

Projet Galt

Spécifications

Version 1.0

23/10/2016

Historique des révisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Auteur** |
| 13/10/2016 | 0.1 | Première description | AC |
| 18/10/2016 | 0.2 | Cas d’utilisation réalisés | AC |
| 19/10/2016 | 0.3 | Première version complète | AC |
| 23/10/2016 | 1.0 | Première version finalisée et révisée | AC |

Sommaire

1 Introduction 6

1.1 Contexte initiale, historique et vision 6

1.2 Mission 6

1.3 Objectifs 6

1.4 Glossaire 6

1.5 Documents de référence 6

2 Description générale 7

2.1 Acteurs 7

2.2 Cas d’utilisations 7

3 Spécifications fonctionnelles 9

3.1 Carte de navigation 9

3.2 Détails des cas d’utilisations 9

4 Spécifications non fonctionnelles 13

4.1 Environnement opérationnel 13

4.2 Contraintes de conception et d’implémentation 13

4.3 Documentation utilisateur 13

4.4 Exigences de performance 14

4.5 Exigences de sécurité 14

4.6 Attributs de qualité 14

# Introduction

## Contexte initial, historique et vision

Cf. Etude d’opportunité et de faisabilité

## Mission

Cf. Etude d’opportunité et de faisabilité

## Objectifs

Cf. Etude d’opportunité et de faisabilité

## Glossaire

### Termes du domaine

Dépendance : Intégration d’un package *x* au sein d’un autre package *y*. On dit que le package *y* dépend de *x*.

Package : Archive contenant des fichiers et des informations nécessaires à l’installation d’un logiciel.

NuGet : Gestionnaire de packages en ligne. Il sera celui utilisé sur le site Galt.

Graphe de dépendances : Graphe orienté acyclique (qui ne possède pas de circuit) représentant l’architecture des dépendances d’un package.

### Termes techniques

Chaîne de build : Outil permettant de vérifier à chaque modification du code source que les modifications ne créent pas une régression dans le logiciel en vue du respect de la continuous integration.

C# : Langage qui permet de réaliser des sites web dynamiques.

ASP .NET Core : framework facilitant la conception d’applications web.

JS (Javascript) : Langage utilisé sur les pages web, s’exécutant côté client.

Vue.js : framework facilitant la conception de SPA.

D3.js : bibliothèque Javascript permettant l’affichage de données graphiques et dynamiques.

Azure : Service cloud permettant le stockage de données

SPA (Single Page Application) : application web accessible via une page web unique

Modèle MVC (Model, View, Controller) : Manière d’organiser son code de façon à respecter le principe de séparation des responsabilités dans une application.

Côté client / côté serveur : Le côté client correspond aux langages s’exécutant sur la machine de l’Utilisateur ; le côté serveur correspond aux langages s’exécutant sur le Système.

