

Projet Java ESIAL 2011-2012



Application cliente Picasa Web pour la gestion de contenus multimédias.

Travail réalisé par :

- Yassine El-Hajri
- Yoann Tritz
- Robin Bressan
- Jocelyn Forest



Sommaire

Application cliente Picasa Web pour la gestion de contenus multimédias.	1
1. Rappels des objectifs.....	2
2. Détails des fonctionnalités de l'application	3
1. Fonctionnalité sur les albums.....	3
2. Fonctionnalités sur les photos.....	4
3. Recherche de photo par mot clé.....	5
3. Architecture du produit et déroulement du projet.....	6
4. Manuel d'utilisation.....	8
Répartition du travail et feuille de temps	9
Conclusion	10
Annexe.....	11

1. Rappels des objectifs

L'objectif du projet est la réalisation en Java d'une application cliente Picasa Web pour l'organisation de contenus multimédias. **Picasa** est un logiciel de gestion d'images, et se décline aussi en une application de visualisation et d'organisation sur le Web de photographies.

L'application doit permettre de visualiser et manipuler des photos rangées dans des albums. Les fonctionnalités de l'application doivent prendre en compte à la fois l'organisation des contenus multimédias, les commentaires, descriptions et tags.

Le projet repose donc une interface graphique solide avec le maximum d'implémentations à réaliser.

2. Détails des fonctionnalités de l'application

Les fonctionnalités implémentées sont celles demandées dans la consigne avec plusieurs options.

Fenêtre de connexion

Nous avons implémenté une fenêtre de connexion Picasa pour accéder à l'application. Si le login et le mot de passe correspondant ne sont pas valides, une fenêtre indique que les identifiants sont incorrectes.

Une fois connecté, une fenêtre de chargement apparaît en attente de de l'application Picasa.

Interface générale de l'application

Menu

Le bouton menu permet soit de se déconnecter, soit de quitter directement l'application.

Aide

Le bouton Aide permet d'accéder au guide qui décrit toutes les fonctionnalités de l'application.

1. Fonctionnalité sur les albums

La fenêtre Album contient tous les albums de l'application. L'utilisateur a la possibilité d'ajouter un album, de supprimer un des albums de la liste ou de visualiser les photos contenues dans l'album sélectionné.

Ajout d'un l'album

Pour ajouter un album dans la liste, l'utilisateur doit lui donner un titre, éventuellement une description, et préciser s'il s'agit d'un album privé ou public.

Suppression d'un album

Pour supprimer un album de la liste, l'utilisateur doit sélectionner l'album à supprimer dans la liste. Une demande de confirmation de suppression de l'album est alors demandée.

Visualisation des photos

Dans l'onglet photos, l'utilisateur peut visualiser en miniature sous forme de liste toutes les photos contenues dans l'album sélectionné.

Ajout des photos dans un album

Dans l'onglet Upload, l'utilisateur peut ajouter une ou plusieurs photos à l'album sélectionné. Un tableau indique les photos sélectionnées, ainsi que leur taille en mémoire. Une fois les photos uploadé dans l'album, le tableau se vide.

Diaporama de l'album

L'utilisateur peut visualiser un diaporama de l'album sélectionné.

2. Fonctionnalités sur les photos

Les fonctionnalités sur les photos sont variées, aussi bien sur la visualisation en elle-même, que sur ses caractéristiques.

Visualisation de la photo

En double cliquant sur la photo choisie, une fenêtre avec la photo apparaissant en grand apparaît. Plusieurs options sont alors possibles :

- Effectuer un zoom sur la photo
- Faire apparaître la photo en plein écran
- Effectuer une rotation droite ou gauche de la photo
- Faire apparaître la photo en noir et blanc
- Flouter la photo
- Mettre en évidence les contours des formes de la photo
- Visualiser le négatif de la photo

L'utilisateur a la possibilité de simplement visualiser ces modifications ou d'uploader la photo modifiée.

Suppression d'une photo

Pour supprimer une photo de l'album, l'utilisateur doit sélectionner la photo à supprimer. Une demande de confirmation de suppression est alors demandée.

Déplacement d'une photo

L'utilisateur a la possibilité de déplacer une photo d'un album à un autre. Il doit alors sélectionner l'album de destination.

Copie d'une photo

L'utilisateur peut copier une photo sélectionnée dans un autre album. Il doit alors sélectionner l'album de destination.

Redimensionnement de la photo

L'utilisateur peut redimensionner la photo sélectionnée en modifiant la largeur et la longueur. La photo modifiée sera alors uploadée.

Description de la photo

L'utilisateur peut ajouter ou modifier la description d'une photo.

Tag de la photo

L'utilisateur peut ajouter des tags sur la photo sélectionnée.

