Contexte Projet One Piece



One Piece Forever est une application informative sur l’univers de One Piece. Tout d’abord rappelons ce que l’on peut trouver dans l’univers de One Piece, une planète constituer à 90% d’eau ou les mers sont réparties en 4 océans correspondant au 4 points cardinaux et un océan central qui traverse toute la planète, Grand Line. Dans ce manga/animé le monde est sous la direction du gouvernement mondial qui à pour armé la Marine qui défend le peuple des pirates qui sont réputé comme méchant et dévastateur. One Piece raconte l’histoire de Luffy qui va voyager d’île en île pour trouver le One Piece, un trésor laisser par le roi des pirates à sa mort, et donc devenir le roi des Pirate de son ère.

Notre application sera assez simple d’utilisation et vise un publique large, autant les connaisseurs de l’œuvre, qui pourrons cliquer directement sur les îles. Ou les gens qui découvre One Piece et qui veulent plus d’information sur les arcs qu’ils ont vu. Ou simplement des personnes qui veulent découvrir cet animé en cliquant au hasard sur des îles et apprendre les choses incroyables qui s’y sont produit.

Donc dans notre application nous retrouverons la carte de ce monde avec les îles que Luffy à traverser. On pourra donc cliquer sur chacune des île ou Luffy s’est rendu pour y avoir une page avec des informations supplémentaires sur les événements qui s’y sont déroulé, on appelle cela des arcs. Sur cette page nous pourrons retrouver les informations essentielles, des images marquantes et une liste des personnages présents. Depuis cette page nous pourrons cliquer sur les personnages pour avoir accès à des informations sur eux, cliquer sur un bouton pour accéder à l’arc suivant ou précédent. Nous pourrons également cliquer sur un bouton pour ajouter des informations ou une image.

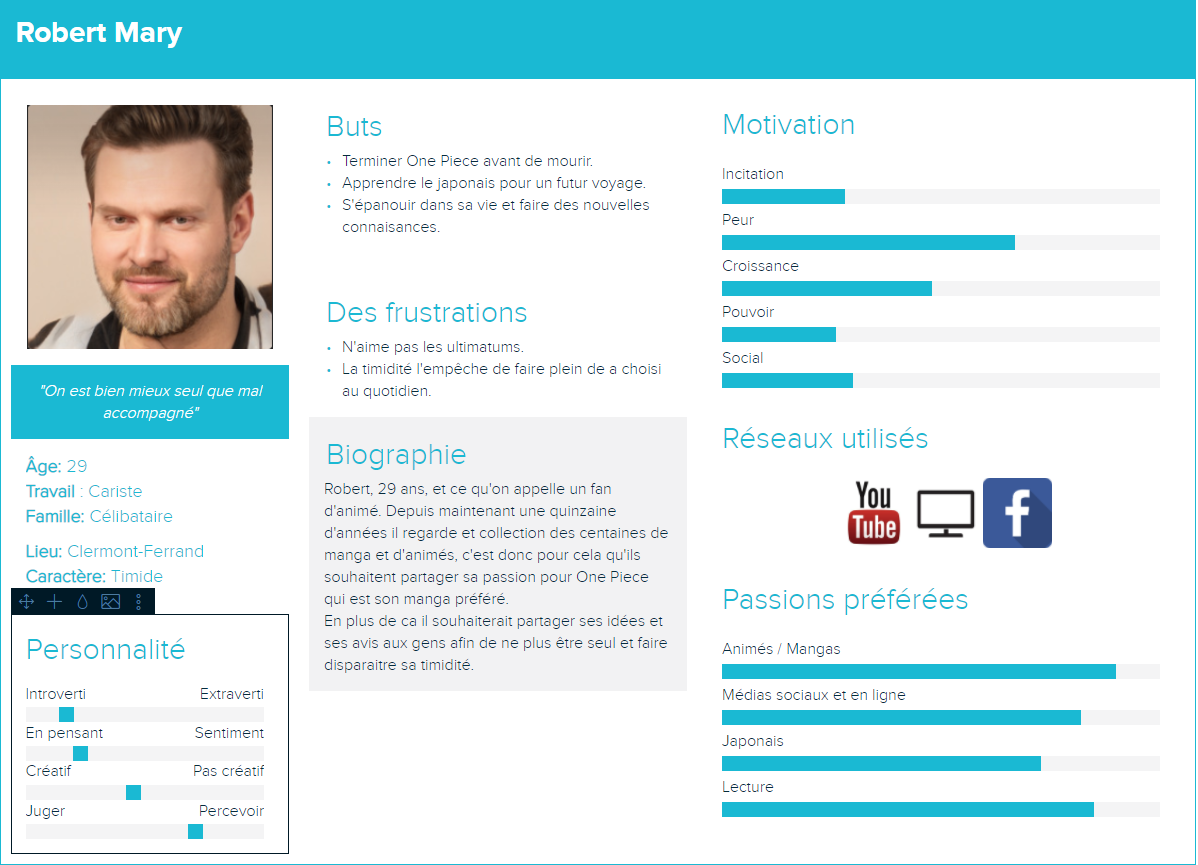
Dans cette application nous pouvons également rechercher un arc si on ne sait pas ou il se situe sur la carte, ainsi qu’un personnage pour accéder à une page d’information sur lui. Sur la page d’information du personnage nous pourrons retrouver les informations essentielles, une liste d’image, ainsi qu’une liste des arcs auxquelles il appartient. Dans cette page également, l’utilisateur pourra cliquer pour ajouter une image, ajouter des informations, sur un arc dans la liste des arcs pour accéder à la page d’information de celui-ci.

**Personnas et user-stories**

Prenons exemple sur Nathan et Robert, deux personnas, pour développer l’utilité de notre application :



Nathan vient de commencer One Piece car on lui dit souvent qu’il a le même tempérament que Luffy le personnage principal. Nathan trouve ça génial et veut donc en savoir plus, connaître l’étendu de l’univers de One Piece il se rend donc dans l’application One Piece Forever et veut en savoir plus sur son personnage préféré et qui lui ressemble, Luffy.



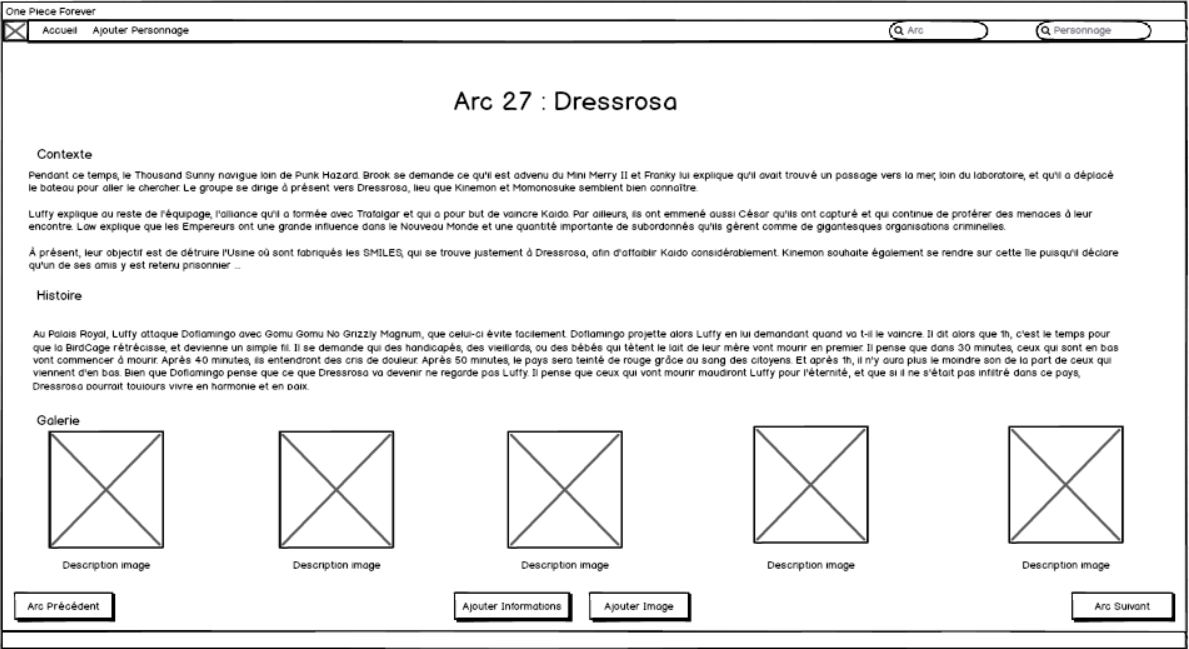
Robert veut avoir à se rappeler des personnages présents dans l’arc d’Enies Lobby. Robert se rend donc dans l’application One Piece forever, afin de trouver les infos nécessaires à sa passion.

