

IMIE ECOLE DE LA FILIÈRE NUMÉRIQUE

IT START 2016-2017

DEVELOPPEMENT LOGICIEL

Projet LeagueStats

CAHIER DES CHARGES

Thomas MALASSIGNE

28 mars 2017





Table des matières

| | | |
|----------|-----------------------------------|-----------|
| 1 | Introduction | 2 |
| 2 | Présentation du projet | 3 |
| 2.1 | Descriptif | 3 |
| 2.2 | Architecture | 4 |
| 2.3 | Fonctionnalités | 5 |
| 2.3.1 | Application Java | 5 |
| 2.3.1.1 | Mockup | 5 |
| 2.3.2 | Version web | 6 |
| 2.3.2.1 | Mockup - Figure 1 | 6 |
| 2.3.2.2 | Mockup - Figure 2 | 7 |
| 2.3.2.3 | Mockup - Figure 2 | 7 |
| 2.3.3 | Capture d'écran | 8 |
| 2.3.3.1 | Screenshoot - Figure 1 | 8 |
| 2.3.3.2 | Screenshoot - Figure 2 | 8 |
| 3 | Outils | 9 |
| 3.1 | Web | 9 |
| 3.2 | API | 9 |
| 3.3 | Application Mobile | 9 |
| 3.4 | Logiciels | 9 |
| 4 | Conclusion | 10 |
| 5 | Annexes | 11 |
| 5.1 | Définition et acronymes | 11 |

1 Introduction

Dans le cadre du projet de fin d'année à réaliser à l'issue de de la formation IT START de l'IMIE Caen, j'ai décidé de réaliser une application mobile et un site web permettant de recenser et d'afficher les informations liés à un jeu vidéo. Le contenu de l'application sera adressé aux joueurs du célèbre jeu vidéo *League of Legends*. Étant moi-même joueur je serais également client de mon application. Cela me permettra de répondre plus facilement aux attentes des utilisateurs en terme de fonctionnalités et de contenu. Pour la réalisation de ce projet, je vais être amené à utiliser différentes applications qui m'étaient encore inconnues avant mon entrée à l'IMIE tel que le JavaScript et l'utilisation d'une API. Concevoir une application avec des langages où je n'ai encore que des connaissances très limités est pour moi un réel défi.

2 Présentation du projet

2.1 Descriptif

Le projet **LEAGUE STATS** a d'abord vu le jour en tant qu'ébauche lors de la conception d'un mini-projet à effectuer en Java. Le but était de se familiariser avec le langage et de s'initier aux technologies annexes nécessaires au développement d'une application. C'est à l'issue de ce premier projet que j'ai décidé de finaliser cette application en développant une version finale disponible également sur navigateur internet. L'objectif de ce projet est d'afficher les informations liés à un champion de l'univers de *League of Legends* dans deux versions distinctes mais liés par une base de données commune. La base de donnée stockera toutes les informations des dits champions. L'application sera développée en Java et l'interface graphique en xml. La version sur navigateur utilisera de l'HTML, du CSS, du PHP et du JavaScript.

