**[Nom de la société]**

TPI - 2017

**OverwatchCollection : La documentation technique**

**Sven Wikberg**

Contenu

[OverwatchCollection : La documentation technique 1](#_Toc485027435)

[Introduction 1](#_Toc485027436)

[Étude d’opportunité 3](#_Toc485027437)

[Pourquoi ce sujet ? 3](#_Toc485027438)

[Ce que ce projet à de plus ! 3](#_Toc485027439)

[Conception 3](#_Toc485027440)

[Analyse de l’environnement 3](#_Toc485027441)

[Détermination de l’arborescence du site et des rubriques 4](#_Toc485027442)

[Maquette Graphique 4](#_Toc485027443)

[Modèle relationnel 9](#_Toc485027444)

[Description des tables 9](#_Toc485027445)

# OverwatchCollection : La documentation technique

# Analyse préliminaire

## Introduction

Le projet OverwatchCollection est basé sur le jeu Overwatch, Overwatch est un jeu de tir, très réactif, à la première personne, aux graphismes colorés et un peu cartoonesques.

Dans Overwatch lorsqu’on commence une partie, il faut d’abord choisir un héros parmi les 23 disponibles, les héros on des capacités et des statistiques différentes. Ils sont regroupé par rôles, les rôles définissent à quoi le héros sert dans l’équipe. Par exemple les « tanks » sont là pour faire en sorte de prendre les dégâts à la place des autres héros, car ils sont plus résistants.

Donc dans ce jeu on peut recevoir des objets cosmétiques à diverses occasions, ces objets permettent de customiser les héros. Par exemple il est possible de recevoir un skin, qui permet d’avoir un modèle diffèrent de celui de base pour son héros (voir figure 1 et 2) ou alors on peut recevoir une phrase spéciale que le héros pourra dire pendant le jeu.

|  |  |
| --- | --- |
| Figure 1, skin de base | Figure 2, skin d'halloween |

## Objectif

Le but du projet OverwatchCollection est d’avoir une sorte d’inventaire de ses objets, car dans le jeu, la façon dont les objets sont ordonnés n’est pas pratique. Il permet donc de voir tous les objets triés de plusieurs façons différentes et de sélectionner ceux que l’on a obtenus dans le jeu afin de les reconnaitre plus facilement ainsi que d’avoir les statistiques sur ceux-ci.

# Analyse

## Pourquoi ce sujet ?

Ce sujet est vaut la peine de se pencher dessus car Overwatch est un jeu très populaire et donc il y a une très grande communauté de joueur autour et les joueurs sont en générale très friands d’objets rares et exclusif afin de les montrer aux autres joueurs. C’est donc normal que les objets cosmétiques dans Overwatch aient beaucoup de succès, surtout avec les évènements spéciaux et la sortie de contenue régulière de la part des développeurs.

Mais le problème c’est qu’il y a vraiment beaucoup d’objets différents et dans le jeu, il n’y a pas de façon simple de voir ou on en est avec tous ces objets, c’est là qu’Overwatch Collection entre en jeu. Il va pouvoir régler ce problème afin de rendre la vie des joueurs plus facile.

## Ce que ce projet à de plus !

Pour l’instant il n’y a pas vraiment de façon de faire un inventaire des objets existants ainsi que des objets obtenu dans le jeu, à part avec un papier et un crayon mais au niveau pratique ce n’est pas le moyen le plus avancé. C’est exactement l’utilité de ce projet, il permet de fait un inventaire de ses objets parmi tous les objets existants dans le jeu.

Il existe déjà des sites qui répertorie certaines informations du jeu, comme par exemple « <http://overwatch.gamepedia.com> » qui contient en autre une liste des héros avec pleins d’informations intéressantes sur chacun d’entre eux ou un liste des skins des héros, mais pas qui permettent d’avoir une liste complète des objets triés, un suivis, des statistiques détaillées par évènement ou par héros sur ceux-ci, comme le propose ce projet.

## Cahier des charges détaillé

### Définition de l’audience

OverwatchCollection est un projet qui a évidemment comme cibles les joueurs du jeu Overwatch, et plus particulièrement les joueurs collectionneurs, c’est-à-dire les joueurs aiment bien avoir un inventaire de ce qu’ils possèdent et surtout ce qu’ils n’ont pas encore.

