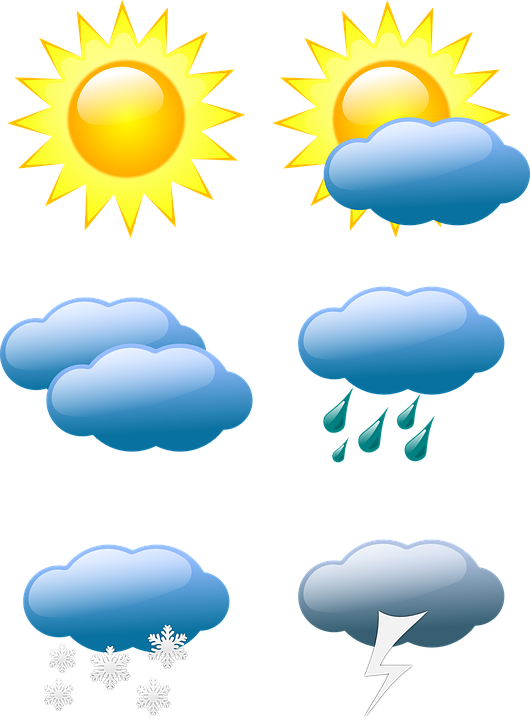
Web service client consumer





|  |  |
| --- | --- |
| Auteur | Brandon Mérillat |
| Date de début de projet | 19/12/16 |
| Date de fin de projet | 06/02/17 |
| Date de fin du rapport | 06/02/17 |
| Version |  |

Table des matières

[1 Introduction 3](#_Toc474143810)

[2 Documentation de développement 3](#_Toc474143811)

[2.1 Cahier des charges 3](#_Toc474143812)

[2.2 Définition des conventions applicables 3](#_Toc474143813)

[2.3 Planning initial 3](#_Toc474143814)

[3 Code clé 4](#_Toc474143815)

[3.1 C# Méthode « RecupereMeteo » 4](#_Toc474143816)

[3.2 Explication choix de code 5](#_Toc474143817)

[3.3 La phase de tests 5](#_Toc474143818)

[4 Mode d'emploi utilisateur 6](#_Toc474143819)

[5 Conclusions 6](#_Toc474143820)

[6 Annexes 7](#_Toc474143821)

[6.1 Journal de bord 7](#_Toc474143822)

[6.2 Cahier des charges détaillé 7](#_Toc474143823)

[6.3 Références 7](#_Toc474143824)

# Introduction

J’ai été mandaté par M. Sacchetti pour réaliser un projet de Météo en PHP ainsi qu’en C# ou en Android.

Le projet consiste à afficher la météo pour une ville se trouvant dans une liste déroulante. Quatre villes doivent se trouver dans la liste par défaut. On peut ajouter des villes qui seront conservée.

Le projet à commencer le 19/12/2016 et doit être rendu le 6/02/2017 à 16h20.

# Documentation de développement

## Cahier des charges

L’application permet d’afficher la météo des prochains 5 jours. Des images doivent illustrer chacun de ces jours. Le choix de la localité doit se faire par une liste déroulante. Par défaut, les villes présentes sont Neuchâtel, La Chaux-de-fonds, Berne et Lausanne.

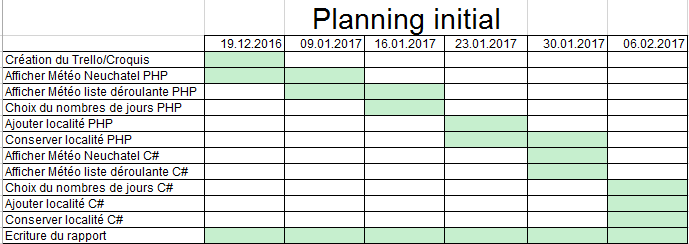
Il faut impérativement développer 2 applications. Une en PHP et une en C# ou Android.

Il doit être possible d’ajouter des localités supplémentaires qui seront conservée au prochain lancement de l’application.

## Définition des conventions applicables

Décrire les conventions des nommages des variables, des fonctions et des fichiers. Cette rubrique permet aussi la définition des standards graphiques applicables.

