PATCHASTIKS

**Mémoire fin de formation**

Table des matières

[Remerciements 3](#_Toc112819156)

[English part 4](#_Toc112819157)

[Abstract 4](#_Toc112819158)

[Technical subject translation 5](#_Toc112819159)

[Introduction 6](#_Toc112819160)

[Compétences couvertes par mon projet 6](#_Toc112819161)

[Etude 6](#_Toc112819162)

[Conception 6](#_Toc112819163)

[Réalisation 6](#_Toc112819164)

[Spécifications fonctionnelles 7](#_Toc112819165)

[Diagramme d’utilisation 7](#_Toc112819166)

[Diagrammes d’activités 9](#_Toc112819167)

[Diagramme de séquence 11](#_Toc112819168)

[Diagramme de classe 11](#_Toc112819169)

[Maquettage 12](#_Toc112819170)

[Arborescence 12](#_Toc112819171)

[Zoning 12](#_Toc112819172)

[Wireframe 16](#_Toc112819173)

[Conception 18](#_Toc112819174)

[Modèle Conceptuel de Données 18](#_Toc112819175)

[Modèle Logique de Données 19](#_Toc112819176)

[Outils Techniques Utilisés 19](#_Toc112819177)

[Fonctionnalités 20](#_Toc112819178)

[Inscription, connexion et déconnexion 20](#_Toc112819179)

# Remerciements

Ce changement de cap avec le début de formation m’emmène aujourd’hui ici, et toute cette année a été pour moi une grande étape que je n’aurais pas pu franchir sans le support de certaines personnes, que je tenais ici à remercier.

Tout d’abord, merci ma famille, me soutenant dans mes choix, leur apport m’est inestimable et je n’aurais jamais pu faire ce que j’aime sans eux.

Je tiens ensuite à remercier les formateurs de l’ADRAR, qui durant toute l’année, on était disponible et à l’écoute, que ce soit pendant les cours, les soirées ou même les week-ends.

Et pour finir merci à ma session, j’ai de la chance d’avoir eu un groupe aussi solidaire, dans les bons et mauvais moments nous venions tous de différentes voies, ce qui a été pour moi le moteur principale du succès de notre groupe.

# English part

## Abstract

*Authors*: Florent Tomps

*Institutions*: Simplify patch notes information

*Abstract body:*

The video game industry is something that is evolving fast, and for every (almost every) change

there are things that we called patch notes, used to summarize what happened when there is

an update. League of Legends is no exception, it needs to add or remove things in order to

keep the casual player and needs adjustment in order to keep the competitive integrity of the

game.

That’s why I’m developing on Patchistiks (Patch / Statistics), a website which is gonna reference change from every patch in an easy understandable way. You will be able to sort by patch, champion or items, and see what has changed between one or more patches. If you take a break from the game or want to check precise information, this will help you a lot (I hope so). Some features will be available like creating an account in order to add a champion or an item as favorite to find them faster from your profile and get notified when they are modified.

To get the different data, Riot Games (producer of League of Legends) made an API

(Application Programming Interface) in order to help the community create content around their games by different ways, some develop website about in game stats, about the game lore and etcetera.

Advantages:

● Accessible and easy to use website.

● Get notified when you there are modifications of your favorite champion.

● Precise information from different patches.

Drawbacks:

● Require internet connection.

● Can’t get the new information if Riot Games API is unavailable.

● The first version will start from the current patch, so it will need time to be really

useful.

## Technical subject translation

Extracted from the official Riot games developer website:

[**https://developer.riotgames.com/docs/lol**](https://developer.riotgames.com/docs/lol)

***English part:***

**“**This set of documentation will help you gain a better understanding of how to use the APIs and developer resources related to League of Legends. It is designed to help you begin developing your own tools and products for the League of Legends player community.

Before starting on a product for League of Legends, there are **two requirements every developer must complete:**

-Read the [general policies](https://developer.riotgames.com/policies/general)

-[Register your product](https://developer.riotgames.com/app-type)

Before you start reading this documentation you need to first [login](https://developer.riotgames.com/login) with your Riot Games account. Once you do, a Developer Portal account is created for you! This action also generates a basic development API key that is associated with your account. Parts of this documentation will reference trying things out in the [API reference](https://developer.riotgames.com/apis) page. While this documentation is by no means exhaustive, we believe it will give you a great start toward building products in the League of Legends ecosystem. As always, if you run into any difficulties or are having technical issues, please join [Developer Discord](https://discord.gg/riotgamesdevrel) for support.

**”**

***French part :***

Cet ensemble de documentation vont vous aidez à avoir une meilleure compréhension de comment utilisé l’API et les ressources développeurs de League of Legends. Il est conçu pour vous aider à commencer le développement de vos propres outils et applications pour la communauté de League of Legends.

Avant de commencer sur un produit pour League of legends, chaque développeur doit remplir deux conditions :

-Lire les conditions générales

-Enregistrez votre produit

Avant de commencer la lecture de cette documentation vous devez vous connecter avec votre compte Riot games. Après l’avoir fait, un compte développeur est créé pour vous ! Cette action génère aussi une clé d’API basique associé à votre compte. Une partie de cette documentation fait référence à des essais dans la page référence de l’API. Tandis que cette documentation ne soit en aucun cas exhaustive, nous pensons qu’elle vous donnera un bon départ pour créer des produits dans l’écosystème de League of Legends. Comment toujours, si vous rencontrez des difficultés où rencontrez des problèmes techniques, rejoignez le discord développeur pour obtenir de l’aide.

# Introduction

Durant cette année, un projet fil rouge nous a été demandé, avec la possibilité de le faire sur un sujet de notre choix, remplissant bien entendu les différents critères de l’examen.

