
Sommaire:

1-Introduction

1.1-Rappel des objectifs

1.2-Présentation des fonctionnalités

2-Arborescence de l'application

3-Présentation des fonctionnalités

3.1-Diagramme des cas d'utilisation

4-Descriptif des fonctionnalités

4.1-Diagramme d'activité

5-Architecture de déploiement

5.1-Diagramme de déploiement

6-Architecture de l'application

6.1-Model

6.2-View

6.3-Controller

1-Introduction

1.1-Rappel des objectifs

Le thème du sujet pour l'application à réaliser est Marvel: <https://developer.marvel.com/docs>.

L'application à réaliser affiche à minima:

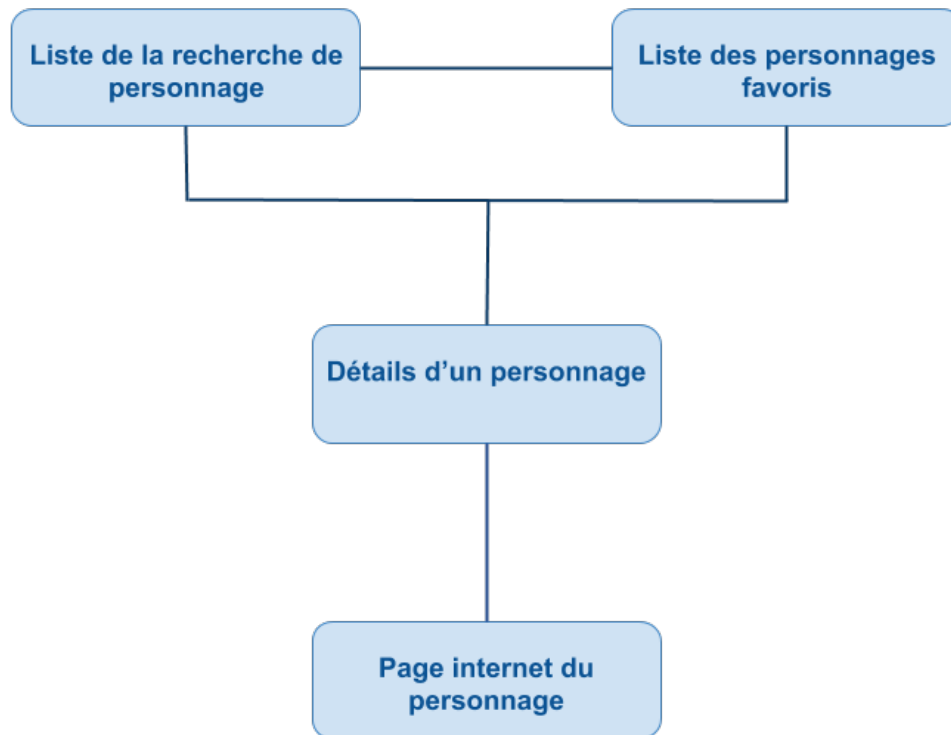
- Une liste
- Le détail d'une des types de données présent dans l'API.

1.1-Présentation des fonctionnalités

L'application que nous proposerons pour ce test devra permettre à l'utilisateur de:

