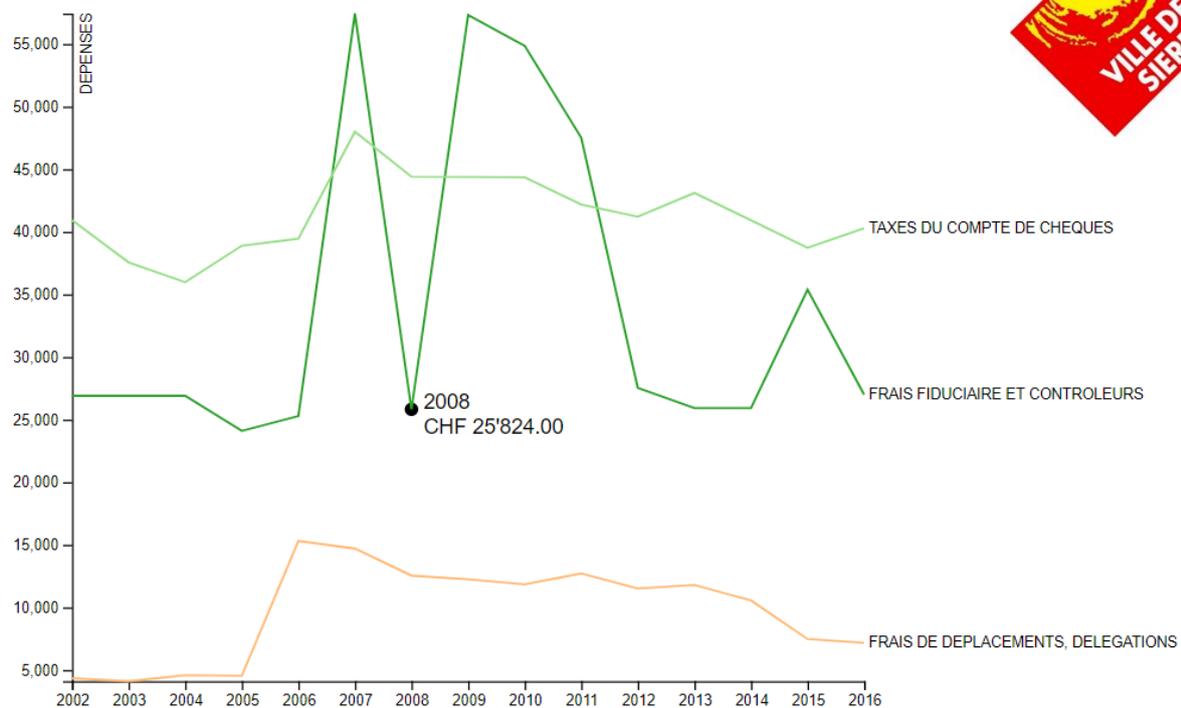


Travail de Bachelor 2017

Plateforme web pour les finances communales



Étudiant : Sandro Mathier

Professeur : Jean-Pierre Rey

Déposé le : 02.08.2017

Résumé

La ville de Sierre, bien connue pour ses vignobles et son climat, compte, comme la plupart des villes et des communes, de nombreux comptes différents dans sa comptabilité.

L'objectif principal de ce travail de Bachelor était de créer un prototype de plateforme web donnant la possibilité d'afficher différentes visualisations des comptes de la ville. Celles-ci leur permettraient de prendre des décisions, si nécessaire. Avant la réalisation, il fallait d'abord comprendre la thématique, analyser les données et choisir les outils et les langages de programmation. En se basant sur le *framework* PHP (*Hypertext Preprocessor*) « Laravel », il serait possible de développer sans problème de nouvelles fonctionnalités. Avec « D3.js » la manipulation des éléments DOM (*Document Object Model*) est facile à gérer. Il existait déjà certains *mock-ups*, plusieurs *proofs of concepts* réalisés avec Excel et des fragments de code JavaScript non documentés. Les cinq pages avec les visualisations différentes à mettre en place étaient les suivantes :

- **Historique : Graphique en lignes** (comptes 2002-2016) ;
- **Heatmap : Carte thermique** (comptes 2002-2016) ;
- **Visual Compta : Graphique en barres** (comptes 2016) ;
- **Sankey** (groupes des comptes 2016) ;
- **Investissements : Bulles** (planifications des investissements 2015-2018).

Avec les visualisations précitées, la ville de Sierre reçoit un nouvel instrument pour analyser ses données fiscales plus rapidement et plus facilement. Si on remplace les données par les données d'une autre commune, il serait aussi possible d'utiliser cet outil pour d'autres communes.

Mots clés : *Data Visualisation*, JavaScript, Finance, Commune, Plateforme Web

Avant-propos

Ce travail de Bachelor a été effectué au cours du sixième semestre de la filière « Informatique de gestion ». Le travail a été proposé par MM. Arnaud Zufferey et Jean-Pierre Rey et a été réalisé entre mai et août 2017.

Vu que j'ai travaillé pendant trois ans à la Commune de Loèche et deux ans chez un fournisseur de logiciel pour les communes haut-valaisans, j'ai éprouvé un très grand intérêt pour ce travail de Bachelor. Il m'a été très bénéfique de connaître les problèmes et les besoins d'une commune. J'étais de plus très motivé par la mise en place d'une solution basée sur une plateforme web. Étant de langue maternelle allemande, la réalisation de ce travail en français était un défi majeur pour moi.

Les objectifs de ce travail sont les suivants :

- Compréhension et appréhension de la thématique et la problématique ;
- Analyse du contexte donné, des outils et des langages de programmation ;
- Réalisation du prototype de la plateforme web.

La difficulté majeure en ce qui concerne les visualisations des données est d'obtenir des représentations significatives. La thématique est très vaste et il fallait donc se concentrer vraiment sur les tâches spécifiées. Il existe beaucoup de développements futurs possibles, par exemple intégrer une option pour ajouter de nouvelles comptabilités pour les années suivantes.

Pour faciliter la lecture de ce document, seule la forme masculine sera utilisée dans le texte.

Remerciements

Je me tiens à remercier sincèrement tous ceux qui m'ont aidé, soutenu et motivé pendant ce travail de Bachelor.

Un très grand merci à MM. Jean-Pierre Rey et Arnaud Zufferey pour leur disponibilité et leur suivi de mon travail, ainsi qu'à la ville de Sierre pour avoir mis à disposition ses données.

Je remercie également MM. Arnaud Zufferey, Florian Evéquo et Daniel Hunacek pour leurs travaux préliminaires. Les *mock-ups* et les prototypes existants m'ont permis d'avancer plus rapidement dans ce travail.

Enfin, je tiens à remercier mes parents pour la relecture de ce travail et leur soutien pendant toute la formation.

Table des matières

Liste des tableaux.....	VIII
Liste des figures.....	IX
Liste des abréviations.....	XI
Introduction.....	1
Présentation de la ville de Sierre.....	1
Situation actuelle.....	1
Solution proposée.....	2
Méthodologie utilisée.....	2
1. Analyse.....	4
1.1 Visualisation et exploration des données.....	4
1.2 Compréhension des données.....	7
1.2.1 Services.....	7
1.2.2 Comptes de fonctionnement.....	7
1.2.3 Plan financier.....	8
1.3 Langages de programmation.....	9
1.3.1 Backend.....	9
1.3.2 Frontend.....	10
1.4 Frameworks.....	11
1.4.1 Backend.....	11
1.4.2 Frontend.....	12
1.5 Base de données.....	14
2. Mock-ups.....	15
2.1 Pages.....	15
2.2 Visualisation.....	18
3. Réalisation.....	21

3.1	Création du projet.....	21
3.1.1	Installation.....	21
3.1.2	Structure du répertoire	21
3.1.3	Base de données et configuration de messagerie	22
3.2	Modèle	23
3.3	Administration des utilisateurs.....	24
3.4	Visualisations	26
3.4.1	Historique	26
3.4.2	Heatmap.....	28
3.4.3	Visual compta	29
3.4.4	Sankey	30
3.4.5	Plan des investissements	31
3.5	Compatibilité des navigateurs	32
3.6	Multi-plateformes.....	33
3.7	Localisation	34
4.	Scrum.....	35
4.1	Product Backlog avec les tâches	35
4.2	Sprints	37
	Conclusion	38
	Résultats du travail	38
	Recommandations, limites et perspectives	39
	Sources	40
	Références.....	41
	Annexe I : Cahier des charges	44
	Annexe II : Contenu du CD	49
	Annexe III : Mock-up historique	50

Sandro Mathier

Déclaration de l'auteur 52

Liste des tableaux

Tableau 1 : La ville de Sierre en chiffres (Ville de Sierre, 2016)	1
Tableau 2 : Ensembles de données, quartet d'Anscombe (données de l'auteur)	4
Tableau 3 : Totaux des services 2016 (Ville de Sierre, 2017)	7
Tableau 4 : Comparaison des frameworks charts (Données de l'auteur)	12
Tableau 5 : Actions traitées par le contrôleur de ressource (Laravel, s.d.)	25
Tableau 6 : Product backlog avec les tâches (Données de l'auteur)	35

Liste des figures

Figure 1 : Logo Ville de Sierre (Ville de Sierre, 2016)	1
Figure 2 : La méthode Scrum, selon Mike Cohn (Cohn, Mountain Goat Software, 2014)	2
Figure 3 : graphiques, quartet d'Anscombe (Kosara, 2011)	5
Figure 4 : Faire mentir la visualisation des données (Parikh, 2014)	5
Figure 5 : Logo PHP (php, s.d.)	9
Figure 6 : Utilisation des langages web du côté serveur (W3Techs.com, 2017)	9
Figure 7: Logo HTML, CSS et JavaScript (e-novations, s.d.)	10
Figure 8 : Logo Laravel (Laravel, s.d.)	11
Figure 9 : Google Trend,s comparaison des framework PHP (Google Trends, 2017).....	11
Figure 10 : Logo D3 (D3.js, s.d.).....	12
Figure 11: Logo Bootstrap (Bootstrap, s.d.)	13
Figure 12 : Logo jQuery et DataTables (doktapepa, 2016)	13
Figure 13 : Logo MySQL (MySQL, s.d.)	14
Figure 14 : Page de bienvenue (données de l'auteur)	15
Figure 15 : Page d'accueil et navigation (Données de l'auteur).....	16
Figure 16 : Connexion et création d'un compte (données de l'auteur).....	17
Figure 17 : Administration des utilisateurs (données de l'auteur)	17
Figure 18 : Mock-up heatmap (Arnaud Zufferey & Florian Evéquoz)	18
Figure 19 : Mock-up, Visual Compta (Arnaud Zufferey)	19
Figure 20 : Mock-up, Sankey (d3noob, 2017)	19
Figure 21 : Mock-up, plan des investissements (Arnaud Zufferey et Florian Evéquoz)	20
Figure 22 : Page vide avec le template (données de l'auteur).....	23
Figure 23 : Tableau de l'administration des utilisateurs (données de l'auteur)	24
Figure 24 : Screenshot graphique évolution (Données de l'auteur).....	27
Figure 25 : tableau des indicateurs (données de l'auteur)	27
Figure 26: Légende heatmap (Données de l'auteur)	28
Figure 27 : capture d'écran heatmap (données de l'auteur)	28
Figure 28 : Capture d'écran visual compta	29
Figure 29 : Capture d'écran Sankey (données de l'auteur).....	30
Figure 30 : capture d'écran, plan des investissements	31

Figure 31 : Utilisations des navigateurs (NetMarketShare, 2017) 32

Figure 32 : Part de marché ordinateurs de bureau contre. mobiles contre tablettes (StatCounter, 2017)..... 33

Figure 33 : en-tête de page en français et allemand (Données de l’auteur) 34

Liste des abréviations

PHP	Hypertext Preprocessor (langage de scripts)
DOM	Document Object Model (interface de programmation)
CRISP-DM	Cross Industry Standard Process for Data Mining (modèle de processus)
HTML	Hypertext Markup Language (langage de balisage)
CSS	Cascading Style Sheet (feuilles de style en cascade)
JS	JavaScript (langage de programmation)
D3	Data Driven Documents (<i>framework</i> pour JavaScript)
SQL	Structured Query Language (langage de requête structurée)
REST	Representational State Transfer (style d'architecture dans le net)
JSON	JavaScript Object Notation (format des données)
CSV	Coma-separated Values (format des données)
PO	Product Owner (un rôle dans Scrum)
PSPI	Potentially Shippable Product Increment (le produit livrable à la fin de chaque sprint)

Introduction

Présentation de la ville de Sierre

La ville de Sierre est située au cœur du Valais, en Suisse. C'est une ville qui bénéficie de multiples atouts liés au vin, au climat, à l'histoire et à sa situation géographique. Sierre se trouve à la frontière linguistique entre le français et l'allemand. Il s'agit de la ville la plus ensoleillée de Suisse avec environ 300 jours d'ensoleillement par an. La ville s'est construite autour de ses quartiers historiques, liés à la transhumance et au Val d'Anniviers. Ainsi, le quartier du Bourg, de Tservetta, de Muraz, de Villa, de Glarey ou encore de Géronde sont des composantes essentielles d'une ville originale, définitivement pas comme les autres. (Office de Tourisme, s.d.)



