

**École polytechnique de Louvain**

# **HaïtiWater**

Développement d'une application web pour gérer  
la distribution de l'eau en Haïti

Auteurs: **Céline DEKNOP, Adrien HALLET , Sébastien STREBELLE**

Promoteurs: **Kim MENS, Sandra SOARES FRAZAO**

Lecteurs: **Benoît DUHOUX, Olivier CARLIER**

Année académique 2018-2019

Master [120] en sciences informatiques

# Table des matières

<b>Abstract</b>	<b>3</b>
<b>1 Introduction</b>	<b>4</b>
<b>2 Contexte</b>	<b>9</b>
2.1 Situation de l'eau en Haïti . . . . .	9
2.2 Gestion actuelle de la distribution d'eau en milieu rural . . . . .	12
2.3 Comparaison avec d'autres pays . . . . .	15
2.4 Comparaison avec des outils existants . . . . .	17
<b>3 Organisation</b>	<b>18</b>
3.1 Approche de travail . . . . .	18
3.2 Méthodologie . . . . .	22
3.3 Répartition des tâches . . . . .	24
<b>4 Analyse des besoins</b>	<b>28</b>
4.1 Besoins fonctionnels . . . . .	28
4.2 Besoins non-fonctionnels . . . . .	31
4.3 Cahier des charges . . . . .	32
<b>5 Implémentation</b>	<b>34</b>
5.1 Choix technologiques . . . . .	34
5.2 La hiérarchie dans l'application . . . . .	42
5.3 Interface utilisateur . . . . .	44
5.4 Client . . . . .	47
5.5 Serveur . . . . .	50
5.5.1 Requêtes . . . . .	50
5.5.2 Mapping Objet-Relationnel . . . . .	50
<b>6 Validation</b>	<b>52</b>
6.1 Vérifications automatiques . . . . .	52
6.2 Vérifications utilisateurs réels . . . . .	53
<b>7 Améliorations futures</b>	<b>57</b>
7.1 Suite du projet . . . . .	57
7.2 Défis rencontrés . . . . .	58
7.3 Propositions . . . . .	58
<b>8 Conclusion</b>	<b>59</b>
<b>Bibliographie</b>	<b>59</b>
<b>A Cahier des charges complet</b>	<b>62</b>

<b>E</b>	<b>Documents de validation</b>	<b>72</b>
E.1	Objectifs . . . . .	72
E.2	Protocole . . . . .	73
E.3	Attentes . . . . .	76

# Abstract

Page : 1

# Chapitre 1

## Introduction

### Contexte

Ce mémoire se place dans le cadre d'un projet de développement financé par l'ARES-CCD dans le cadre de ses projets de synergie. Les partenaires du projet sont L'ONG Protos<sup>1</sup>, L'UCL et l'UEH. Protos a pour objectif, entre autres, d'améliorer l'accès à l'eau potable en milieu rural afin d'aider le développement de plusieurs pays du monde. Un des pays dans lesquels Protos s'est engagée est Haïti.

Une succession de crises politiques et catastrophes naturelles ces dernières décennies ont rendu l'accès à l'eau potable, entre autres, particulièrement complexe dans ce pays des Antilles. En 2010, un violent séisme a laissé le pays en ruine, détruisant beaucoup d'infrastructures, y compris de distribution d'eau. Des incertitudes politiques entravent la reconstruction de ces installations et les populations ne sont pas toujours aidées par les services publics pour assurer la distribution de l'eau, particulièrement dans les zones rurales. C'est une des raisons pour lesquelles l'ONG Protos est active dans le pays.

Protos a contacté l'UCL afin de l'aider dans un projet de développement en Haïti. L'objectif est de réaliser un système logiciel pilote pour la gestion de la distribution d'eau potable en zone rurale. En effet, aucune gestion centralisée organisée par l'Etat n'existe pour ces zones, éloignées des grandes agglomérations. Des réseaux existent, constitués de points de prélèvement d'eau, de conduites de

---

1. [www.protos.ngo](http://www.protos.ngo)

distribution d'eau et de fontaines situées dans les villages, mais la gestion publique de ceux-ci n'est pas opérationnelle. Dès lors, les organismes locaux en charge de ces réseaux en assurent la gestion de manière indépendante, en s'appuyant comme ils le peuvent sur les moyens et les acteurs locaux. Une des principales conséquences est un taux de recouvrement extrêmement faible des factures (inférieur à 20% en moyenne).