Commentaires de la photo

L'utilisateur peut visualiser et ajouter des commentaires sur la photo selon ses droits.

Localisation de la photo

L'utilisateur peut visualiser sur une carte le lieu où la photo a été prise.

Impression de la photo

L'utilisateur peut imprimer la photo sélectionnée.

3. Recherche de photo par mot clé

L'utilisateur peut effectuer une recherche de photo par tag. En tapant un mot clé, on sélectionne les photos correspondantes dans les albums privés ou publics. Afin de limiter en nombre l'apparition des photos, l'utilisateur peut indiquer combien il souhaite en faire apparaître au maximum.

3. Architecture du produit et déroulement du projet

Le projet s'est effectué sur 3 mois avec entre-temps quatre itérations qui ont permis de faire état de l'avancement du projet. Nous allons détailler les différentes étapes de l'avancement du projet.

1^{ère} étape : Renseignement et utilisation des bibliothèques de l'API existantes, surtout pour la connexion et la synchronisation avec PicasaWEB.

2^{ème} étape : Notre implémentation repose sur l'architecture MVC. Nous avons donc décidé de concevoir avant tout le modèle de l'architecture et d'effectuer nos premiers tests en ligne de commande. Nous avons utilisé l'outil d'analyse JUnit. L'implémentation du modèle a pris peu de temps au niveau de sa conception, mais a nécessité plus de temps pour respecter le code de l'API.

Les premières classes implémentées sont AlbumsModel, PhotoModel, Comment, Tag et ConnectModel.

Nous avons des difficultés au niveau des tags car les ITag ne sont pas à jour.

3^{ème} étape : Continuation et optimisation de l'implémentation du modèle en ajoutant les nouvelles méthodes. Le problème des ITag n'est pas toujours pas résolu. Sinon, les tests implémentés sont tous validés.

Les deux premières itérations ont donc été consacrées à une bonne documentation de l'API et à la programmation du modèle de notre architecture. Notre modèle étant complet, nous pouvons commencer l'interface graphique.

Nous faisons communiquer les vues avec le modèle grâce au patron de conception Observer. Ainsi la vue est toujours au courant des changements procédés.

4^{ème} étape : Nous entamons l'implémentation de l'interface graphique et la fenêtre de connexion. Nous séparons tous les composants de la fenêtre (login, mot de passe, structure de la fenêtre) dans des classes associées. Nous implémentons en même temps les classes du Controller correspondante. Nous utilisons également une fenêtre de chargement implémentée à l'aide de threads.

Concernant la fenêtre principale, nous regroupons tous les composants dans la MainFrame. Les différents composants sont les vues correspondant aux modèles définis : PictureBox, AlbumsView, CommentView, TagView. La classe GestionView permet la visualisation des trois onglets Photos, Upload et Details sur la fenêtre principale.

Au niveau des tags et des descriptions, nous avons leur affichage sur la fenêtre principale. Nous ne pouvons pas encore en ajouter ou supprimer.

Nous rencontrons un problème pour l'Upload des photos et pour récupérer leurs détails. Nous nous renseignons sur la classe abstraite AbstractTableModel et décidons de compléter le modèle avec les classes DtModel et UploadTableModel.

Nous commençons également à gérer le clic droit qui fera apparaître toutes les options possibles sur les photos.

5^{ème} étape : Nous entamons la recherche de photo par tag avec la classe PhotoSearch du modèle et la classe de Controller et de vue correspondante.

Par ailleurs, nous arrivons à créer un nouvel album(NewAlbumFrame), supprimer un album et à visualiser une photo en grand dans une autre fenêtre. La visualisation se fait temporairement avec un OptionPane et non une vraie fenêtre de visualisation.

La gestion du clic droit est maintenant au point et toutes les options sur les photos sont maintenant disponibles.

La gestion des tags et des descriptions est maintenant finalisée. Nous pouvons désormais les modifier, ajouter ou supprimer.

6^{ème} étape : Nous ajoutons de nombreuses classes dans le package Vue et Controller pour les différentes implémentations sur la visualisation des images. Par exemple, Redim Controller permet le redimensionnement de la photo. ModifimgController permet de lancer les boutons de retouche de la photo.

La visualisation de la photo se fait maintenant dans une vraie fenêtre de visualisation.

Au niveau de la connexion, nous avons également un contrôle avec la touche Enter.

Pour les opérations lourdes, nous améliorons la fluidité grâce à l'implémentation de threads et de fenêtre d'animation. Ainsi, par exemple, pour uploader une photo retouchée, une fenêtre d'attente apparaît.