Figure 1 : Logo Ville de Sierre (Ville de Sierre, 2016)

Tableau 1 : La ville de Sierre en chiffres (Ville de Sierre, 2016)

Habitants	16'711 (31.12.2015)
Superficie	19,17 km ²
Densité	871 hab./km ²
Altitude	533 m

Situation actuelle

La comptabilité de la ville de Sierre comprend plus de 1 200 comptes différents. Au vu de ce grand nombre de comptes, il est difficile d'obtenir une vue d'ensemble. Il manque également un outil permettant de comparer les différents comptes pour une période donnée. Il n'existe actuellement que quelques graphiques Excel. Ceux-ci représentent surtout les chiffres clés de la commune, comme les dettes par habitant ou la marge d'autofinancement. Ces graphiques sont plutôt destinés à permettre aux habitants de mieux comprendre la situation financière de la commune.

La visualisation des finances aidera la commune à comprendre mieux et plus rapidement ses données et à prendre des décisions. La plateforme web est donc un instrument prévu pour les membres de la commission des finances de Sierre.

Solution proposée

L'objectif est de mettre en place une plateforme web regroupant les différentes visualisations qui ont été élaborées avant ce travail. Sur cette plateforme, les utilisateurs peuvent se connecter, puis interagir avec les différents graphiques. Le but principal est d'obtenir une vue d'ensemble de tous les comptes, de l'état financier, ainsi de la planification des investissements. Si cette solution fait ses preuves, elle pourrait être également mise en place par d'autres communes.

Méthodologie utilisée

Ce travail a été réalisé avec une version simplifiée de la méthode « Scrum », une méthode Agile qui est souvent utilisée pour mener des projets dans le monde de l'informatique. Cette méthode utilise un processus de développement itératif. (Cohn, Mountain Goat Software, s.d.)

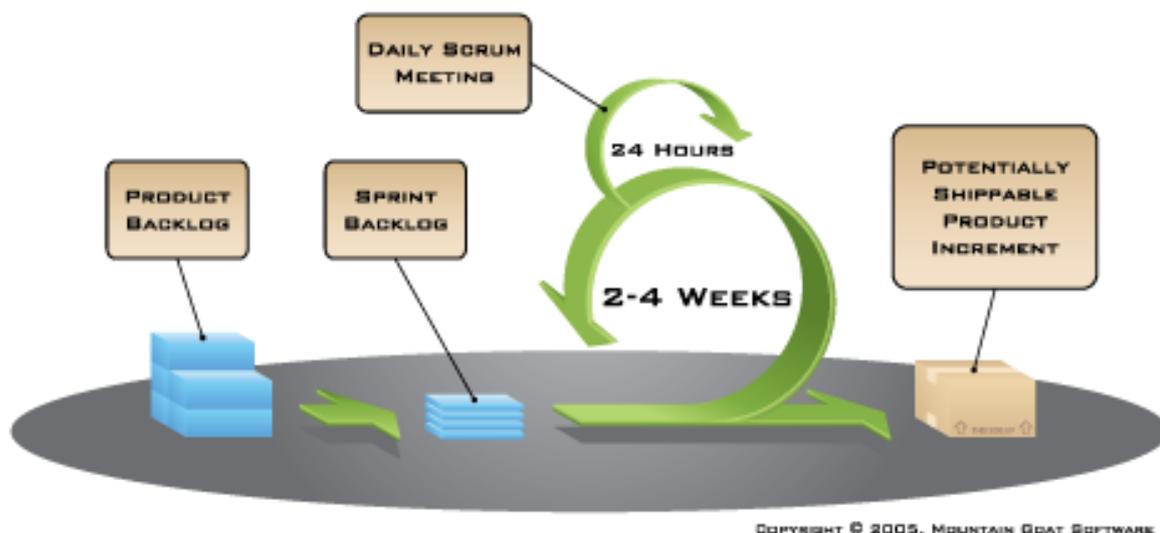


Figure 2 : La méthode Scrum, selon Mike Cohn (Cohn, Mountain Goat Software, 2014)

Le projet a été découpé en plusieurs étapes, les « sprints ». Chaque sprint a duré de deux semaines et a contenu plusieurs *user stories*. Les diverses *user stories* réalisées dans un sprint sont définies dans le *product backlog*. Pour chaque *user story*, la complexité est estimée avec des *story points*. À la fin de chaque sprint, le résultat a été présenté au PO, M. Arnaud Zufferey, pour faire la rétrospective du sprint.

Avec cette méthodologie, l'échange entre le client et le développeur est beaucoup plus simple. À la fin de chaque sprint, le client peut voir une version totalement fonctionnelle du produit et il peut faire des remarques.

En ce qui concerne le traitement des données, la méthode CRISP-DM a été utilisée. Le fonctionnement de cette méthode est expliqué dans le chapitre suivant.

1. Analyse

1.1 Visualisation et exploration des données

L'objectif de la visualisation des données est de permettre de comprendre les données plus rapidement et plus facilement. Selon Alain Fernandez « la Data Visualisation est une technique d'exploration et d'analyse des données numériques à l'aide de graphiques » (Fernandez, s.d.).

Avec le célèbre Quartet d'Anscombe, le statisticien Francis Anscombe a réussi à démontrer l'importance des graphiques pour l'analyse des données, déjà en 1973. Dans son exemple, il étudie quatre ensembles de données qui ont plus ou moins les mêmes propriétés statistiques, mais qui apparaissent très différents, dès qu'on les représente sous forme graphique (Anscombe, 1973).

Tableau 2 : Ensembles de données, quartet d'Anscombe (données de l'auteur)

I		II		III		IV	
x	y	x	y	x	y	x	y
10	8.04	10	9.14	10	7.46	8	6.58
8	6.95	8	8.14	8	6.77	8	5.76
13	7.58	13	8.74	13	12.74	8	7.71
9	8.81	9	8.77	9	7.11	8	8.84
11	8.33	11	9.26	11	7.81	8	8.47
14	9.96	14	8.1	14	8.84	8	7.04
6	7.24	6	6.13	6	6.08	8	5.25
4	4.26	4	3.1	4	5.39	19	12.5
12	10.84	12	9.13	12	8.15	8	5.56
7	4.82	7	7.26	7	6.42	8	7.91
5	5.68	5	4.74	5	5.73	8	6.89

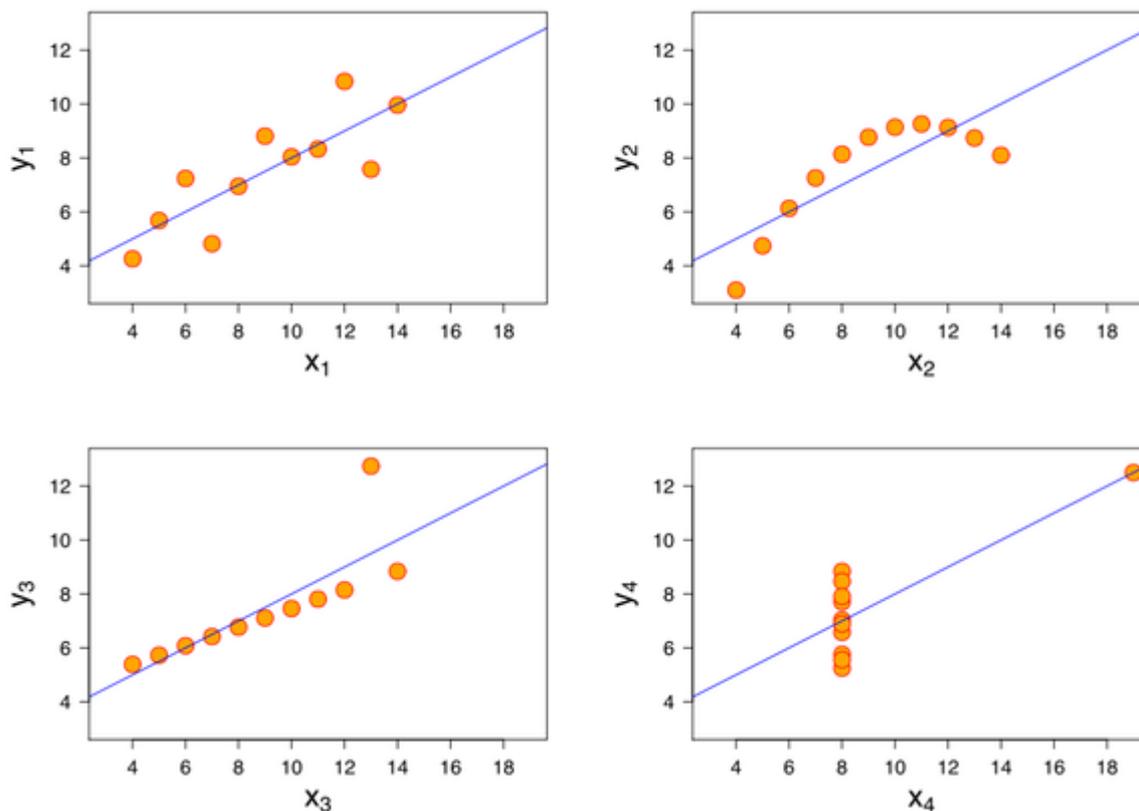


Figure 3 : graphiques, quartet d'Anscombe (Kosara, 2011)

Les données du tableau 2 sont difficiles à comprendre, alors qu'avec les graphiques (figure 3) on voit immédiatement les différences entre les quatre ensembles de données. La visualisation n'est pas aussi précise que les statistiques, mais l'affichage peut être beaucoup plus pratique pour découvrir des structures intéressantes (Kosara, 2011).

Il existe de nombreuses possibilités différentes de représenter des données visuellement. Les défis majeurs sont de disposer de visualisations qui représentent la vérité, ainsi que de graphiques significatifs. L'exemple de la figure 4 montre deux graphiques avec les mêmes données, mais dont le résultat visuel est très différent.

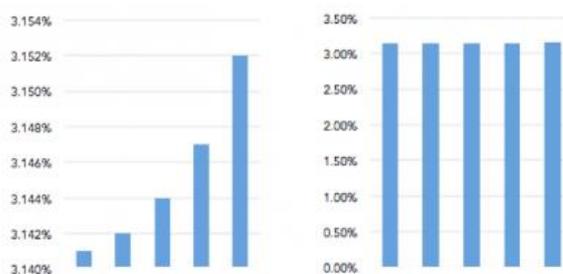


Figure 4 : Faire mentir la visualisation des données (Parikh, 2014)

Si on veut éviter ces risques, il faut nécessairement comprendre les données. Les formes de présentation des données ont déjà été choisies préalablement à ce travail et pour cette raison, nous ne les développerons pas plus avant ici.

Pour l'exploration des données, l'entreprise IBM a développé la méthode CRISP-DM. Cette méthode se décompose en six étapes (Gaultier, 2016) :

1. La compréhension du problème métier

La première étape, qui a déjà été effectuée en ce qui nous concerne, consiste à définir les problèmes à résoudre ou à améliorer.

2. La compréhension des données

Après avoir trouvé les problèmes, il faut analyser les données pour établir de quelle façon on veut les traiter. Dans ce travail, nous verrons ceci au paragraphe suivant.

3. La préparation des données

Une fois que les données sont claires, il faut les mettre au bon format pour pouvoir les utiliser dans la plateforme web. Dans notre cas, on traduit les données au format JSON.

4. La modélisation

L'ensemble de la réalisation est compris dans cette étape.

5. L'évaluation

Au cours de l'étape d'évaluation, l'objectif est de tester tout ce qui a été réalisé et d'établir si la réalisation correspond aux besoins.

6. Le déploiement

Enfin, la solution réalisée est mise en production pour être fournie aux utilisateurs finaux. Cette étape n'est pas effectuée dans le cadre de ce travail.

La méthode CRISP-DM est, comme Scrum, une méthode agile et itérative. Après la sixième étape, le cycle recommence donc avec la compréhension du problème métier.

1.2 Compréhension des données

La ville de Sierre a fourni sa comptabilité de 2002 à 2016, ainsi que la planification des investissements des années 2015 à 2018. Selon la méthode CRISP-DM, l'étape suivante est celle de la compréhension des données.

1.2.1 Services

Les comptes sont repartis entre plusieurs services et comprennent les charges, ainsi que les recettes. Les totaux des services représentent les montants de tous les comptes. Avec un diagramme de Sankey, les chiffres suivants vont être visualisés sur la plateforme web pour permettre de les comprendre plus rapidement.

Tableau 3 : Totaux des services 2016 (Ville de Sierre, 2017)

N° compte	Libellé	Charges	Recettes
10	Administration générale	CHF 5'677'853.75	CHF 2'251'291.47
20	Sécurité publique	CHF 7'666'230.42	CHF 4'227'841.52
30	Enseignement et formation	CHF 14'790'515.60	CHF 1'683'153.45
40	Affaires culturelles, sports, culte	CHF 8'500'817.64	CHF 1'134'328.70
50	Actions sociales et médico-sociales	CHF 10'101'755.21	CHF 189'501.60
60	Eau et énergie	CHF 9'247'692.59	CHF 9'097'206.65
70	Travaux publics, env., municipalité	CHF 18'663'823.18	CHF 9'310'291.49
90	Finances et impôts	CHF 14'326'351.61	CHF 57'730'212.80

1.2.2 Comptes de fonctionnement

Chaque compte a un numéro qui est composé de quatre chiffres. Prenons par exemple le compte « Indemnités du conseil » avec le numéro **10.101.300.02** :

- 10** indique le service dont il s'agit
- 101** désigne la section
- 300** renseigne sur la nature du compte
- 02** constitue la sous-rubrique

Le troisième groupe de chiffres est particulièrement important pour la bonne compréhension des informations. Quatre valeurs sont utilisées. Elles révèlent la nature du compte (Ville de Sierre, 2016, p. 59).