J’ai fait le choix de prendre pour sujet League of Legends, un jeu vidéo en ligne. League of Legends existant depuis maintenant plus de 10 ans, de nombreux patchs apportant divers ajouts, suppressions ou modifications.

J’ai eux pour idée de développer un site permettant de référencer les différentes modifications, donnant la possibilité de retrouver facilement les changements sur les champions, objets, ou autres éléments disponibles dans le jeu.

## Compétences couvertes par mon projet

### Etude

- Recherche des différentes options possibles pour la création de l’application

- Etude des différentes applications déjà existantes

### Conception

- Création des différentes maquettes

- Réalisation des différentes chartes visuelles

- Mise en place des différents diagrammes d’activités

- Préparation du modèle logique de données

### Réalisation

- Création de la base de données

- Réalisation d’une interface utilisateur responsive

- Développer la partie back-end de l’application, concernant l’utilisation des différentes données disponibles

# Spécifications fonctionnelles

## Diagramme d’utilisation

Le premier diagramme créé durant la conception de mon projet et le diagramme d’utilisation, autrement nommé « Use Case »

Celui-ci comporte les fonctionnalités disponibles en fonction du type d’utilisateur, avec un code couleur ayant pour but de différencier les possibilités :

- Rouge pour l’affichage

- Bleu pour la gestion du compte

- Cyan pour la gestion des favoris

- Orange pour les paramètres de notifications

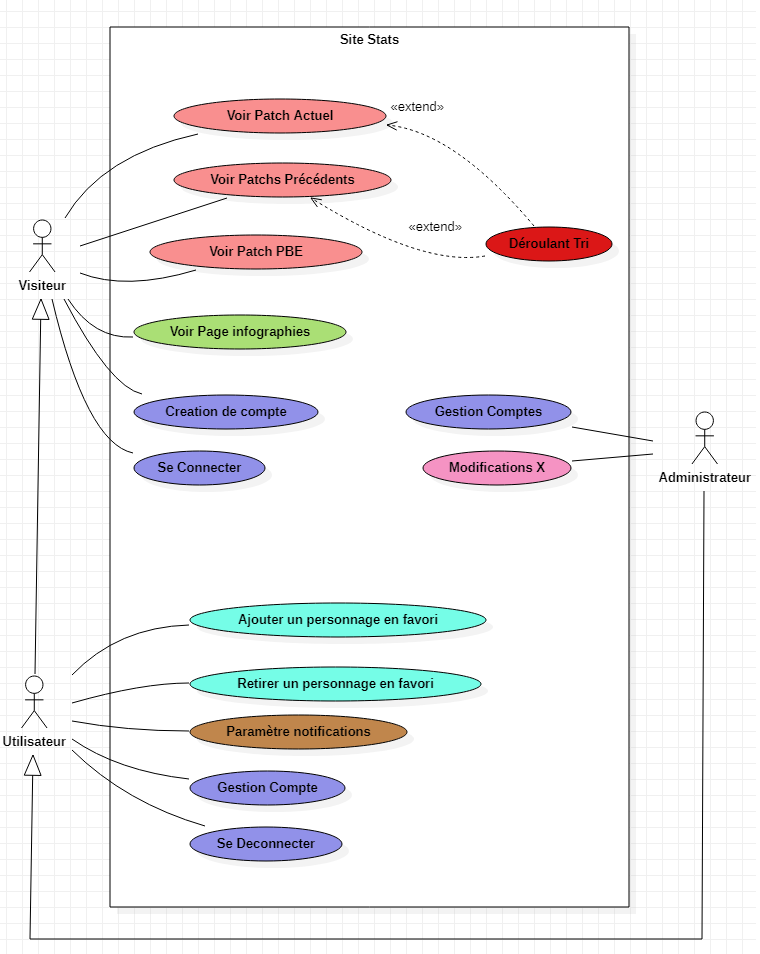
On peut trouver 3 acteurs sur ce diagramme

En premier « le visiteur », un utilisateur non connecté, ayant la possibilité de voir les différentes pages disponibles sur le site, s’inscrire et ce connecté.

Ensuite « l’utilisateur » qui est connecté, peut en plus de navigué sur le site paramétrer les notifications ainsi que d’ajoute des personnages en favoris, pour avoir un accès plus rapide ainsi qu’une notification si celui-ci subit un changement.

Pour finir « l’administrateur », un super utilisateur ayant en plus des droits des deux utilisateurs précédents, accéder aux modifications du site et des comptes utilisateurs.

Ce diagramme a été une source de réflexion, et n’est certainement pas définitif, le site évoluant au fil du temps, le besoin d’ajout de fonctionnalités ainsi que d’utilisateurs sera potentiellement nécessaire.