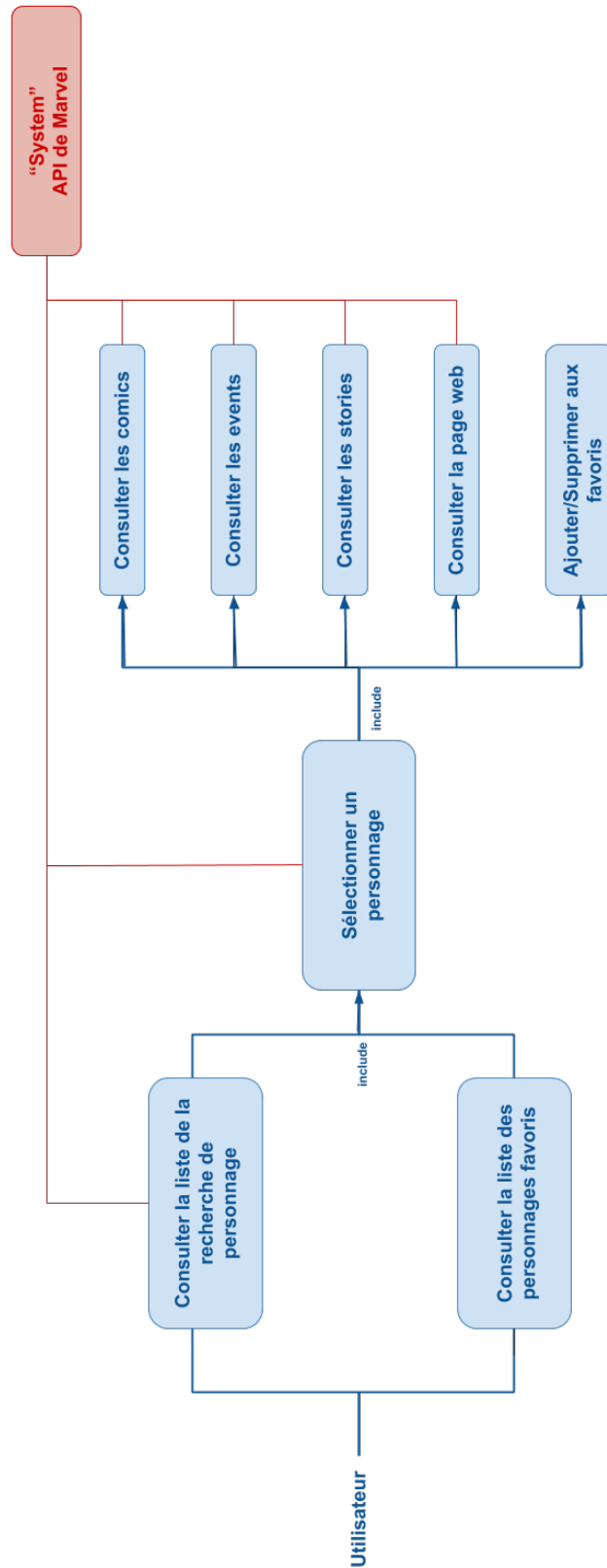
- Rechercher les personnages Marvel par leur nom
- Sélectionner le personnage voulu dans la liste proposée par l'API
- Voir les informations complémentaires sur ce personnage disponibles sur l'API:
 - Description
 - Comics
 - Events
 - Stories
- Sauvegarder un personnage dans les favoris
- Supprimer un personnage des favoris
- Consulter la liste des favoris
- Voir les informations complémentaires sur ce personnage sauvegardé disponibles sur l'API:
 - Description
 - Comics
 - Events
 - Stories
- Accéder à la page internet du personnage.