**Sketchs**

Page principal :



Page description d’arc :

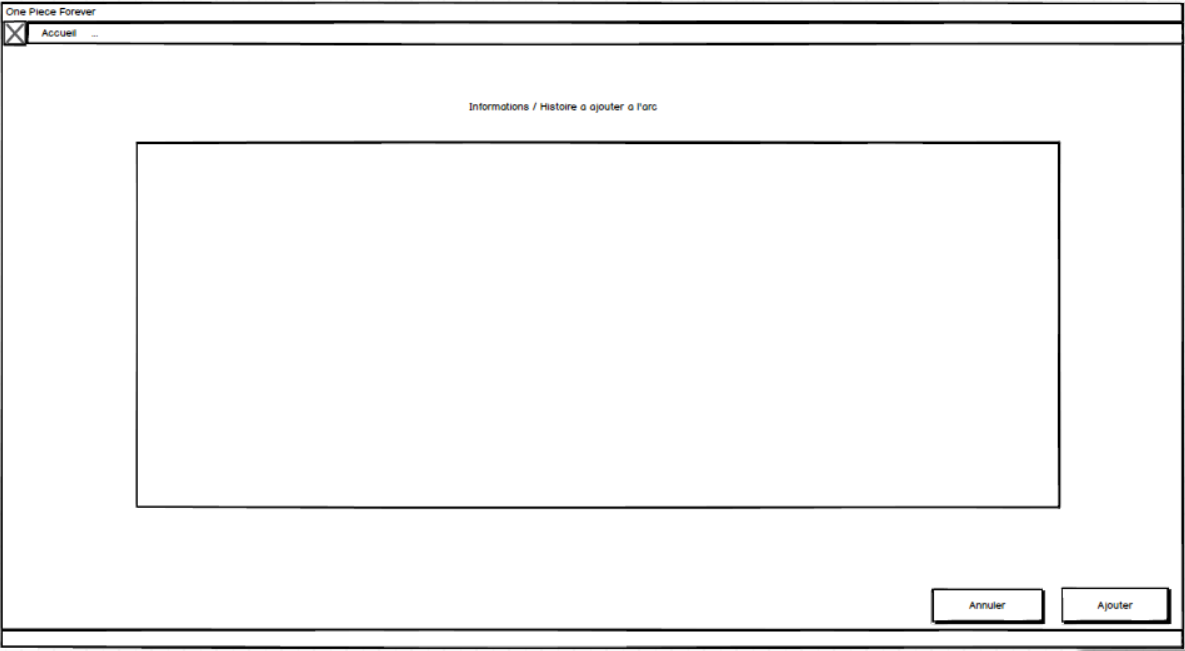


Page description personnages :

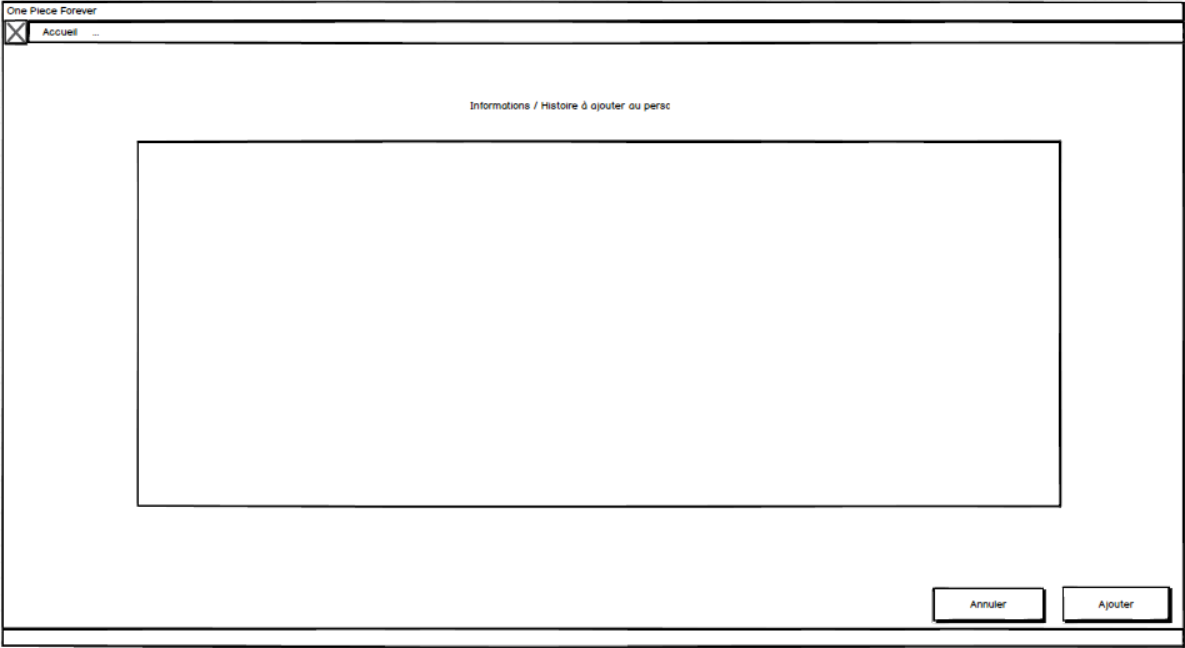


Page ajout d’image arc :

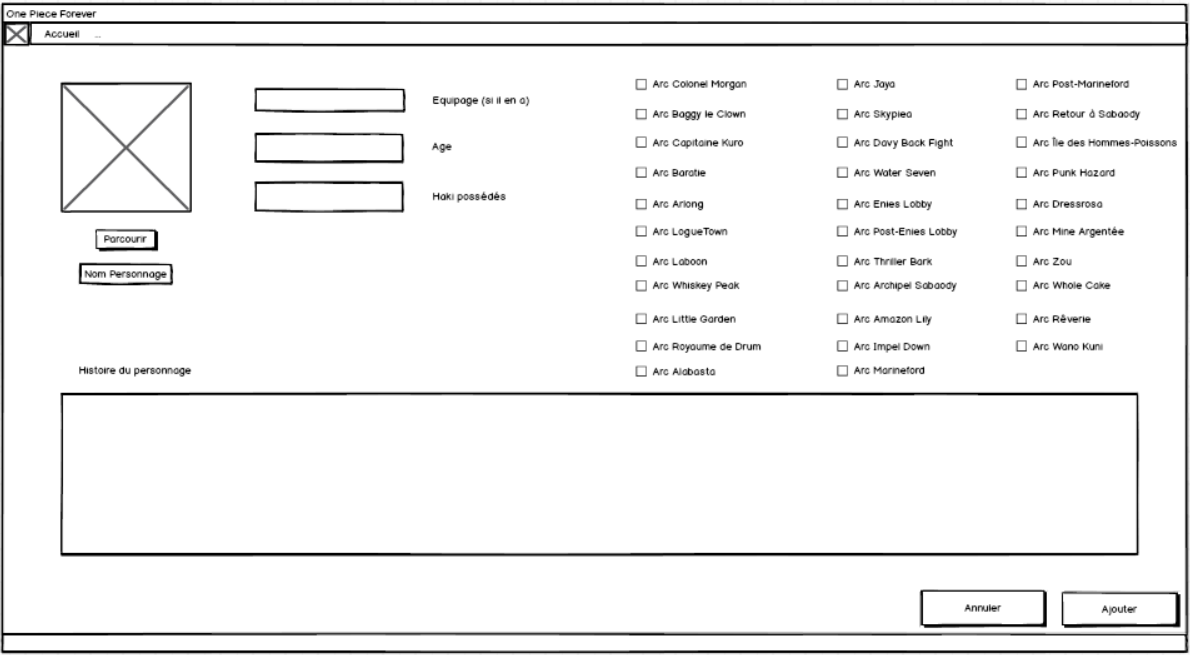


Page ajout d’information arc :