300	charges de fonctionnement
400	revenus de fonctionnement
500	dépenses d'investissement
600	recettes d'investissement

Un compte contient donc soit des charges, soit des revenus, selon le troisième groupe de chiffres. Le montant représente les charges ou les revenus effectifs.

1.2.3 Plan financier

La loi sur les communes du 5 février 2004 définit, dans son article 79, le principe de la planification financière pour les communes. Son contenu est le suivant :

¹ Le conseil municipal établit, pour une durée de quatre ans au moins, une planification financière qu'il porte à la connaissance de l'assemblée primaire ou du conseil général.

² Cette planification financière donne une vue d'ensemble de l'évolution prévisible des recettes et des dépenses courantes, des investissements, ainsi que de la fortune et de l'endettement. (Ville de Sierre, 2016, p. 1)

Les comptes du plan financier sont composés de deux nombres. Le premier nombre indique de nouveau le service (20 correspond à Sécurité publique). Le deuxième nombre indique le compte d'investissement (20 correspond à Police municipale). Dans cet exemple, le compte n° 20.20 contient tous les investissements concernant la police municipale.

Contrairement aux comptes de fonctionnement, chaque compte peut comporter des dépenses et des recettes, par exemple si un projet est subventionné.

1.3 Langages de programmation

1.3.1 Backend

PHP



Figure 5 : Logo PHP
(php, s.d.)

La partie *backend* dans ce projet n'a que peu d'importance, étant donné que l'objectif est surtout d'avoir des visualisations dynamiques des finances communales, qui sont traitées dans le *frontend*. La plateforme web pourrait être réalisée dans plusieurs langages, comme le PHP, l'ASP.NET, le Java, le Ruby, le Python, le JavaScript ou de nombreux autres.

Ce projet peut être réalisé avec chacun de ces langages, mais le PHP est le langage web le plus populaire. Selon W3Techs, qui fait des enquêtes sur l'utilisation de divers types de technologies web, le PHP est utilisé pour plus que 82 % des sites examinés par leur algorithme (W3Techs.com, 2017).

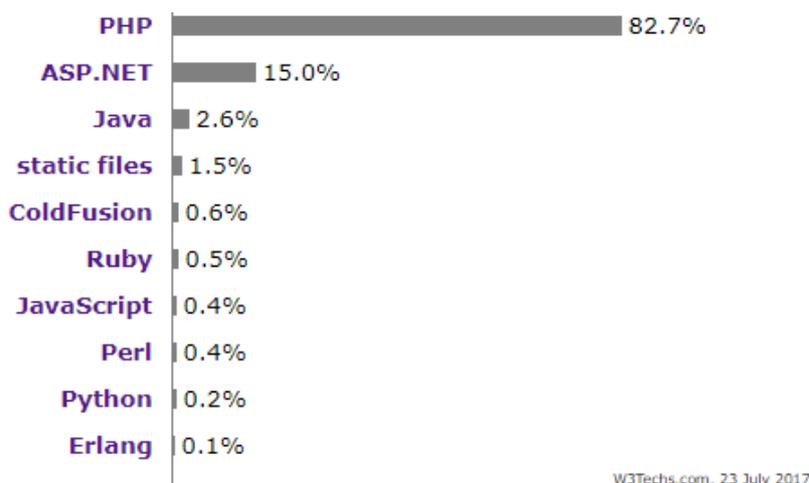


Figure 6 : Utilisation des langages web du côté serveur (W3Techs.com, 2017)

En plus d'être populaire, le langage PHP est également facile à apprendre et, avec ces nombreux *framework*, il permet de créer une plateforme web rapidement. La performance du PHP a été considérablement améliorée avec la version 7. Selon des tests de CCM Benchmark, les gains entre PHP 5.6 et PHP 7 peuvent atteindre 50 % de temps d'exécution et de consommation de mémoire (Crochet-Damais, 2016).

Pour ces raisons, le PHP (Version 7.0) a été choisi comme langage de programmation dans la partie *backend*.

1.3.2 Frontend

HTML, CSS et JavaScript

Le HTML et le CSS sont les bases de la création de pages web (W3C, s.d.). À cela s'ajoute le JavaScript, qui est utilisé pour les interactions avec les utilisateurs. Le HTML est responsable de la structure du contenu et le CSS de la présentation, tandis que le

JavaScript gère le comportement. Le JavaScript est un langage de programmation léger qui est normalement exécuté normalement par le navigateur du client (Phang, 2015, p. 33). Le choix de ces trois langages peut se justifier par un manque de réelles alternatives réelles. Aucune alternative ne peut offrir tant de documentation sur Internet, ainsi que des *frameworks*.



Figure 7: Logo HTML, CSS et JavaScript
(e-novations, s.d.)

1.4 Frameworks

1.4.1 Backend

Laravel



Figure 8 : Logo Laravel (Laravel, s.d.)

De nombreux *frameworks* PHP différents permettent d'accélérer le développement, mais le plus populaire, selon une comparaison de Google Trends, est « Laravel », un *framework* entièrement *open source* sous licence MIT.

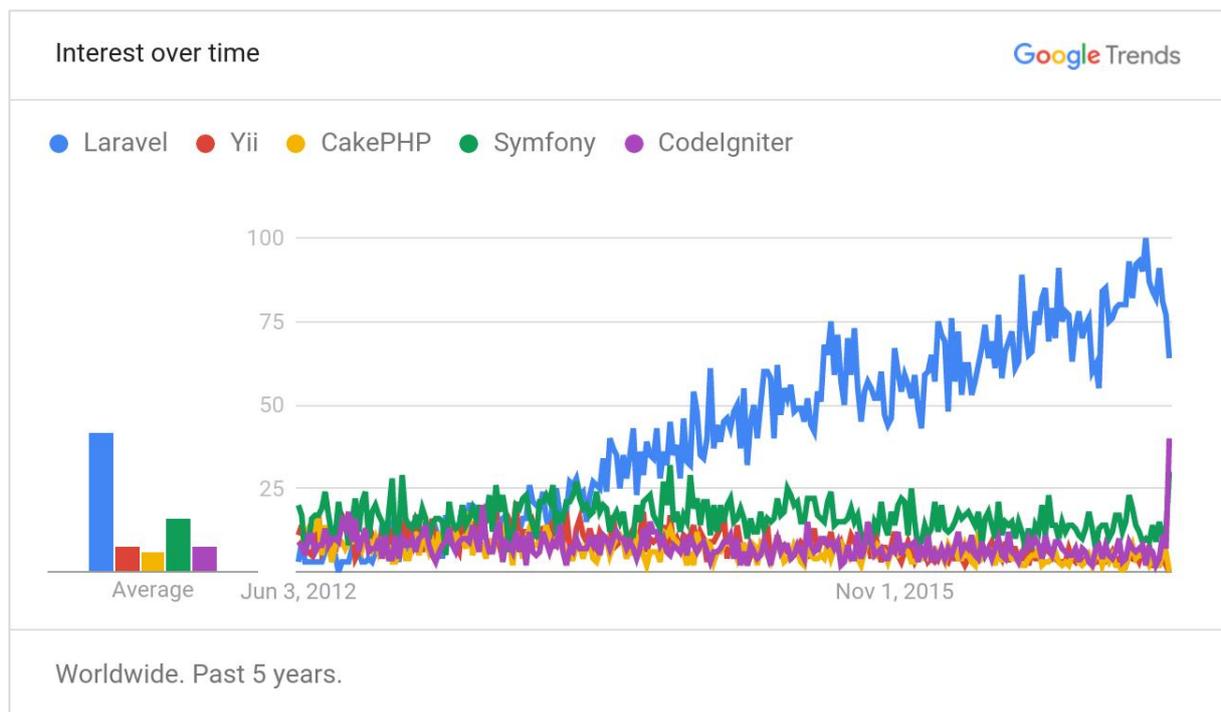


Figure 9 : Google Trends comparaison des framework PHP (Google Trends, 2017)

En raison de sa grande popularité, Laravel offre une grande communauté et une excellente documentation (Garbade, 2016). Laravel propose toutes les fonctionnalités nécessaires, par exemple une option préconçue pour l'authentification, le multilinguisme, le principe MVC (*Model/View/Controller*) qui facilite le débogage et le développement de la plateforme en séparant la logique et la présentation, ou encore un moteur de *templates*, Blade. Le code dans Laravel est bien organisé, réutilisable et maintenable, tout comme la structure et le contenu de la base de données. La localisation permet de créer simplement une plateforme multilingue.

1.4.2 Frontend

D3.js

La plus grande partie de ce projet s'effectue directement dans le *frontend*, afin d'interagir avec les utilisateurs. Le but est de créer les différentes visualisations des données de la commune de Sierre souhaitées par le client.



Figure 10 : Logo D3 (D3.js, s.d.)

Comme pour le *backend*, il existe de nombreux cadres applicatifs JavaScript pour mettre en place ces fonctionnalités. Les plus connus d'entre eux ont été analysés et testés afin de trouver la meilleure solution pour ce travail.

Les critères d'évaluations étaient la fonctionnalité (0-10 points), la facilité d'utilisation (0-5 points), les documentations dans l'Internet, la communauté et le prix. Avec 26 points, D3.js s'est révélé être le meilleur *framework* pour ce projet.

Tableau 4 : Comparaison des frameworks charts (Données de l'auteur)

Logiciel	Fonctionnalité	Facilité d'utilisation	Support	Communauté	Prix	Total
D3.js	10	3	4	5	3	26
Google Charts	8	4	5	3	3	23
Highcharts	8	4	5	5	2	24
Charts.js	6	5	5	4	3	23

D3 est un *framework* très étendu et n'a presque pas des limites. « D3.js » est une bibliothèque pour JavaScript sous la licence « BSD », permettant de créer toutes les visualisations requises.

D3 n'est pas uniquement utilisé pour créer des graphiques, comme Google Charts, Highcharts ou Charts.js, mais propose aussi des fonctionnalités pour manipuler tous les éléments du DOM (D3.js, s.d.). La transformation des données est plus simple et, par rapport autres bibliothèques, D3.js permet de faire plus de modifications.

Bootstrap

Bootstrap est sous licence MIT et utilise le HTML, le CSS et le JavaScript pour simplifier le développement *frontend*. Le point fort de Bootstrap est qu'il est complètement adaptatif. L'affichage est donc optimisé pour toutes les plateformes : moniteurs d'ordinateur, smartphones, tablettes, etc. De plus, de nombreux éléments ont déjà été prédéfinis, ce qui permet aux développeurs d'économiser beaucoup de travail (Bootstrap, s.d.).



Figure 11: Logo Bootstrap
(Bootstrap, s.d.)

DataTables et jQuery

Étant donné le grand nombre de comptes différents, l'affichage d'un tableau avec des fonctionnalités avancées est logique. DataTables, également un projet *open source* sous licence MIT, comporte des fonctions pour les filtres de recherche, la sélection, le tri et la pagination. DataTables étant basé sur jQuery, on doit également implémenter jQuery pour profiter des fonctionnalités de DataTables.



Figure 12 : Logo jQuery et DataTables
(dokterpepa, 2016)

1.5 Base de données

MySQL

MySQL est la base de données la plus commune parmi les bases de données *open source*. Des organisations comme Google, Facebook et Adobe utilisent MySQL pour ses bases de données (MySQL, s.d.). Il s'agit également de la base de données la plus utilisée avec le PHP. MySQL utilise la syntaxe standard SQL et fonctionne très bien avec les applications de toute taille. Avec un petit nombre de requêtes, le choix des bases de données a peu d'importance. Pour ce travail, la base de données ne contient qu'un tableau utilisateur et un tableau pour la réinitialisation des mots de passe.



Figure 13 : Logo MySQL (MySQL, s.d.)

Si le développement de la plateforme web est poursuivi après ce travail, les migrations de Laravel permettent de faire évoluer facilement cette base de données.

2. Mock-ups

Avant de réaliser la plateforme web, la création des *mock-ups* nous permet déjà de nous faire une meilleure idée de la solution. Avec « Balsamiq Mockups », un logiciel libre de l'entreprise Balsamiq Studios, la génération de pages est rapide et très simple (Balsamiq Studios, s.d.). Cet outil a été utilisé pour la création des pages. Les *mock-ups* pour les visualisations ont déjà été créés par les personnes citées dans les remerciements. Le but est d'avoir une présentation uniforme en utilisant uniquement les *frameworks* choisis au chapitre précédent.

2.1 Pages

Page de bienvenue



Figure 14 : Page de bienvenue (données de l'auteur)

La page de bienvenue contient que peu d'éléments. C'est la première page que les utilisateurs voient. De là, ils ont la possibilité de se connecter ou de s'enregistrer.