2.2 Architecture

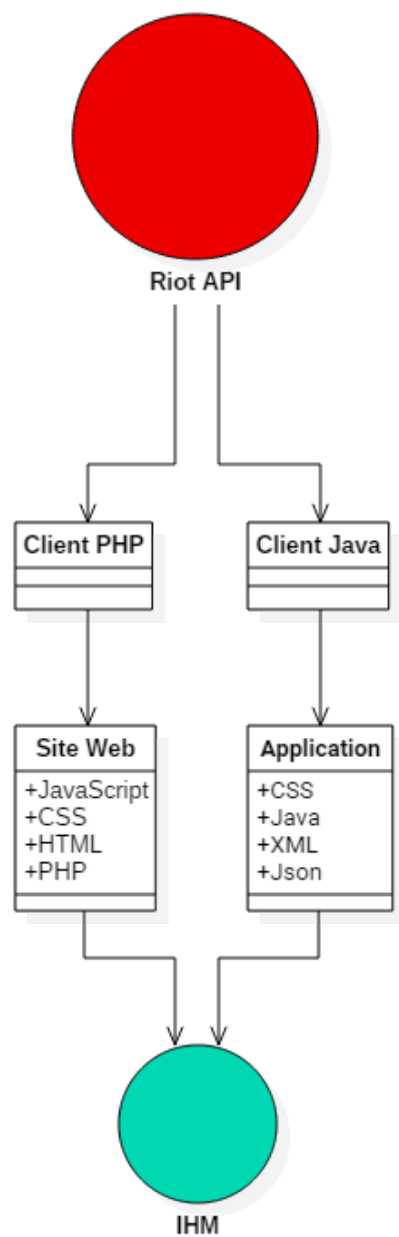


Diagramme représentant le fonctionnement de l'application **LEAGUE STATS**

2.3 Fonctionnalités

L'application et la version sur navigateur auront accès au même contenu via une base de données. Des requêtes SQL communes seront présentes pour interagir avec cette dernière afin d'afficher et de modifier les éléments présents sur la table de la base de données.

2.3.1 Application Java

L'interface graphique de l'application permettra des interactions plus simple avec les utilisateurs. Celle-ci comportera un menu avec des boutons permettant de naviguer dans l'application. L'utilisateur pourra cliquer sur le bouton "Champions" et l'application affichera la liste de tous les champions. Il sera ensuite possible de sélectionner un champion et l'application retournera toutes les informations relatives à celui-ci. Un bouton d'aide sera également présent afin d'aiguiller les utilisateurs novices.

2.3.1.1 Mockup

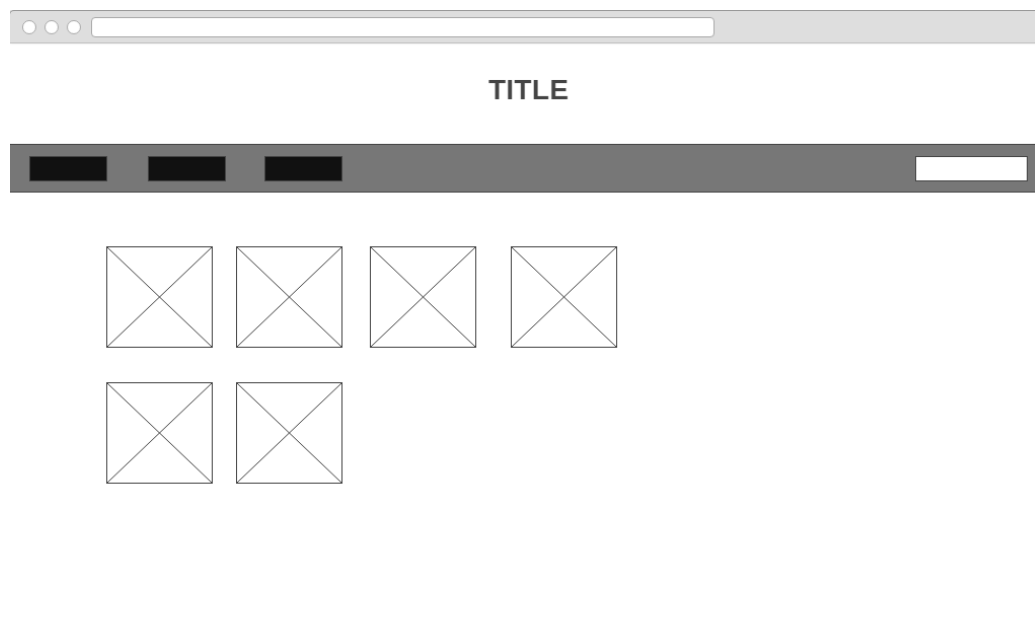


Ébauche de la première interface graphique réalisé en JavaFX

2.3.2 Version web

La version web a pour objectif la simplicité d'utilisation et le respect du style graphique. L'index sera consacrée à la description du projet et à l'explication des différentes fonctionnalités. L'internaute pourra ensuite utiliser les boutons présents sur le site requêtes afin de visualiser du contenu présents sur la base de données. Une partie du site sera réservée aux membres, préalablement connectés avec leur comptes administrateurs, afin que l'ajout, la suppression et la modification du contenu ne soit possible que par eux. Une page de téléchargement sera également présente afin de permettre aux internautes de télécharger l'application directement sur leur ordinateur. L'utilisation de JavaScript, pour afficher le contenu de façon dynamique, et, du framework Bootstrap pour rendre la navigation plus agréable seront nécessaires.