Page ajout d’information personnage :



Page ajout personnage :



Page ajout image personnage :



**Story board**

Page principal :



1

Quand l’utilisateur va ouvrir l’application, il voit apparaître la carte du monde de One Piece, dans cette fenêtre l’utilisateur va pouvoir :

* Il peut cliquer sur une île pour avoir les informations concernant l’arc de cette île. (1)
* Il peut cliquer sur « ajouter personnage » pour ajouter un personnage à l’application. (2)
* Il peut rechercher un arc et peut également rechercher un personnage. (3)

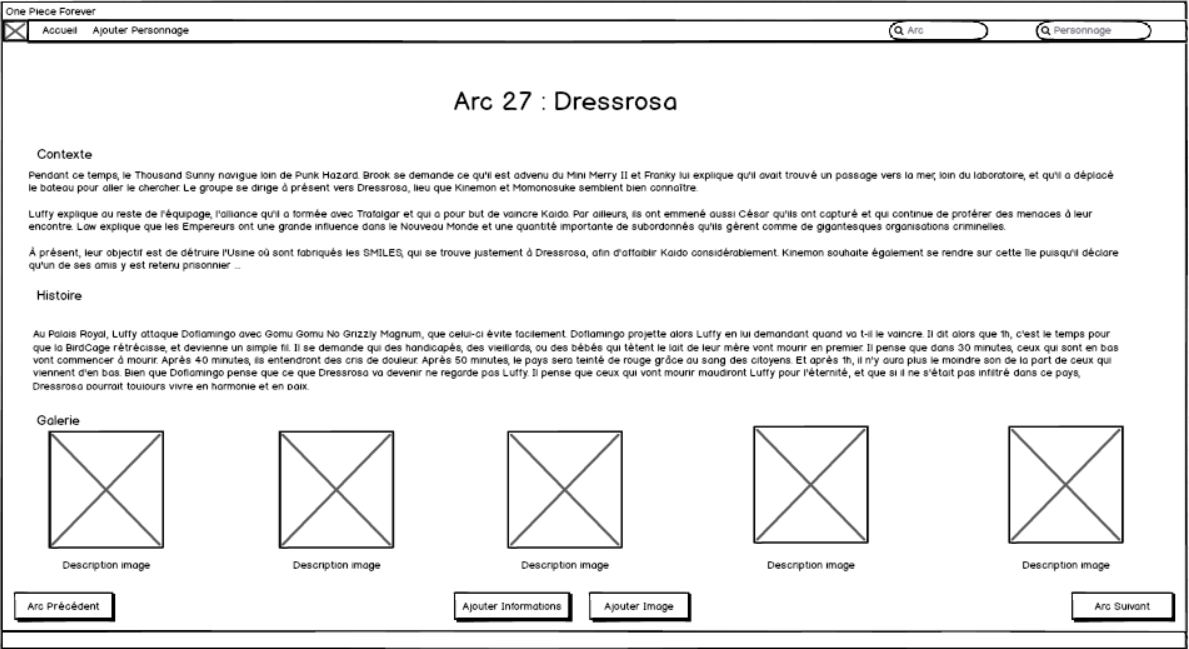
7

6

2

Page description d’arc :

1



5

5

3

Ici l’utilisateur va pouvoir :

4

* Avoir accès aux informations de l’arc en question, ici Dressrosa pour l’exemple (1)
* Retourner à l’accueil, sélectionné un autre arc ou autres fonctionnalité de la page d’accueil (2)
* Ajouter des informations à cet arc (3)
* Ajouter une image à cet arc (4)
* Accéder aux informations de l’arc précédent ou suivant (5)
* Ajouter un personnage à l’application (6)
* Rechercher un personnage ou un arc dans l’application (7)

5

2

1

Page description personnages :



4

3

Dans cette page l’utilisateur peut :

* Retourner à l’accueil, sélectionné un autre arc ou autres fonctionnalités de la page d’accueil (1)
* Ajouter un personnage à l’application (2)
* Ajouter une image à ce personnage (3)
* Ajouter des informations à ce personnage (4)
* Rechercher un personnage ou un arc dans l’application (5)

6

Page ajout d’image arc :



5

4

2

1

3

L’utilisateur va pouvoir exécuter les actions suivantes :

* Parcourir ses fichiers pour trouver l’image qu’il veut ajouter à l’arc (1)
* Donner un nom à son image (2)
* Décrire et expliquer pourquoi cette image correspond à l’arc (3)
* Ajouter les informations saisies à l’application (4)
* Annuler la saisie et revenir à la page précédente (5)
* Revenir à l’accueil et annuler la saisie (6)

4

Page ajout d’information arc :Une image contenant texte

Description générée automatiquement

2

3

1

L’utilisateur peut, dans cette page :

* Saisir les informations concernant l’arc qu’il veut ajouter (1)
* Annuler la saisie et revenir à la page précédente (2)
* Ajouter la saisie à la page d’information de l'arc (3)
* Retourner à l’accueil et annuler la saisie (4)

4

Page ajout d’information personnage :

1

Une image contenant texte

Description générée automatiquement

3

2

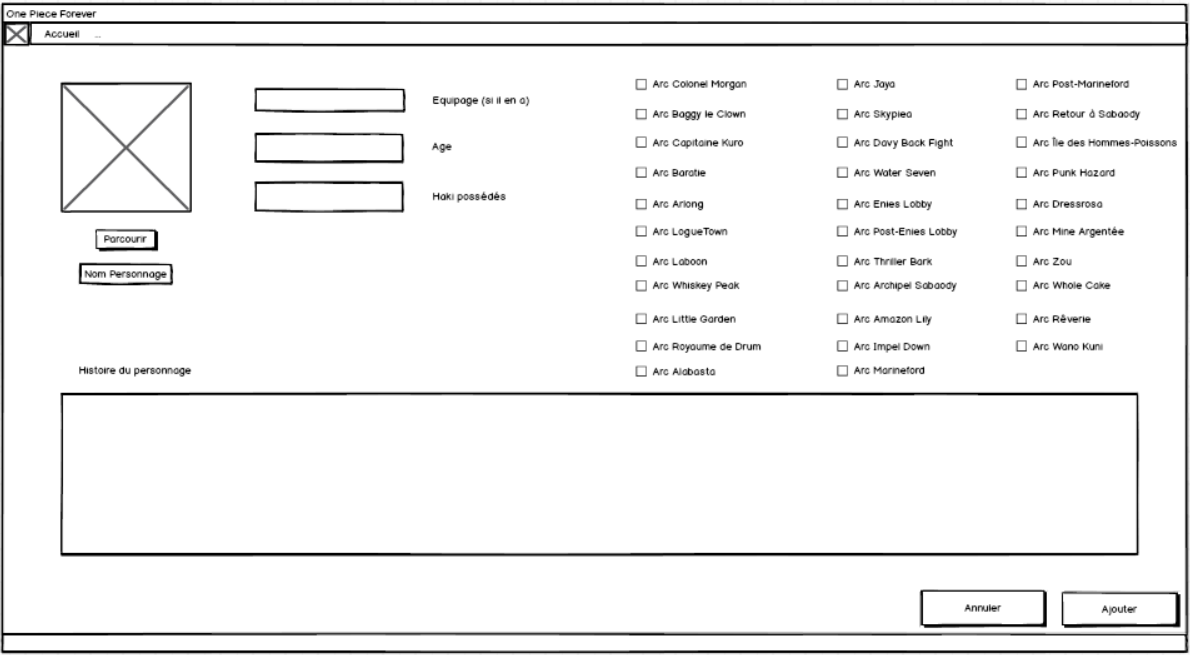
Dans cette page de saisie l’utilisateur peut :

* Saisir des informations concernant le personnage (1)
* Annuler la saisie et revenir à la page précédente (2)
* Ajouter la saisie à la page d’information du personnage de l’application (3)
* Retourner à l’accueil et annuler la saisie (4)

4

1

Page ajout personnage :