### Définition du contenu et de fonctionnalités

Les fonctionnalités sont regroupées en quatre parties, premièrement le groupe de fonctionnalité accessible par les trois prochaines parties. Deuxièmement la partie « invité », c’est la partie qui contient les fonctionnalités accessibles en tant qu’invité. Ensuite la partie « utilisateur », c’est la partie contient les fonctionnalités accessibles lorsqu’on a créé un compte et que l’on est connecter sur ce compte. Et finalement la partie « administrateur » qui regroupe les fonctionnalités accessible en tant qu’administrateur.

#### Toutes catégories

* affichage des utilisateurs du site
* affichage des objets cosmétiques d’Overwatch par héro
* affichage des objets cosmétiques d’Overwatch par événement
* affichage des objets cosmétiques d’Overwatch qui ne sont lié à aucun héro
* affichage des informations de base des héros
* affichage des informations de base des événements

#### Invité

* création un compte
* identification

#### Utilisateur

* possibilité cliquer sur n’importe quel objet cosmétique non sélectionné afin de le sélectionner
* possibilité cliquer sur n’importe quel objet cosmétique sélectionné afin de le désélectionner
* modification des informations des son compte
* affichage des objets sélectionné différent afin de les reconnaitre
* affichage de statistiques sur les objets sélectionnés

#### Administrateur

* bannissement d’un utilisateur
* dé- bannissement d’un utilisateur
* suppression d’un compte
* affichage des utilisateurs bannis
* affichage des utilisateurs non bannis

### Maquette préliminaire

# Conception

## Analyse de l’environnement

Ce projet a principalement été réalisé avec le logiciel « Visual Studio Code », en tout cas pour tout le html, le php ainsi que le css. Ce logiciel a été utilisé principalement pour des raisons d’habitude, depuis plus d’un an je fais mes projets web qu’avec ça.

Au niveau du serveur Apache, PHP ainsi que MySQL c’est le logiciel « EasyPHP » qui a été utilisé, pour les mêmes raisons d’habitude, la base de données a elle été gérée avec « phpMyAdmin ».

En ce qui concerne les sauvegardes du projet, c’est « GitHub Desktop » qui a été utilisé, car c’est vraiment facile a utilisé lorsqu’on fait un projet tout seul et il s’occupe de faire les versions de façon très claires et pratiques.

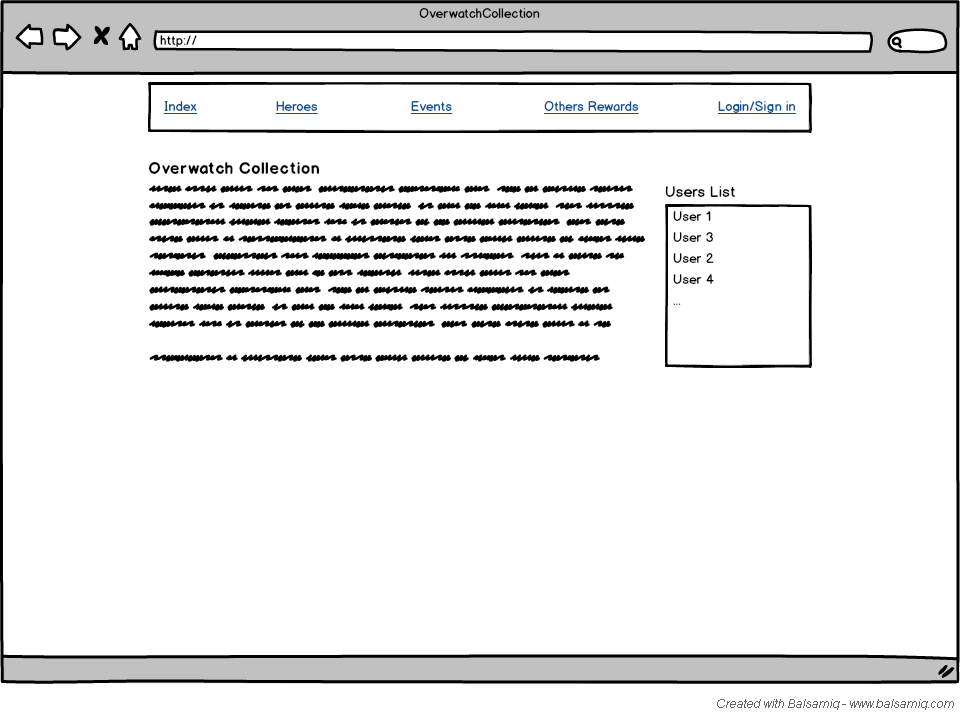
Pour la documentation j’ai utilisé l’application web « draw.io » pour le modèle relationnel ainsi que l’arborescence du site et « balsamiq mockup 3 » pour les maquettes graphiques.

