# Table des matières

[Table des matières 1](#_Toc188225520)

[Table des figures 2](#_Toc188225521)

[INTRODUCTION 3](#_Toc188225522)

[1 Etude De L’existant 4](#_Toc188225523)

[1.1 Logiciels existants pour la lecture de bandes dessinées 4](#_Toc188225524)

[1.1.1 Présentation des outils populaires 4](#_Toc188225525)

[1.1.2 Points forts et limites des solutions existantes 5](#_Toc188225526)

[1.2 Formats de fichiers utilisés 5](#_Toc188225527)

[1.2.1 Formats d’archives 5](#_Toc188225528)

[1.2.2 Formats d’images 5](#_Toc188225529)

[1.2.3 Limites des formats actuels 5](#_Toc188225530)

[1.3 Technologies et algorithmes pertinents 6](#_Toc188225531)

[1.3.1 Algorithmes de redimensionnement d’image 6](#_Toc188225532)

[1.3.2 Gestion de la mémoire et mise en cache 6](#_Toc188225533)

[1.3.3 Outils et bibliothèques existants 6](#_Toc188225534)

[1.4 Synthèse et opportunités pour le projet 6](#_Toc188225535)

[2 ANALYSE DES BESOINS 7](#_Toc188225536)

[3 CONCEPTION GLOBALE 7](#_Toc188225537)

[3.1 Architecture logicielle 7](#_Toc188225538)

[3.2 Diagramme de contexte 7](#_Toc188225539)

[3.3 Diagramme de cas d’utilisation 7](#_Toc188225540)

[3.4 Diagramme de classes global 7](#_Toc188225541)

[3.5 Diagramme de déploiement 7](#_Toc188225542)

[4 CONCEPTION DETAILLEE 7](#_Toc188225543)

[4.1 Gestion des livres et fichiers 8](#_Toc188225544)

[4.2 Diagrammes de séquence 8](#_Toc188225545)

[4.3 Diagramme de communication 8](#_Toc188225546)

[5 TEST ET VALIDATION 8](#_Toc188225547)

[6 CAHIER D’ANALYSE 9](#_Toc188225548)

[7 CAHIER DE CONCEPTION 9](#_Toc188225549)

[CONCLUSION 20](#_Toc188225550)

# Table des figures

# INTRODUCTION

Les bandes dessinées, au croisement de l’art graphique et de la narration, occupent une place privilégiée dans le cœur des amateurs de littérature visuelle. Avec l’évolution des formats numériques, la nécessité d’outils dédiés pour leur lecture et leur création s’est accentuée. Ce rapport présente le développement d’un projet intitulé **Comic Book Reader/Writer**, une application visant à offrir une expérience optimale pour lire, gérer, et créer des bandes dessinées numériques.

Le projet repose sur des objectifs ambitieux : une prise en charge étendue des formats d’archives et d’images courants tels que .CBR, .CBZ, .JPEG, et .PNG, tout en garantissant une interface intuitive et performante. En outre, il intègre des fonctionnalités avancées comme le redimensionnement asynchrone des pages, l’application de filtres pour améliorer la lisibilité des textes, et la gestion d’une bibliothèque numérique enrichie.

L’approche adoptée pour ce projet se concentre sur l’évolution et la modularité, avec une architecture logicielle permettant d’ajouter facilement de nouvelles fonctionnalités. Ce rapport retrace les différentes étapes de conception et de réalisation de cette application, en s’appuyant sur une analyse approfondie des besoins utilisateurs et des technologies disponibles.

# Etude De L’existant

Avant d’entamer le développement de l’application Comic Book Reader/Writer, une étude approfondie de l’existant s’impose pour comprendre les outils disponibles, identifier leurs forces et faiblesses, et positionner notre solution de manière unique. Cette analyse se concentre sur les logiciels populaires existants pour la lecture et la gestion de bandes dessinées numériques, les formats de fichiers utilisés, ainsi que les technologies et algorithmes pertinents.