3

2

Dans cette page d’ajout de personnage l’utilisateur peut :

* Saisir les informations du personnage qu’il souhaite intégrer à l’application (1)
* Annuler la saisie et revenir à la page précédente (2)
* Ajouter la saisie à l’application et ouvre la page du personnage créé (3)
* Retourner à l’accueil et annuler la saisie (4)

3

6

Page ajout image personnage :



4

5

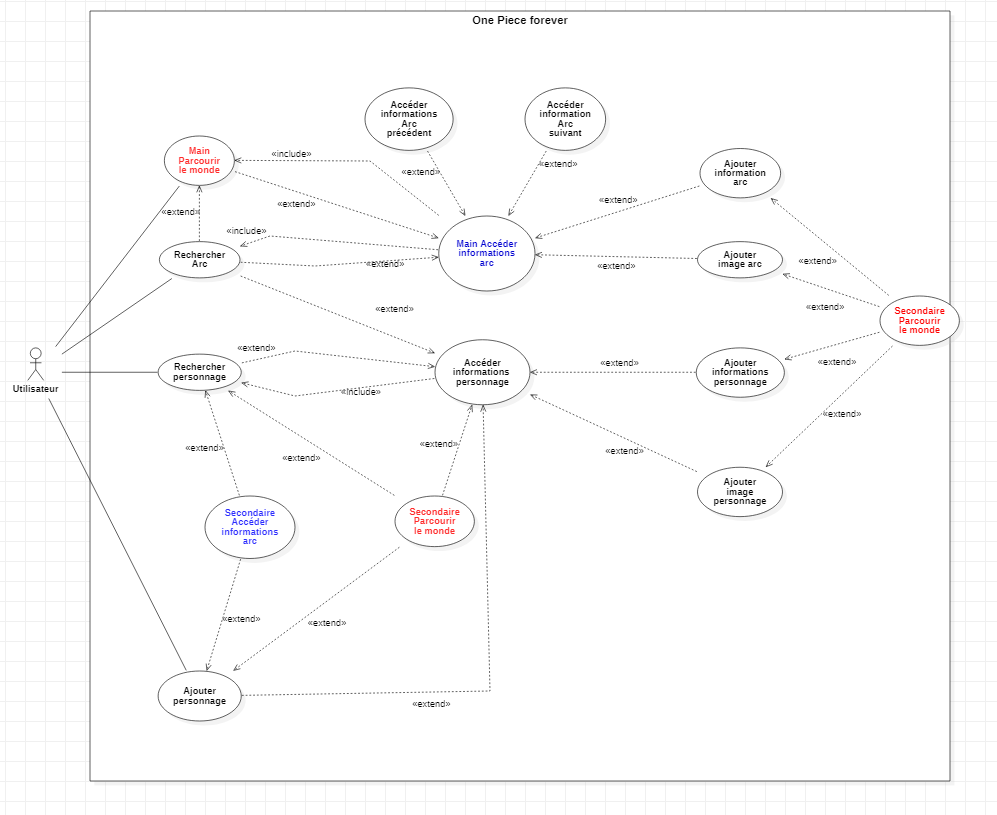
2

1

Dans cette page l’utilisateur va pouvoir :

* Parcourir ses fichiers pour trouver l’image qu’il veut ajouter au personnage (1)
* Donner un nom à son image (2)
* Décrire et expliquer pourquoi ajouter cette image au personnage (3)
* Ajouter les informations saisies à l’application (4)
* Annuler et revenir à la page précédente (5)
* Revenir à l’accueil et annuler la saisie (6)

**Diagramme de cas**

****

Pour un souci de compréhension les cas de même couleur avec le mot secondaire sont présents pour éviter de faire croiser les flèches vers le main. Cela revient à relier les cas reliés au « secondaire au cas « main ».

**Description du diagramme de cas**

**Cas « Parcourir le monde » :**

|  |  |
| --- | --- |
| Nom | Parcourir le monde |
| Objectif | Sélectionner sur la carte du monde de One Piece l’arc sur lequel on veut avoir des information |
| Acteurs principaux | Utilisateur |
| Acteurs secondaire |  |
| Conditions initial | - Avoir le pointeur de la souris dans la page de la carte |
| Scénario d’utilisation | - Cliquer sur une île (fin 1, sinon fin 2) |
| Conditions de fin | 1. Ouvre la fenêtre d’information de l’arc demander 2. Reste dans la page d’accueil en attendant une action de l’utilisateur |

**Cas « Rechercher un arc » :**

|  |  |
| --- | --- |
| Nom | Rechercher un arc |
| Objectif | Chercher un arc dans tous les arcs de One Piece |
| Acteurs principaux | Utilisateur |
| Acteurs secondaire |  |
| Conditions initial | - Être dans la barre de recherche d’arc |
| Scénario d’utilisation | - Entrer un nom d’arc existant (fin 1)  - Entrer un nom d’arc inexistant (fin 2) |
| Conditions de fin | 1. Ouvre la fenêtre d’information de l’arc demander 2. Affiche qu’il n’y pas d’arc à ce nom |

**Cas « Rechercher un personnage » :**

|  |  |
| --- | --- |
| Nom | Rechercher un personnage |
| Objectif | Chercher un personnage de One Piece |
| Acteurs principaux | Utilisateur |
| Acteurs secondaire |  |
| Conditions initial | - Être dans la barre de recherche de personnage |
| Scénario d’utilisation | - Entrer un nom de personnage existant (fin 1)  - Entrer un nom de personnage inexistant  (fin 2) |
| Conditions de fin | 1. Ouvre la fenêtre d’information du personnage demander 2. Affiche qu’il n’y pas de personnages à ce nom |

**Cas « Ajouter un personnage » :**

|  |  |
| --- | --- |
| Nom | Ajouter un personnage |
| Objectif | Ajoute un personnage dans l’application |
| Acteurs principaux | Utilisateur |
| Acteurs secondaire |  |
| Conditions initial | - Cliquer sur l’icône d’ajout de personnage  - Être dans la fenêtre de saisie de personnage |
| Scénario d’utilisation | - Entrer toutes les informations nécessaires pour la création du personnage (nom, prénom, situation (pirate, marine, lambda, …), dans quel arc, informations complémentaire)  - Cliquer sur « ajouter » (fin 1)  - Cliquer sur « annuler » (fin 2)  - Cliquer sur « accueil » (fin 3)  - Cliquer sur « ajouter personnages » (fin 4) |
| Conditions de fin | 1. Enregistre le personnage et ouvre la page d’information de ce personnage 2. Annule la saisie d’un personnage et retourne à la page précédente 3. Ouvre la page d’accueil et annule la saisie 4. Ouvre une nouvelle page d’ajout d’un personnage et annule la saisie |

**Cas « Ajouter information arc » :**

|  |  |
| --- | --- |
| Nom | Ajouter information arc |
| Objectif | Permet d’ajouter des informations sur un arc donné |
| Acteurs principaux | Utilisateur |
| Acteurs secondaire |  |
| Conditions initial | - Être dans la fenêtre d’information d’un arc  - Cliquer sur l’icône d’ajout d’information  - Être dans la fenêtre de saisie d’information |
| Scénario d’utilisation | - Cliquer sur « ajouter » (fin 1)  - Cliquer sur « annuler » (fin 2)  - Cliquer sur la zone de texte « Informations / Histoire à ajouter à l’arc » (fin 3)  - Cliquer sur « accueil » (fin 4)  - Cliquer sur « ajouter personnages » (fin 5) |
| Conditions de fin | 1. Enregistre les informations saisies et retourne dans la page d’information de l’arc 2. Annule la saisie d’informations et retourne dans la page d’information de l’arc 3. Permet à l’utilisateur d’entrer les informations qu’il souhaite sur l’arc 4. Ouvre la page d’accueil et annule la saisie 5. Ouvre la page d’ajout d’un personnage et annule la saisie |