Page d'accueil et navigation



Figure 15 : Page d'accueil et navigation (Données de l'auteur)

La page d'accueil présente aux utilisateurs les différentes pages de la plateforme. D'ici, les utilisateurs peuvent naviguer vers les pages contenant les visualisations. La barre de navigation est visible sur tous les sites et l'utilisateur voit directement sur quelle page il se trouve grâce à la couleur de fond rouge. Par ailleurs, l'utilisateur peut se déconnecter ou modifier la langue grâce aux liens en haut à droite.

Connexion et création d'un compte

Login

Email

Password

Remember

[Forgot password](#)

Register

Name

Email

Password

Password confirm

Figure 16 : Connexion et création d'un compte (données de l'auteur)

Les pages de connexion et de création d'un compte sont très similaires à celles de la plupart des sites web. Les champs de saisie inutiles, par exemple la date de naissance, sont omis. L'utilisateur a la possibilité de rester connecté et, quand il oublie son mot de passe, il peut le réinitialiser.

Administration des utilisateurs

Name	Mail	Level	Action
Sandro Mathier	sandro@mathier.ch	Admin	<input type="button" value="OK"/> <input type="button" value="Trash"/>
Arnaud Zufferey	arnaud@zufferey.ch	Membre	<input type="button" value="OK"/> <input type="button" value="Trash"/>
New User	new@user.ch	New	<input type="button" value="OK"/> <input type="button" value="Trash"/>

Figure 17 : Administration des utilisateurs (données de l'auteur)

Ce tableau est uniquement destiné aux administrateurs de la plateforme. Avant qu'un utilisateur puisse bénéficier des différentes fonctions, l'administrateur doit lui donner les droits. L'administrateur a également la possibilité de supprimer un utilisateur.

2.2 Visualisation

Historique

La page de l'historique se compose de trois parties. La première comporte deux tableaux, dont le premier affiche tous les comptes de la ville de Sierre. L'utilisateur peut sélectionner les comptes qu'il veut visualiser et les est transmettre à l'autre tableau. Les comptes de ce deuxième tableau vont être visualisés sous forme de graphique linéaire dans la deuxième partie. Le graphique montre les montants des années 2002 à 2016 par défaut. L'utilisateur à l'option de sélectionner de choisir des périodes. La dernière partie montre des indicateurs pour ces comptes, qui peuvent être exportés en format CSV. Le *mock-up* de cette fonction se trouve dans l'annexe III.

Heatmap

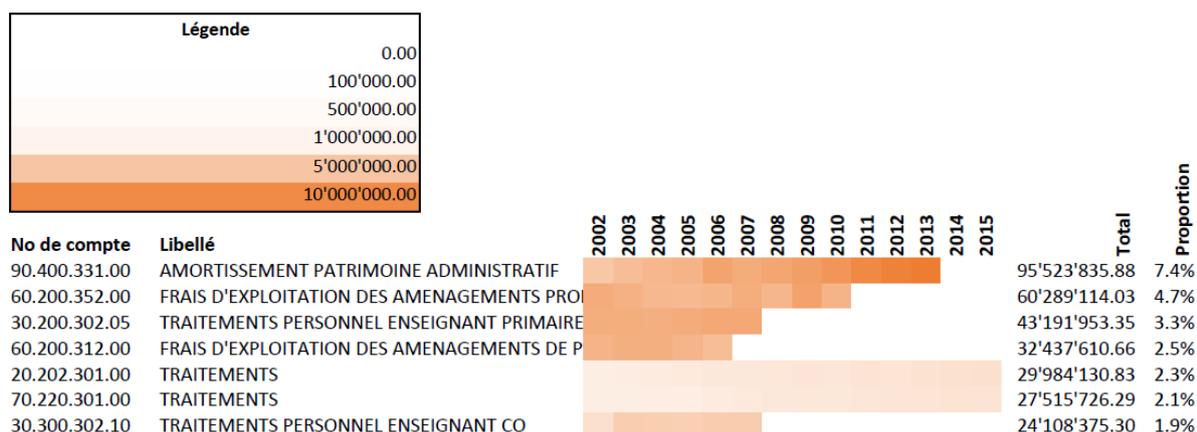


Figure 18 : Mock-up heatmap (Arnaud Zufferey & Florian Evéquoz)

Avec la *heatmap*, les cellules du tableau reçoivent une couleur d'arrière-plan selon la valeur de chaque cellule. L'intensité de la couleur représente selon la légende le montant de chaque compte pour chaque année.

Visual Compta

10.101.300.01	INDEMNITES DU CONSEIL	422'375	<div style="width: 100%;"></div>
10.101.300.04	INDEMNITES DU CONSEIL GENERAL	59'840	<div style="width: 14%;"></div>
10.101.303.00	CHARGES SOCIALES	51'867	<div style="width: 12%;"></div>
10.101.317.00	DEDOMMAGEMENTS	18'000	<div style="width: 4%;"></div>
10.101.300.02	INDEMNITES DES COMMISSIONS	10'710	<div style="width: 2.5%;"></div>
10.101.300.03	INDEMNITES DES SCRUTATEURS	7'965	<div style="width: 1.9%;"></div>
10.101.310.01	IMPRIMES ET FOURNITURES CONSEIL GENERAL	3'748	<div style="width: 0.9%;"></div>
Total		574505	

Figure 19 : Mock-up, Visual Compta (Arnaud Zufferey)

Dans la page Visual Compta, les comptes sont triés par valeur et sont visualisés sous forme de diagramme à barres. Le compte avec la plus grande valeur est indexé à 100 % et les autres comptes seront visualisés en fonction du premier compte.

Sankey

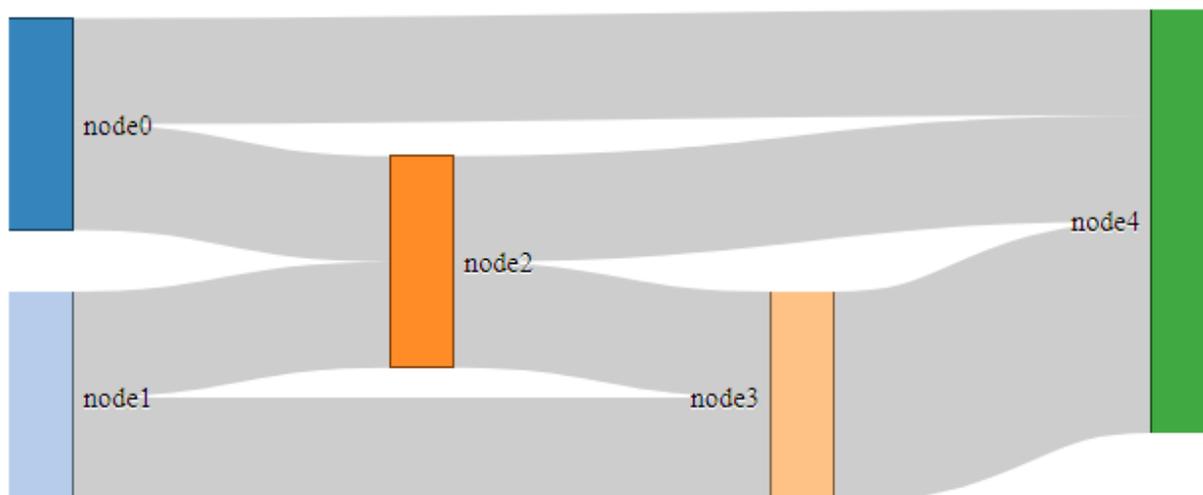


Figure 20 : Mock-up, Sankey (d3noob, 2017)

Le diagramme de Sankey affiche les charges et les revenus pour chaque service. Le résultat montre la pondération de chaque service et la différence entre charges et revenus.

Planification des investissements

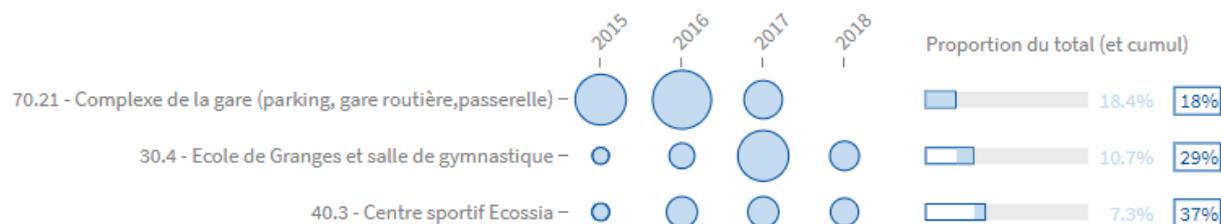


Figure 21 : Mock-up, plan des investissements (Arnaud Zufferey et Florian Evéquoz)

Le plan des investissements comprend les investissements majeurs sur quatre ans. Le montant de chaque compte est affiché avec une bulle par année. Plus une bulle est grande, plus l'investissement est élevé. Le pourcentage montre la somme des investissements de chaque compte par rapport au total des investissements.

3. Réalisation

3.1 Création du projet

3.1.1 Installation

Avant de pouvoir réaliser ce projet, les outils suivants ont été installés sur la machine du développeur :

- **Ampps** (serveur web, PHP, MySQL, phpMyAdmin)
- **JetBrains PHPStorm** (environnement de développement)
- **Composer** (gestionnaire de paquets libre)
- **Laravel** (framework PHP)

Pour créer un nouveau projet de Laravel en utilisant Composer, il suffit de saisir la commande suivante dans un interpréteur de commandes à la racine du serveur web :
composer create-project --prefer-dist Laravel/Laravel nomdusite

3.1.2 Structure du répertoire

Après avoir exécuté cette commande, un nouveau projet avec le *framework* Laravel est visible. Laravel organise toute la structure du répertoire. Voici une vue d'ensemble des dossiers les plus importants :

Dans le dossier « **app** » se trouvent tous les modèles, les contrôleurs, les *repositories* et les intergiciels. Toutes les configurations, comme le nom de l'application, la connexion de la base de données, l'adresse e-mail ou les sessions peuvent être paramétrés sous « **config** ». La navigation entre les différentes pages de la plateforme est réglée sous « **routes** ». Tous les fichiers CSS, JS, les données, les images et les polices se trouvent dans le dossier « **public** », tout comme le fichier `.htaccess`. Pour obtenir un affichage de la plateforme, les vues peuvent être créées et gérées dans « **resources** ». Tous les fichiers de traduction se trouvent également dans ce dossier. Les migrations de la base de données sont enregistrées dans le dossier « **database** ». Le *framework* Bootstrap est déjà intégré au dossier « **bootstrap** ».

3.1.3 Base de données et configuration de messagerie

Pour créer des configurations plus facilement, il existe un fichier « .env » dans Laravel pour les paramètres les plus importants. Ce fichier se trouve à la racine du projet.

La première configuration à faire consiste à saisir les paramètres de la base de données. Une base de données vide a été créée en avance directement dans phpMyAdmin. Pour ce projet, les configurations sont les suivantes :

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=fincom
DB_USERNAME=root
DB_PASSWORD=mysql
```

Les configurations du fichier .env sont intégrés dans le fichier approprié. S'il n'y a pas de spécification dans le fichier .env, Laravel utilise par défaut les informations déclarées après la variable :

```
'mysql' => [
    'driver' => 'mysql',
    'host' => env('DB_HOST', '127.0.0.1'),
    'port' => env('DB_PORT', '3306'),
    'database' => env('DB_DATABASE', 'forge'),
    'username' => env('DB_USERNAME', 'forge'),
    'password' => env('DB_PASSWORD', ''),
```

Avec cette configuration, la plateforme web est connectée avec la base de données.

Pour envoyer des e-mails par le biais de la plateforme, un compte Gmail a été créé. Laravel a besoin de connaître les informations de connexion de l'adresse e-mail. Ces informations sont également déposées dans le fichier .env.