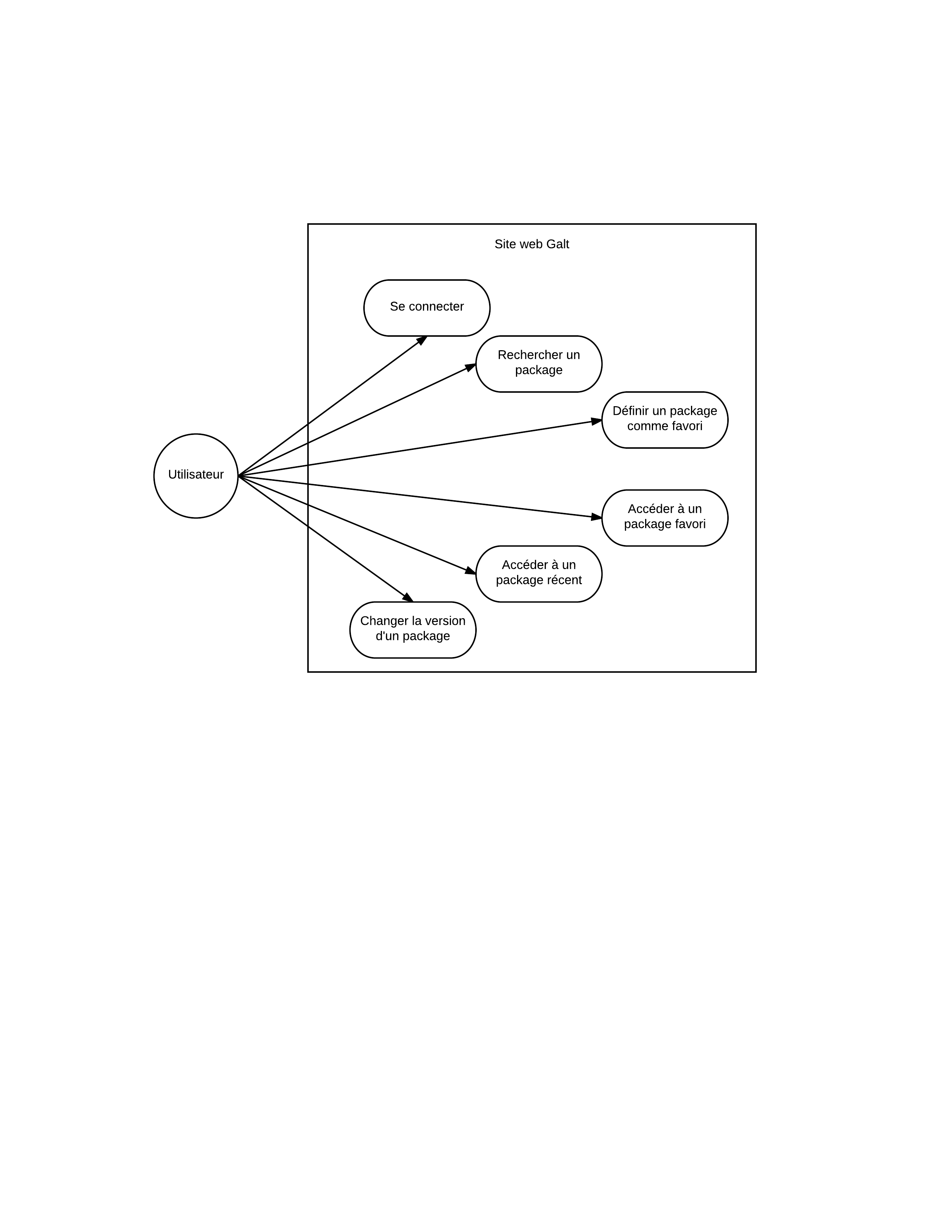
# Description générale

## Acteurs

|  |  |
| --- | --- |
| **Nom** | **Description** |
| Utilisateurs | Développeurs ayant besoin de connaître l’ensemble des dépendances de leur package ainsi que l’actualité des mises à jour de celles-ci |

## Cas d’utilisations

### Diagramme de cas d’utilisation



### Listes des cas d’utilisations

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom du cas d’utilisation** | **Acteur(s)** | **Priorité** | **Risque** |
| Se connecter | Utilisateur  Système | Haute | Moyen |
| Rechercher un package | Utilisateur  Système | Haute | Haut |
| Définir un package comme favori | Utilisateur  Système | Moyenne | Moyen |
| Accéder à un package favori | Utilisateur  Système | Moyenne | Faible |
| Accéder à un package récent | Utilisateur  Système | Faible | Moyen |
| Changer la version d’un package | Utilisateur  Système | Moyenne | Haut |

# Spécifications fonctionnelles

## Carte de navigation

Nous avons réalisé notre diagramme de navigation sur le site Lucidchart :

<https://www.lucidchart.com/invitations/accept/1e09256e-a96f-477b-a76a-8832e0678b13>

Nous avons également réalisé des maquettes de nos vues :

<https://classic.moqups.com/acharrier@intechinfo.fr/W1B9DKyP>

## Détails des cas d’utilisations

### Nom du cas

Se connecter

#### Description :

L’Utilisateur se connecte au site Galt grâce à son compte GitHub.

#### Acteur principal :

L’Utilisateur qui se connecte.

#### Acteurs secondaires :

L’API GitHub qui connecte l’Utilisateur, le Système qui redirige l’Utilisateur vers l’API GitHub.

#### Préconditions :

L’Utilisateur est sur n’importe quelle page du site.