## Logiciels existants pour la lecture de bandes dessinées

### Présentation des outils populaires

Les principaux logiciels disponibles incluent :

* CDisplayEx: Un logiciel largement utilisé pour lire des bandes dessinées dans des formats comme .CBR et .CBZ. Il offre une interface simple et légère, mais manque de fonctionnalités avancées telles que l’organisation des bibliothèques ou la modification des ouvrages.
* ComicRack : Une solution complète permettant non seulement de lire des bandes dessinées mais aussi de les organiser en bibliothèque. Cependant, le projet n’est plus maintenu, ce qui limite son attrait à long terme.
* YACReader : Un outil multiplateforme qui inclut une interface de lecture ergonomique et une gestion efficace des bibliothèques, mais il n’offre pas de capacités de modification des ouvrages.
* SumatraPDF : Bien qu’il ne soit pas spécifiquement conçu pour les bandes dessinées, ce lecteur PDF supporte les formats .CBR et .CBZ, mais avec des fonctionnalités limitées en termes d’optimisation pour les contenus graphiques.

### Points forts et limites des solutions existantes

|  |  |  |
| --- | --- | --- |
| Fonctionnalités | Points forts des solutions actuelles | Limites observées |
| Formats pris en charge | Large support des formats .CBR et .CBZ | Peu de prise en charge d'autres formats comme .PDF ou .TIFF |
| Navigation | Interfaces intuitives pour lire page par page | Navigation parfois lente sur des ouvrages volumineux |
| Qualité d’affichage | Lecture en pleine page et options de zoom | Peu d’optimisations pour améliorer la lisibilité du texte |
| Organisation | Gestion de bibliothèques (ComicRack, YACReader) | Manque d’intégration des métadonnées et des tags |
| Personnalisation | Options de zoom et de mode de lecture | Absence de filtres avancés pour améliorer les images |

## Formats de fichiers utilisés

### Formats d’archives

Les formats d’archives .CBR et .CBZ sont les standards pour les bandes dessinées numériques.

* .CBR : Basé sur le format .rar, il offre une compression efficace mais nécessite des bibliothèques spécifiques pour la décompression.
* .CBZ : Basé sur le format .zip, plus universel et supporté par la majorité des logiciels.

### Formats d’images

Les bandes dessinées numériques contiennent généralement des images compressées dans divers formats :

* JPEG (.jpg) : Format populaire pour les images compressées avec pertes, mais souvent sujet à des artefacts visuels, notamment autour du texte.
* PNG (.png) : Format sans perte adapté aux images nécessitant une haute qualité.
* TIFF (.tiff) : Format utilisé pour les images monochromes ou nécessitant une précision élevée, bien que plus lourd en termes de taille.

### Limites des formats actuels

* Les formats d’archives comme .CBR et .CBZ ne contiennent pas de métadonnées enrichies (par exemple, tables des matières ou tags).
* La qualité d’affichage des formats compressés (JPEG) peut être insuffisante pour les textes et les détails graphiques.

## Technologies et algorithmes pertinents

### Algorithmes de redimensionnement d’image

La qualité du redimensionnement est cruciale pour garantir une bonne expérience utilisateur. Les algorithmes courants incluent :

* Nearest Neighbor (proche voisin) : Rapide mais produit des images pixélisées.
* Bilinear Interpolation : Offre un rendu plus lisse mais peut manquer de netteté pour le texte.
* Lanczos Resampling : Algorithme avancé qui préserve mieux les détails, adapté aux bandes dessinées avec des graphismes complexes.

### Gestion de la mémoire et mise en cache

* Les images décompressées peuvent consommer une grande quantité de mémoire. Une stratégie de mise en cache efficace est essentielle pour garantir la fluidité lors de la navigation entre les pages.
* Le préchargement asynchrone des pages suivantes améliore considérablement l’expérience utilisateur.

### Outils et bibliothèques existants

* ImageMagick : Bibliothèque puissante pour le traitement des images (redimensionnement, filtrage).
* Libarchive : Pour manipuler divers formats d’archives, y compris .zip et .rar.
* Qt et wxWidgets : Cadres populaires pour développer des interfaces graphiques multiplateformes.

## Synthèse et opportunités pour le projet

L’analyse de l’existant met en évidence un besoin de solutions innovantes dans plusieurs domaines :

* Élargissement des formats supportés : Les outils actuels se limitent souvent à .CBR et .CBZ. Le support de formats comme .PDF, .TIFF, et .PNG serait un atout majeur.
* Amélioration de la qualité d’affichage : Intégrer des filtres spécifiques pour améliorer la lisibilité des textes (correction d’artefacts JPEG, suppression de bruit) est essentiel.
* Personnalisation et enrichissement des ouvrages : Ajouter des métadonnées comme des tables des matières ou des tags pour organiser les bibliothèques est une opportunité clé.
* Performances et fluidité : L’utilisation de techniques modernes comme le préchargement asynchrone et la mise en cache intelligente peut significativement améliorer l’expérience utilisateur.