## Détermination de l’arborescence du site et des rubriques

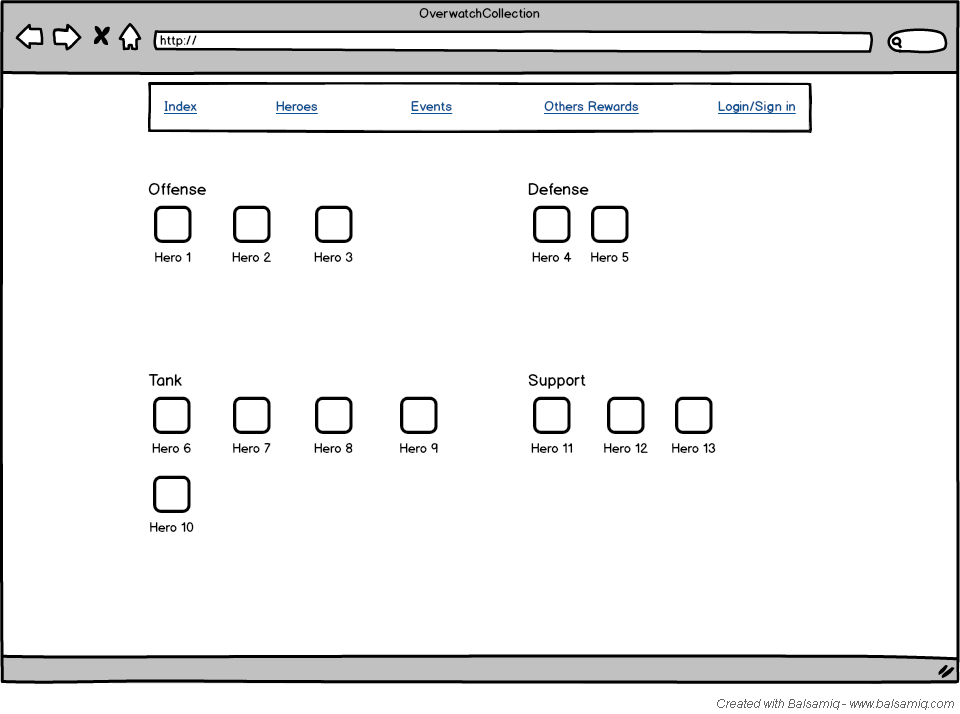


## Maquette Graphique

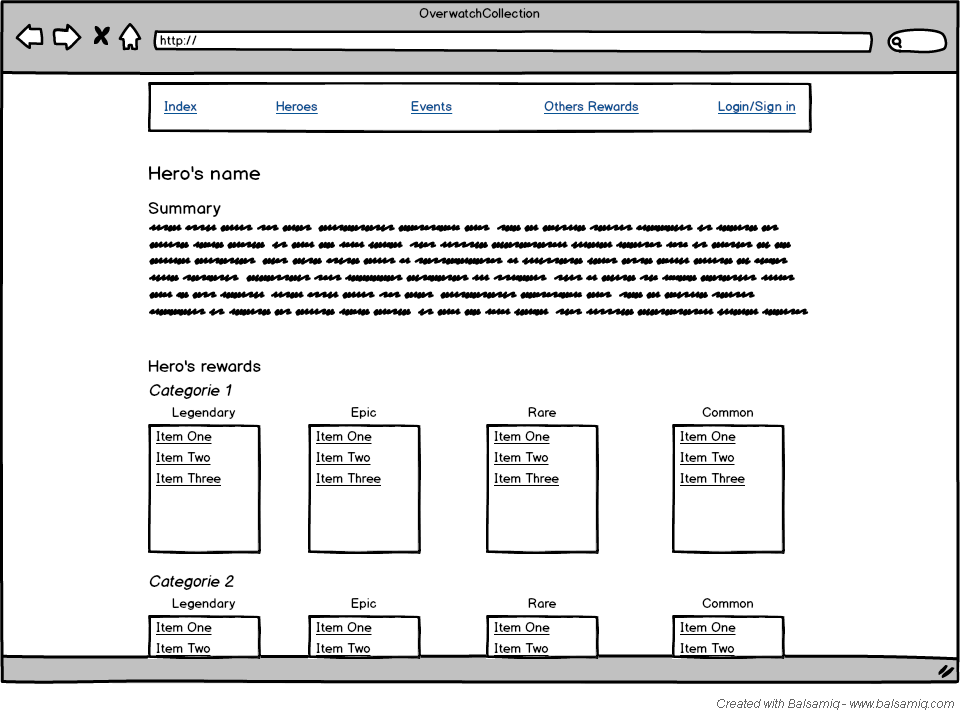
#### Page d’accueil



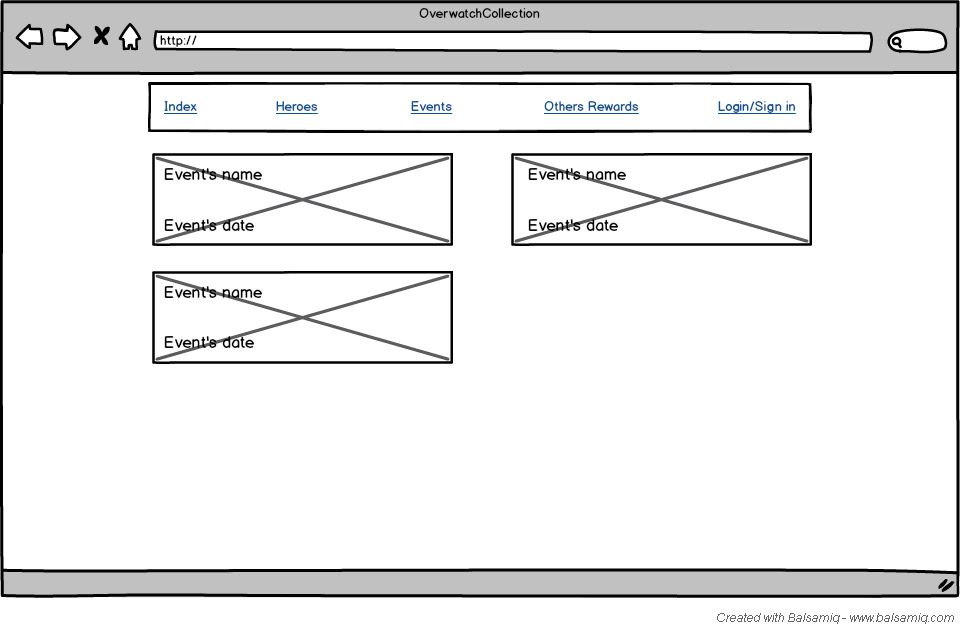