2.3.2.1 Mockup - Figure 1



Page web affichant la liste des champions sous forme d'icônes

2.3.2.2 Mockup - Figure 2



Seconde page affichant les informations d'un champion et ses compétences


2.3.2.3 Mockup - Figure 3



Troisième page permettant de télécharger l'application et les différentes ressources liées au projet

2.3.3 Capture d'écran

2.3.3.1 Screenshoot - Figure 1

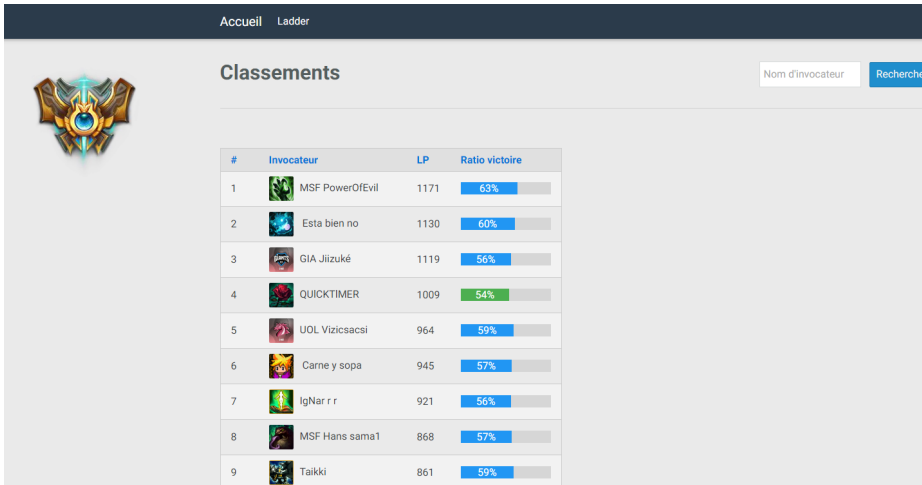


HentioR - EUW

| Mode de jeu | Victoire | Tués | Assistances | Minions |
|-----------------|----------|-------|-------------|---------|
| Unranked | 1510 | 24029 | 16885 | 379034 |
| AramUnranked5x5 | 150 | 3784 | 5173 | |
| RankedSolo5x5 | 128 | 1821 | 1254 | 46641 |
| Unranked3x3 | 58 | 830 | 504 | 9647 |
| CAP5x5 | 47 | 689 | 516 | 13744 |
| RankedFlexSR | 31 | 515 | 288 | 11562 |
| URF | 22 | | | |
| CoopVsAI | 20 | 215 | 196 | 1637 |
| RankedFlexTT | 3 | 35 | 22 | 548 |
| RankedTeam3x3 | 0 | | | |

Page affichant les statistiques d'un invocateur selon sa région

2.3.3.2 Screenshoot - Figure 2



Accueil Ladder

Classements

Nom d'invocateur Rechercher

| # | Invocateur | LP | Ratio victoire |
|---|-----------------|------|----------------|
| 1 | MSF PowerOfEvil | 1171 | 63% |
| 2 | Esta bien no | 1130 | 60% |
| 3 | GIA Jiizuké | 1119 | 56% |
| 4 | QUICKTIMER | 1009 | 54% |
| 5 | UOL Vizicsacsi | 964 | 59% |
| 6 | Carne y sopa | 945 | 57% |
| 7 | IgNar r r | 921 | 56% |
| 8 | MSF Hans sama1 | 868 | 57% |
| 9 | Taikki | 861 | 59% |

Représentation graphique du pallier challenger.

2.3.3.3 Screenshoot - Figure 3

```

<tbody>
<tr>
<?php
usort($result, "cmp");

for ($i=1; $i < 200; $i++) {
    $sum = $result[$i];
    $sumonerName = $sum->playerOrTeamName;
    $lp = $sum->leaguePoints;
    $wins = $sum->wins;
    $losses = $sum->losses;
    $ratio = intval(($wins/($wins+$losses))*100);

    if ($i%2) {
        echo '<tr class="pure-table-odd">';
    }
    else {
        echo '<tr>';
    }
    echo "<td class='tdid'>".$i;
    echo "<td>".<img class='sumicone' src='http://avatar.leagueoflegends.com/suu/$sumonerName.png' />".$sumonerName."</td>";
    echo "<td>".$lp."</td>";
    if ($ratio > 55) {
        echo "<td>".<div class='w3-light-grey' style='width:140px'>".<div class='w3-container w3-blue w3-center' style='width:$ratio%>$ratio%</div>".</div>".</td>";
    }
    elseif ($ratio < 50) {
        echo "<td>".<div class='w3-light-grey' style='width:140px'>".<div class='w3-container w3-red w3-center' style='width:$ratio%>$ratio%</div>".</div>".</td>";
    }
    elseif ($ratio >= 50 && $ratio <= 55) {
        echo "<td>".<div class='w3-light-grey' style='width:140px'>".<div class='w3-container w3-green' style='width:$ratio%>$ratio%</div>".</div>".</td>";
    }
    }
    echo "</tr>";
}
}
</tr>
</tbody>

```

Code html/php permettant un affichage de tous les joueurs du pallier challenger. Classe progress bar permettant une représentation du ratio victoire/défaite d'un joueur.