2-Arborescence de l'application



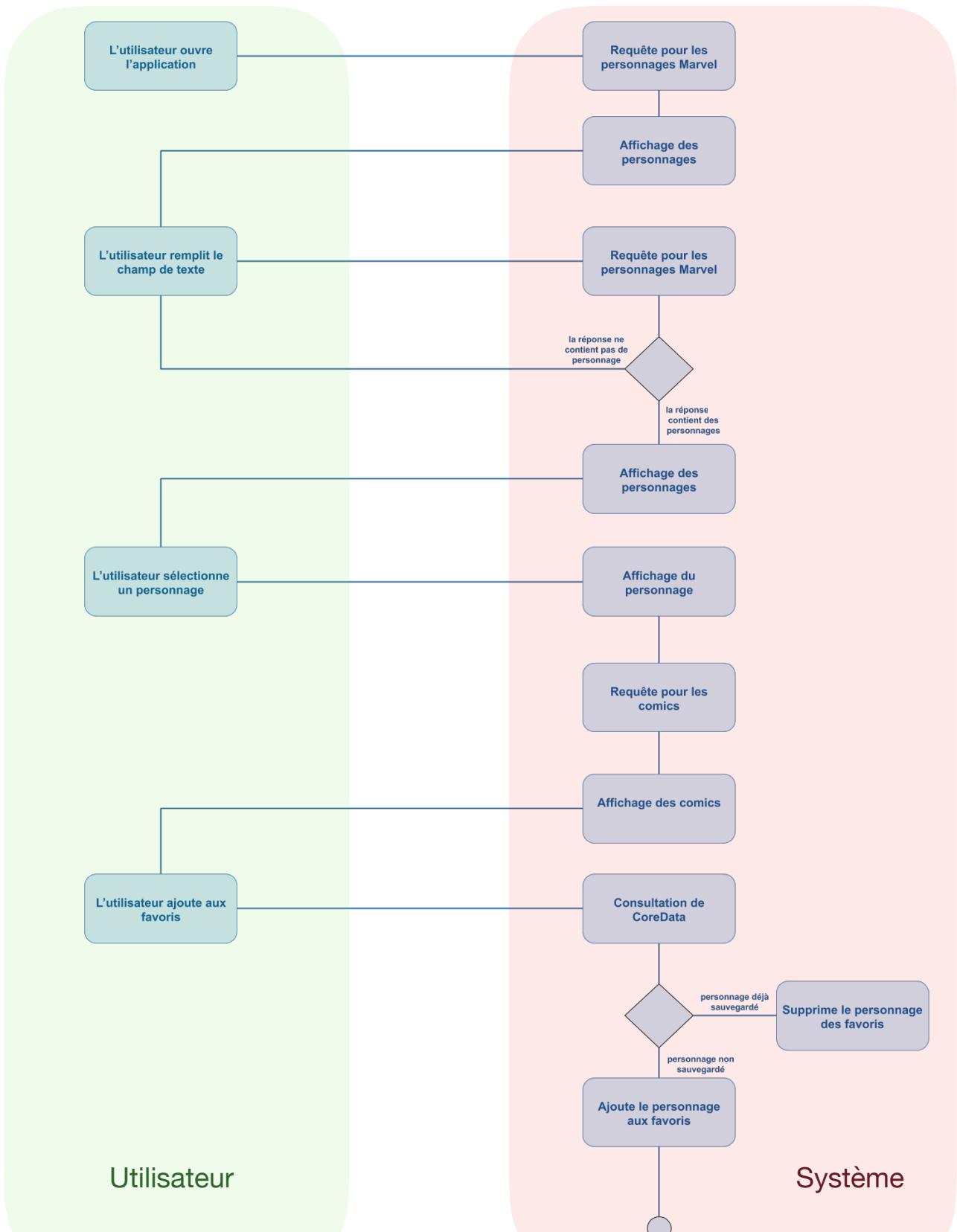
3-Présentation des fonctionnalités

3.1-Diagramme des cas d'utilisations



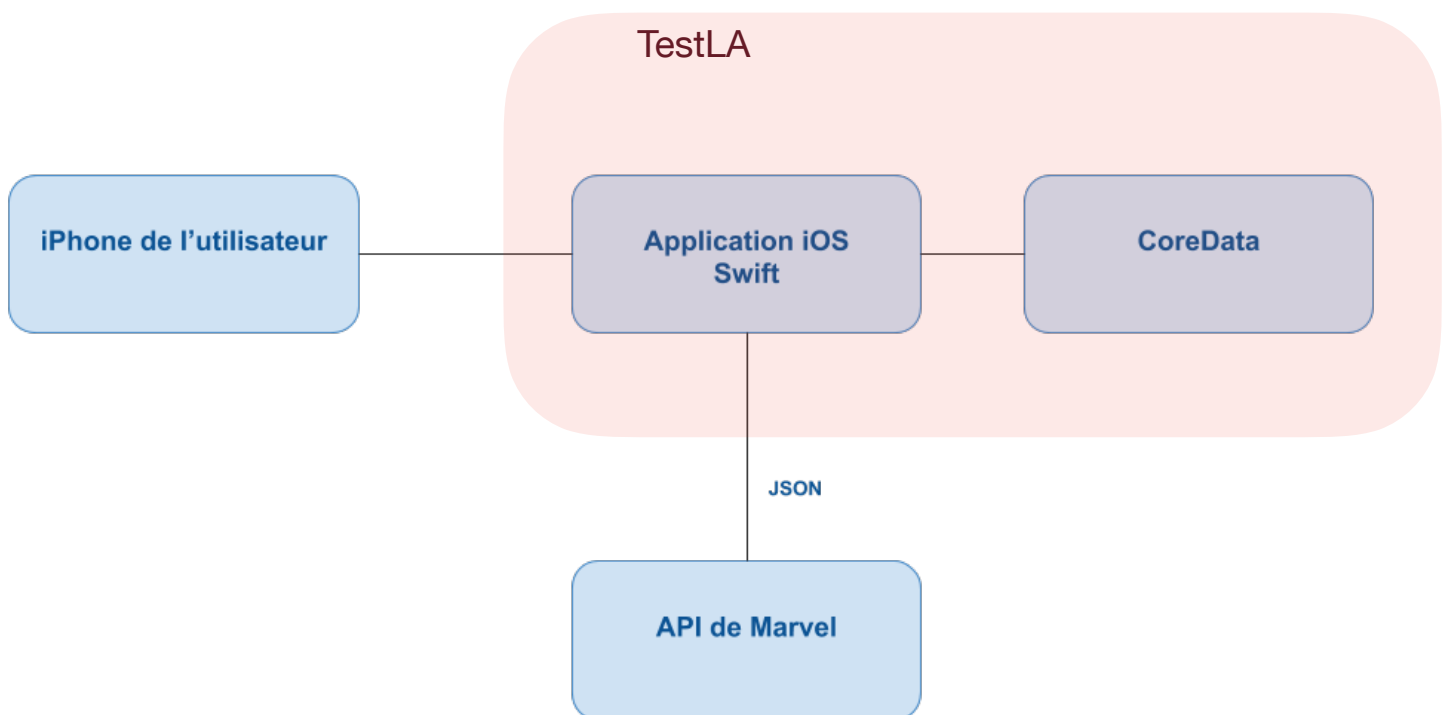
4-Descriptif des fonctionnalités

4.1-Diagramme d'activité - Ajouter un héros aux favoris



5-Architecture de déploiement

5.1-Diagramme de déploiement



6-Architecture de l'application

6.1-Model

Le modèle contient un dossier nécessaire à CoreData.

Il contient également:

- Le fichier **ComicData.swift** qui est la structure de réponse des requêtes pour les comics, les events et les stories.
- Le fichier **MarvelData.swift** qui est la structure de réponse des requêtes pour les personnages
- le fichier **MarvelURL.swift** qui contient structure **MarvelURL**, cette structure contient les différentes variables (String) composant les URL nécessaires aux requêtes pour les personnages et aux requêtes pour les comics, events et stories.

6.1-Model(suite)

- Le fichier `MarvelService.Swift` qui contient la classe `MarvelService`, cette classe contient les fonctions `getMarvel` et `getComics` qui sont les appels réseaux envoyé à l'API de Marvel. Elle utilise `MarvelURL`, `ComicData` et `MarvelData`.
- Le fichier `Extensions` qui contient une extension de `UIViewController` permettant de simplifier l'affichage d'alerte, et une extension `UIView` permettant de convertir une `UIView` en `UIImage`.

6.2-View

La View contient les fichiers `Lauchscreen.storyboard` et `Main.storyboard`.

Il contient également:

- Le fichier `MarvelTableViewCell.swift` contenant la classe `MarvelTableViewCell` permettant de configurer les cellules de la tableView présentant les personnages
- Le fichier `FavoriteTableViewCell.swift` contenant la classe `FavoriteTableViewCell` permettant de configurer les cellules de la tableView présentant les personnages favoris