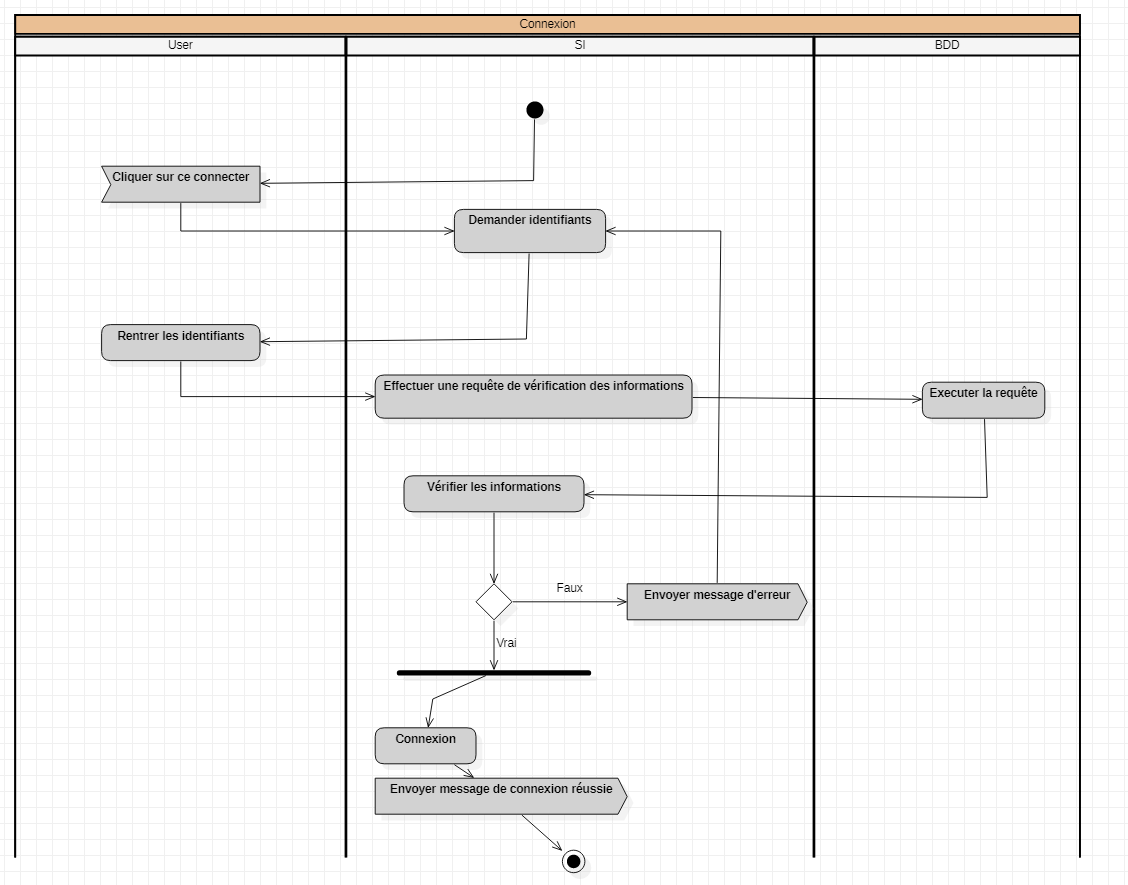
*Use Case*

## Diagrammes d’activités

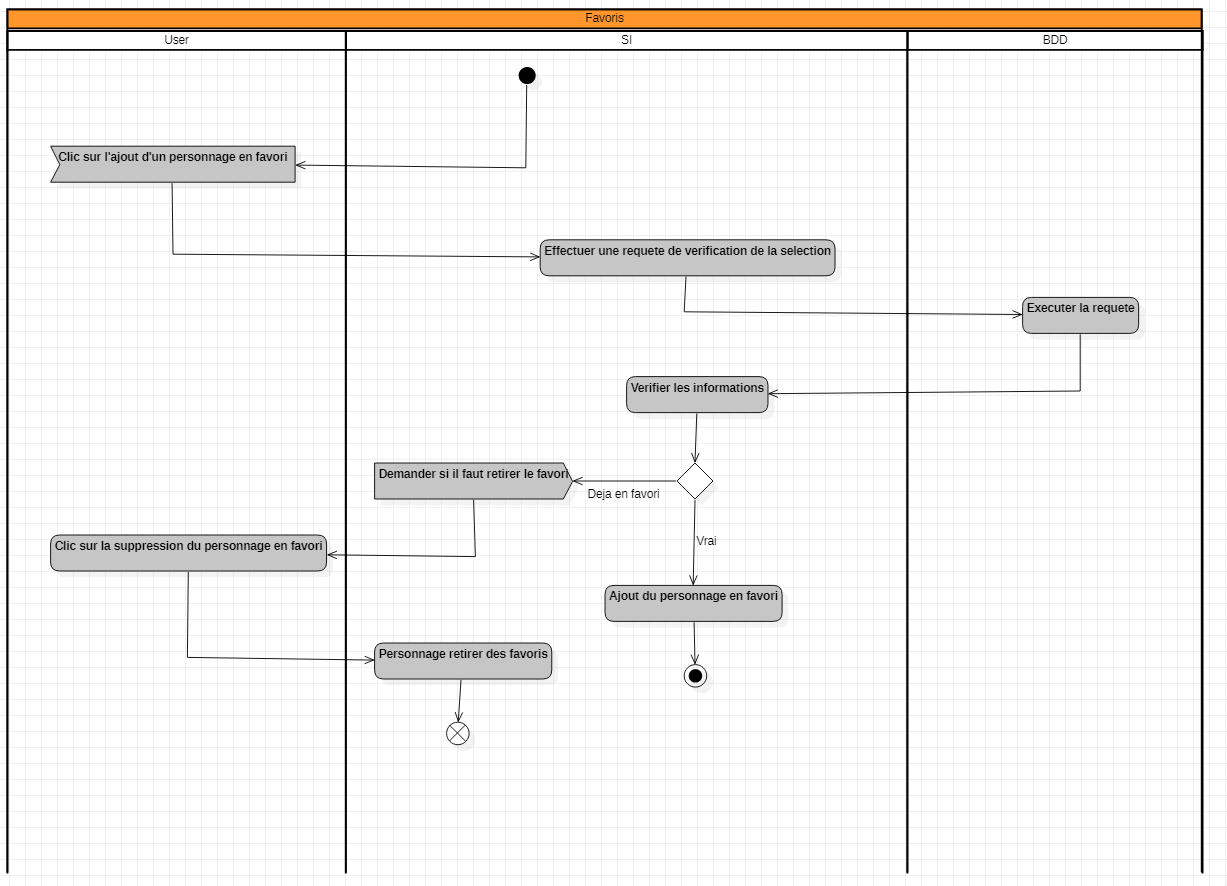
Les diagrammes d’activités sont présents pour décrire le comportement d’une fonctionnalité, comment réagit le système quand une action se passe.