L'ONG Protos intervient dans ce cadre, pour proposer un appui à ces organismes locaux afin de mieux organiser cette distribution. Grâce à une meilleure organisation, il sera possible d'améliorer le taux de recouvrement des factures liées à la distribution d'eau, ce qui permettra d'assurer des rentrées financières à ces organismes qui sont également en charge de la maintenance physique des réseaux, mais sans recevoir de réels moyens de la part de l'Etat.

Ce projet est prévu sur trois ans, une première année pour la conception d'un prototype, une deuxième année pour finaliser le système tester son déploiement sur place et une troisième année pour le finaliser. Ce mémoire constitue la première partie de ce projet.

## Problématiques

La structure hiérarchique des différents acteurs de la distribution de l'eau, gouvernementaux ou non, est assez complexe. Ces acteurs doivent s'organiser, individuellement et en groupes, pour arriver à une gestion efficace des ressources. Dans ce mémoire, nous nous concentrons sur trois axes afin d'améliorer la coordination entre ces acteurs.

**Communication** afin de permettre aux différents acteurs de s'informer à partir de la même plateforme.

**Collaboration** pour uniformiser le format des données et donner un rôle défini à chacun.

**Stockage** de manière à rassembler l'information sous forme numérique, permettant une meilleure sauvegarde des données, et à terme de les traiter en large volume de manière statistique.

## Motivation

Nous réalisons ce travail dans le cadre de nos études de Master en Sciences Informatiques. Le but de ce travail est de conclure celles-ci, et de mettre tous nos apprentissages en application dans un projet à grande échelle.

De plus, il s'agit d'un projet réel, avec de véritables acteurs, enjeux et un objectif d'utilisation à long terme. Cela entraîne de nouvelles problématiques pour nous, qui ne sont pas abordées dans le reste de notre formation.

Ce mémoire nous permet d'avoir une première expérience de développement d'une application web, partant de rien sauf des attentes de nos clients, et intégrant toutes les parties et les étapes de son développement.

Non contents de parfaire notre formation, nous avons pour objectif d'être utiles à Haïti. Nous espérons que l'application développée dans le cadre de ce mémoire pourra être utilisée sur place et avoir un réel impact positif, aidant la distribution de l'eau à Haïti et son développement.

## Objectif

Le but final de ce mémoire est de délivrer une application à l'ONG Protos et aux acteurs de la distribution de l'eau à Haïti. Nous visons un déploiement de cette base applicative sur le terrain, et que les futures équipes de développement pourront travailler à sa maintenance et évolution.

Nous espérons également que cette application permettra à Protos comme aux Haïtiens d'avoir un bon exemple de solution logicielle. Cela pourrait les aider à voir comment un logiciel peut être un appui dans leur gestion, et leur permettre de bien repenser leurs besoins lors de projets futurs.

## Approche

Avant de commencer le développement de l'application, nous avons effectué une longue phase de recherches. Cette phase est séparée en deux parties.

Durant la première partie, nous avons observé ce qu'il se faisait dans les systèmes existants de la gestion de l'eau en Europe pour avoir une meilleure vue d'ensemble du travail à accomplir et des possibilités. En seconde partie, nous avons analysé les documents fournis par l'ONG Protos afin d'avoir une idée de la solution actuellement déployée en Haïti et des améliorations possibles.

La phase de développement suivante fut l'analyse fonctionnelle et la conception. Nous avons établi nos propositions en termes de fonctionnalités pour l'application, ainsi qu'en termes de fenêtres et contrôles pour les présenter à Protos sous forme d'un cahier des charges.

Après validation de l'interface et des fonctionnalités, nous sommes passés à la phase de réalisation, durant laquelle nous avons implémenté ces fonctionnalités et les systèmes nécessaires à leur bon fonctionnement. Cette phase a été la plus longue étant donné le nombre de fonctionnalités à implémenter et la nécessité de documenter à la fois le code pour les équipes de développement ainsi que l'interface pour nos futurs utilisateurs.

Enfin, la validation. Nous avons présenté l'application et les fonctionnalités que nous avons implémentées à des utilisateurs, afin d'obtenir des retours et de nous assurer que l'application que nous avons développée répondait aux attentes.