**Cas « Ajouter informations personnage »**

|  |  |
| --- | --- |
| Nom | Ajouter information personnage |
| Objectif | Permet d’ajouter des informations sur un personnages donné |
| Acteurs principaux | Utilisateur |
| Acteurs secondaire |  |
| Conditions initial | - Être dans la fenêtre d’information d’un personnage  - Cliquer sur l’icône d’ajout d’informations  - Être dans la fenêtre de saisie d’informations |
| Scénario d’utilisation | - Cliquer sur « ajouter » (fin 1)  - Cliquer sur « annuler » (fin 2)  - Cliquer sur la zone de texte « Informations / Histoire à ajouter au personnage » (fin 3)  - Cliquer sur « accueil » (fin 4)  - Cliquer sur « ajouter personnages » (fin 5) |
| Conditions de fin | 1. Enregistre les informations saisies et retourne dans la page d’information du personnage 2. Annule la saisie d’informations et retourne dans la page d’information du personnage 3. Permet à l’utilisateur d’entrer les informations qu’il souhaite sur le personnage 4. Ouvre la page d’accueil et annule la saisie 5. Ouvre la page d’ajout d’un personnage et annule la saisie |

**Cas « Ajouter image personnage »**

|  |  |
| --- | --- |
| Nom | Ajouter image personnage |
| Objectif | Permet d’ajouter des images sur un personnages donné |
| Acteurs principaux | Utilisateur |
| Acteurs secondaire |  |
| Conditions initial | - Être dans la fenêtre d’information d’un personnage  - Cliquer sur l’icône d’ajout d’image  - Être dans la fenêtre de saisie d’image |
| Scénario d’utilisation | - Cliquer sur « ajouter » (fin 1)  - Cliquer sur « annuler » (fin 2)  - Cliquer sur « parcourir » (fin 3)  - Cliquer sur la zone de texte « Nom de l’image » (fin 4)  - Cliquer sur la zone de texte « Description de l’image et explication » (fin 5)  - Cliquer sur « accueil » (fin 6)  - Cliquer sur « ajouter personnages » (fin 7) |
| Conditions de fin | 1. Enregistre l’image et les informations saisies 2. Annule la saisie d’image et retourne dans la fenêtre d’information du personnage 3. Permet à l’utilisateur de parcourir ses fichiers pour trouver la photo qu’il veut ajouter 4. Permet à l’utilisateur d’entrer le nom de son image 5. Permet à l’utilisateur d’entrer les informations et les explications concernant l’image qu’il veut ajouter à au personnage 6. Ouvre la page d’accueil et annule la saisie 7. Ouvre la page d’ajout d’un personnage et annule la saisie |

**Cas « Ajouter image arc »**

|  |  |
| --- | --- |
| Nom | Ajouter image arc |
| Objectif | Permet d’ajouter des images sur un arc donné |
| Acteurs principaux | Utilisateur |
| Acteurs secondaire |  |
| Conditions initial | - Être dans la fenêtre d’information de l’arc  - Cliquer sur l’icône d’ajout d’image  - Être dans la fenêtre de saisie d’image |
| Scénario d’utilisation | - Cliquer sur « ajouter » (fin 1)  - Cliquer sur « annuler » (fin 2)  - Cliquer sur « parcourir » (fin 3)  - Cliquer sur la zone de texte « Nom de l’image » (fin 4)  - Cliquer sur la zone de texte « Description de l’image et explication » (fin 5)  - Cliquer sur « accueil » (fin 6)  - Cliquer sur « ajouter personnages » (fin 7) |
| Conditions de fin | 1. Enregistre l’image et les informations saisies 2. Annule la saisie d’image et retourne dans la fenêtre d’information de l’arc 3. Permet à l’utilisateur de parcourir ses fichiers pour trouver la photo qu’il veut ajouter 4. Permet à l’utilisateur d’entrer le nom de son image 5. Permet à l’utilisateur d’entrer les informations et les explications concernant l’image qu’il veut ajouter à l’arc 6. Ouvre la page d’accueil et annule la saisie 7. Ouvre la page d’ajout d’un personnage et annule la saisie |

**Cas « Accéder information arc »**

|  |  |
| --- | --- |
| Nom | Accéder information arc |
| Objectif | Permet d’accéder aux différentes informations d’un arc donné |
| Acteurs principaux | Utilisateur |
| Acteurs secondaire |  |
| Conditions initial | - Être dans la fenêtre d’information de l’arc  - Regarder et lire les informations que l’on recherche sur cet arc |
| Scénario d’utilisation | - Cliquer sur « ajouter image » (fin 1)  - Cliquer sur « ajouter information » (fin 2)  - Cliquer sur « arc précédent » (fin 3)  - Cliquer sur « arc suivant » (fin 4)  - Cliquer sur « accueil » (fin 5)  - Cliquer sur « ajouter personnages » (fin 6)  - Clique dans la barre de recherche de personnage ou d’arc (fin 7) |
| Conditions de fin | 1. Ouvre la page pour ajouter une image à l’arc 2. Ouvre la page pour ajouter des informations sur l’arc 3. Ouvre la page d’information de l’arc précédent 4. Ouvre la page d’information l’arc suivant 5. Ouvre la page d’accueil 6. Ouvre la page d’ajout d’un personnage 7. Permet à l’utilisateur de rechercher un arc ou un personnage |

**Cas « Accéder information Personnage »**

|  |  |
| --- | --- |
| Nom | Accéder information personnage |
| Objectif | Permet d’accéder aux différentes informations d’un personnage donné |
| Acteurs principaux | Utilisateur |
| Acteurs secondaire |  |
| Conditions initial | - Être dans la fenêtre d’information du personnage  - Regarder et lire les informations que l’on recherche sur cet arc |
| Scénario d’utilisation | - Cliquer sur « ajouter image » (fin 1)  - Cliquer sur « ajouter information » (fin 2)  - Cliquer sur « accueil » (fin 3)  - Cliquer sur « ajouter personnages » (fin 4)  - Clique dans la barre de recherche de personnage ou d’arc (fin 5) |
| Conditions de fin | 1. Ouvre la page pour ajouter une image au personnage 2. Ouvre la page pour ajouter des informations sur le personnage 3. Ouvre la page d’accueil 4. Ouvre la page d’ajout d’un personnage 5. Permet à l’utilisateur de rechercher un arc ou un personnage |

**Description écrite de l’architecture**

Pourquoi avoir découpé votre application en plusieurs packages ?

Quel est le rôle de chaque package ?

Pourquoi tel package dépend de tel autre ?

On a choisi de découper notre application en plusieurs paquets pour pouvoir gérer l’encapsulation et la maintenance. Le fait de la répartir ainsi limite les risques de conflits en cas de nouvelles versions. Les paquets sont liés entre eux de manière que tout fonctionne bien et en respectant l’encapsulation. Chaque paquet a son rôle :

* UnitsTests pour faire les tests unitaires de nos classes. Il dépend de ClassLibrary car ClassLibrary contient toutes les classes que UnitsTests doit tester.
* Test\_Personnage est là pour tester la class personnage Il dépend de ClassLibrary car ClassLibrary contient les classes que Test\_Personnage doit tester.
* Test\_Arc est là pour tester la class arc. Il dépend de ClassLibrary car ClassLibrary contient les classes que Test\_Arc doit tester.
* Test\_DataContract est là pour tester la persistance de l’application
* ClassLibrary est là pour rassembler toutes les classes du modèle de notre application et ne dépend de personnes.
* DataContractPersistance est là pour faire la persistance de l’application, il dépend de ClassLibrary qui contient sont interface.
* Stub est là pour charger les données de l’application en cas de besoin (problème avec DataContractPersistane). Mais ne permet pas la sauvegarde. Il dépend de ClassLibrary qui contient sont interface.
* Projet1 est là pour rassembler toutes les classes des vues et le navigateur, il est dépendant de ClassLibrary car il contient les classes qui font fonctionner ses vues tel que le manager. Il est dépendant de Converteur qui permet de convertir les enums de navigateur (qui est dans Projet1) et qui fait donc la navigation de l’application. Il dépend également des paquets Image, ImagePerso, ImageArc et Icône car ils contiennent des données nécessaires à l’application. Il est aussi dépendant de stub car il contient les données qui permettent de charger l’application.
* Converteur est là pour convertir l’enum du navigateur en vue XAML
* Image est là pour stocker les images de notre application
* ImagePerso est là pour stocker les images des personnages de l’application
* ImageArc est là pour stocker les images des arcs de l’application
* Icône est là pour stocker les icônes de l’application

Pourquoi vous avez utilisé une interface/héritage à tel endroit ?