## Planning initial



# Code clé

## C# Méthode « RecupereMeteo »

private void RecupererMeteo(string Ville)

{

Meteo MeteoReponse = new Meteo();

try

{

Connexion au site.

string endPoint = @"http://www.prevision-meteo.ch/services/json/" + Ville;

var client = new RestClient(endPoint);

var json = client.MakeRequest();

object objResponse = JsonConvert.DeserializeObject(json, typeof(Meteo));

//Converti dans le type requis

MeteoReponse = (Meteo)objResponse;

int nbJours = Convert.ToInt16(cboPrevision.SelectedValue);

EffacerTout();

//La boucle for permet de savoir combien de jours on affiche.

for(int i = 0; i <= nbJours; i++)

{

//Le switch affiche en fonction de i le nombre de jours. Ecriture des prévisions dans les textboxs et affichage des images.

switch (i)

{

case 0: //Le 0 est là car lorsqu’on lance l'application il se peut que le programme vienne ici. Dans ce cas, comme nous n'avons pas défini de jour, il pourrait générer une erreur sans ce « case ».

break;

case 1:

tbxMeteoJour0.Text = AfficherPrevision\_Jour\_0(MeteoReponse);

img\_day\_0.Source = AfficherIcone(MeteoReponse.fcst\_day\_0.icon);

break;

case 2: tbxMeteoJour1.Text = AfficherPrevision\_Jour\_1(MeteoReponse);

img\_day\_1.Source = AfficherIcone(MeteoReponse.fcst\_day\_1.icon);

break;

case 3:

tbxMeteoJour2.Text = AfficherPrevision\_Jour\_2(MeteoReponse);

img\_day\_2.Source = AfficherIcone(MeteoReponse.fcst\_day\_2.icon);

break;

case 4:

tbxMeteoJour3.Text = AfficherPrevision\_Jour\_3(MeteoReponse);

img\_day\_3.Source = AfficherIcone(MeteoReponse.fcst\_day\_3.icon);

break;

case 5:

tbxMeteoJour4.Text = AfficherPrevision\_Jour\_4(MeteoReponse);

img\_day\_4.Source = AfficherIcone(MeteoReponse.fcst\_day\_4.icon);

break;

default:

break;

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

## Explication choix de code

PHP

* Switch : Le switch a été utilisé pour ne pas prendre trop de temps sur la partie PHP. Il est également utile car permet de gérer l’intrusion d’un jour supérieur ou inférieur dans l’URL (7 jours / -2 jours). Ce qui ajoute une petite vérification d’erreur.
* Cookie : Les cookies sont utiles car ils permettent de sauvegarder d’une session à l’autre les choix de la ville. Qui plus est, on peut choisir la durée de vie de ces derniers et ils sont enregistrés en local. Ce qui permet à deux utilisateurs distincts de ne pas voir les mêmes villes.

C#

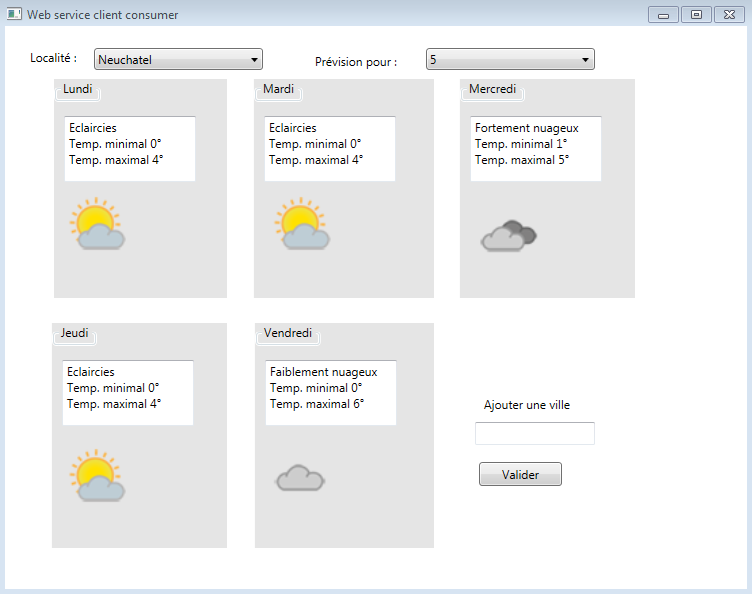
* Fichier Texte : Le fichier texte permet de garder une trace en locale des villes que l’on a ajoutées.

## La phase de tests

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test à effectuer** | | **Résultat**  **escompté** | **Résultat**  **obtenu** | **Constatation** |
| Généralités | | | | |
| C# | Afficher les prévisions pour Neuchâtel sur 5 jours. | Affichage des 5 jours | Affichage des 5 jours | OK |
| Afficher les prévisions pour une ville se trouvant dans la liste déroulante. 5 jours | Affichage des 5 jours pour la bonne ville. | Affichage des 5 jours pour la bonne ville | OK |
| Les villes par défaut sont présentes | 4 villes | 4 villes | OK |
| Ajouter de Geneve | Geneve est ajouté. | Geneve est ajouté | OK |
| Relancer le programme pour vérifier que l’ajout de Geneve est sauvegardé | Geneve est sauvegardé. | Geneve est sauvegardé. | OK |
| PHP | Afficher les prévisions pour Neuchâtel sur 5 jours. | Affichage des 5 jours | Affichage des 5 jours | OK |
| Afficher les prévisions pour une ville se trouvant dans la liste déroulante. 5 jours | Affichage des 5 jours pour la bonne ville. | Affichage des 5 jours pour la bonne ville | OK |
| Les villes par défaut sont présentes | 4 villes | 4 villes | OK |
| Ajouter Geneve | Geneve est ajouté | Geneve est ajouté | OK |
| Relancer le programme pour vérifier que l’ajout de Geneve est sauvegardé | Geneve est sauvegardé. | Geneve est sauvegardé | OK |
| Changer la valeur du jour dans l’URL | Le programme ne plante pas. Et affiche un message disant de choisir le jour. | Le programme ne plante pas. Et affiche un message disant de choisir le jour. | OK |