Ci-dessous trois diagrammes d’activités présentant les fonctionnalités suivantes :

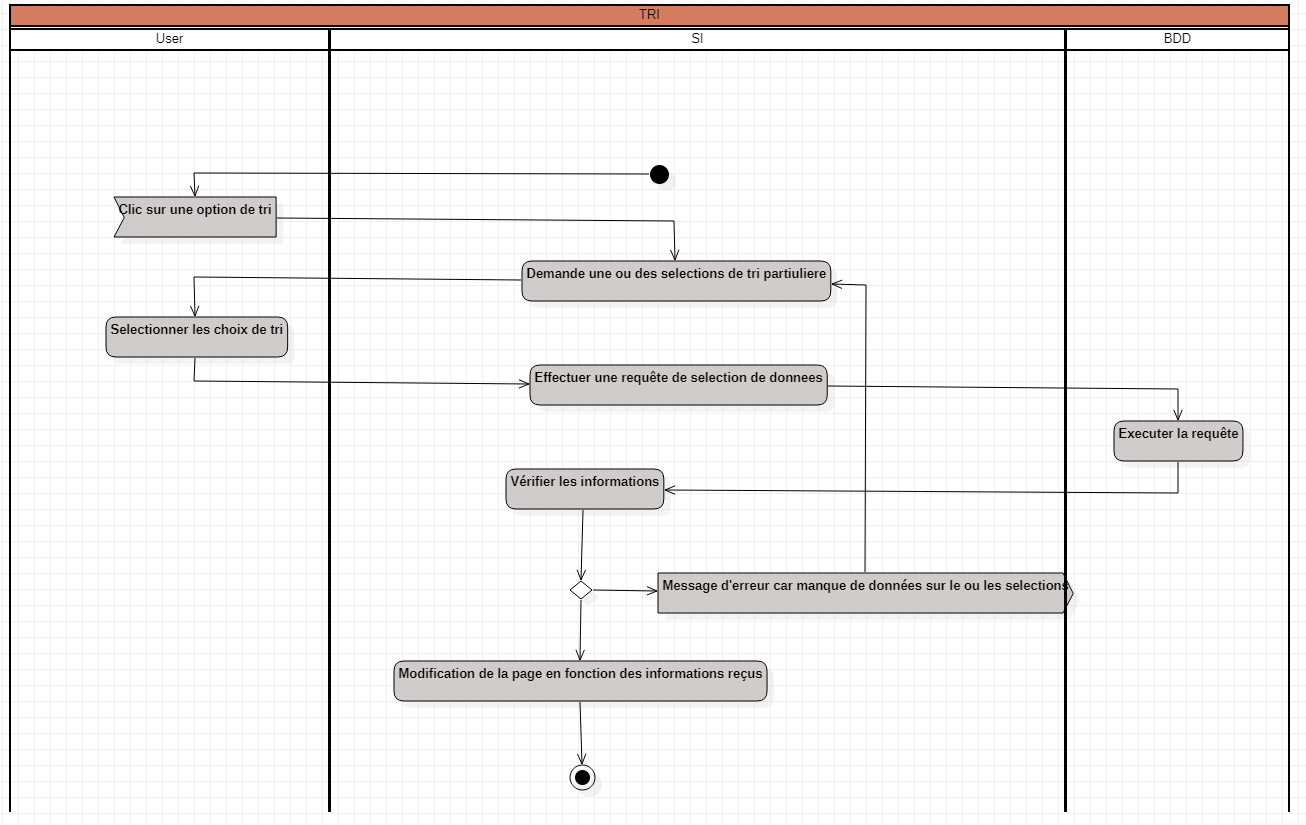
**Connexion**

****

**Ajouts favoris**

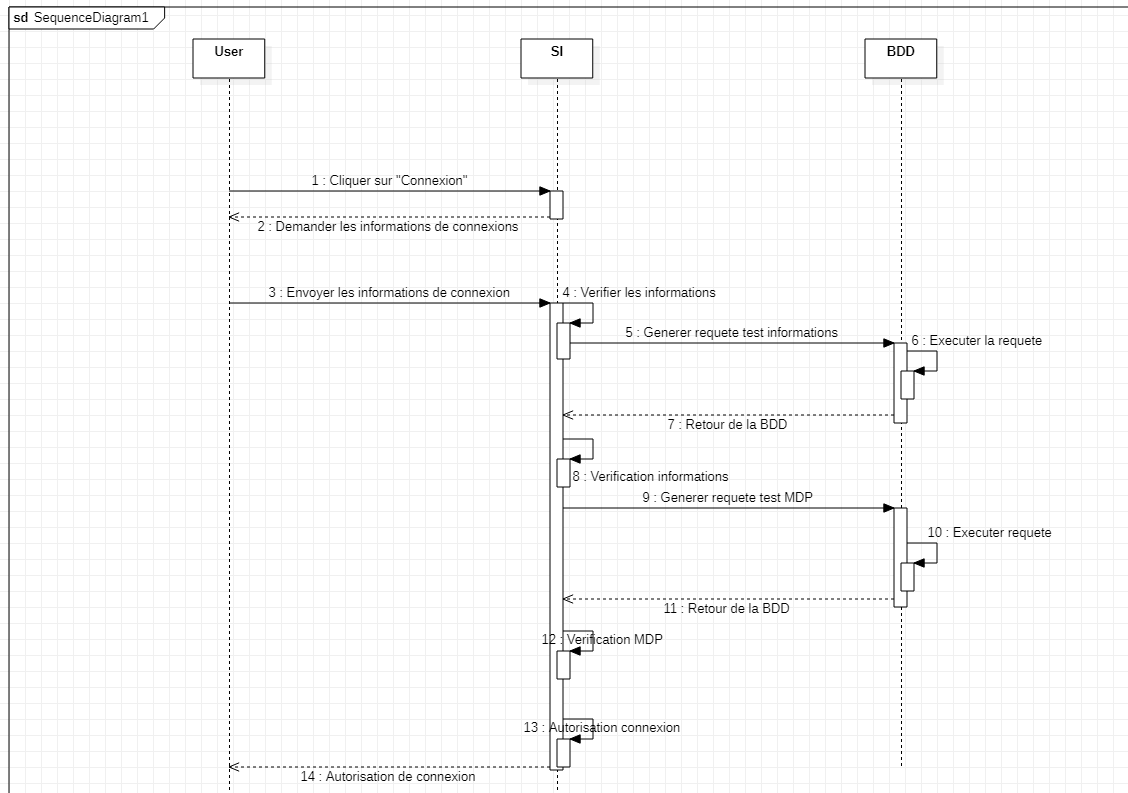
****

**Tri**

****

## Diagramme de séquence

Le diagramme de séquence sert à détailler les interactions entre les utilisateurs et le système, il se lit de haut en base, avec des échanges d’information entre ici l’utilisateur, le système d’information et la base de données