```
MAIL_DRIVER=smtp
MAIL_HOST=smtp.gmail.com
MAIL_PORT=587
MAIL_USERNAME=finances.sierre@gmail.com
MAIL_PASSWORD=vds2017finance$
MAIL_ENCRYPTION=tls
```

La fonction des e-mails est utilisée pour réinitialiser le mot de passe d'un utilisateur, quand celui-ci oublie son mot de passe.

3.2 Modèle

Avant de créer les différentes pages, un *template* est défini. L'utilisation d'un *template* facilite la conception de la plateforme et assure une cohérence entre les différentes pages (Essique, 2017).

Blade est le moteur de *template* intégré dans Laravel. Les parties de la plateforme qui sont équivalentes sur tous les sites, par exemple la navigation, sont définies sur la page *template*. Pour interagir avec la page *template*, Blade utilise des directives. L'intégration de la page *template* sur un autre site s'effectue avec la directive « @extends('layouts.app') ».

Les parties qui sont différentes sont affichées dans la page *template* avec « @yield('content') ». Le contenu différent est créé dans chaque page entre les directives « @section('content') » et « @endsection ». La désignation entre les parenthèses définit les différentes sections. Le *template* créé dans ce projet comprend les sections contenues et script.

La figure 22 montre un exemple d'une page vide qui met en place le *template* créé avec Blade.

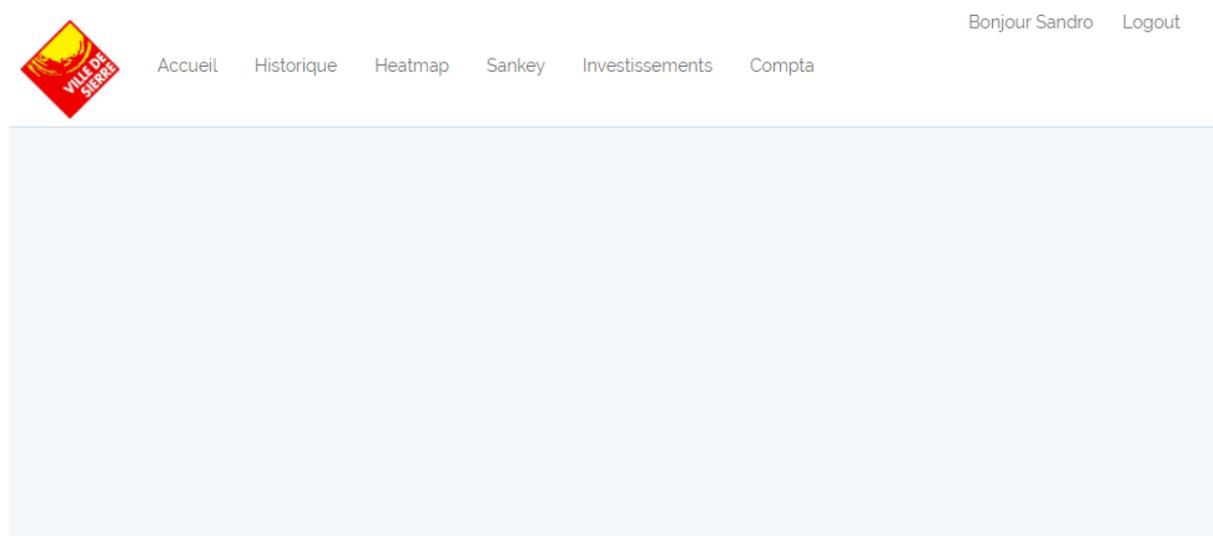


Figure 22 : Page vide avec le template (données de l'auteur)

3.3 Administration des utilisateurs

Pour éviter l'accès à la plateforme web par des personnes indésirables, l'accès est restreint en administrant les utilisateurs. Laravel permet cette fonctionnalité avec les deux commandes `PHP artisan make:auth` et `PHP artisan migrate` (Laravel, s.d.).

La première commande active l'authentification préfabriquée qui contient les pages de la connexion, l'enregistrement, le modèle pour les utilisateurs, ainsi que les contrôleurs et les intergiciels. La validation des données, ainsi que la vérification de l'utilisateur sont déjà prêtes l'emploi. Par ailleurs ; le traitement des mots de passe oubliés avec l'envoi d'un e-mail pour la réinitialisation du mot de passe ne nécessite que des changements mineurs. Les sessions et cookies sont même déjà intégrés. La deuxième commande est responsable de la migration de la base de données. Elle insère le tableau des utilisateurs et le tableau des mots de passe oubliés dans la base de données. En ajoutant un champ « level » aux utilisateurs, on empêche que chaque utilisateur puisse s'enregistrer et avoir directement un accès complet à la plateforme. Ce n'est que quand l'administrateur accepte un nouvel utilisateur que celui-ci peut voir les visualisations. L'administrateur est informé en cas de nouvel enregistrement en recevant une notification sur la page d'accueil. La figure 23 montre l'affichage du tableau de l'administration des utilisateurs avec les différentes options.

Liste des utilisateurs

Nom	Mail	Level	Changement de level	Supprimer
Sandro	sandromathier@hotmail.com	Admin	Enlever les droits administrateur	Supprimer
Sandro Mathier	sandro.mathier@students.hevs.ch	Membre	Donner les droits administrateur	Supprimer
Sandro2	sandro@mathier.ch	Membre	Donner les droits administrateur	Supprimer
Test User	user@test.ch	En attente	Accepter utilisateur	Supprimer

Figure 23 : Tableau de l'administration des utilisateurs (données de l'auteur)

Les fonctions derrière ces différents boutons ont été créées à l'aide d'un contrôleur de ressources RESTful. Avec cette aide, il suffit de déclarer un seul chemin d'accès pour traiter toutes les actions qui sont montrées dans le tableau 5 avec l'exemple d'une photo.

Tableau 5 : Actions traitées par le contrôleur de ressource (Laravel, s.d.)

Verb	Path	Action	Route Name
GET	/photo	index	photo.index
GET	/photo/create	create	photo.create
POST	/photo	store	photo.store
GET	/photo/{photo}	show	photo.show
GET	/photo/{photo}/edit	edit	photo.edit
PUT/PATCH	/photo/{photo}	update	photo.update
DELETE	/photo/{photo}	destroy	photo.destroy

Avec les différents niveaux des utilisateurs, on peut facilement contrôler l'accès aux différentes pages ou même aux fonctions dans les pages. L'exemple suivant montre la simplicité de la restriction d'accès avec Laravel :

```
Route::get('heatmap', ['middleware' => 'user', function() {
    return view('visualisations/heatmap');
}]);
```

Avec la spécification de l'intergiciel « user » dans le routage, l'accès à ce lien est réservé directement aux utilisateurs acceptés par l'administrateur.

3.4 Visualisations

La partie plus importante de ce travail était la réalisation de visualisations. Cinq visualisations différentes ont déjà été définies, avant la réalisation de ce travail. Les pages suivantes ont été réalisées selon les mock-ups et les prototypes fournis.

3.4.1 Historique

La première page permet aux utilisateurs d'afficher les évolutions des comptes de 2002 à 2016. Les données ont déjà été préparées sous forme d'un fichier JSON dans le prototype. Ce fichier JSON comprend une entrée pour chaque compte (voir l'exemple ci-dessous) avec le numéro et le nom du compte, les valeurs pour chaque année et un type. Le type représente s'il s'agit d'un compte de charges (type 0) ou un compte de revenus (type 1).