#### Page des héros



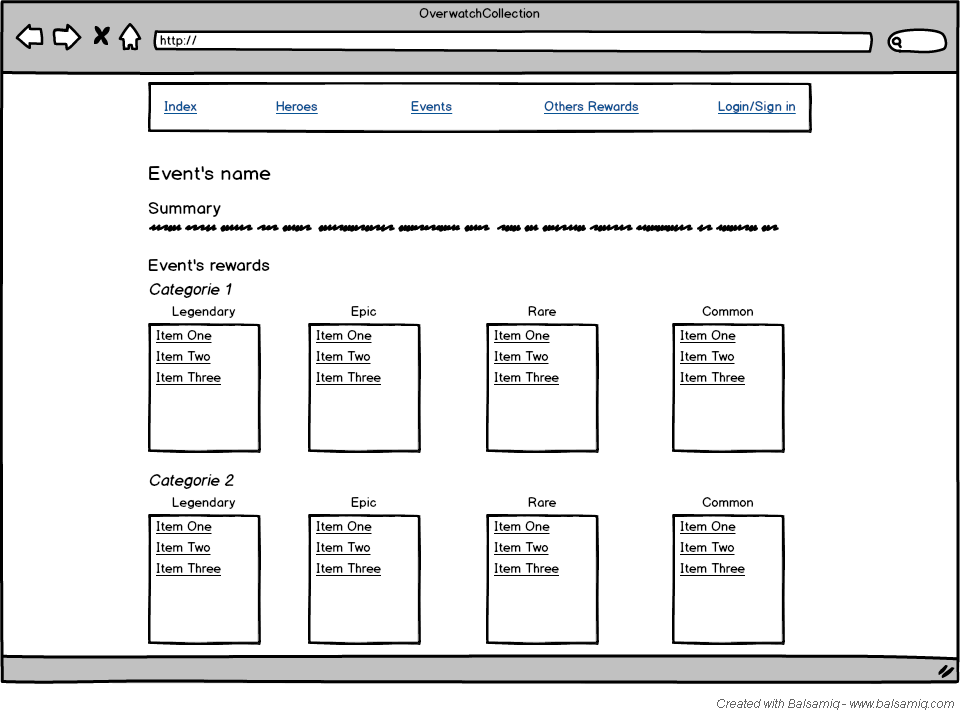
#### Page d’un héros



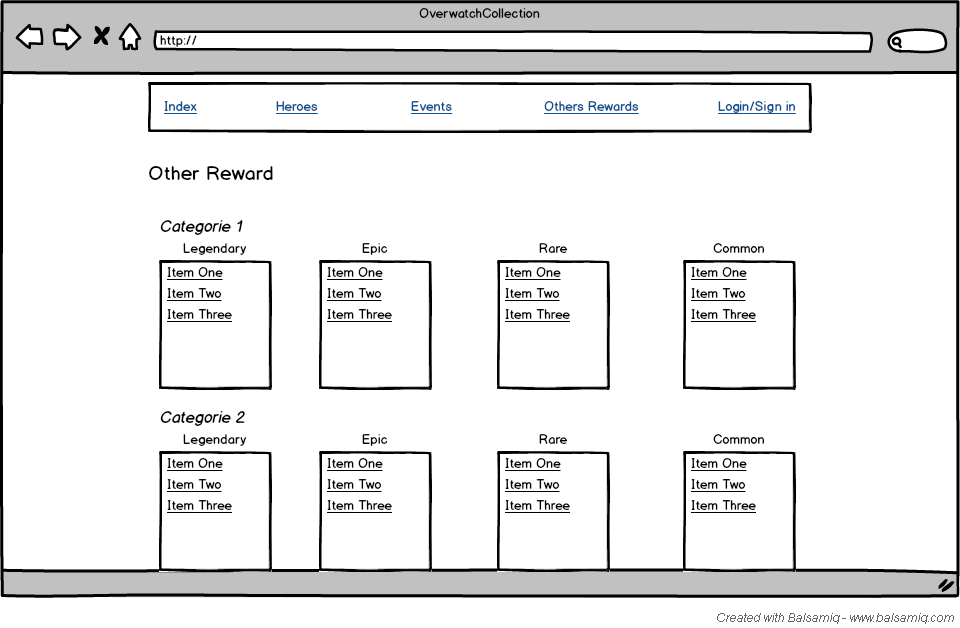
#### Page des évènements



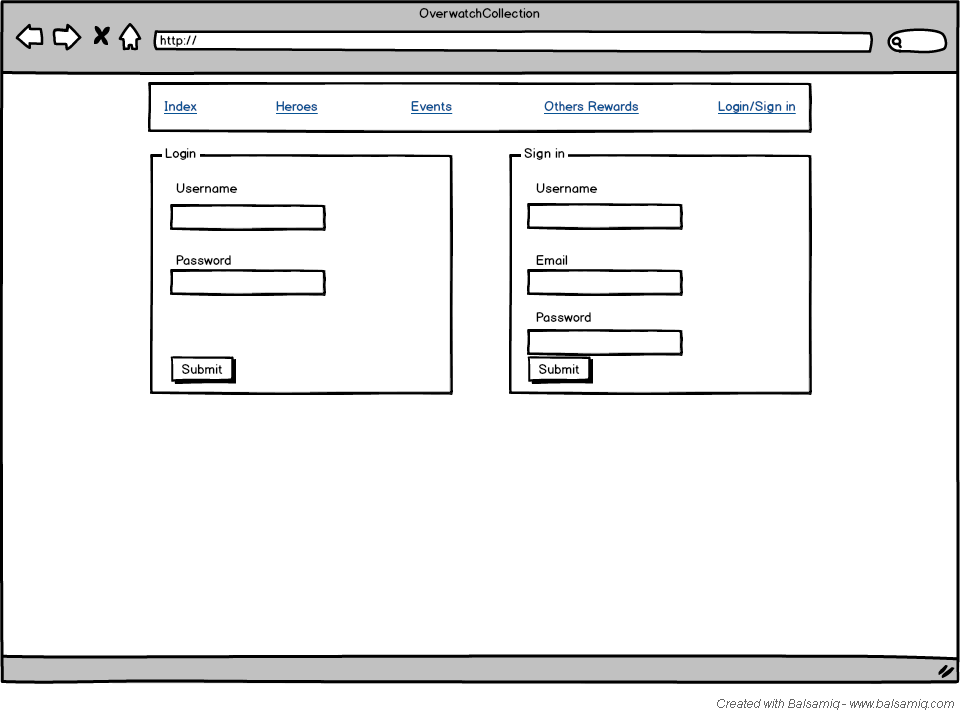
#### Page d’un évènement



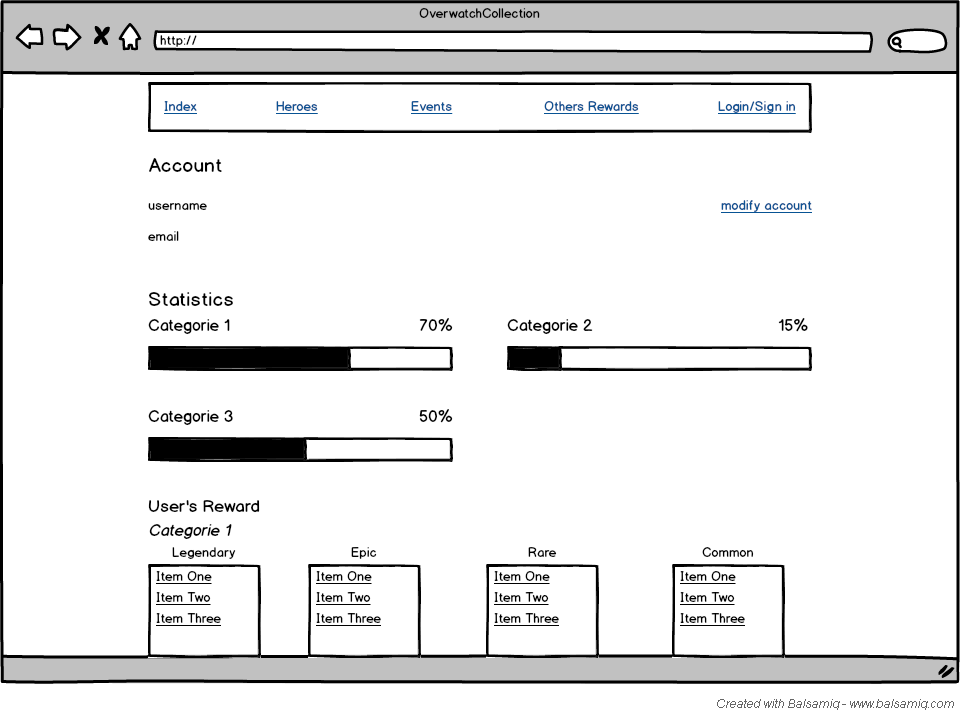