# ANALYSE DES BESOINS

Exigences fonctionnelles, non fonctionnelles….

# CONCEPTION GLOBALE

## Architecture logicielle

Description des modules principaux : interface utilisateur, gestion des fichiers, traitement des images, etc.

## Diagramme de contexte

## Diagramme de Paquetage

## Diagramme de cas d’utilisation

## Diagramme de classes global

## Diagramme de déploiement

 Infrastructure matérielle et logicielle nécessaire.

 Exemple : composants sur un système local (application, bibliothèques tierces, etc.).

# CONCEPTION DETAILLEE

## Gestion des livres et fichiers

Description du flux de travail pour charger, afficher et organiser les livres

## Diagrammes de séquence

## Diagramme de communication

Échanges entre les objets pour ouvrir et afficher un fichier .CBZ ou .CBR.

Interface utilisateur et navigation

## Traitement des images

Explication des algorithmes utilisés pour le redimensionnement et le filtrage des images.

Diagramme d’activités : Application d’un filtre

Etapes nécessaires pour charger une image, appliquer un filtre, et afficher le résultat

## Gestion des bibliothèques

Gestion des collections de livres, tags et métadonnées

# TEST ET VALIDATION

## Stratégie de test

## Méthodes de validation des fonctionnalités principales (tests unitaires, tests fonctionnels)

Diagramme d’activités : Workflow de test

# CAHIER D’ANALYSE

# CAHIER DE CONCEPTION

Les exigences de l’application :

1. L’application doit prendre en charge les différents formats utilisés pour les bandes dessinées (CBR, CBZ …, TIFF, PDF envisageable)
2. L’application doit offrir une interface agréable et performante permettant de parcourir les différentes pages d’un livre au format .CBR ou au format .CBZ
   1. Prendre en charge éventuellement d’autres formats tels que TIFF, PDF…
3. L’application doit offrir un rendu graphique de qualité et un confort de lecture :
   1. Masquer le temps pris par les différentes tâches
   2. Anticiper la lecture et le redimensionnement des pages, en permettant de débuter la lecture du document sans pour autant attendre que l’ensemble des pages soit chargé en mémoire…
4. L’application devra offrir une qualité d’affichage soignée
   1. utiliser des filtres pour rehausser le texte
5. L’application devra créer des ouvrages au format .CBR, .CBZ : assembler différentes images et les placer dans un fichier de type .CBZ, .CBR
6. L’application devra offrir la possibilité d’ajouter une table des matières (
   1. Proposer une extension du format de fichier, typiquement par l’ajout d’un fichier toc (table of content) au format XML par exemple dans la liste des fichiers contenus dans l’archive .CBZ ou .CBR.
7. L’application devra éventuellement prendre en charge la gestion de bibliothèques
   1. Permettre l’annotation des documents par des tags
   2. Gérer plusieurs livres en cours de lectures
8. L’application doit être évolutive : etendre progressivement les formats souhaités ainsi que les algorithmes de traitement d’images

CLASSONS LES EN EXIGENCES FONCTIONNELLES ET NON FONCTIONNELLES

1. Exigences fonctionnelles

De façon peu exhaustive, nous avons essayé de regrouper les différentes fonctionnalités des applications CDisplayEx et YACReader que nous avons pensé essentielles à implémenter pour notre application.