- Le fichier `MarvelDetailsHeaderTableViewCell.swift` contenant la classe `MarvelDetailsHeaderTableViewCell` permettant de configurer la première cellule de la tableView contenant les formations détaillés d'un personnage (l'image et le nom)
- Le fichier `MarvelDetailsMiddleTableViewCell.swift` contenant la classe `MarvelDetailsMiddleTableViewCell` permettant de configurer la deuxième cellule de la tableView contenant les formations détaillés d'un personnage (a description, les boutons et le segmentedControl)
- Le fichier `MarvelDetailsFooterTableViewCell.swift` contenant la classe `MarvelDetailsFooterTableViewCell` permettant de configurer les autres cellules de la tableView contenant les formations détaillés d'un personnage (les comics, events et stories)

6.3-Controller

Le Controller contient:

- Le fichier `MarvelSearchViewController.swift` contenant la classe `MarvelSearchViewController` qui contrôle la vue de recherche de personnages. Il se sert de `MarvelService`, `MarvelURL`, et `MarvelTableViewCell`
- Le fichier `MarvelDetailsViewController.swift` contenant la classe `MarvelDetailsViewController` qui contrôle la vue des détails d'un personnage. Il se sert de `MarvelService`, `MarvelURL`, `MarvelDetailsHeaderTableViewCell`, `MarvelDetailsMiddleTableViewCell`, `MarvelDetailsFooterTableViewCell` et également de `CoreData`.
- Le fichier `MarvelFavoriteViewController.swift` contenant cela classe `MarvelFavoriteViewController` qui contrôle la vue des personnages favoris. Il se sert de `CoreData`, `MarvelService`, `MarvelURL` et de `FavoriteTableViewCell`.
- Le fichier `MarvelWebLinkViewController` contenant la classe `MarvelWebLinkViewController` qui contrôle la vue allant sur le site internet Marvel.