```
"CHARGES" : "10.101.300.01",  
"Col1" : "INDEMNITÉS DU CONSEIL",  
"type" : 0,  
"2002" : 337713.0,  
"2003" : 345947.0,  
"2004" : 354603.0,  
"2005" : 376100.0,  
"2006" : 390396.85,  
"2007" : 389560.0,  
"2008" : 392683.5,  
"2009" : 437375.0,  
"2010" : 430275.4,  
"2011" : 429631.05,  
"2012" : 422375.95,  
"2013" : 425722.15,  
"2014" : 421575.6,  
"2015" : 433405.05,  
"2016" : 441646.7
```

Pour voir le graphique, l'utilisateur doit d'abord sélectionner un compte ou plusieurs comptes dans le premier tableau et le transmettre dans le deuxième tableau avec le bouton correspondant. Pour trouver les comptes désirés plus rapidement, des fonctions permettent de trier et filtrer le tableau par numéro ou par libellé.

Ensuite, une visualisation sous forme de graphique linéaire va être créée automatiquement avec le(s) compte(s) sélectionné(s). L'axe Y est ajusté automatiquement selon les valeurs des comptes. Une infobulle avec le montant correspondant à l'année s'affiche dès que l'utilisateur place son curseur sur une ligne. L'utilisateur a également la possibilité de changer la période qu'il veut afficher dans le graphique. L'affichage peut être choisi entre les montants absolus et un pourcentage relatif basé sur la première année.

La capture d'écran suivante (figure 24) montre exemplairement une comparaison de deux comptes différents. L'évolution représentée montre clairement les différences de chaque compte par année et peut être analysée plus facilement par les membres de la commission finances que dans un tableau.

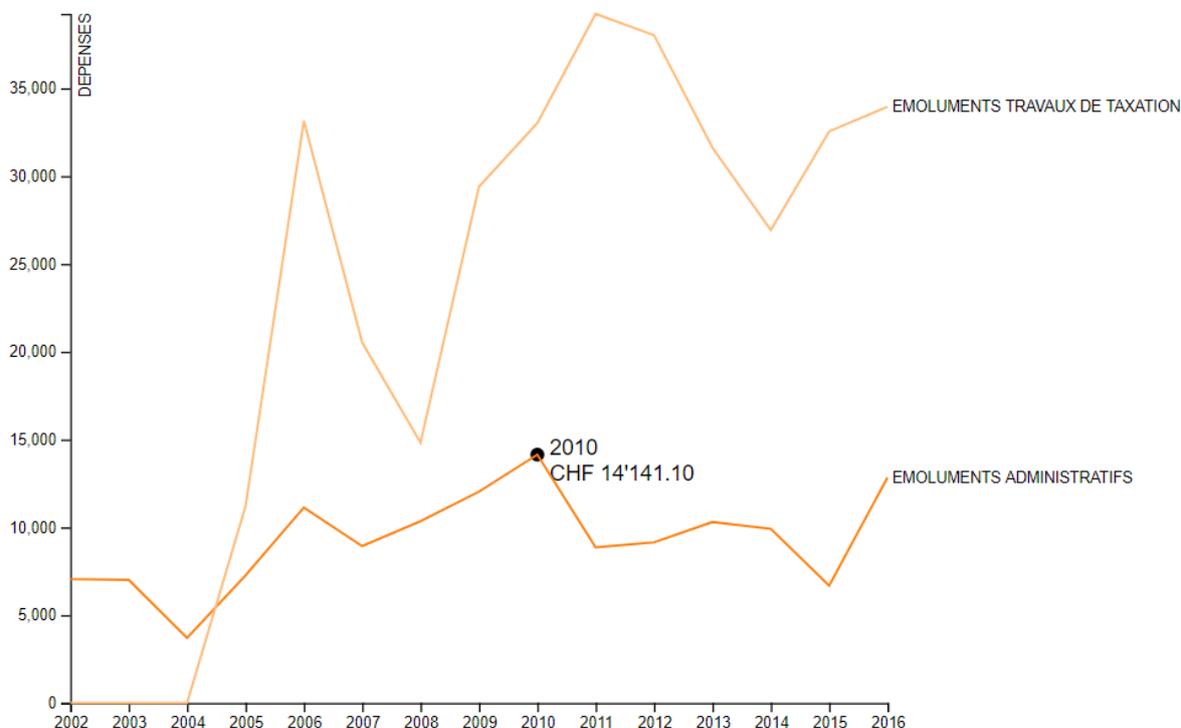


Figure 24 : Screenshot graphique évolution (Données de l'auteur)

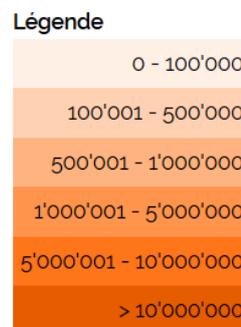
En même temps que le graphique est généré, le troisième tableau de cette page est rempli avec les indicateurs des comptes choisis. Ce tableau montre le minimum, le maximum et la moyenne des montants de chaque compte. En plus elle indique le taux de progression de la première à la dernière année.

Compte	Min	Max	Moyenne	Taux de progression
INDEMNITES DES COMMISSIONS	CHF 5'486.80	CHF 21'148.00	CHF 13'317.40	-69.42 %
INDEMNITES DES SCRUTATEURS	CHF 3'248.40	CHF 18'496.40	CHF 10'872.40	153.97 %
INDEMNITES DU CONSEIL	CHF 337'713.00	CHF 441'646.70	CHF 389'679.85	30.78 %

Figure 25 : tableau des indicateurs (données de l'auteur)

3.4.2 Heatmap

La page *heatmap* donne aux utilisateurs un tableau comportant les valeurs de tous les comptes des années 2002 à 2016. Puisqu'il s'agit des mêmes données que pour l'historique, les deux pages utilisent le même fichier avec les données en JSON. Au lieu d'afficher des nombres, les valeurs sont remplacées par des couleurs. Chaque couleur correspond à une valeur, selon la légende de la figure 26. L'intensité de la couleur augmente avec la valeur. Cet outil permet aux utilisateurs de détecter visuellement les différences entre les comptes et les années, en fonction de la couleur.



L'illustration 27 montre par exemple le compte 40.310.503.01 qui n'avait pas des valeurs élevées jusqu'en 2010. Il a ensuite augmenté jusqu'en 2014 et, depuis lors, la valeur descend de nouveau. Pour les communes, il est intéressant de voir s'il y a des écarts afin de pouvoir les expliquer. Dans ce cas, c'était la rénovation des Bains de Géronde qui était responsable de la différence.

Figure 26: Légende heatmap (Données de l'auteur)

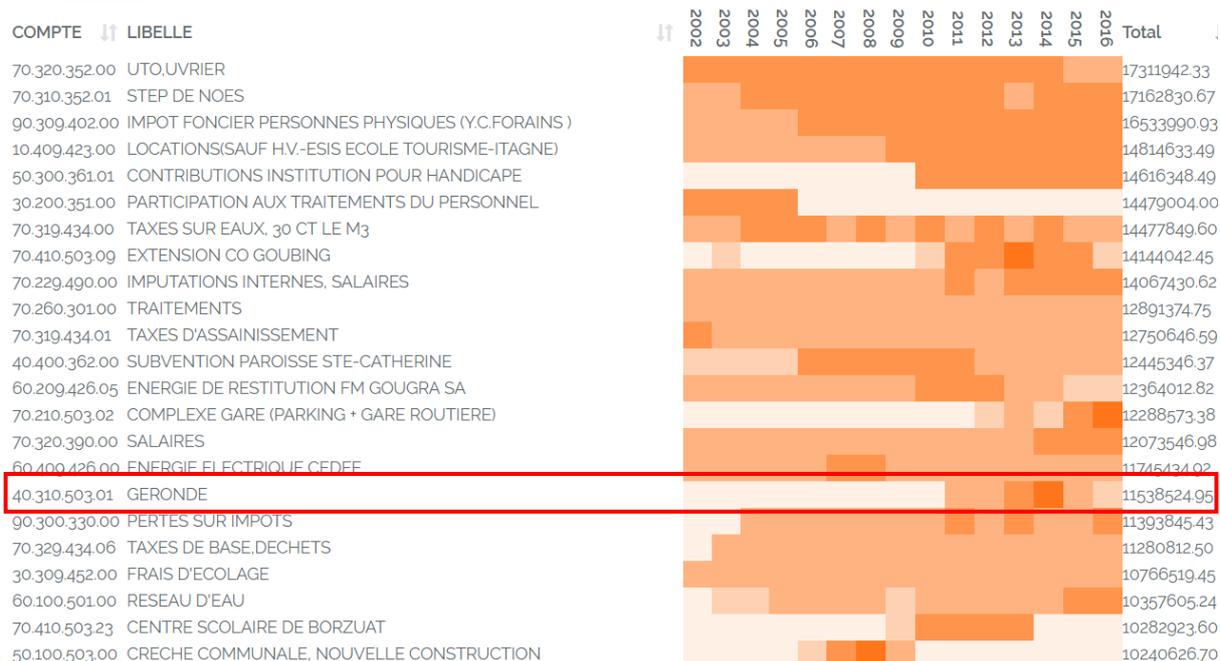


Figure 27 : capture d'écran heatmap (données de l'auteur)

3.4.3 Visual compta

À la différence des visualisations historique et *heatmap*, le *visual compta* ne comprend que les chiffres d'une seule année. Les données restent toujours les mêmes, sauf que, dans ce tableau, uniquement l'année 2016 est affichée. Les colonnes de ce tableau ne peuvent pas être triées, parce que les valeurs sont toujours triées par ordre décroissant. Ceci est important pour la visualisation sous forme de barres, parce que le premier compte représente toujours l'indice. La barre des comptes qui suivent est en fonction de part en pourcentage de l'indice. Pour cette raison la barre du premier compte est toujours pleine. Pour comparer les comptes d'une section, l'utilisateur peut mettre le numéro de la section dans le champ de recherche et le tableau va être mis à jour automatiquement.

La figure 28 montre cet exemple avec la section 70.320 - Élimination des ordures :



Figure 28 : Capture d'écran visual compta

3.4.4 Sankey

A Sankey diagram is a graphic illustration of flows, like energy, material or money flows. Usually the flows are illustrated as arrows. The width of the arrows is proportional to the size of the represented flow. Sankey diagrams are a better way to illustrate which flows represent advantages and what flows are responsible for waste and emissions. (Stenum GmbH, s.d.)

Pour la page Sankey, les données ne resteront plus les mêmes. Les données nécessaires existaient seulement au format Excel. Pour traiter les données sur la plateforme web, celles-ci ont été mises au format JSON en ligne avec les services du site suivant : <http://sankey-diagram-generator.acquireprocure.com/>. Le fichier récupéré est composé avec des nœuds et des liens. Chaque service à deux nœuds qui ont une liaison avec les charges et les recettes.

Pour afficher un diagramme de Sankey, le plug-in Sankey de D3 a été installé (<https://github.com/d3/d3-sankey>). Le diagramme représente les services de la Ville de Sierre en fonction de leur montant. Plus le montant du service est élevé, plus son rectangle est grand. L'illustration des relations des différents services permet aux utilisateurs de voir où la commune a les plus grandes dépenses et les revenus les plus élevés.

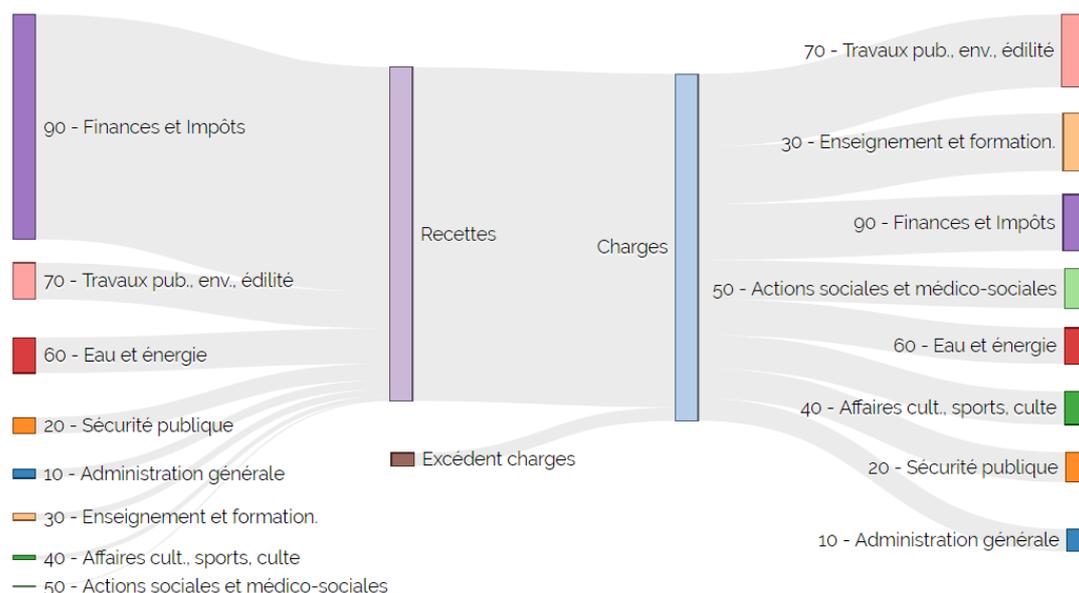


Figure 29 : Capture d'écran Sankey (données de l'auteur)

L'utilisateur peut également afficher un graphique net. Dans ce cas selon le montant le plus élevé, le service est soit du côté charges, soit du côté recettes.

3.4.5 Plan des investissements

Les comptes de fonctionnement qui ont été utilisés pour les visualisations des autres pages ne sont pas les mêmes que ceux pour la planification des investissements. Les données qui ont été fournies au format csv ont été passées au format JSON grâce aux services du site <http://www.convertcsv.com/csv-to-json.htm>. Elles contiennent les investissements prévus des années 2015 à 2018.

Le tableau de cette page montre une bulle par année pour chaque compte. La bulle représente le montant de l'investissement. Le total des investissements du compte en fonction du total de tous les investissements est indiqué par un pourcentage. Les pourcentages sont totalisés dans la dernière colonne. Étant donné que les comptes des investissements peuvent aussi comporter des recettes, les utilisateurs peuvent afficher un graphique qui les envisage également. Avec cette option, une deuxième bulle apparaît, en vert, dans les comptes concernés. Pour voir le montant exact correspondant à une bulle, il suffit de passer le curseur de la souris sur la bulle désirée, ce qui fait apparaître une infobulle.

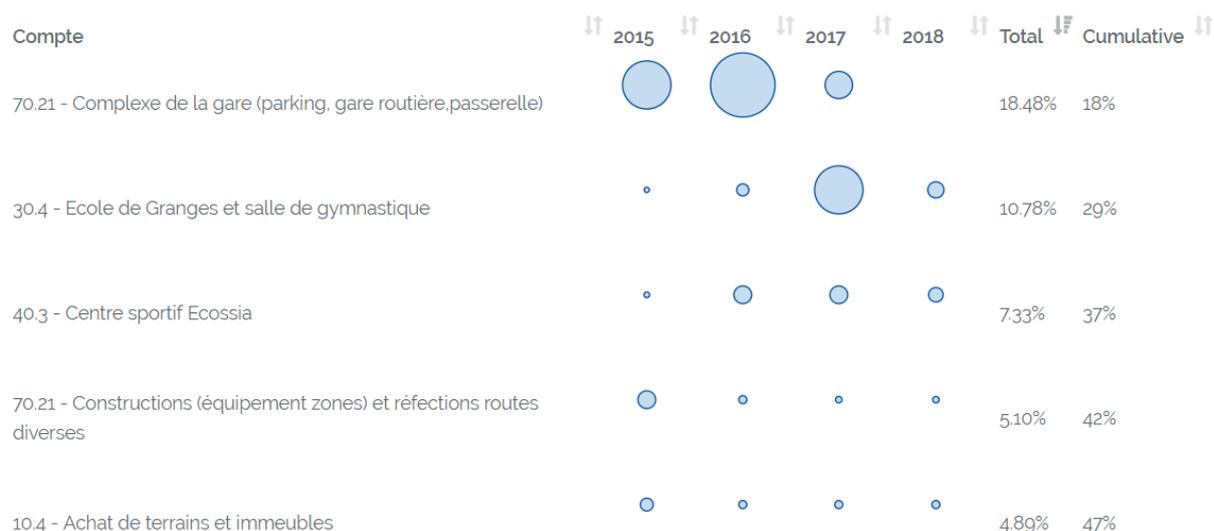


Figure 30 : capture d'écran, plan des investissements

3.5 Compatibilité des navigateurs

Pour obtenir un affichage correct et garantir la fonctionnalité de la plateforme, il faut savoir quel navigateur est utilisé chez les clients. Selon la plateforme « netmarketshare », le navigateur plus populaire est Google Chrome avec 59,49 %, suivi d'Internet Explorer avec 16,84 %, puis de Mozilla Firefox avec 12,02 % (NetMarketShare, 2017). Ces pourcentages montrent l'utilisation des navigateurs de bureau au niveau mondial.

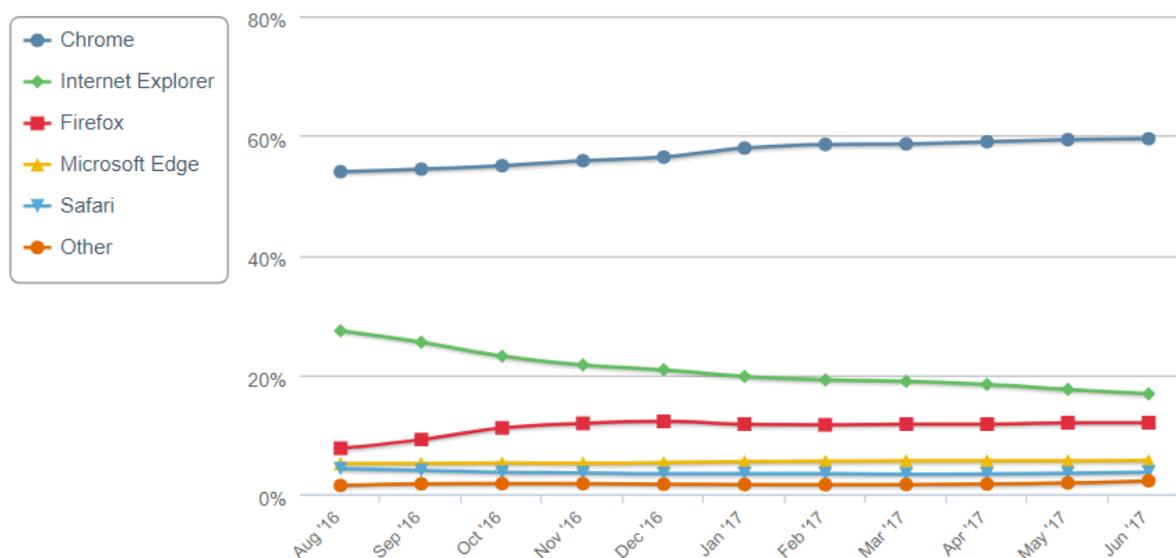


Figure 31 : Utilisations des navigateurs (NetMarketShare, 2017)

Avec les cinq navigateurs les plus populaires (Google Chrome, Internet Explorer, Mozilla Firefox, Microsoft Edge et Safari), on couvre 97,72 % de part de marché. La fonctionnalité et l'affichage de la plateforme web ont été optimisés pour l'ensemble des cinq navigateurs.

Selon les tests, la plateforme peut être utilisée sans restriction avec les navigateurs des versions suivantes :

- Google Chrome 59.0.3071.115
- Mozilla Firefox 54.0.1
- Microsoft Edge 38.14393.1066.0
- Safari 5.1.7 (7534.57.2)
- Internet Explorer 11.1480.14393.0

Les versions plus anciennes n'ont pas été prises en comptes, ce qui peut causer des discordances.

3.6 Multi-plateformes

En plus des différences entre les différents navigateurs, il faut également respecter l'importance des divers appareils mobile. Tandis que les mobiles ont déjà dépassé les ordinateurs de bureau au niveau du trafic Internet global, en Suisse, les ordinateurs de bureau ont encore une petite longueur d'avance selon le service « StatCounter » (StatCounter, 2017).

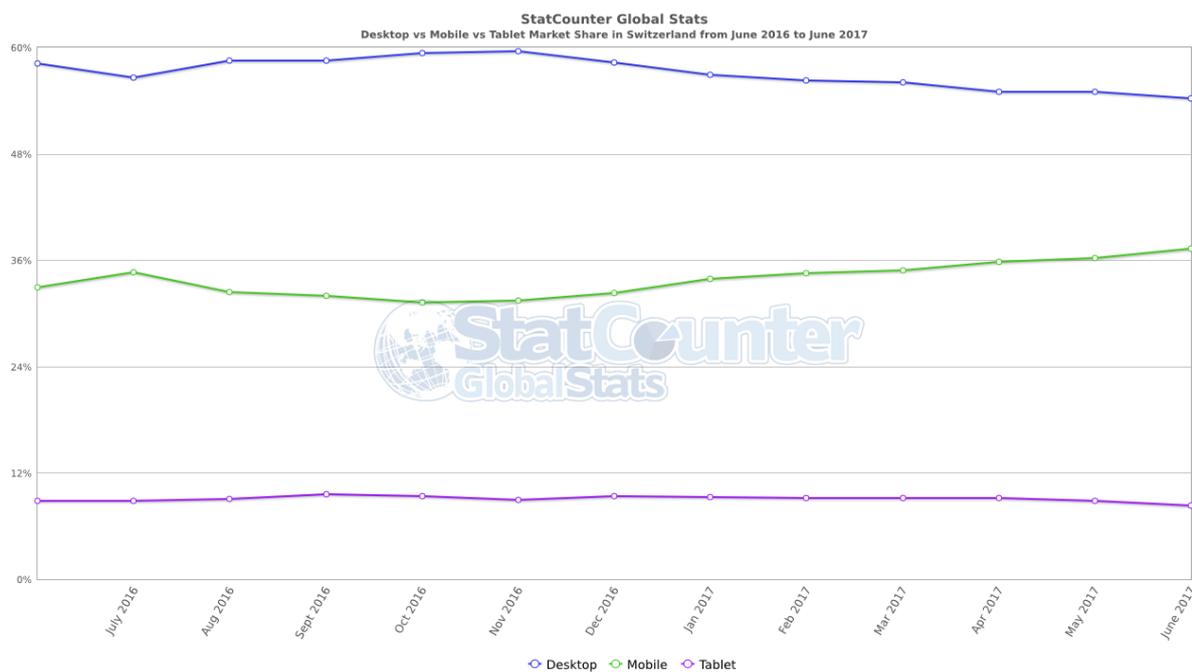


Figure 32 : Part de marché ordinateurs de bureau contre. mobiles contre tablettes (StatCounter, 2017)

Avec le *framework* Bootstrap, l'adaptation des différentes dimensions des écrans peut être gérée sans problème. Par contre, adapter l'affichage des tableaux et des visualisations à l'écran n'est utile qu'à un certain degré pour pouvoir voir tous les détails. Pour cette raison, une barre de défilement a été ajoutée. La navigation n'est plus visible en permanence en mode mobile, mais peut toujours être appelée avec un clic sur l'icône « ≡ ».