D’abord**, CDisplayEx** est une référence dans le domaine des liseuses de bandes dessinées, elle a certaines spécificités notamment :

|  |  |
| --- | --- |
| **Lecture optimisée** | **Affichage et confort de lecture** |
| Lecture des fichiers dans les archives sans extraction manuelle. | Affichage en double page ou simple page. |
| Détection automatique de l’orientation des pages pour un affichage correct. | Zoom intelligent : ajustement automatique à la largeur ou à la hauteur de l’écran. |
| Navigation rapide grâce à des raccourcis clavier et souris. | Gestion des bandes dessinées japonaises (lecture de droite à gauche). |

Ensuite, CDisplayEx ne prenant pas en charge la gestion de bibliothèques, nous nous sommes penchés sur un autre logiciel, assez complet, multiplateformes et performantes comme modèles pour avoir une idée de comment celles-ci étaient implémentées. Ceci est un ensemble de fonctionnalités notables que nous pensons implémenter.

|  |  |
| --- | --- |
| **Gestion de bibliothèque** | **Expérience utilisateur** |
| Organisation des bandes dessinées par dossier et métadonnées. | Possibilité de marquer une page ou de mettre des favoris. |
| Création d’une bibliothèque locale avec des images de couverture. | Historique de lecture : savoir où l’utilisateur s’est arrêté. |
| Système de **tags** et de recherche avancée. |  |

**Tableau des exigences fonctionnelles de l’application**

Classées en 4 grandes catégories :

1. Lecture et Navigation
2. Affichage
3. Modification des ouvrages
4. Gestion de bibliothèques

|  |  |  |  |
| --- | --- | --- | --- |
| Lecture et Navigation | Affichage : interface agréable, performante, rendu graphique de qualité, confort de lecture | Modification des ouvrages | Gestion de bibliothèques |
| Doit Support des formats : .CBR, .CBZ, et extension future vers .pdf, .TIFF | Paramètres pour régler les marges. | Extraction et assemblage des pages | Organisation par dossier ou tags : annoter les documents par des tags. |
| Doit pouvoir afficher les premières pages sans délai. (renvoie à la lecture asynchrone en fait) | Parcourir les différentes pages d’un livre : Transitions fluides entre les pages. | Ajout et édition des tables des matières. | Ajout et édition des tables des matières. |
| L’Affichage doit être flexible :  simple page, double page, ou défilement continu (Cff YarcReader) | Options de filtres pour : Rehausser les textes, Optimiser les graphiques. | Création de livres personnalisés. | Recherche avancée par nom, tag, ou métadonnées. |
| Doit permettre la navigation facile :  Page suivante/précédente, dernière page, première page…  Accès direct à n’importe quelle page. | Détection automatique de l’orientation des pages. |  | Création de bibliothèques locales avec images de couverture. |
| Zoom intelligent : ajustement à la largeur, hauteur, ou plein écran. |  |  |  |
| Lecture inversée pour les bandes dessinées japonaises : (envisageable) |  |  |  |

Exigences non fonctionnelles

1. Exigences non fonctionnelles

4 principales exigences que nous pouvons scinder comme suit :

|  |  |  |
| --- | --- | --- |
| Performances | Architecture logicielle | Ergonomie et expérience utilisateur |
| Doit être Rapide et fluide  Chargement asynchrone des livres : les premières pages doivent être disponibles immédiatement, même si le reste du fichier est en cours de traitement.  Mise en cache des images déjà redimensionnées et filtrées pour éviter des recalculs inutiles.  Temps de réponse minimal (masquer le temps pour : Appliquer des filtres ou redimensionner une page.  Débuter la lecture du document sans pour autant que le reste soit terminé | Doit être modulaire et extensible  Conception modulaire permettant l’ajout de nouveaux formats de fichiers sans modification majeure du code : Par exemple, ajouter facilement des algorithmes de traitement d’image ou des formats comme .7z, .png, ou .pdf.  Utilisation de classes abstraites pour gérer les différents types de fichiers (images et archives) :  Classe abstraite pour les archives avec des spécialisations pour .rar, .zip, etc.  Classe abstraite pour les images, permettant le support d’extensions futures. | Confort de lecture  Transitions fluides entre les pages pour éviter des interruptions dans la navigation. |
| Optimisation mémoire  Gestion efficace des images décompressées, qui peuvent occuper jusqu’à 10 fois leur taille d'origine.  Réduction de l’occupation mémoire par l’implémentation de techniques de compression temporaire ou par la libération des ressources inutilisées. | Compatibilité  Gestion des fichiers mal identifiés (vérification via la signature binaire plutôt que l’extension).  Prise en charge des fichiers compressés contenant des images sans extension explicite. | Interface utilisateur  Interface intuitive et adaptée aux écrans tactiles (gestes pour tourner les pages, zoomer, etc.). |
|  | Qualité et évolutivité  Préparation pour des améliorations futures : Prise en charge des bandes dessinées japonaises (lecture de droite à gauche). |  |