*Diagramme de séquence pour la connexion d’un utilisateur*

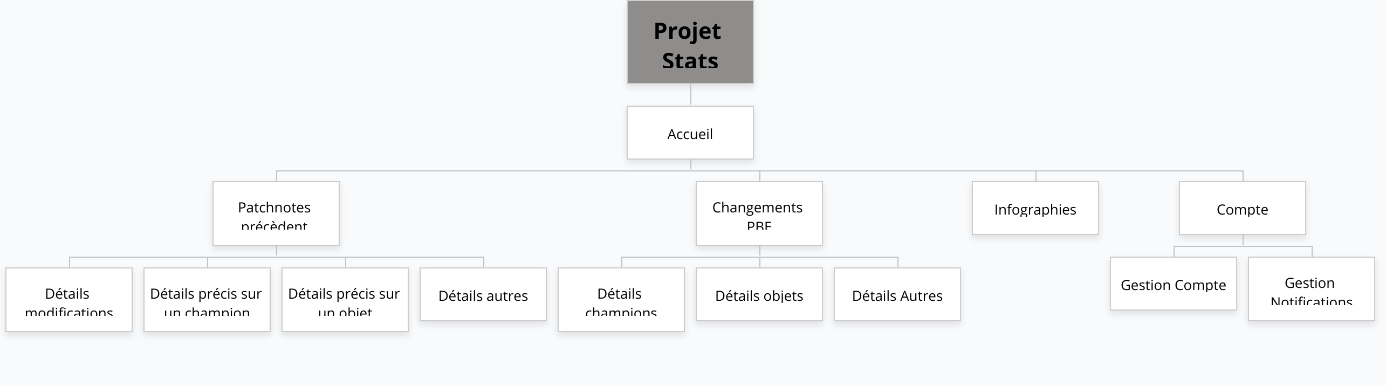
## Diagramme de classe

Le diagramme de classes est présent pour décrire la structure d’un système, modélisant les classes, attributs, opérations ainsi que la relation entre les différents objets.

## Maquettage

### Arborescence

Dans un premier temps, l’arborescence est présente pour représenter la liaison entre les différentes pages du site.