* Dans notre application nous n’utilisons aucun héritage étant donné que nous n’en avons pas obligatoirement besoin.

Cependant nous avons utilisé 2 interfaces, la première est « INotifyPropertyChanged » que le Manager implémente car celui-ci est primordial si l’on veut constater les changements effectués avec le binding sur nos pages, donc sans celui-ci nous ne pourrions pas voir nos modifications en direct sur la page mais il faudrait attendre un peu ce qui est dommage pour une application comme celle-ci.

Une deuxième interface est « IPersistanceManager », le manager l’utilise en propriété, celle-ci est là pour prendre en compte la persistance dans notre application et ainsi éviter que l’on ait du code en dur dans notre modèle.

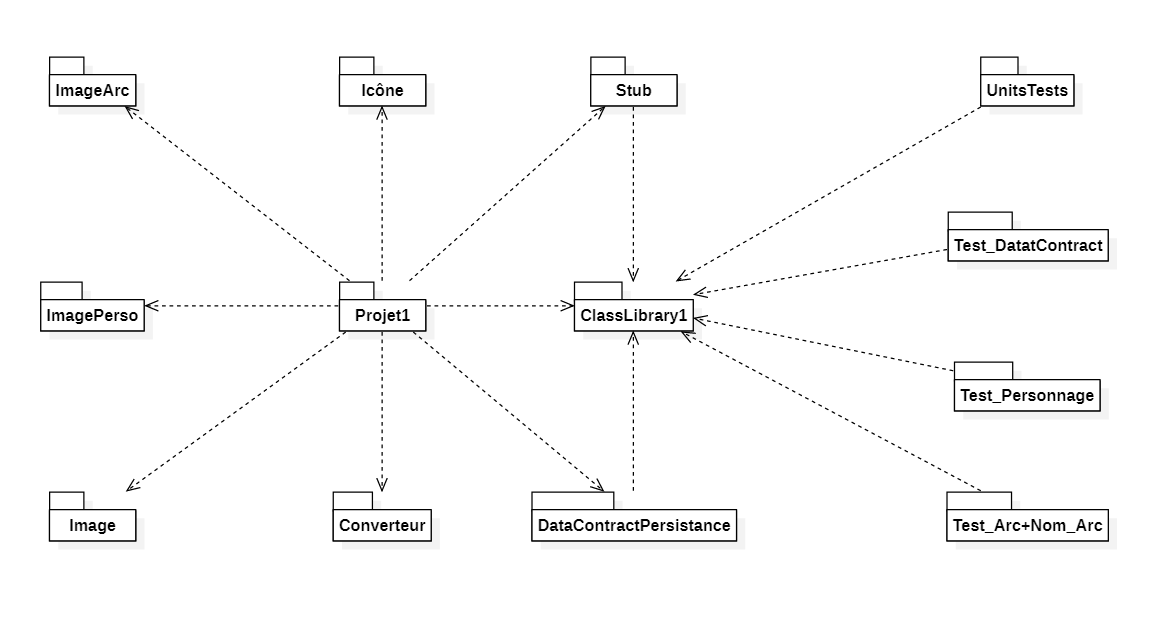
Pourquoi vous avez utilisé tel patron de conception ? A quoi sert-il ?

* Nous n’avons utilisé aucun patron de conception.

Pourquoi vous avez utilisé tel type de collection ?

* Alors nous avons principalement utilisé des List que ce soit pour la liste des arcs, la liste des personnages de l’application ou bien même la liste des images de la galerie des personnages ou des arcs. On a privilégié des List car celles-ci étaient parfaites par rapport à l’idée des listes que l’on souhaitait, c’est-à-dire, une liste dans laquelle on peut ajouter un élément facilement sans avoir à le trir étant donné que nous ne cherchons pas à faire de tri. Il est aussi simple dans une List de faire une fonction de recherche avec equals ce qui nous arrange assez bien pour les fonctions de recherche dont l’application dispose.
* Notre application dans son design utilise beaucoup de grilles ce qui nous aide beaucoup afin de faire des sections avec un minimum de responsive mais c’est aussi utile pour gérer la place que prend nos éléments sur la page. Il y a aussi quelques ListBox qui nous sont très utiles pour afficher les listes de personnages et d’images de chaque arc, on a préféré utiliser les ListBox car c’est plus simple de faire le binding dessus correctement et puis c’est beaucoup plus propre qu’une liste de TextBox.
* Ensuite notre application se découpe en une page principale avec le menu uniquement pour faire en sorte qu’il y ait seulement la partie du bas de la fenêtre qui change afin de ne pas perdre les utilisateurs. C’est aussi beaucoup plus simple de faire cela plutôt que d’ouvrir une nouvelle fenêtre à chaque fois.
* Nous avons par la suite utilisé un navigateur regroupant une énumération des différents états possibles par notre application. L’énumération est transformée grâce à un convertisseur qui convertit ces éléments en UserControl. Cette navigation nous a grandement aidé pour améliorer la fluidité de l’application ainsi que son utilisation pour les futurs utilisateurs.

**Diagramme de paquetage**



Ce diagramme de paquetage regroupe tous les paquets de notre application One Piece Forever.

Donc nous avons la partie **ClassLibrary1**, qui comprend toutes nos classes qui sont décrites dans le diagramme de classes.

Nous avons pour le moment que 3 paquets de **test**, qui regroupent tous les autres, qui sont toutes les classes et le test du **DataContract**.

Ensuite nous avons **Projet1** qui symbolise notre application, elle est donc dépendante de tous les autres paquets dont elle a besoin pour fonctionner.

Dans le paquet **converteur** nous avons pour le moment qu’un seul convertisseur qui nous sert pour la navigation. Mais nous en prévoyons un autre pour changer la syntaxe de l’énumération NomHaki.

Ensuite nous avons le paquet **Image** où l’on met toutes les images utiles à l’application, tels que la carte du monde ou les logos de menus.

Les paquets **ImagePerso** et **ImageArc** portent bien leurs noms car ce sont les dossiers où nous avons placé les images des personnages et des arcs.

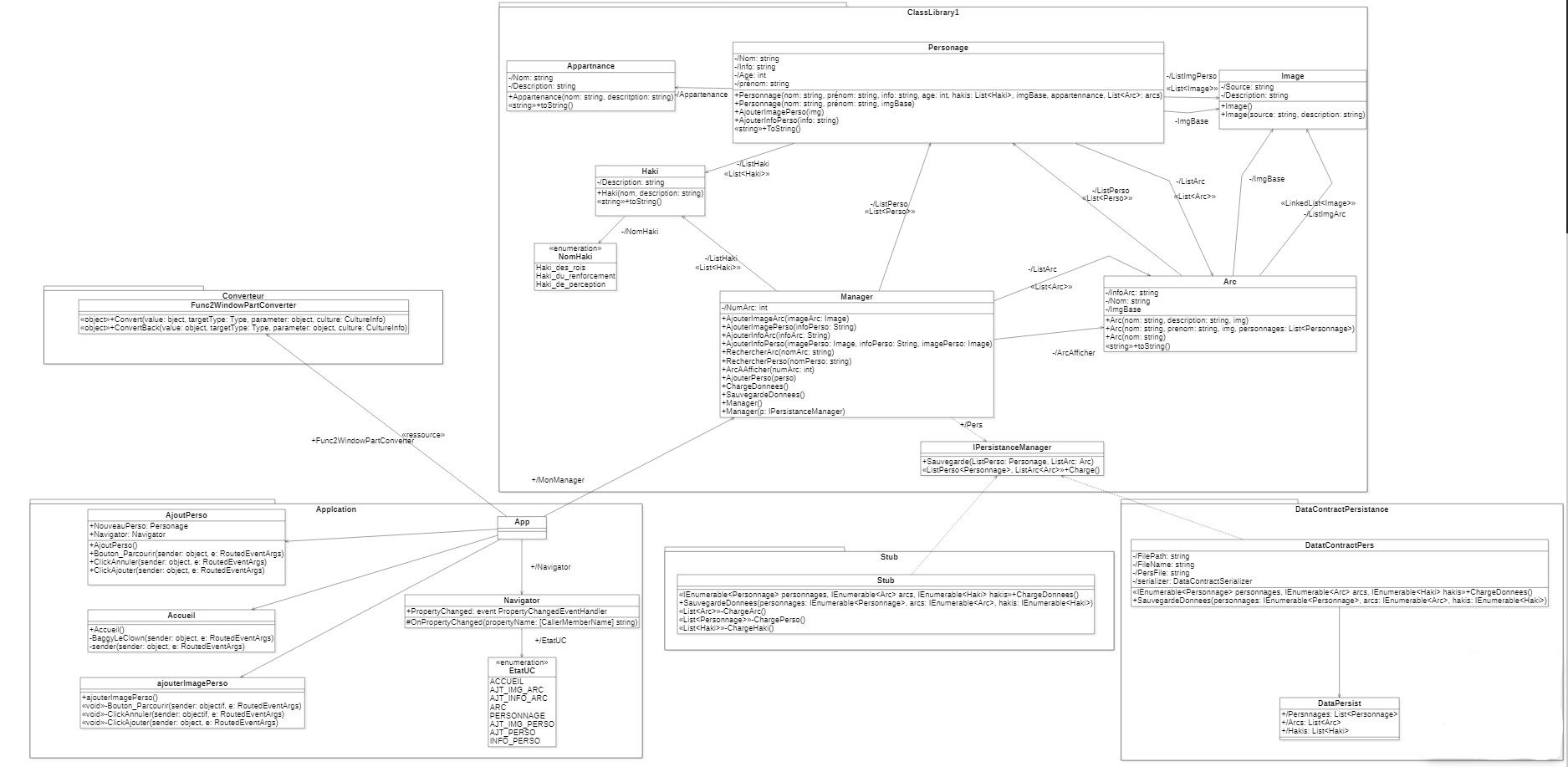
Le paquet **Icône** est le paquet où nous avons mis les différentes icônes utiles telles que celles que l’on utilise pour les boutons.