Les services :

1. Ouverture d’un livre de manière asynchrone en permettant de consulter les premières pages avant que le chargement du livre soit terminé.
2. Redimensionnement des images à l’échelle de l’écran.
3. Possibilité d’afficher plusieurs pages à l’écran (au moins deux pages). Penser notamment à soigner le redimensionnement des images si les deux images ne sont pas exactement à la même taille pour donner l’impression de deux images au même format.
4. Pouvoir sélectionner un filtre de redimensionnement adapté au document (ie. plus adapté au texte ou plus adapté aux graphiques).
5. Pouvoir feuilleter l’ouvrage, aller à la page suivante, précédente, première page et dernière page de l’ouvrage, et ce le plus naturellement possible.
6. Pouvoir extraire des pages d’un ouvrage.
7. Pouvoir créer un ouvrage en assemblant des pages.
8. Pouvoir ajouter une table de matière aux ouvrages existant.
9. Pouvoir ajouter des tags aux ouvrages pour savoir quel est le style. Pourquoi pas faire une interface qui associe à un ouvrage la référence sur la page présente sur le site bedetheque.com.

Questions :

* C’est quoi un noyau applicatif

MODELISATION UML :

Afin de donner quelques points de repères sur le positionnement et les liens entre

tous les diagrammes d’UML, nous donnons ici notre propre vision en proposant un

regroupement des diagrammes en quatre ensembles suivant leur finalité :

• description du système : huit diagrammes ;

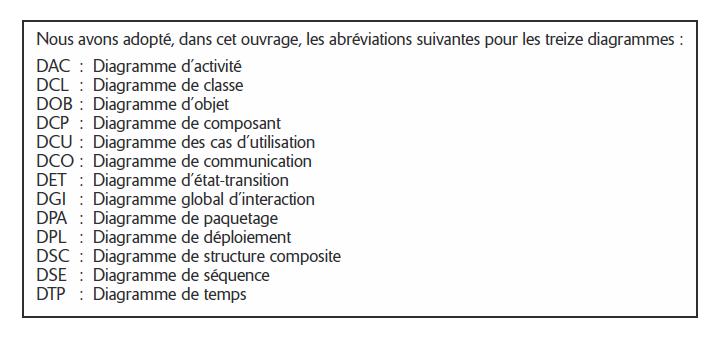
• architecture technique : deux diagrammes ;

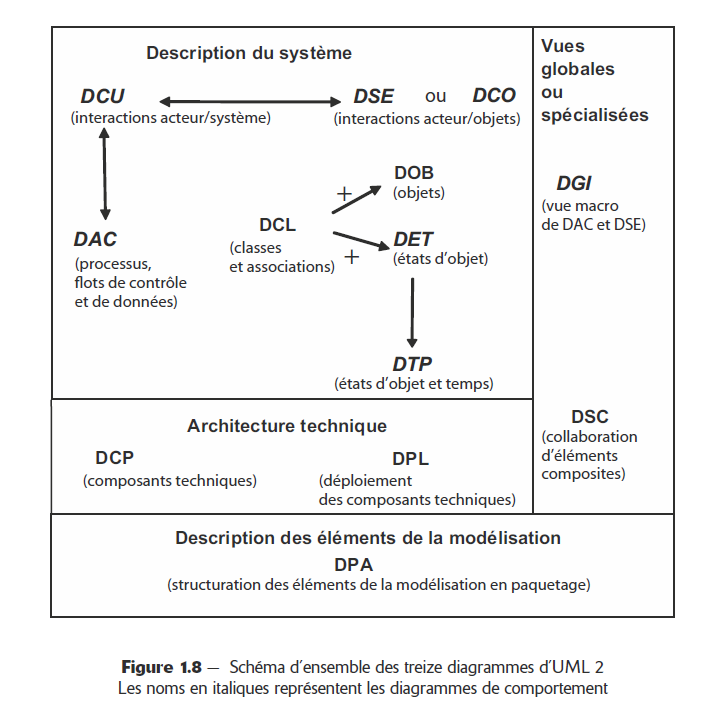
• vues globales ou spécialisées : deux diagrammes ;

• partition d’éléments de la modélisation : un diagramme.