#### Page des autres objets



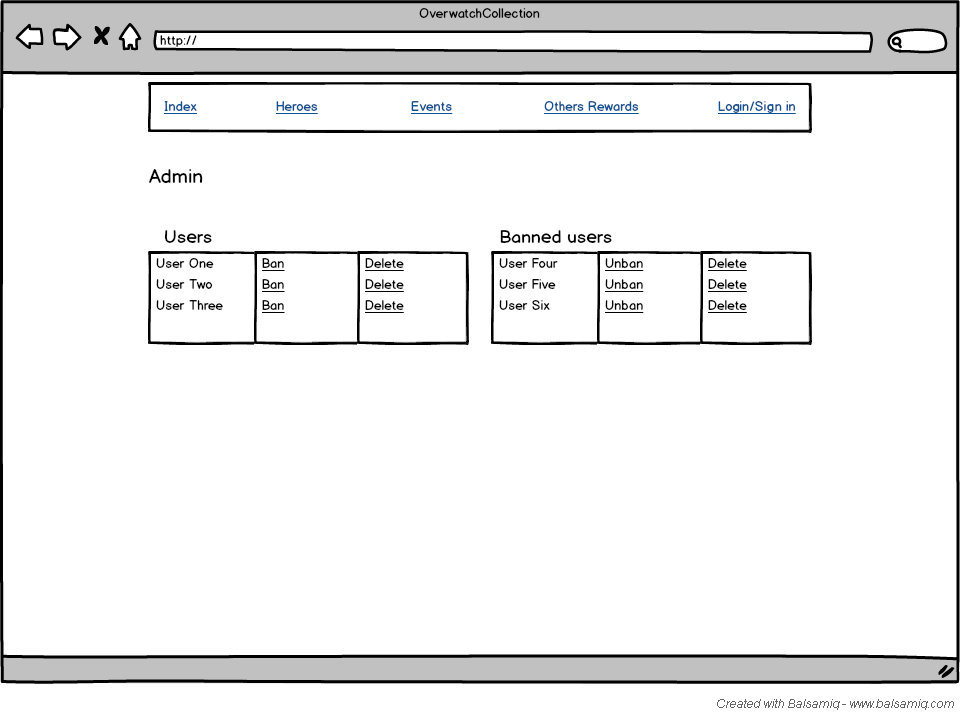
#### Page de connexion/inscription



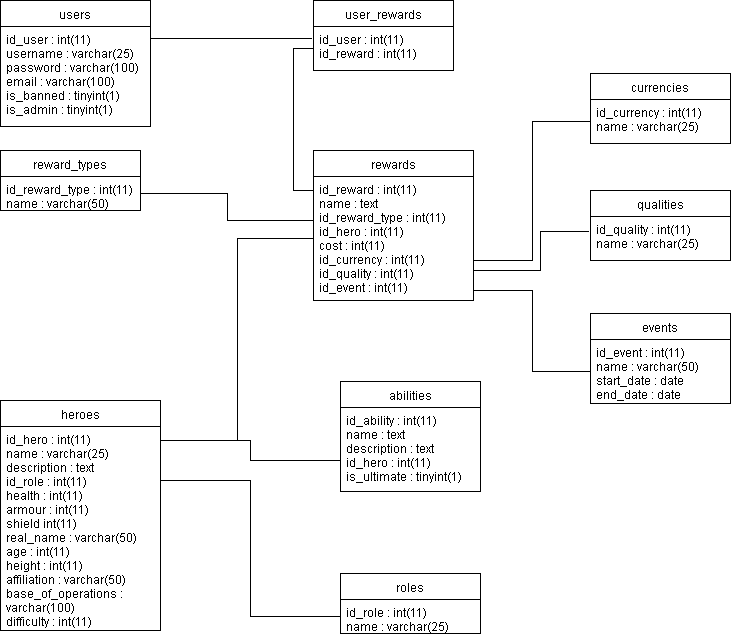
#### Page du compte



#### Page administrateur



### Modèle relationnel



### Description des tables

#### Table users

|  |  |  |
| --- | --- | --- |
| Nom du champ | Type | Description |
| id\_user | int(11), Non null, AI, PK | Identifiant de l’utilisateur |
| username | varchar(25), Non null | Nom de l’utilisateur |
| password | varchar(100), Non null | Mot de passe de l’utilisateur |
| email | varchar(100), Non null | Email de l’utilisateur |
| is\_banned | tinyint(1), Non null | Définit si l’utilisateur est banni ou pas |
| is\_admin | tinyint(1), Non null | Définit si l’utilisateur est admin ou pas |