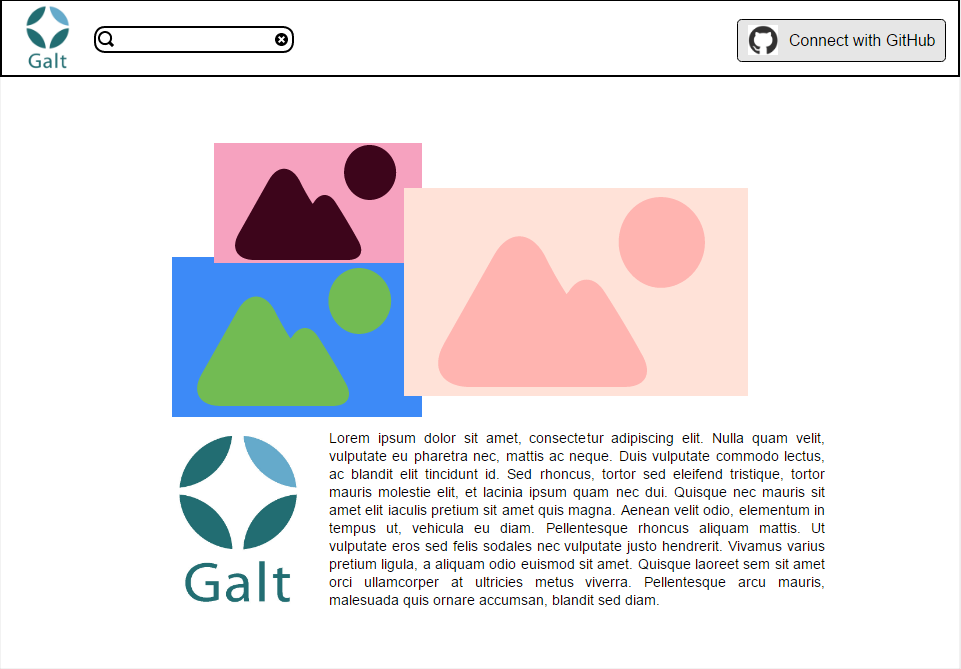
#### Postconditions (garantie en cas de succès) :

Le Système a connecté l’Utilisateur.

#### Scénario nominal :

1. Le cas débute lorsque l’Utilisateur clique sur « Connect with GitHub ».
2. Le Système redirige vers la connexion vers l’API GitHub.
3. L’API GitHUB connecte l’Utilisateur.
4. Le Système renvoie la page d’accueil avec l’Utilisateur connecté.

#### Interface utilisateur:



Page d’accueil lorsque l’Utilisateur n’est pas connecté



Page d’accueil lorsque l’utilisateur est connecté

#### Fréquence :

Régulière

### Nom du cas

Rechercher un package

#### Description :

L’Utilisateur recherche un package présent sur NuGet.

#### Acteur principal :

L’Utilisateur qui recherche le package.

#### Acteurs secondaires :

Le Système qui affiche le package.

#### Préconditions :

L’Utilisateur est sur la page d’accueil (ou n’importe quelle autre page du site).

#### Postconditions (garantie en cas de succès) :

Le Système renvoie la page du package.

#### Scénario nominal :

1. Le cas débute lorsque l’Utilisateur clique dans la barre de recherche.
2. L’Utilisateur tape le nom du package qu’il recherche.
3. Le Système affiche les résultats (4 au maximum) correspondants sous la barre de recherche et « More results ».
4. L’Utilisateur clique sur le package souhaité.
5. Le Système affiche la page du package.

#### Extensions:

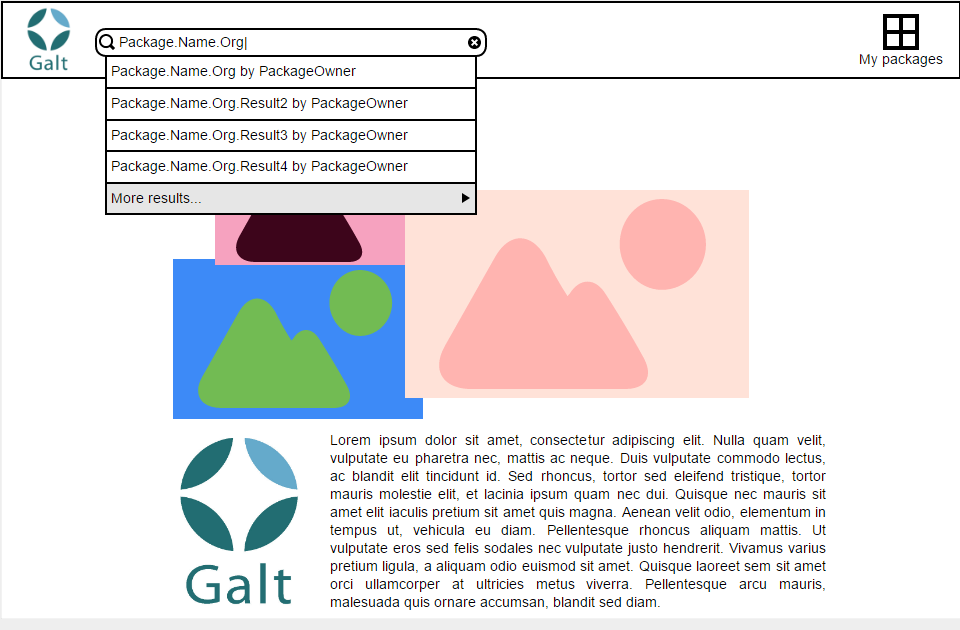
2a . Si le Système ne trouve pas de résultat, alors :