Le schéma proposé reprend les treize diagrammes en les répartissant sur les quatre

ensembles définis





1. Diagrammes structurels ou statiques :
2. Diagramme de classe et diagramme d’objet
3. Diagramme de composants
4. Diagramme de déploiement
5. Diagramme de paquetage
6. Diagramme de structure composite

Critère d’évaluation :

1. Avoir un code bien modulaire et bien séparable
2. Comic book reader : architecture MVP
3. Clé : possibilité d’ajout de formats de fichiers différents
4. Testabilité du système (test unitaire et d’intégration) : possibilité d’écrire des tests (gtest) pour pouvoir tester les différents modules du programme
5. Qualité et factorisation du code
6. Bon fonctionnement de ce qui a été implanté, le programme de doit pas planter (dans le cas où par exemple on implante un fichier dont le format n’est pas supporté
7. Ne pas réinventer
8. Lister les fonctionnalités et les bibliothèques et pouvoir justifier

La couche infrastructure : pour les traitements bas niveau

Les suffixes :

1. Processor : lorsque les classes traite des données complexes de manière spécialisée (opérations spécifiques sur un type particulier de données), gère uniquement ces traitements spécifiques et non la gestion d’ensemble ou l’intéraction directe avec d’autres parties du système
2. Builder : pour la construction d’objets complexes
3. Manager : utilisé pour des classes qui gerent un ensemble d’objets ou des fonctionnalités complexes. Elles ont des responsabilités de coordination et facilitent l’accès ou la manipulation d’instances d’autres classes

BookManager : gère la collection de livres

LibraryManager : Coordonne les livres dans la bibliothèque

CacheManager : gère le stockage temporaire des pages

1. Handler : utilisé pour des classes qui interagissent avec des ressources externes ou traitent des fichiers ou des données brutes. Elles agissent comme une interface technique entre le système et une ressource externe. Exemples :
   1. FileHandler : interagit avec les fichiers sur le disque
   2. ArchiveHandler : gère la lecture et l’ecriture dans les archives CBZ/CBR

Architecture en couches avec des modules spécialisés

Couche présentation (interface graphique)

Couche métier (logique applicative)

Couche accès aux données :

* Lecture et écriture des formats d’images (.jpeg, .png, .bmp)
* Décodage des formats d’images(.jpeg, .png, .bmp, etc)
* Encodage des fichiers pour sauvegarde ou création de nouveaux livres

Couche infrastructure (services et bibliothèques)

* Chargement asynchrone des fichiers
* Optimisation des performances (par exemple : utilisation du OP4 pour redimensionnement)
* Gestion de la compatibilité entre les différents systèmes OS

Fournit les services fondamentaux pour les couches supérieures, comme le traitement d’images ou la gestion mémoire (les aspects bas niveaux)

Principe de séparation des préoccupations :

Pourquoi avoir séparé Page, PageLoader, PageView

* Lisibilité : chaque classe doit avoir une responsabilité unique et identifiable
* Modularité : Les composants peuvent évoluer ou être remplacés indépendamment
* Réutilisabilité : Les classes comme Loader ou view peuvent être utilisés ailleurs
* sans dépendre directement de Page

# CONCLUSION

Le développement du **Comic Book Reader/Writer** illustre la manière dont une application bien conçue peut répondre aux attentes variées des amateurs de bandes dessinées numériques. Ce projet a permis d’implémenter des fonctionnalités essentielles comme la lecture fluide des formats .CBR et .CBZ, tout en introduisant des options novatrices pour la personnalisation de l’expérience utilisateur, telles que l’ajout de tags, le support des BD japonaises, et l’intégration de tables des matières.

Grâce à une architecture modulaire et extensible, cette application ouvre la voie à de futures évolutions, comme l’intégration de nouveaux formats ou l’optimisation des performances avec des technologies graphiques avancées. Ce projet a ainsi démontré la faisabilité et la pertinence d’une solution numérique adaptée aux besoins spécifiques des passionnés de bandes dessinées.

En conclusion, ce travail représente non seulement une solution technique mais également une plateforme évolutive qui pourrait servir de base à des outils encore plus sophistiqués. Ce projet reflète l’engagement envers une approche centrée sur l’utilisateur, mariant créativité et efficacité technique.