# Mode d'emploi utilisateur

Utilisation de l’application C# :



Par défaut, le programme affiche la météo des 5 prochains jours à Neuchâtel.

Pour changer la localité, cliquez sur la liste déroulante où « Neuchâtel » est marqué (Se référer à la capture d’écran.). Pour changer le nombre de prévision souhaiter, cliquez sur la liste déroulant où « 5 » est affiché (Se référer à la capture d’écran.).

Pour ajouter une ville, inscrivez votre ville en bas à droite, puis cliquez sur « Valider ». Un message apparaîtra à l’écran pour vous indiquer que la ville a été ajoutée.

La ville ainsi ajoutée apparaîtra tout en bas de la liste et sera sauvegardée pour votre prochaine connexion.

Utilisation de l’application PHP :



Voici l’affichage par défaut de l’application PHP. Pour choisir la ville, cliquez sur le « Neuchâtel » (Référence à la capture d’écran) puis choisissez quel villes vous souhaiter se voir. Sélectionnez ensuite le nombre de jours. Pour ce faire, cliquez sur « Aujourd’hui » puis sélectionnez le nombre de jour désiré. Cliquez ensuite sur « Valider » pour voir apparaître la météo.

Pour ajouter une ville, entrez la ville désirée dans le champ texte, puis cliquez sur « Ajouter ». ATTENTION, le nom de la ville ne doit pas comporter d’accentuation pour la plupart des villes. Les villes doivent également être écrites dans la langue de la ville. (Bâle -> Basel car situé en Suisse allemande).

# Conclusions

Les deux applications devant être rendues sont opérationnelles. Elles permettent toutes deux d’afficher les prévisions météorologiques sur 5 jours. Toutes les fonctionnalités devant être ajoutées sont fonctionnelles, on peut ajouter une ville, la choisir dans la liste, choisir le nombre de jours que l’on souhaite voir s’afficher et les villes ajoutées restent lors du second lancement des applications. Des images illustrent chaque jour. En revanche, les 2 applications ne gèrent pas encore les erreurs.

Amélioration potentielle: Ajout des sécurités relatives à l’entrée de la ville. Vérification de la ville pour savoir si elle existe ou non. Optimisation de certaines parties du code. Permettre d’ajouter une ville dans la bonne écriture sans générer d’erreurs (Ex : Écrire Bâle au lieu de Basel).

Ressenti : Le projet était très intéressant à réaliser. Le fait de devoir développer 2 applications utilisant un langage de programmation différent est un type de projet qu’il me plairait à refaire. En revanche, j’aurais dû travailler un peu plus sur la partie PHP pour rendre le code un peu plus « propre ». En effet, l’utilisation du switch rend le code plutôt vulgaire. Si je suis amené à travailler sur ce projet à nouveau ou à pouvoir l’améliorer, c’est la partie que je changerais en priorité. Sinon, le fait de devoir se mettre dans les conditions d’un TPI a rendu le projet beaucoup plus attractif et a permis un engagement plus important.

# Annexes

## Journal de bord

Inscrivez ici le détail de votre avancement quotidien sous la forme d’un journal de travail détaillé de manière à ce que le lecteur puisse retracer votre travail d’une manière quotidienne.

## Cahier des charges détaillé

Le client veut une application permettant d’afficher la météo des prochains 5 jours. Des images doivent illustrer chacun de ces jours.

Le choix de la localité doit se faire par une liste déroulante. Par défaut les villes présentes sont Neuchâtel, La Chaux-de-fonds, Berne et Lausanne.

Vous devez impérativement développer 2 applications :

* La première en PHP, en tenant compte de la responsivité et donc le Framework Bootstrap.
* Une deuxième application : soit en C# soit en Android.

L’utilisateur doit pouvoir ajouter des localités supplémentaires, ces localités seront conservées au prochain lancement de l’application. Proposez des solutions différentes et choisissez celles-qui vous convient le mieux.

## Références

Image :

Météo : <https://pixabay.com/fr/pr%C3%A9visions-m%C3%A9t%C3%A9o-m%C3%A9t%C3%A9o-soleil-nuage-146472/> Image de la météo.

C# PHP : <https://www.youtube.com/watch?v=3uuZhHXSW1Y> L’image provient de la vidéo.