1. Le Système « No results » sous la barre de recherche.

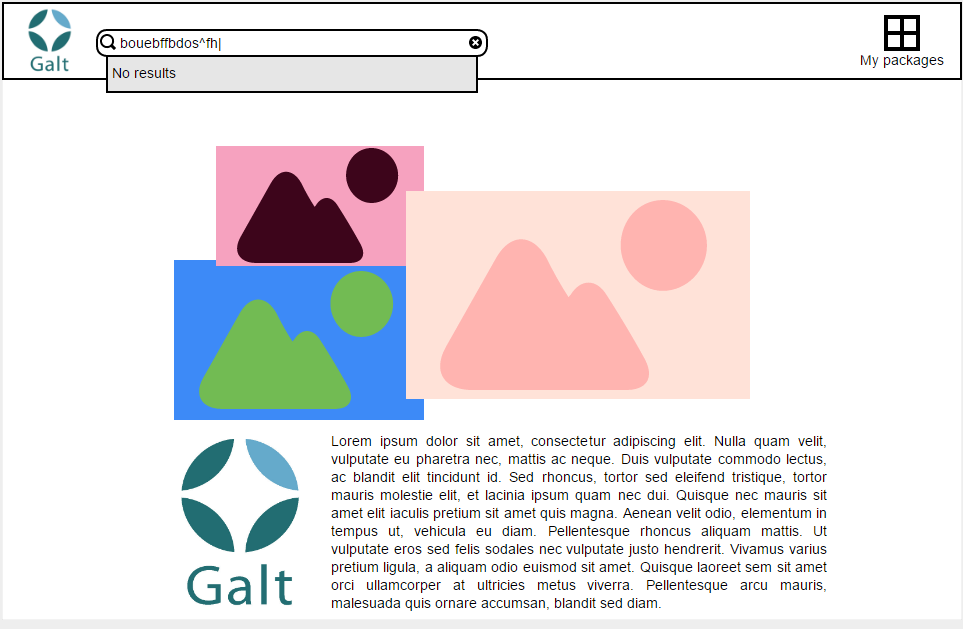
4a . Si l’Utilisateur clique sur « More results » ou appuie sur Entrée, alors :

1. Le Système affiche une nouvelle page avec tous les résultats.
2. L’Utilisateur clique sur le package souhaité.
3. Le Système affiche la page du package.

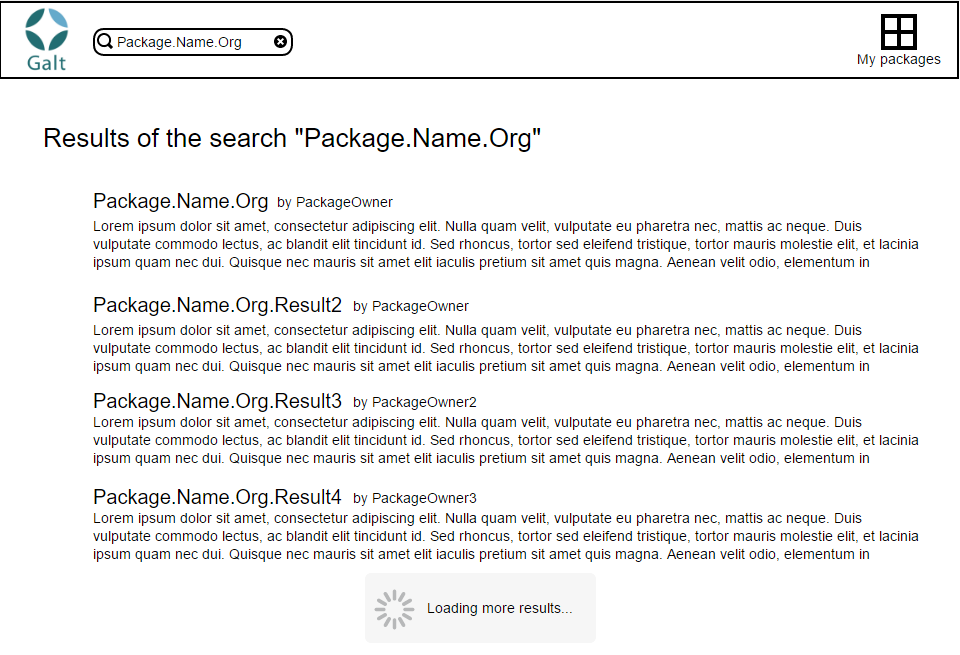
#### Interface utilisateur:



Page d’accueil lorsque l’Utilisateur effectue une recherche



Page d’accueil lorsque le Système ne trouve pas de résultat après une recherche de l’Utilisateur



Page qui s’affiche lorsque l’Utilisateur a cliqué sur « More results »

#### Fréquence :

Régulière

### Nom du cas

Définir un package comme favori

#### Description :

L’Utilisateur cherche à définir un package comme favori

#### Acteur principal :

L’Utilisateur qui cherche à ajouter un package dans ses favoris

#### Acteurs secondaires :

Le Système qui ajoute le package aux favoris de l’Utilisateur.

#### Préconditions :

L’Utilisateur est sur la page d’un package.

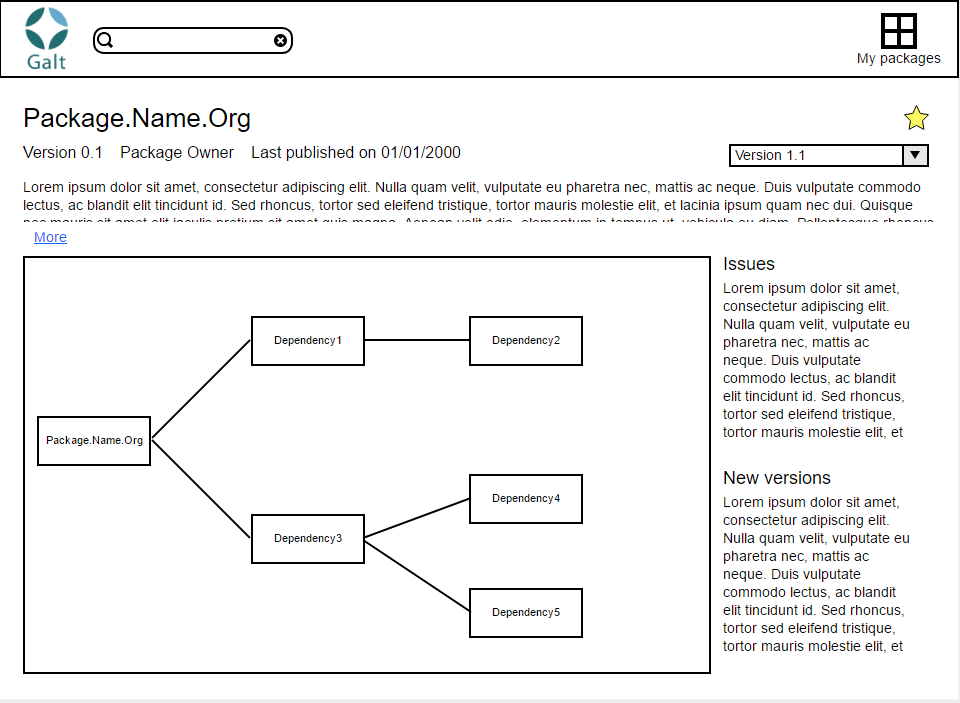
#### Postconditions (garantie en cas de succès) :

Le package est enregistré dans les favoris de l’Utilisateur

#### Scénario nominal :

1. Le cas débute lorsque l’Utilisateur clique sur le bouton gris en forme d’étoile.
2. Le Système ajoute le package actuel dans les favoris de l’Utilisateur.
3. Le Système transforme le bouton gris en bouton jaune.

#### Interface utilisateur:



Page « My packages » lorsque l’Utilisateur a ajouté le package en favori (l’étoile est jaune)

#### Fréquence :

Occasionnelle

### Nom du cas

Accéder à un package favori

#### Description :

L’Utilisateur cherche à accéder à un package enregistré dans ses favoris.