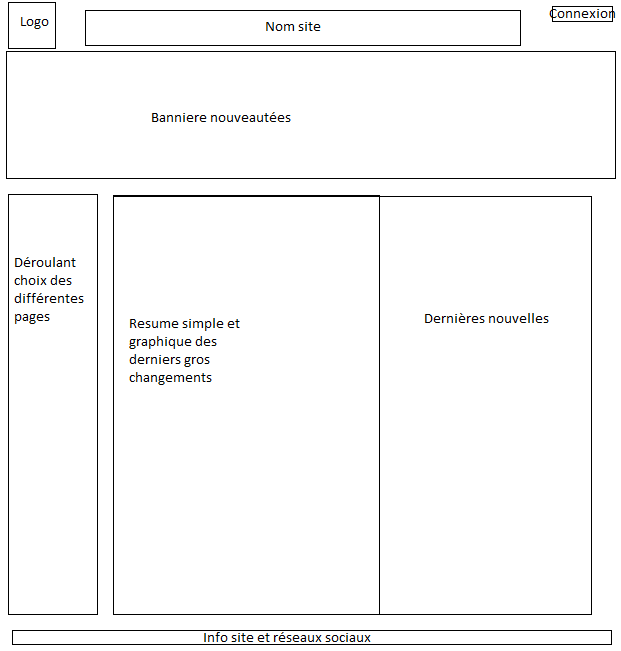
### Zoning

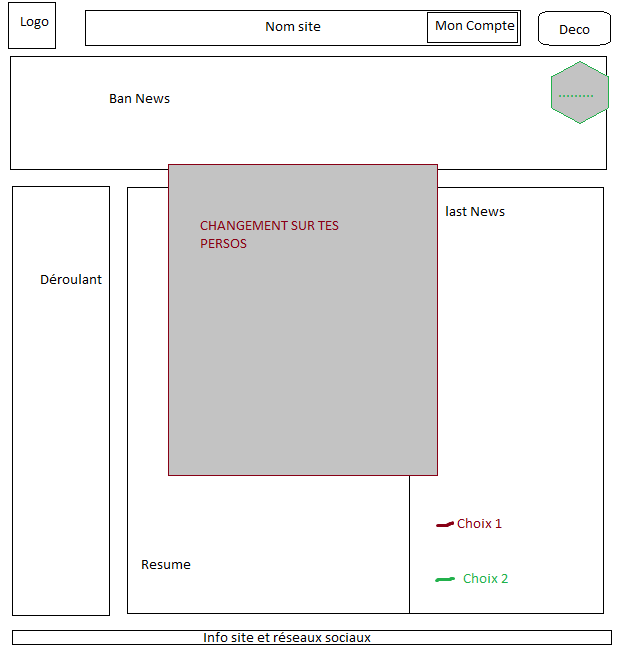
Le zoning est une étape extrêmement importante dans la conception du site, c’est le premier aperçu du rendu du site de façon simplifié pour connaître l’organisation des différentes pages.

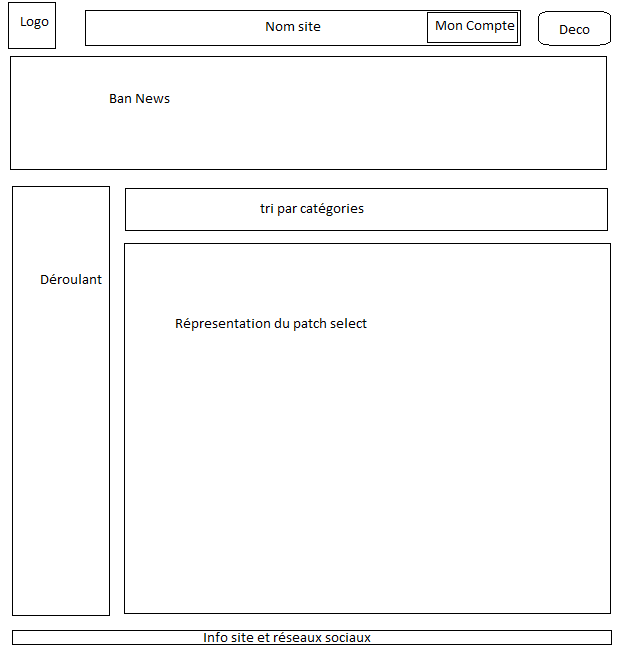
Les pages présentes ci-dessous représentent la page d’accueil sous deux vues, la première en étant déconnecté, contenant le logo, le nom du site ainsi qu’un bouton de connexion en haut de la page, une bannière en dessous avec les dernières nouvelles, un déroulant contenant les différentes pages disponibles et un résumé du dernier patch et nouvelles plus détailler que présente sur la bannière (principalement les futures modifications connues)

La page connectée est similaire, avec en plus le popup contenant des annonces et les modifications concernant les personnages ajoutés en favoris.

La page suivante concerne l’affichage des différents patchnotes, contenant au-dessus des informations concernant le patch une option de tri, permettant d’affine la recherche d’informations.