Le paquet **UnitsTests** comprend les tests unitaires des différentes classes de **ClassLibrary**, il est donc dépendant.

Le paquet **DataContract**, est le paquet de sauvegarde et de chargement des données de l’application par défaut.

Le paquet **stub**, est un paquet pour charger les données utiles à l’application mais on ne peut pas les sauvegarder. Il est donc très utile si on a un problème avec le fichier de sauvegarde du DataContract.

**Diagramme de classes**

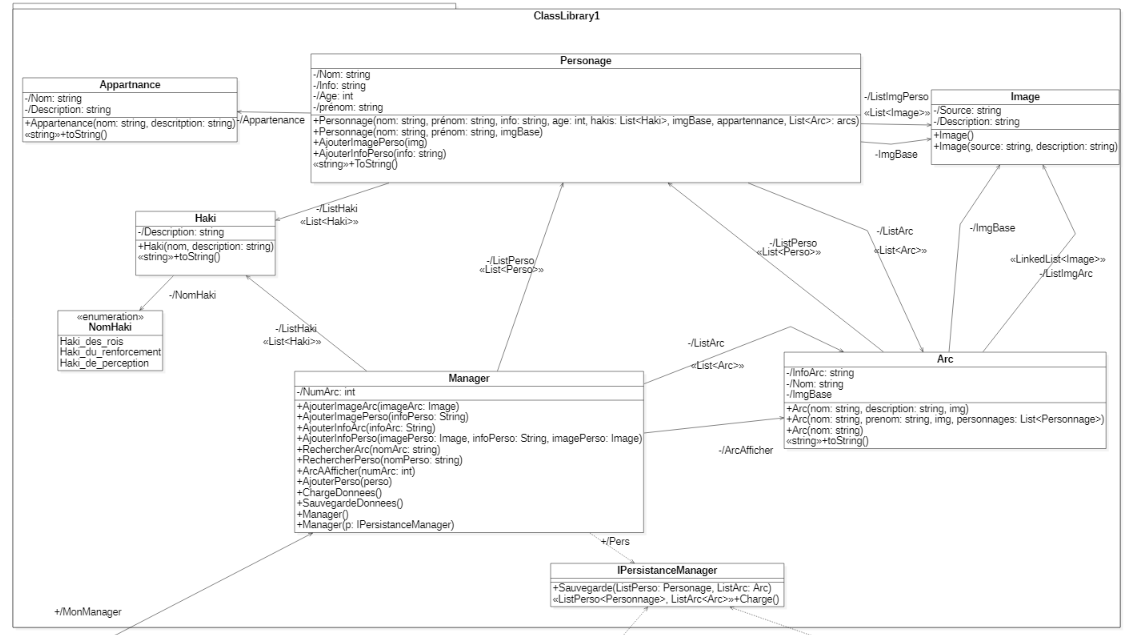


Voici donc notre diagramme de classe complet. Il comporte 5 paquets : Application, ClassLibrary1, Converteur, Stub et DataContractPersistance. Bien sûr dans ce diagramme de classe des éléments ne figurent pas par soucis de place, nous n’arrivions pas à tout faire tenir donc on a préféré enlever certaines vues ainsi que certaines méthodes peu utiles tels que les toString ou les constructeurs que l’on n’a pas forcément toujours précisés.

Le paquet Converteur est relié à l’Application par les ressources de app.xaml. Il ne contient qu’un seul convertisseur Func2WindowPartConverter.cs pour le moment. Nous en rajouterons un pour convertir les NomHaki de la classe Haki qui sont issus de l’énumération NomHaki.

Le paquet ClassLibrary1 est relié à l’Application par le manager qui est déclaré en propriété dans App.xaml.cs.

Pour les paquets Stub et DataContractPersistance la même technique a été utilisée pour les relier à ClassLibrary, la classe principale des deux paquets implémente l’interface IPeristanceManager. On précise que ces deux paquets sont prévus pour le stockage des données. Le paquet stub ne peut que charger les données dans l’application. Tandis que le paquet DataContractPersistance lui permet de charger mais aussi de sauvegarder les données de l’application.

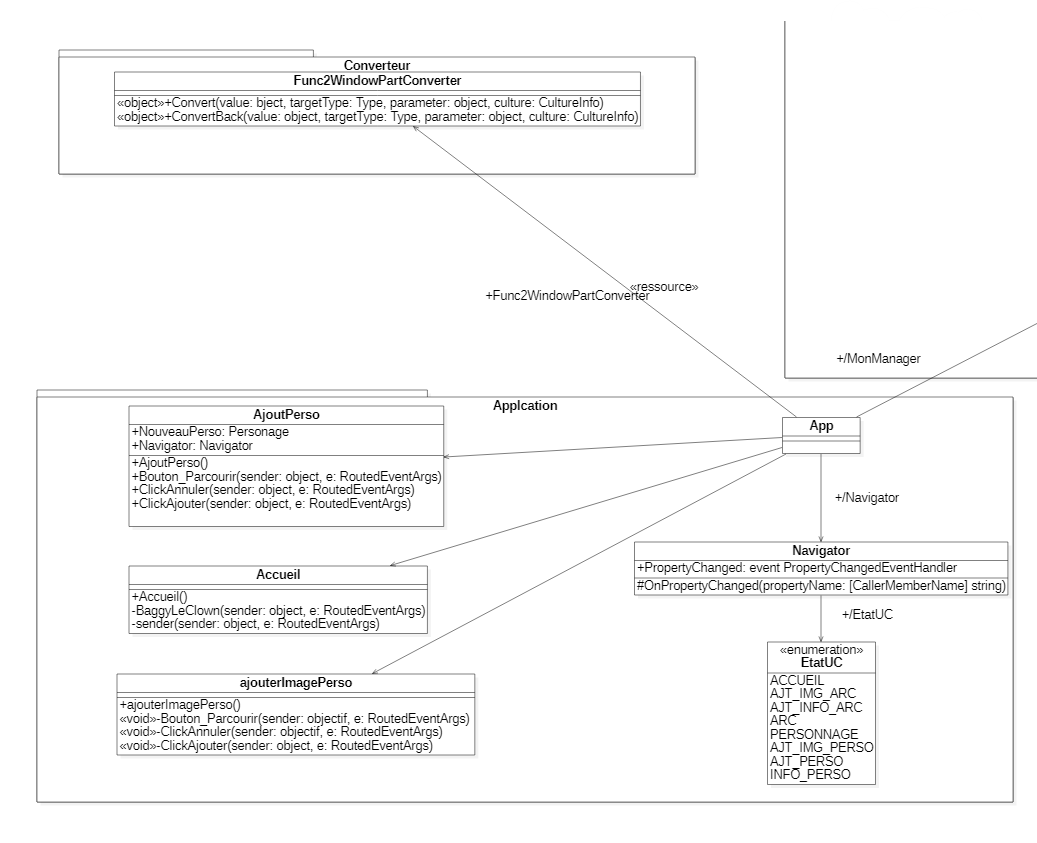


Maintenant rentrons un peu plus dans le détail de ClassLibrary :