#### Acteur principal :

L’Utilisateur qui cherche à accéder au package.

#### Acteurs secondaires :

Le système qui affiche le package.

#### Préconditions :

L’Utilisateur a enregistré le package dans ses favoris.

#### Postconditions (garantie en cas de succès) :

Le Système renvoie la page du package.

#### Scénario nominal :

1. Le cas débute lorsque l’Utilisateur clique sur « My packages ».
2. L’Utilisateur cherche le package désiré dans la liste des packages favoris.
3. L’Utilisateur clique ensuite sur le bouton « View » en face du package qu’il souhaite afficher.
4. Le Système affiche le package.

#### Extensions:

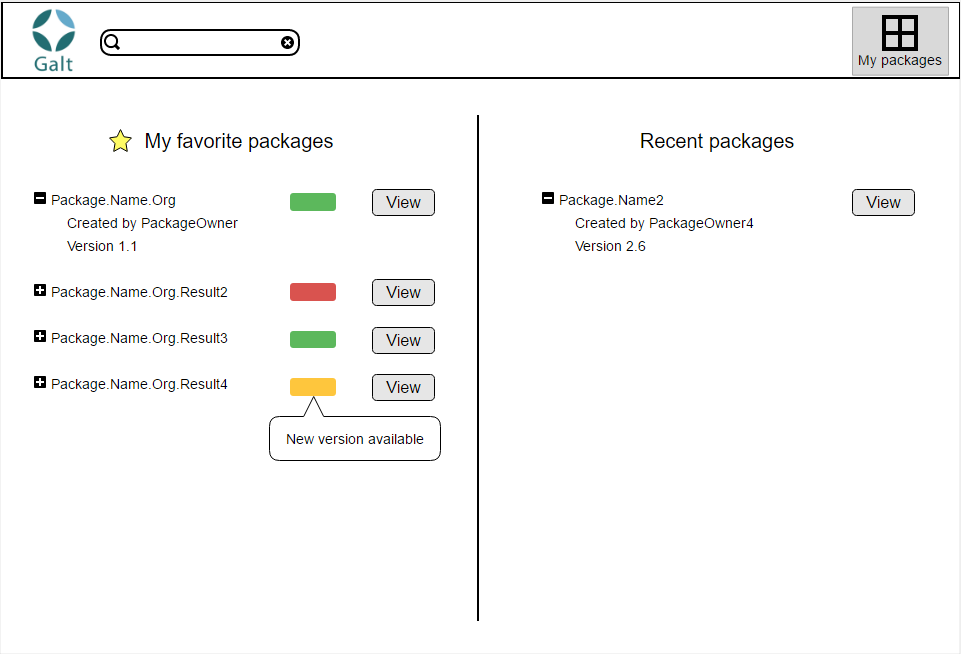
2a . Si l’Utilisateur passe sa souris sur le voyant (vert, jaune ou rouge), alors :

1. Le Système affiche une bulle d’informations : vert = rien à signaler, jaune = nouvelle version disponible et rouge = conflit de versions.

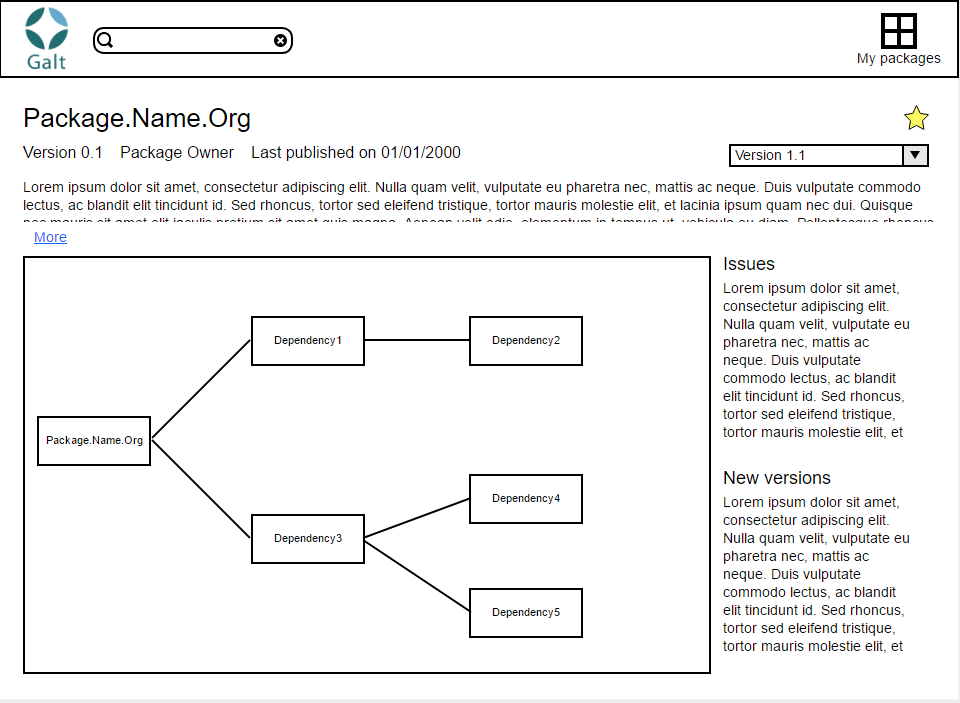
2b . Si l’Utilisateur clique sur le (+) à côté du nom du package, alors :