3.7 Localisation

Un objectif de ce travail était de prévoir le fonctionnement multilingue du site. Dans Laravel, ce fonctionnement est mis en place avec la localisation. Comme pour l'authentification, il suffit une ligne de commande pour installer cette partie. Par contre, des configurations supplémentaires étaient nécessaires, notamment à cause de la traduction des mots dans les fichiers JavaScript qui posaient un grand problème. Ce problème a pu être résolu avec l'implémentation du répertoire de Rubens Mariuzzo qui fait une copie lisible pour JavaScript des fichiers de langue (<https://github.com/rmariuzzo/Laravel-JS-Localization>).

Les données de la commune n'ont pas été traduites dans ce travail. Pour tous les autres textes et libellés, les traductions ont été faites avec un dossier par langue. Chaque dossier contient les mêmes fichiers avec les mêmes variables. Le changement de langue peut être effectué dans l'en-tête de page en cliquant sur l'abréviation de chaque langue (DE = allemand, FR = français). La figure 33 montre l'en-tête de page traduit en français et en allemand.

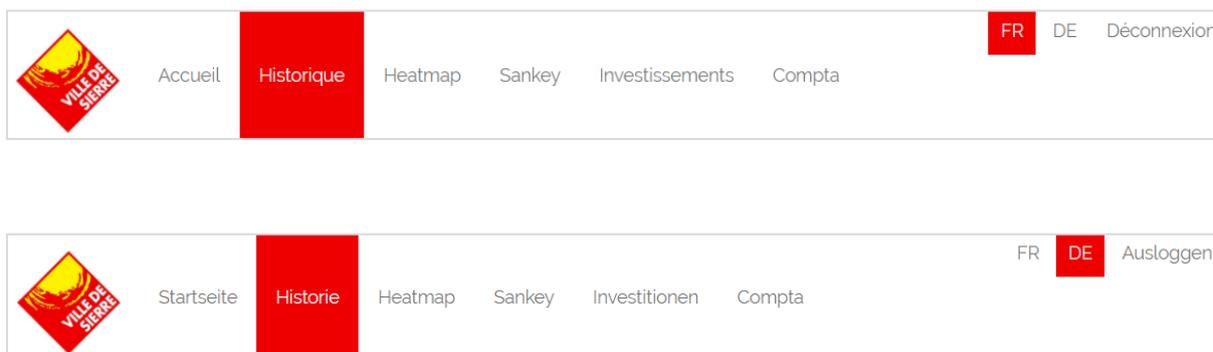


Figure 33 : en-tête de page en français et allemand (Données de l'auteur)

4. Scrum

4.1 Product Backlog avec les tâches

Pour la réalisation de ce travail, une version simplifiée de Scrum a été appliquée. Le tableau 6 montre le *product backlog* utilisé. Les *user stories* ont un fond bleu clair, tandis que toutes les autres lignes représentent des tâches sans avantage direct pour le client selon le PO.

Tableau 6 : Product backlog avec les tâches (Données de l'auteur)

US Nr.	Theme	User Stories			Acceptance Criteria	Priority	Status (1-3)	Story Points	Sprint	MoSCoW
		As an/a ...	I want to ...	so that ...						
1	Preparation	Developer	Prepare the work environment	I can make a release planning and I have a source control	Environment working	1000	3	3	0	Must
2	Analyze	Developer	Analyze the needs for the Municipality of Sierre	I know how the solution should look like and what the users should be able to do	Documented problem	900	3	5	0	Must
3	Analyze	Developer	Find the best fitting tools, language and libraries	I can easily develop a solution for the problem	Tools, language and libraries chosen and justified	800	3	13	0	Must
4	Analyze	Developer	Test different visualizations	I find the most meaningful visualizations	Documented visualizations	700	3	5	0	Must
5	Application	Developer	Draw mockups	I can have a better overview of the application	Documented mockups	650	3	3	0	Must
6	Application	User	Log in to the site	I can access to the platform	User can connect to the platform	600	3	5	1	Must
7	Application	User	have a navigation	I can choose the next action	User can navigate through the different views	500	3	5	1	Must
8	Application	Developer	be able to reuse the csv-data in the website	I can visualize them	Data integrated as json-files or inside a database	200	3	13	1	Must

9	Application	User	use the platform on mobile devices	I can access it from everywhere	Responsive website	100	3	3	2	Must
10	Application	User	see the different accounts in a table	I can see all the accounts	table with all accounts	90	3	8	2	Must
11	Application	User	filter / sort the table	I can show the accounts that I want in an order that I want	table is filterable / sortable	85	3	2	2	Must
12	Application	User	see a history diagram where I can select which Accounts I want to visualize	Compare multiple accounts	interactive history diagram	80	3	13	2	Must
13	Application	User	see the visualized investment plan	I know where the municipality invests in the next years	visualized investment plan	72	3	5	3	Must
14	Application	User	see a heatmap	I see for which accounts we spend the most money	interactive heatmap	76	3	8	3	Must
15	Application	User	see the visual compta	I have an ordered (by amount) list for each accounts group	interactive visual compta	74	3	5	3	Must
16	Application	User	see a sankey diagram	I can navigate through the accounts	interactive sankey diagram	78	33	13	4	Must
17	Application	User	be able to change the language	I can see the platform in my language	Multilingual platform	70	3	3	4	Must
18	Application	User	see a dashboard	I have an overview about the most important things	Dashboard with essential information	65	1	5	4	Could
19	Application	Admin	be able to add / remove accountings	I have the possibility to manage the different years	add / remove function implemented	60	1	13	4	Won't
20	Finishing	Developer	write all the documentation	I can give back all the required deliverables	Bachelor work accepted	40	3	13	5	Must

4.2 Sprints

Le travail a été divisé en plusieurs sprints. Dans chaque sprint, plusieurs *user stories* et tâches ont été effectuées. À la fin de chaque sprint, le PSPI a été présenté au PO.

Sprint 0 :

Dans le sprint 0, l'objectif était de comprendre la problématique et les besoins, de choisir les outils et les langages de programmation et de se faire une idée de la présentation du site avec des mock-ups. De plus, l'environnement de programmation a été préparé.

Sprint 1 :

Au sprint 1, la structure du squelette de la plateforme et l'administration des utilisateurs ont été créées.

Sprint 2 :

La première visualisation, c'est-à-dire l'historique, a été mise en place dans le sprint 2. Étant donné qu'elle servait de base aux autres visualisations, cette réalisation a pris de temps.

Sprint 3 :

Les visualisations suivantes, la heatmap et le visual compta, ont été réalisés dans l'itération du sprint 3.

Sprint 4 :

Dans le sprint 4, les dernières visualisations, le Sankey et la planification des investissements, ont été réalisés. De plus, le fonctionnement multilingue a été mis en place.

Sprint 5 :

Le dernier sprint servait à terminer la documentation et à achever le travail. Les dernières corrections ont été faites.

Conclusion

Résultats du travail

Pour ce travail, l'objectif était de réaliser une plateforme web donnant à la ville de Sierre un outil décisionnel pour ses finances. Cet outil est prévu pour être utilisé par les membres de la commission des finances.

Avant de réaliser cette plateforme, j'ai choisi les différents outils et langages de programmation. Après avoir analysé les différents outils et langages, j'ai décidé d'utiliser la combinaison suivante :

- **Backend** : Langage PHP/*framework* Laravel
- **Frontend** : Langages HTML, CSS et JavaScript/*frameworks* D3.js, DataTables et jQuery, Bootstrap
- **Base de données** : MySQL

Les visualisations élaborées permettent aux utilisateurs d'obtenir une simple vue d'ensemble des finances communales. Les utilisateurs peuvent interagir avec les différentes visualisations dynamiques, ce qui était plus difficile avec les graphiques Excel. Tandis que la solution existante se concentrait plutôt sur l'ensemble des comptes et des chiffres, la plateforme que j'ai réalisée permet également de récupérer les informations visuelles de chaque compte.

Scrum est une méthode très utile pour les projets informatiques, qui doit cependant être adaptée aux besoins des équipes de développement, en particulier le *sprint review*, c'est-à-dire le rendez-vous avec le PO à la fin de chaque sprint qui donne aux développeurs un *feedback* direct. Ceci évite d'avoir des points de vue différents.

Recommandations, limites et perspectives

Avant la mise en production de cette plateforme, je recommande de développer une fonctionnalité qui permettrait de récupérer les données d'une manière dynamique. Cela n'était pas le but de ce travail et n'a pas été désiré par le PO. Par contre, pour une utilisation productive, il doit être possible d'insérer de nouvelles données et de gérer les données existantes. Cette fonctionnalité serait également utile pour mettre cette solution à la disposition d'autres communes.

La plateforme web est limitée en raison de la thématique très vaste aux cinq visualisations présentées. Il est possible de développer d'autres visualisations si nécessaires. Les visualisations réalisées peuvent également être développées. Par exemple, on pourrait ajouter les sections et les comptes aux différents services dans le diagramme de Sankey.

Avec ces extensions supplémentaires, cette plateforme pourrait devenir un outil de décision assez important pour la ville de Sierre, ainsi pour les autres communes.

Sources

Ville de Sierre	www.sierre.ch
PHP	www.php.net
MySQL	www.mysql.com
Ampps	www.ampps.com
Composer	www.getcomposer.org
Laravel	www.laravel.com
D3.js	www.d3js.org
jQuery	www.jquery.com
Bootstrap	www.getbootstrap.com
DataTables	www.datatables.net

Références

Anscombe, F. J. (1973, février). Graphs in Statistical Analysis. *The American Statistician*, p. 5.

Récupéré sur Eageyeyes: <https://eageyeyes.org/criticism/anscombes-quartet>

Balsamiq Studios. (s.d.). *balsamiq*. Récupéré sur balsamiq: <https://balsamiq.com/>

Bootstrap. (s.d.). *getbootstrap*. Récupéré sur getbootstrap: <http://getbootstrap.com/>

Cohn, M. (2014, juin 6). *Mountain Goat Software*. Récupéré sur Mountain Goat Software:

<https://www.mountaingoatsoftware.com/presentations/an-introduction-to-scrum>

Cohn, M. (s.d.). *Mountain Goat Software*. Récupéré sur Mountain Goat Software:

<https://www.mountaingoatsoftware.com/agile/scrum>

Crochet-Damais, A. (2016, novembre 10). *Journal du Net (JDN)*. Récupéré sur Journal du Net:

<http://www.journaldunet.com/web-tech/developpeur/1152109-php-7-la-future-version-majeure-de-php-au-crible/>

D3.js. (s.d.). *d3js*. Récupéré sur d3js: <https://d3js.org/>

d3noob. (2017, mai 15). *Blocks*. Récupéré sur Blocks:

<https://blocks.org/d3noob/013054e8d7807dff76247b81b0e29030>

doktapepa. (2016, mars 18). *doktapepa*. Récupéré sur doktapepa:

<https://doktapepa.wordpress.com/2016/03/18/filtrer-par-colonne-dans-jquery-datatables/>

e-novations. (s.d.). *e-novations*. Récupéré sur e-novations:

<https://www.enovations.fr/clients-web-riches-1/>

Essique, H. (2017, juin 29). *anthedesign*. Récupéré sur anthedesign:

<https://www.anthedesign.fr/creation-de-sites-internet/creation-site-internet-template/>

Fernandez, A. (s.d.). *Piloter.org*. Récupéré sur Piloter.org: <http://www.piloter.org/business-intelligence/data-visualisation.htm>

Sandro Mathier

Garbade, D. M. (2016, juin 7). *opensource.com*. Récupéré sur *opensource.com*:
<https://opensource.com/business/16/6/which-php-framework-right-you>

Gaultier, D. (2016, février 9). *businessdecision*. Récupéré sur *businessdecision*:
<http://blog.businessdecision.com/bigdata/2016/02/methode-crisp-la-cle-de-la-reussite-en-data-science/>

Google Trends. (2017). *Google Trends*. Récupéré sur *Google Trends*:
<https://trends.google.de/trends/explore?cat=730&q=%2Fm%2F0jwy148,Yii,%2Fm%2F09t3sp,%2Fm%2F09cjcl,%2Fm%2F02qgdkj>

Kosara, R. (2011, février 13). *Eagereyes*. Récupéré sur *Eagereyes*:
<https://eagereyes.org/criticism/anscombes-quartet>

Laravel. (s.d.). *Laravel*. Récupéré sur *Laravel*:
<https://laravel.com/docs/5.1/controllers#restful-resource-controllers>

Laravel. (s.d.). *Laravel*. Récupéré sur *Laravel*: <https://laravel.com/docs/5.4/authentication>

MySQL. (s.d.). *MySQL*. Récupéré sur *MySQL*: <https://www.mysql.com/fr/why-mysql/>

NetMarketShare. (2017, juin). *NetMarketShare*. Récupéré sur *NetMarketShare*:
<https://www.netmarketshare.com/>

Office de Tourisme, S. (s.d.). *Sierre Tourisme*. Récupéré sur *Sierre Tourisme*:
<https://www.sierretourisme.ch/tourisme/sierre-capitale-suisse.html>

Parikh, R. (2014, avril 14). *Heapanalytics*. Récupéré sur *Heapanalytics*:
<https://blog.heapanalytics.com/how-to-lie-with-data-visualization/>

Phang, C. L. (2015). *Web Coding Bible*. Malaysia.

php. (s.d.). *php*. Récupéré sur *php*: <http://ch1.php.net/download-logos.php>

StatCounter. (2017, juin). *StatCounter*. Récupéré sur *StatCounter*:
<http://gs.statcounter.com/platform-market-share/desktop-mobile-tablet>

Sandro Mathier

Stenum GmbH. (s.d.). *Stenum*. Récupéré sur Stenum:
<http://www.stenum.at/en/?id=software/sankey/sankey-glossar>

Ville de Sierre. (2016). *Budget 2017*. Sierre.

Ville de Sierre. (2016). *Plan financier 2017-2020*. Sierre.

Ville de Sierre. (2017). *Comptes 2016*. Sierre.

W3C. (s.d.). *W3C*. Récupéré sur W3C: <https://www.w3.org/standards/webdesign/htmlcss>

W3Techs.com. (2017, juillet 23). *W3Techs.com*. Récupéré sur W3Techs.com:
https://w3techs.com/technologies/overview/programming_language/all

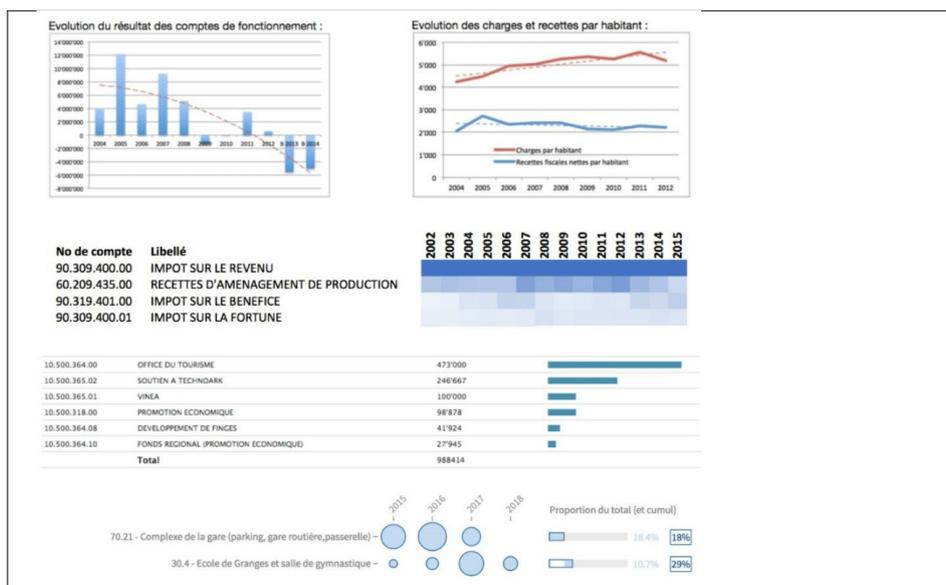
Annexe I : Cahier des charges

Filière Informatique de gestion
 Studiengang Wirtschaftsinformatik

Travail de Bachelor 2017

Plateforme web pour les finances communales

Cahier des charges



Etudiant : Sandro Mathier

Professeur : Jean-Pierre Rey

Table des matières

1	Description	1
2	Objectifs.....	2
3	Phases.....	3
3.1	Préparation	3
3.2	Analyse.....	3
3.3	Réalisation.....	3
3.4	Achèvement.....	3

Cahier des charges – Travail de Bachelor 2017 – Sandro Mathier

1 Description

Plusieurs visualisations innovantes pour les finances communales (sankey, heatmap, visual compta, bulles PIM, etc.) ont été esquissées ces dernières années à l'IIG (Florian Evequoz, Arnaud Zufferey). Il existe certains mock-ups, quelques proof of concept sous Excel et des bouts de code Javascript non documentés.

Le but de ce Travail de Bachelor est de regrouper et transformer ces idées en « web app » d'aide à la décision pour les finances communales.

Le travail vise d'être suffisamment abouti pour permettre à l'IIG de déposer une annonce d'invention et/ou de lancer un projet CTI avec un partenaire (comme T2i).

Cahier des charges – Travail de Bachelor 2017 – Sandro Mathier

2 Objectifs

Le but de ce projet consiste à développer une plateforme web qui permet aux utilisateurs d'avoir plusieurs visualisations interactives des finances communales pour leur donner un aperçu rapide et facile. Ces visualisations devraient les aider à prendre des décisions si nécessaires.

Les différentes visualisations prévues sont les suivantes :

- historique
- sankey
- heatmap
- visual compta
- bulles PIM

Pour accéder aux différentes visualisations, les utilisateurs peuvent utiliser la navigation de la plateforme. L'accès à la plateforme sera aussi possible avec des appareils mobiles.

Cahier des charges – Travail de Bachelor 2017 – Sandro Mathier

3 Phases

3.1 Préparation

Dans la première phase, le but est d'organiser les étapes du projet, de préparer les documents administratifs et de comprendre la problématique de la gestion des finances dans une commune, ainsi que ses besoins.

Livrables : Document "Données du travail de bachelor" signé, product backlog et cahier des charges

3.2 Analyse

La phase d'analyse est prévue pour analyser et choisir les outils (p. ex. Knime), les langages (PHP, JS, HTML5, ...) et les bibliothèques (D3, HighCharts, JQuery, Bootstrap, ...) les plus adaptés pour ce projet. Pour visualiser les finances communales, il faut déterminer les représentations les plus significatives.

Livrables : Les chapitres d'analyse dans le rapport

3.3 Réalisation

Après la phase d'analyse, nous allons développer la plateforme web. La réalisation est divisée en plusieurs sprints de 2 semaines. À la fin de chaque sprint, il y a une revue avec le product owner du produit (Arnaud Zufferey), à qui les progrès seront présentés.

Livrables : Les chapitres de chaque sprint dans le rapport et la plateforme web développée.

3.4 Achèvement

Dans la phase de finalisation les dernières corrections sont effectuées et le rapport est complété.

Livrables : Le rapport final et tous les fichiers du projet

Réalisé le 5 Mai 2017
Sandro Mathier

Validé le 15 Mai 2017
Jean-Pierre Rey



HES-SO Valais-Wallis • rue de la Plaine 2 • 3960 Sierre
+41 27 606 89 11 • info@hevs.ch • www.hevs.ch

3 | Page

Annexe II : Contenu du CD

Le CD joint au travail contient tous les fichiers de ce travail et le projet réalisé.

À la **racine** vous trouvez le rapport en PDF (TB_Mathier_Sandro_2017) et le poster de présentation de ce travail de Bachelor.

Dans le dossier **docs** se situent le product backlog, le planning et le journal de bord.

Le dossier **projet** contient le projet réalisé, les instructions pour l'installation du projet et un script SQL pour insérer la base de données.

Annexe III : Mock-up historique

Ville de Sierre - explorateur des finances

Veillez sélectionner les comptes dont vous voulez voir l'évolution :

 *filtre les lignes ci-dessous*

- [-] 10 Administration générale *datatables.net*
- [+] 10.101 Autorités
- [+] 10.102 Généralités
- [+] 10.200 Hôtel de Ville
- [+] 10.400 Cadastre
- [+] 10.500 Economie et tourisme

- [+] 20 Sécurité publique
- [+] 30 Enseignement et formation
- [+] 40 Affaires cult., sports, culte
- [+] 50 Actions sociales et médico-sociales
- [+] 60 Eau et énergie
- [+] 70 Travaux pub., env., édilité
- [+] 90 Finances et impôts

Comptes sélectionnés :

Glisser & déposer les comptes à comparer ici

Public cible : en priorité les membres de la commission des finances de la Ville de Sierre

Délai : 5 mars pour la 1^{re} version

Le numéro de compte peut être considéré comme identifiant unique.

Les libellés des comptes peuvent évoluer, trouver un moyen de donner l'historique.

Ville de Sierre - explorateur des finances

Veillez sélectionner les comptes dont vous voulez voir l'évolution :

 *le filtre peut se faire aussi sur le numéro*

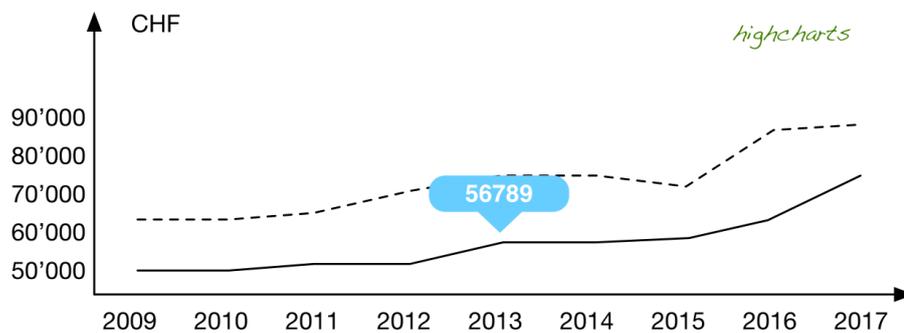
- 40.31 Piscine de Guillamo
- 40.310.312.01 Piscine, énergie
- 40.310.390.02 Piscine, consommation d'eau
- 40.319.434.00 Entrées piscine

Comptes sélectionnés :

- 40.25 Théâtre les Halles (charges)
- 40.30 Sports (charges)

Période de à

Affichage des valeurs : absolu (CHF) relatif (indice base 100)



Légende & indicateurs

40.25 Théâtre les Halles **+6%**
 40.30 Sports **+4%**

[Exporter CSV](#)

indicateurs : min, max, moyenne, tx progression..

Déclaration de l'auteur

Je déclare par ce document que j'ai effectué le travail de Bachelor ci-annexé seul, sans autre aide que celles dûment signalées dans les références et que je n'ai utilisé que les sources expressément mentionnées. Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe du responsable de la filière et du professeur chargé du suivi du travail de Bachelor, y compris au partenaire de recherche appliquée avec lequel j'ai collaboré, à l'exception des personnes qui m'ont fourni les principales informations nécessaires à la rédaction de ce travail et que je cite dans la partie « Remerciements » de ce document.

Salquenen, le 31 juillet 2017

Sandro Mathier