2.3.3.4 Screenshoot - Figure 4

```

function __construct($key,$region,$returnJSON=false){
    $this->api_key=trim($key);
    $this->region=trim(strtolower($region));
    $this->returnJSON=$returnJSON;
}

public function getSummonerByName($Name){
    /* Get summoner objects mapped by standardized summoner name for a given list of summoner names */
    return $this->getData($this->getDataUrl('summoner/by-name/'.(is_array($Name)?implode(',',$Name):$Name),'1.4'));
}

public function getSummoner($SummonerID,$page='') {
    /* Get summoner objects mapped by summoner ID for a given list of summoner IDs
    (string) $page => ['','masteries','runes','name']
    */
    return $this->getData($this->getDataUrl('summoner/'.(is_array($SummonerID)?implode(',',$SummonerID):$SummonerID).'/'.$page,'1.4'));
}

public function getSummonerStats($SummonerID,$isRanked=false,$season=false){
    /* Get player stats by summoner ID */
    return $this->getData($this->getDataUrl('stats/by-summoner/'.$SummonerID.'/'.(isRanked?'ranked':'summary'),'1.3',$season?array('season'=>$season):array()));
}

public function getTeams($SummonerID){
    /* Get teams mapped by summoner ID for a given list of summoner IDs */
    return $this->getData($this->getDataUrl('team/by-summoner/'.(is_array($SummonerID)?implode(',',$SummonerID):$SummonerID),'2.4'));
}

```

Client PHP de l'API. On trouve ici la fonction construct et des exemples des différentes méthodes disponibles.

3 Outils

Le projet **LEAGUE STATS** a nécessité l'exploitation d'un bon nombre de langages et d'applications diverses et variés dans le but de répondre aux exigences attendues en terme de fonctionnalités et d'ergonomie. Voici la liste exhaustive de toutes les technologies utilisés durant le développement du projet.

3.1 Web

- HTML
- CSS
- PHP
- JavaScript

3.2 API

- Client en PHP de l'API
- Client en JAVA de l'API

3.3 Application Mobile

- Java
- XML
- SpringBoot

3.4 Logiciels

- Adobe Photoshop
- Paint
- Latex
- StarUML
- IntelliJ IDEA
- Android Studio

4 Conclusion

Ce projet est pour moi une réelle opportunité de développer mes compétences et d'acquérir de l'expérience en gestion de projet. L'apprentissage de nouveaux langages et framework est pour moi un véritable atout dans la conception d'application et de site web. Mon objectif étant de savoir développer une application de la conceptualisation à la publication dans un temps imparti. Je suis convaincu que ce projet est porteur et qu'il saura convaincre les utilisateurs de *League of Legends*. La partie développement ne sera pas déliée de problème mais ma motivation est telle que je pense pouvoir parvenir à palier aux différentes difficultés que je pourrai rencontrer.

5 Annexes

5.1 Définition et acronymes

API : API est un acronyme pour Applications Programming Interface.

Une API est une interface de programmation qui permet de se « brancher » sur une application pour échanger des données.

CSS : Cascading Style Sheets, feuilles de styles en cascade en français, est un langage informatique qui sert à présenter et styliser le contenu d'une site internet. JavaFX utilise également le CSS.

Framework : Ensemble de bibliothèques permettant le développement rapide d'applications.

IntelliJ : IDE écrit en java permettant de créer des projets de développement dans n'importe quel langage.

JavaFX : JavaFX est la la bibliothèque de création d'interface graphique officielle du langage Java.

MOBA : De l'anglais "Multiplayer Online Battle Arena". Il s'agit d'un genre essentiellement multijoueur, qui se joue généralement avec deux équipes de cinq joueurs. L'objectif pour chaque équipe est de détruire la structure principale de l'équipe adverse, au moyen des personnages contrôlés par chaque joueur et avec l'aide des unités contrôlées par l'ordinateur.