1. Le Système affiche les détails du package (créateur et version enregistrée).

#### Interface utilisateur:



Page « My packages » avec les différents packages enregistrés comme favoris



Page « Package » qui s’affiche lorsque l’Utilisateur a cliqué sur « View »

#### Fréquence :

Régulière

### Nom du cas

Accéder à un package récent

#### Description :

L’Utilisateur cherche à accéder à un package qu’il a consulté récemment.

#### Acteur principal :

L’Utilisateur qui cherche à accéder au package.

#### Acteurs secondaires :

Le Système qui affiche le package.

#### Préconditions :

Le package est parmi les cinq packages que l’Utilisateur a visité dernièrement.

#### Postconditions (garantie en cas de succès) :

Le Système renvoie la page du package.

#### Scénario nominal :

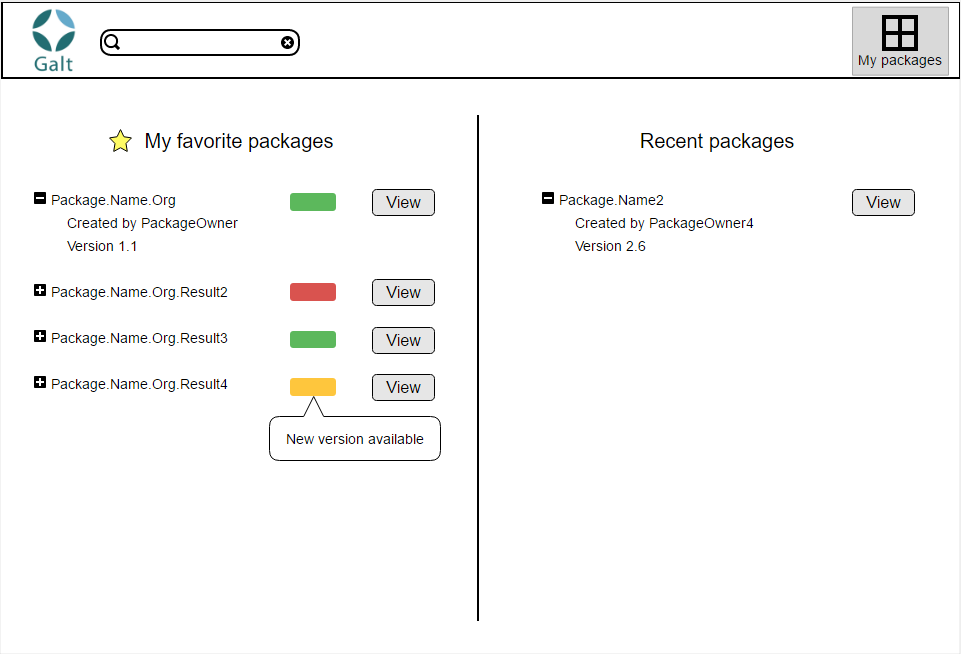
1. Le cas débute lorsque l’Utilisateur clique sur « My packages ».
2. L’Utilisateur cherche le package désiré dans la liste des packages récents.
3. L’Utilisateur clique ensuite sur le bouton « View » en face du package qu’il souhaite afficher.
4. Le Système affiche le package.

#### Extensions:

2a . Si l’Utilisateur clique sur le (+) à côté du nom du package, alors :

1. Le Système affiche les détails du package (créateur et version enregistrée).

#### Interface utilisateur:



Page « My packages » avec les différents packages consultés récemment

#### Fréquence :

Rare

### Nom du cas

Changer la version d’un package

#### Description :

L’Utilisateur souhaite accéder à une version antérieure du package.

