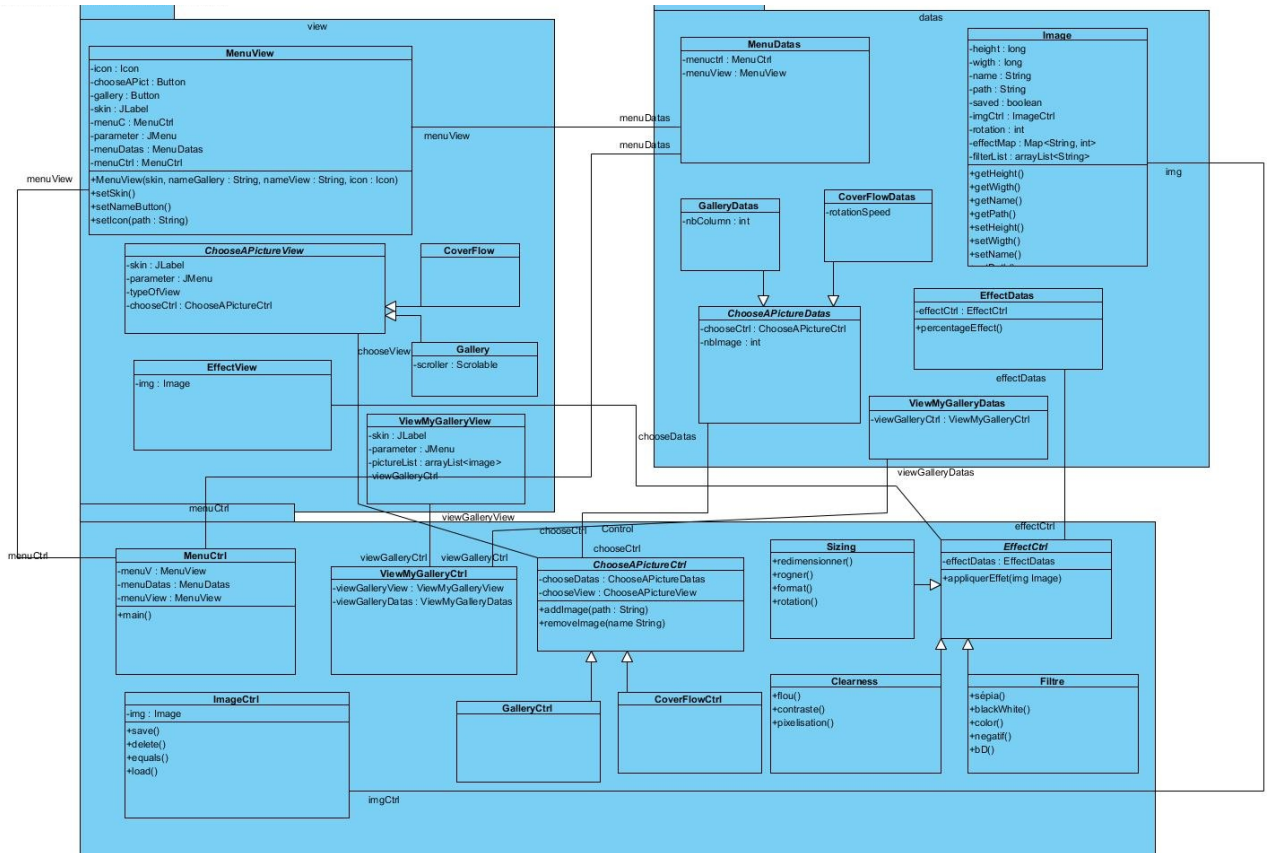


DIAGRAMME DE CLASSE DU LOGICIEL :



VI.Scénarios de test

I) ANNULER UNE ACTION

Pré-requis :

Image chargée

Effet flou (50%) appliqué

Test :

Clic sur bouton annuler dernière action

Résultat :

Être précis sur le résultat attendu : Comment vérifier le résultat ?

Résultat méthode equals() entre l'image d'origine et la nouvelle = true

II) ROTATION 90°

Pré-requis :

Image chargée

Test :

Effet rotation (2 fois)

Résultat :

attribut 'rotation' = 180°

résultat equals() entre l'image d'origine et la nouvelle = false

III) PARAMÈTRE 3 COLONNES SUR LA GALLERIE

Pré-requis :

Etre sur la page 'gallery' avec paramètre 2 colonnes ou Coverflow

Test :

Cliquer sur paramètre (icone en haut à droite)

Choisir sur la partie 'gallery' : 3 colonnes

Résultat :

3 colonnes d'affichées

attribut nbColumn de la class GalleryDatas = 3

attribut typeOfView de la class ChooseAPictureView = "gallery"

IV) APPLIQUER UN FILTRE

Pré-requis :

Image chargée.

Test :

Appliquer un filtre quelconque. Prenons ici pour exemple le filtre "Sépia".

Résultat :

L'image est désormais modifiée avec un filtre. Pour le vérifier, regardons de plus près l'attribut "filterList" de la classe "Image". Celui-ci doit contenir la chaîne de caractère "Sepia". De plus nous devons exécuter la méthode "equals" afin de s'assurer que l'image précédente et l'image avec le filtre sepia sont belle et bien différente. Equals() doit donc renvoyer "false"

V) APPLIQUER UN EFFET

Pré-requis :

Image chargée.

Test :

Appliquer un effet quelconque. Prenons ici pour exemple l'effet "Flou".

Résultat :

L'image est désormais modifiée avec un filtre. Pour le vérifier, regardons de plus près l'attribut "effectMap" de la classe "Image". Celui-ci doit contenir la chaîne de caractère "Flou" ainsi que le pourcentage auquel nous avons poussé l'effet lorsque nous avons appliqué ce dernier. De plus nous devons exécuter la méthode "equals" afin de s'assurer que l'image précédente et l'image avec l'effet "Flou" sont belle bien différente. Equals() doit donc renvoyer "false".

VII.Planning prévisionnel

	Nom de la tâche	Début	Fin
1	Présentation du projet	17/03/14	17/03/14
2	Rétro-conception PDA	17/03/14	28/03/14
3	Réflexion des IHM	29/03/14	03/04/14
4	Diagramme de séquence boite noire	29/03/14	03/04/14
5	Architecture et diagramme d'analyse	04/04/14	10/04/14
6	Recensement des scénarios de test	11/04/14	17/04/14
7	Rédaction du Cahier d'Analyse	18/04/14	21/04/14
8	Rendu du Cahier d'Analyse	22/04/14	22/04/14
9	Bilan de la phase d'analyse	22/04/14	25/04/14
10	Analyse du code du PDA	22/04/14	25/04/14
11	Déploiement de logiciels java	22/04/14	10/05/14
12	Conception des test unitaires	11/05/14	18/05/14
13	Bilan de la préparation au codage	19/05/14	23/05/14
14	Début du codage	24/05/14	24/05/14
15	Codage du logiciel	24/05/14	18/06/14
16	Préparation à la présentation orale	14/06/14	18/06/14
17	Rendu du projet	19/06/14	19/06/14
18	Présentation orale	20/06/14	20/06/14