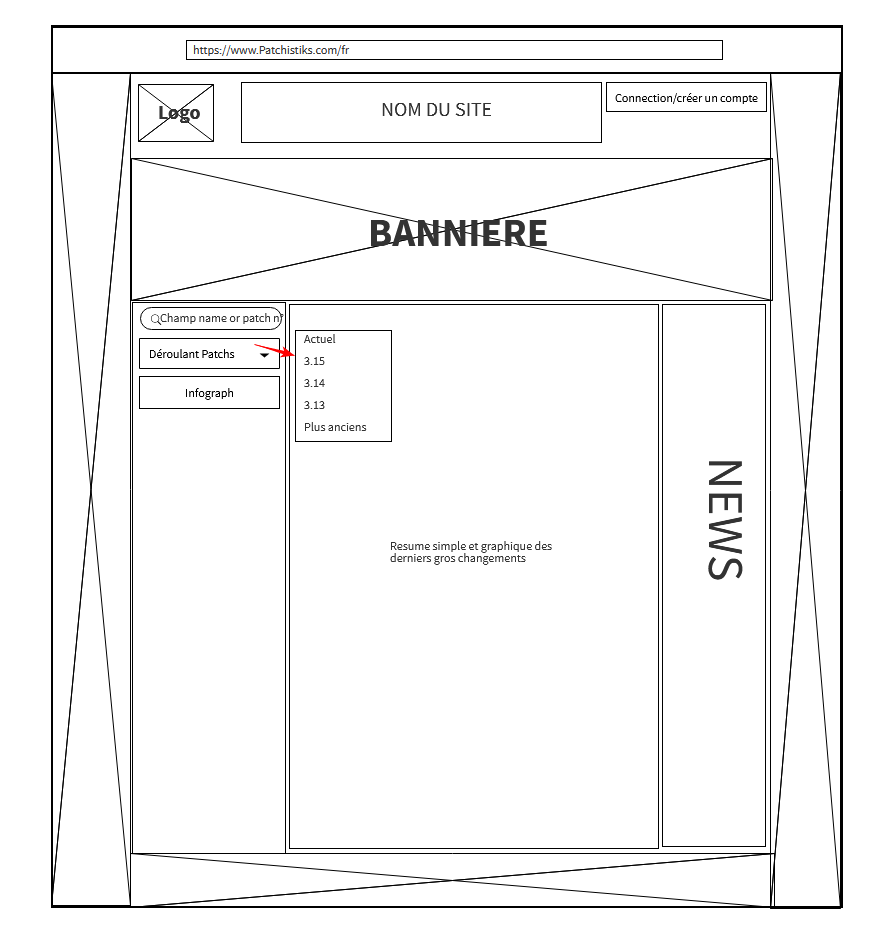


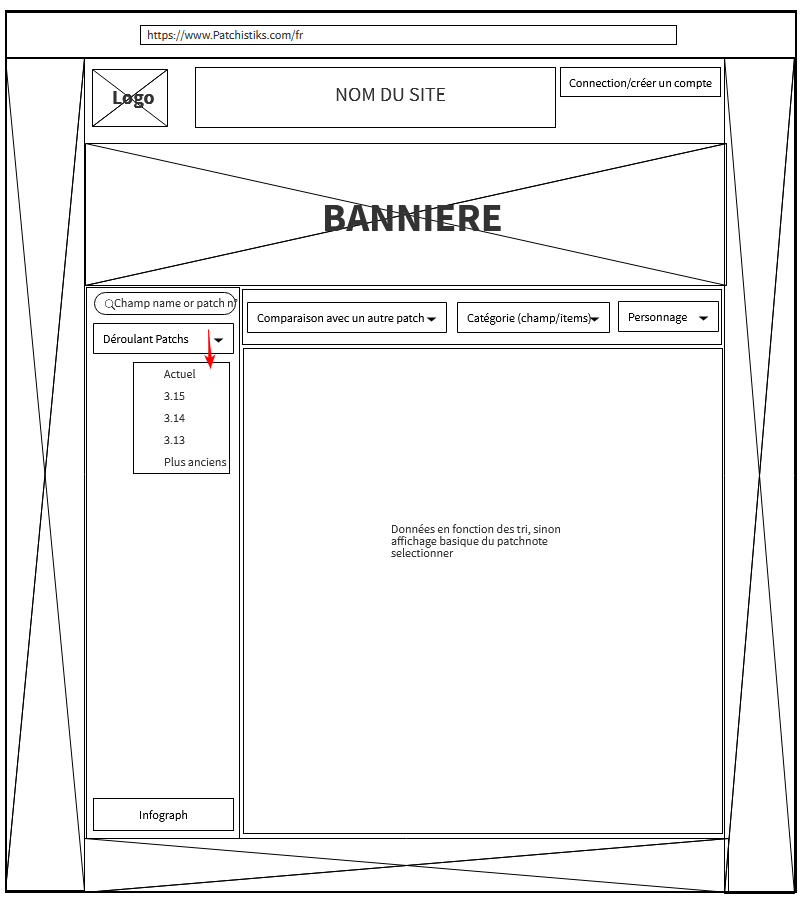




## Wireframe

Le wireframe est zoning avec plus de détail, en restant simple et sans détail.





# Conception

## Modèle Conceptuel de Données

Le Modèle Conceptuel de Données (MCD) est une partie primordiale dans la conception d’un projet, il s’agit de toute la structure de la base de données représenté par des tables liées les unes aux autres.

Le MCD représentant ma base de données comporte dix tables différentes :

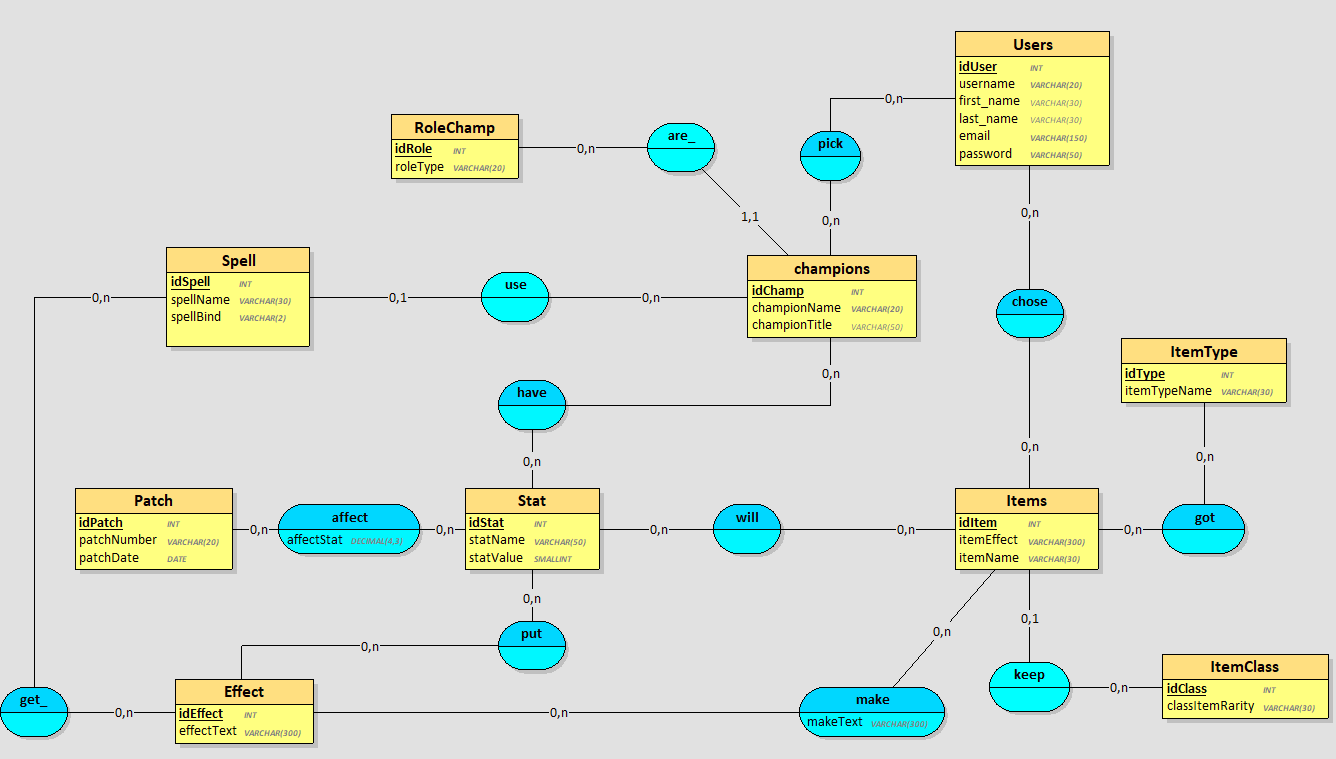
* Users
* Champions
* RoleChampions
* Spell
* Effect
* Stats
* Items
* ItemType
* ItemClass