#### Acteur principal :

L’Utilisateur qui cherche à accéder à une autre version du package.

#### Acteurs secondaires :

Le Système qui affiche le package avec la version souhaitée.

#### Préconditions :

Le package possède au moins une autre version que la version actuelle. L’Utilisateur est sur la page d’un package.

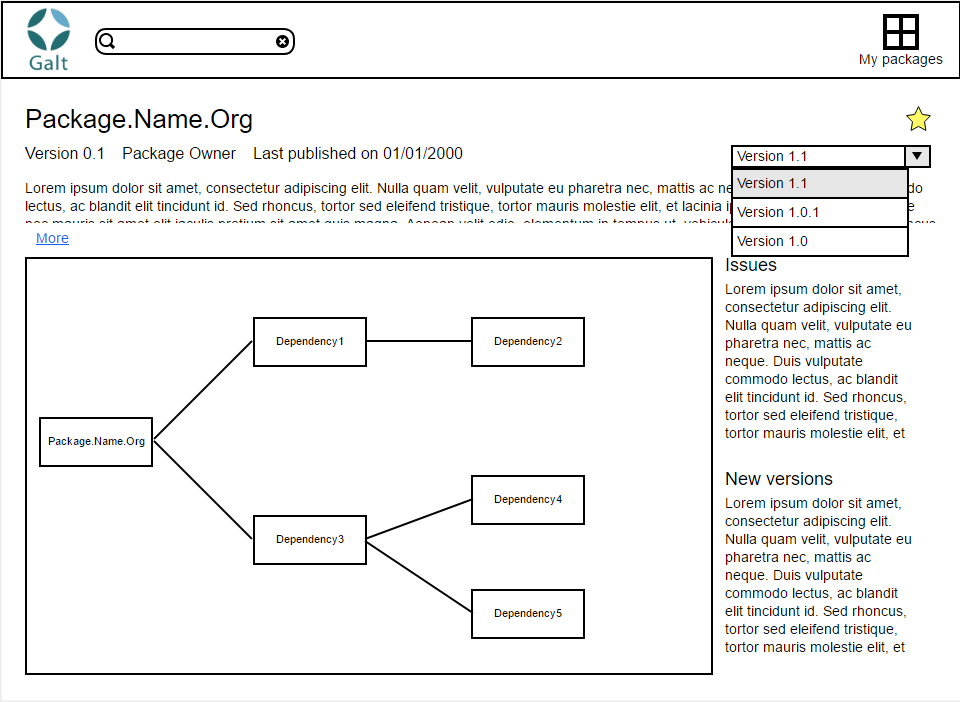
#### Postconditions (garantie en cas de succès) :

Le Système renvoie la page du package avec la version souhaitée.

#### Scénario nominal :

1. Le cas débute lorsque l’Utilisateur clique sur le menu déroulant de la version.
2. Le Système affiche les différentes versions disponibles pour le package.
3. L’Utilisateur clique ensuite sur la version qu’il souhaite.
4. Le Système affiche de nouveau la page du package avec la version souhaitée.

#### Interface utilisateur:



Page d’un package lorsque l’Utilisateur clique sur la liste des versions.

#### Fréquence :

Rare

# Spécifications non fonctionnelles

## Environnement opérationnel

L’équipe utilisera le logiciel Microsoft Visual Studio 2015 comme environnement de travail et Windows comme système d’exploitation (versions 7, 8.1 et 10). Le site sera accessible pour des utilisateurs du monde entier via Google Chrome (dernière version) et Mozilla Firefox (dernière version). Nous utiliserons le service Microsoft Azure Tables pour notre base de données.

## Contraintes de conception et d’implémentation

Technologies :

* ASP .NET Core
* Vue.js 2.0.3
* d3.js 3.5.17
* webpack
* API NuGet

Langages :

* C#
* HTML
* Less

Outils :

* Visual Studio 2015
* Git

BDD : Azure Tables

OS : Windows 7 à 10

Echange d’information : JSON

## Attributs de qualité

* + - * Possibilité d’ajouter d’autres outils en plus de NuGet tels que MVNRepository.
      * Site web en single page, ergonomique et simple d’utilisation grâce à un nombre de pages réduit et un design épuré.
      * Serveur web stable, peu de possibilités de bug grâce aux tests unitaires