Pour avoir une vision globale de l'architecture du projet et pour montrer le respect de la structure MVC de notre programmation, le diagramme de classe d'une des fonctionnalités de l'application est donné en annexe. Il ne s'agit que d'une version très simplifiée des principales classes et de leurs interactions entre elle. Les trois packages représente le modèle (en jaune), le Controller (en rouge) et la vue (en bleu).

4. Manuel d'utilisation

Pour se connecter, identifiez-vous avec le login et le mot de passe correspondant.

Manipulation des albums

Pour créer un nouvel album, cliquez sur le bouton de création de l'album. Une fenêtre apparaît. Inscrivez le titre de votre nouvel album, éventuellement une description et précisez si vous le souhaitez privé ou public.

Pour supprimer un album, cliquez sur l'album choisi, puis sur le bouton supprimer.

Pour consulter un album, cliquez sur l'album choisi. Il apparaît alors dans le premier onglet la liste des photos en miniature, dans le second l'Upload des photos et dans le dernier le détail des photos.

Pour ajouter une ou plusieurs photos dans un album, cliquez sur l'onglet Upload, ajoutez les photos souhaitées puis Uploader.

Pour lancer un diaporama de l'album, cliquez droit sur l'album, puis sur lancer diaporama. Un diaporama de l'album se lancera.

Manipulation des photos

Pour visualiser une photo dans une nouvelle fenêtre, double cliquez sur la photo miniature. Plusieurs options vous offre à vous : vous pouvez effectuer un zoom sur la photo, la faire apparaître en plein écran, flouter la photo, avoir le négatif ou mettre en évidence les contours de la photo. Si vous souhaitez conserver les modifications, cliquez sur enregistrer. La nouvelle photo sera ajoutée alors à l'album.

Pour déplacer, copier ou supprimer, cliquez droit sur la photo et sélectionnez votre choix.

Pour redimensionner la photo, cliquez sur redimensionner. Sélectionnez alors la nouvelle largeur et longueur souhaitée. Pour enregistrer la nouvelle photo, cliquez sur valider.

Pour imprimer la photo, cliquez sur imprimer, puis sélectionnez le mode d'apparence souhaité.

Pour voir ou modifier les commentaires de la photo, cliquez sur commentaires. Si vous souhaitez savoir où la photo a été prise, cliquez sur Localisation.

Pour modifier/ajouter la description, les tags de la photo, cliquez à droite sur les boutons correspondants.

Pour effectuer une recherche par tag, entrez votre mot clé. Les photos correspondantes apparaîtront.

Quitter l'application

Pour quitter l'application, cliquez sur menu déconnexion/quitter.

Répartition du travail et feuille de temps

Nous avons donc disposé de 3 mois pour réaliser ce projet avec un travail suivi et continu.

-Analyse du sujet, renseignement sur l'API et première conception du projet : **10 heures** en commun

-Importation de toutes les librairies et implémentation du modèle. Travail effectué en binôme pour séparer les tâches : **20 heures**

-Finition du modèle et résolution des bugs : Robin Bressan, Jocelyn Forest : **10 heures**

-Début de l'implémentation de l'interface graphique : Yoann Tritz, Yassine El-Hajri : **20 heures**

-Gestion des composants principaux de l'interface graphique : Yoann Tritz, Yassine El-Hajri : **30 heures**

-Gestion de l'Upload et des détails de la photo : Robin Bressan, Jocelyn Forest : **20 heures.**

-Gestion des multiples options disponibles sur les photos : Yoann Tritz : **30 heures.**

-Gestion de la fenêtre de tag et de description : Yassine El-Hajri : **20 heures.**

-Résolution du problème de la fenêtre de commentaires, résolution des bugs existants et rédaction du rapport : Robin Bressan, Jocelyn Forest : **20 heures.**

Total

Jocelyn Forest et Robin Bressan : 80h

Yassine El-Hajri : 100h

Yoann Tritz : 110h

Conclusion

Ce projet nous a permis de développer et de conduire en groupe un projet informatique pendant 3 mois. De la conception du projet jusqu'au résultat final, ce travail nous a permis de concilier nos outils de développeur afin d'aboutir à un produit logiciel aux fonctionnalités multiples.

L'évolution de notre travail a été évaluée selon un processus itératif afin que nous puissions gérer dans le temps l'état d'avancement de notre projet. La complexité et la longueur du projet nécessitait une répartition et une synchronisation du travail entre les membres du groupe.

Ce projet nous donne ainsi donc l'opportunité de rendre compte des travaux qui pourront nous être demandé en entreprise et de la manière de les effectuer.

Le projet est disponible au lien suivant :

<http://redmine.esial.uhp-nancy.fr/svn/java-g9>

Annexe

(Nous n'avons pas fait figuré les opérations pour chaque classe, par souci de clarté).