Les tables dans un premier temps contiennent différents attributs, comme par exemple un id, élément souligné, qui est « l’identité » de la table. On le nomme « Primary Key », que l’on peut voir comme une ligne dans un fichier excel, c’est une valeur qui s’auto-incrémente pour chaque ajout d’une nouvelle donnée dans la table.

Les tables sont mises en relation, et possèdent des cardinalités présentes pour identifier quel type d’association sont présentes entre elles.

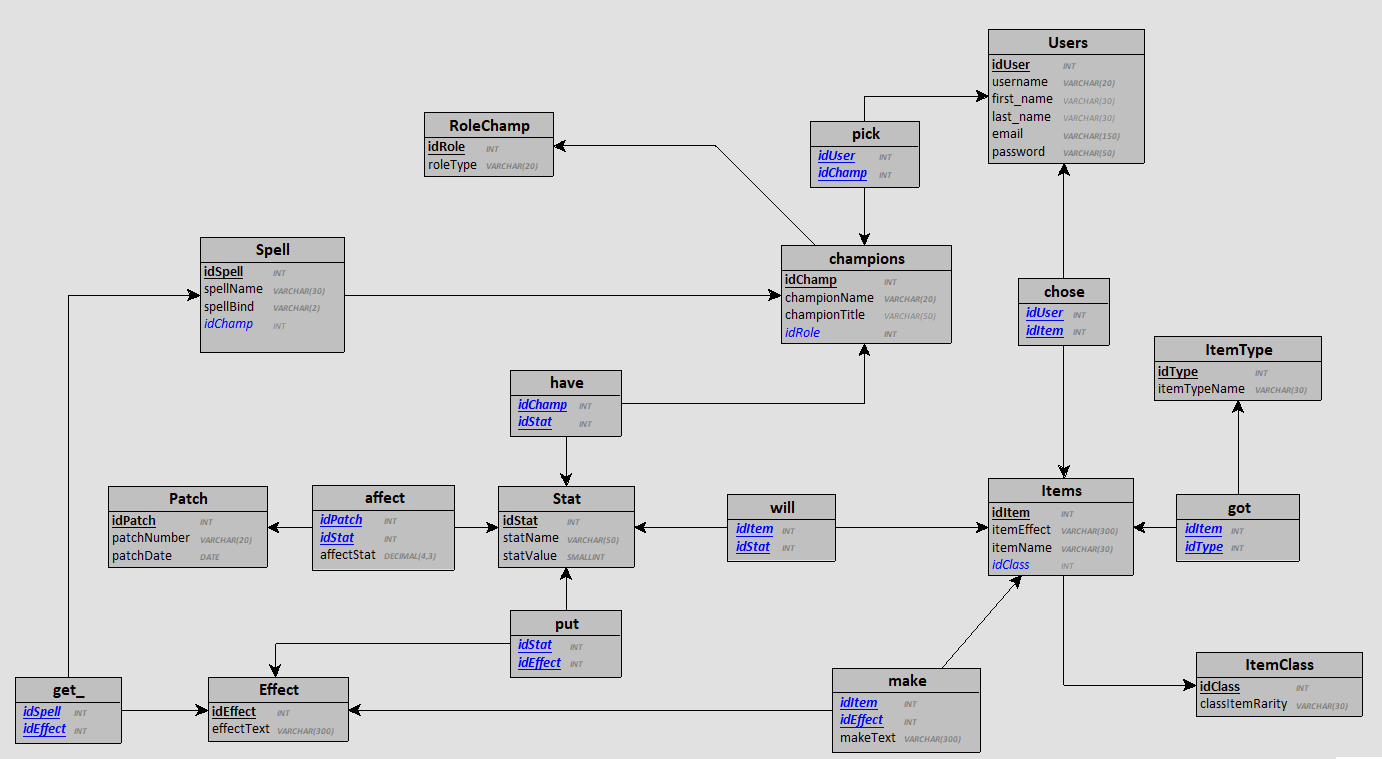
Par exemple entre « Champions » et « Spell », on peut voir 0, N indiquant qu’un champion peut avoir 0 à une infinité de capacités, tandis que 0,1 ne peut être attribué qu’à un seul champion au maximum.

Ce MCD est la version actuelle de comment la base de données de mon projet doit être.



## Modèle Logique de Données

Le Modèle Logique de Données (MLD) est une version du MCD permettant de représenter les données d’un système d’information.



# Outils Techniques Utilisés

|  |  |
| --- | --- |
|  | **VSCODE**  IDE gratuit  et open source développer par Microsoft |
|  | **STAR UML**  Logiciels permettant  de schématiser  des diagrammes UML |
|  | **GITHUB** site web de service de versionnage de code |
|  | **DJANGO** Framework Python  similaire au modèle MVT (Models/ View/ Template) |

# Fonctionnalités

# Inscription, connexion et déconnexion