#### Table reward\_types

|  |  |  |
| --- | --- | --- |
| Nom du champ | Type | Description |
| id\_reward\_type | int(11), Non null, AI, PK | Identifiant de la catégorie d’objet cosmétique |
| name | varchar(50), Non null | Nom de la catégorie de cosmétique |

#### Table heroes

|  |  |  |
| --- | --- | --- |
| Nom du champ | Type | Description |
| id\_hero | int(11), Non null, AI, PK | Identifiant du héros |
| name | varchar(25), Non null | Nom/Pseudo du héros |
| description | text, Non null | Description du héros |
| id\_role | int(11), Non null, FK | Identifiant du rôle du héros |
| health | int(11), Non null | Points de vie du héros |
| armour | int(11), Non null | Points d’armure du héros |
| shield | int(11), Non null | Points de bouclier du héros |
| real\_name | varchar(50), Non null | Vrai nom (et prénom) du héros |
| age | int(11), Non null | Age du héros |
| height | int(11), Null | Taille, en cm, du héros |
| affiliation | varchar(50), Null | Organisation du héros |
| base\_of\_operations | varchar(100), Null | Base d’opération/ Lieu d’habitation du héros |
| difficulty | int(11), Non null | Difficulté du héros, en jeu |

#### Table user\_rewards

|  |  |  |
| --- | --- | --- |
| Nom du champ | Type | Description |
| id\_user | int(11), Non null, FK |  |
| id\_reward | int(11), Non null, FK |  |

#### Table rewards

|  |  |  |
| --- | --- | --- |
| Nom du champ | Type | Description |
| id\_reward | int(11), Non null, AI, PK | Identifiant de l’objet cosmétique |
| name | text, Non null | Nom de l’objet cosmétique |
| id\_reward\_type | int(11), Non null, FK | Identifiant de la catégorie de l’objet cosmétique |
| id\_hero | int(11), Null, FK | Identifiant du héros de l’objet cométique |
| cost | int(11), Null | Cout de l’objet cosmétique |
| id\_currency | int(11), Null, FK | Identifiant de la monnaie de l’objet cosmétique |
| id\_quality | int(11), Non null, FK | Identifiant de la qualité de l’objet cosmétique |
| id\_event | int(11), Null, FK | Identifiant de l’événement de l’objet cosmétique |

#### Table abilities

|  |  |  |
| --- | --- | --- |
| Nom du champ | Type | Description |
| id\_abilities | int(11), Non null, AI, PK | Identifiant de la capacité |
| name | text, Non null | Nom de la capacité |
| description | text, Non null | Description de la capacité |
| id\_hero | int(11), Non null, FK | Identifiant du héros à qui appartient la capacité |
| is\_ultimate | tinyint(1), Non null | Définit si la capacité est une capacité ultime ou pas |

#### Table roles

|  |  |  |
| --- | --- | --- |
| Nom du champ | Type | Description |
| id\_role | int(11), Non null, AI, PK | Identifiant du rôle |
| name | varchar(25), Non null | Nom du rôle |

#### Table currencies

|  |  |  |
| --- | --- | --- |
| Nom du champ | Type | Description |
| id\_currency | int(11), Non null, AI, PK | Identifiant de la monnaie |
| name | varchar(25), Non null | Nom de la monnaie |

#### Table qualities

|  |  |  |
| --- | --- | --- |
| Nom du champ | Type | Description |
| id\_quality | int(11), Non null, AI, PK | Identifiant de la rareté/qualité |
| name | varchar(25), Non null | Nom de la rareté/qualité |

#### Table events

|  |  |  |
| --- | --- | --- |
| Nom du champ | Type | Description |
| id\_event | int(11), Non null, AI, PK | Identifiant de l’événement |
| name | varchar(50), Non null | Nom de l’événement |
| start\_date | date, Non null | Date de début de l’événement |
| end\_date | date, Non null | Date de fin de l’événement |