Ce paquet contient :

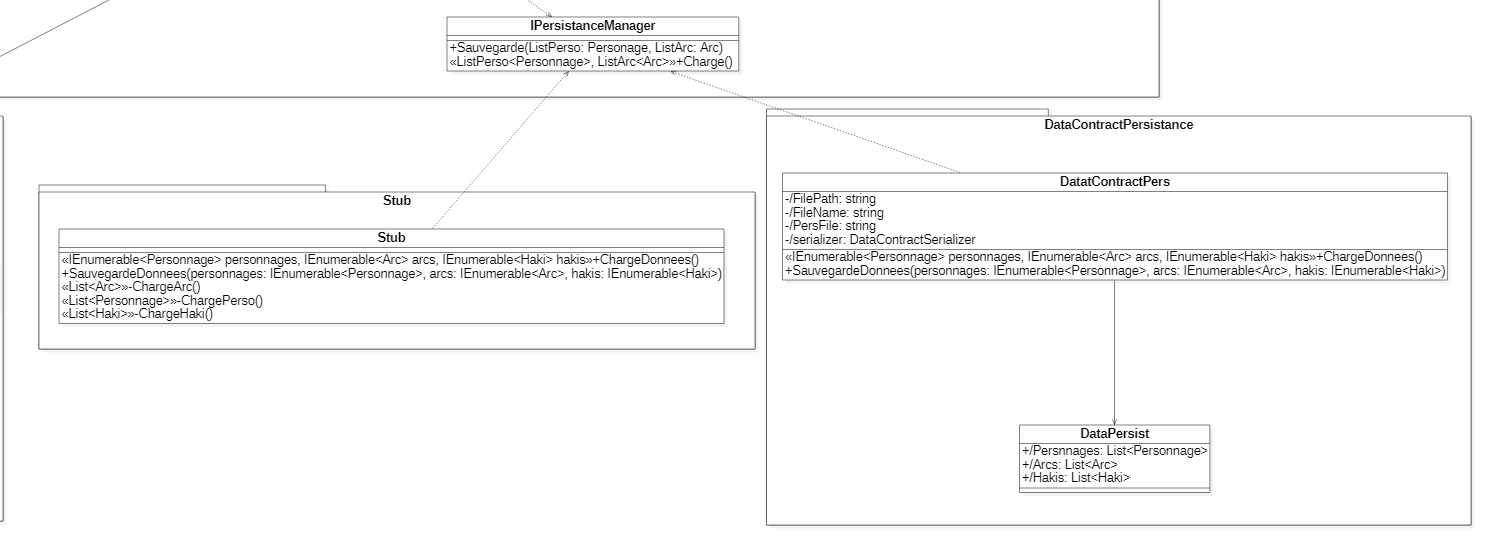
* 6 classes
* 1 énumération
* 1 interface

Toutes les classes sont là pour présenter les personnages et les arcs sauf le manager qui réunit toutes les classes. En effet le manager a 3 collections importantes, une pour les personnages, une pour les arcs et une pour regrouper les différents hakis. Toutes les collections de notre application sont des listes car nous n’avions pas besoin ni de dictionnaire, ni de collection complexe, de simple liste suffisait. Nous pouvons préciser que le Manager est relié au IPersistanceManager en l’implémentant mais aussi en ayant une propriété qui nous permet de changer de mode de sauvegarde ou de récupération. Nous avons également mis toutes nos propriétés en privé pour respecter l’encapsulation.



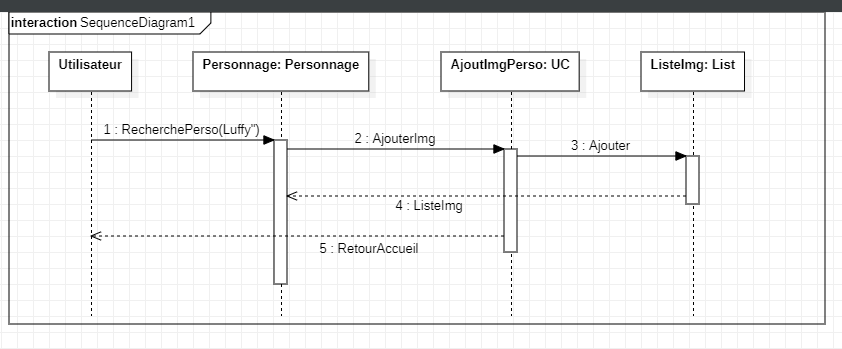
Maintenant voyons plus le paquet Application, qui est appelé Projet dans le diagramme de paquetage. Il est lié à ClassLibrary avec la propriété MonManager. Il est ensuite relié à toutes les vues de notre application que nous n’avons pas pu représenter ici par soucis de place. Nous avons gardé celle qui représente le plus notre application :

* AjoutPerso, qui nous permet d’ajouter des personnages les boutons ont toujours les mêmes paramètres, le nouveau personnage est là pour pouvoir être ajouté et le navigateur pour naviguer sur une autre page.
* Accueil qui contient un des boutons pour chacun de nos arcs, nous ne les avons donc pas tous représentés ici.
* ajoutImagePerso qui comprend des boutons qui ont tous les mêmes paramètres



Et pour finir sur ce qui est plutôt de la partie 4, la persistance. Nous avons donc, dans un premier temps, fait le **Stub** pour pouvoir charger les données qui sont rentrées en dur dessus. Et dans un second temps, le **DataContractPersistance** pour pouvoir avoir un fichier de chargement mais aussi de sauvegarde. La classe Stub permet de charger les données qu’elle contient. DataContract permet de charger et sauvegarder les données dans un fichier dont toutes les infos sont en propriété, le nom : FileName, le chemin : FilePath. PersFile est là pour combiner le nom et le chemin d’accès. Nous utilisons la classe DataPersist pour pouvoir charger toutes les données dans le même fichier.

**Diagramme de séquence**



Ici nous pouvons voir le diagramme de séquence dans le cas où nous ajoutons une image a la galerie du personnage qui se prénomme « Luffy », tout d’abord on doit rechercher le personnage de Luffy dans la barre de recherche ensuite si le personnage existe celle-ci nous envoie sur la page du personnage. Ensuite on doit cliquer sur le bouton « Ajouter Image » afin de lui ajouter une image à sa galerie, tout ceci va ouvrir un nouveau UserControl et donc une fois l’image ajoutée et sa description on n’a plus qu’à appuyer sur le bouton « Ajouter » ce qui va ajouter l’image dans la galerie est aussi nous ramener à la page du personnage afin que l’on voit l’image rajouter dans la galerie.

**Ergonomie**

Pour l’ergonomie, toutes les fonctionnalités de l’application sont disponibles en quelques clics dès l’écran d’accueil. De plus le fait de mettre la carte du monde de One Piece rend l’application plus intéressante et orignale, cela donne envie de cliquer sur chaque île pour se documenter sur les évènements qui s’y sont produits. Les fonctionnalités sont également claires et définies avec des boutons qui portent leur nom pour éviter toutes confusions et pour rendre l’application le plus simple d’utilisation possible.

**Accessibilité**

Au sujet de l’accessibilité, l’application est très simple d’utilisation par exemple avec une carte en page d’accueil qui regroupe toutes les îles et qui évite ainsi de chercher et se promener sur la carte. Ensuite l’utilisation de l’application est aussi assez aisée avec des boutons très clairs et pas cachés donc c’est bien plus simple pour l’utilisateur lambda de s’y retrouver.