Au cours de la phase de réalisation, nous avons pu collaborer avec un stagiaire venant d'Haïti. Cela nous a permis de comprendre plus directement les besoins sur place, complétant ainsi les informations plus théoriques que nous avions obtenues des documents de Protos. Cette vision plus pratique nous a permis d'avoir une meilleure idée de la réalité du terrain.

## Contribution

Grâce à notre travail de fin d'études, nous avons permis à l'ONG Protos ainsi qu'aux acteurs de terrain en Haïti d'avoir un premier contact avec le monde du



développement logiciel. Nous espérons que celui-ci les aidera à comprendre les problématiques liées à la création de logiciels informatiques pour leurs projets futurs.

Nous avons également proposé une application pour la gestion de l'eau potable à Haïti. Nous espérons qu'elle pourra servir de base de travail, permettant aux acteurs de terrain d'avoir un jour un outil les aidant dans leur travail quotidien et apportant une aide au développement national.

Nous espérons avoir contribué à l'amélioration de la gestion de l'eau en Haïti, mais ce travail a assurément contribué au développement de nos compétences ; en tant que développeurs, de par l'utilisation de librairies et frameworks peu ou pas connus, en tant qu'analystes grâce au problème réel posé et à la nécessité de comprendre les besoins. La coordination du groupe, la planification du travail et la communication avec les clients ont également beaucoup apporté à notre développement professionnel.

## **Plan**

Dans ce document, nous allons exposer le déroulement de notre travail de fin d'études. Tout d'abord, nous allons revenir sur le contexte de ce travail. Nous allons ensuite détailler notre approche pour sa réalisation. Après cela, nous étayerons notre analyse des besoins de l'ONG Protos et des acteurs haïtiens. Ensuite, nous expliquerons toutes les étapes et décisions de notre implémentation. Nous reviendrons sur les techniques que nous avons utilisées afin de valider notre implémentation. Nous clôturerons ce mémoire par une rétrospective sur le travail et une ouverture aux possibilités futures.

# Chapitre 2

## Contexte

### 2.1 Situation de l'eau en Haïti

Haïti est un des pays les plus pauvres de la planète, qui peine à se remettre du séisme de 2010 qui a ravagé le pays et en particulier sa capitale, Port-au-Prince. Éducation, économie, habitat, etc, sont autant de défis que le pays doit relever. L'eau potable, son assainissement, sa gestion et sa distribution font partie de ces défis, en particulier dans les zones rurales. Les zones d'action de l'ONG Protos sont particulièrement touchées. La majorité des familles s'approvisionne en eau à des fontaines publiques (96% des foyers) et effectue une marche de plus de 30 minutes (58% des foyers). Il est de plus extrêmement difficile de maintenir et développer le réseau de distribution d'eau potable dans certaines régions en raison de la pauvreté extrême d'une grande partie de l'île. La zone étudiée de *Passe Catabois* (au nord du pays), par exemple, a un taux de recouvrement des factures de 11% seulement. Si la situation est complexe, il est toutefois possible d'identifier des facteurs expliquant l'état, entre autres, des infrastructures d'eau potable.

### Problèmes environnementaux

Située dans les Caraïbes au sud-est de la République de Cuba et des États-Unis d'Amérique, la République d'Haïti occupe un territoire de  $27.750km^2$  sur la partie occidentale de l'île d'Hispaniola. Le pays entier souffre de sa position

géographique [8] qui le place dans un risque permanent de catastrophes naturelles. La proximité de failles tectoniques et de courants aériens et marins favorisant la formation de tempêtes tropicales produisent une succession de cataclysmes. De multiples tempêtes et séismes frappent l'île fréquemment. Le tristement célèbre tremblement de terre du 12 janvier 2010, dont le nombre de victimes est approximé à 250.000 morts et autant de blessés, a, à lui seul, causé des dégâts estimés de 8 à 14 milliards USD [5]. Ces catastrophes périodiques détruisent les infrastructures et mettent à mal l'économie du pays, empêchant notamment un développement du réseau d'eau potable suffisant pour répondre aux besoins.

Le climat haïtien n'aide pas la situation. Météorologiquement parlant, Haïti affronte une alternance entre saison très chaude et saison moins chaude, les deux provoquant fréquemment des sécheresses. Les différentes régions du pays ont des climats variés, obligeant l'adoption de plusieurs mesures pour sauvegarder la population et rend difficile une politique commune de gestion des ressources. Il convient de noter que l'aridité qui résulte de ce climat est accentuée par les activités humaines. La déforestation massive et rapide du pays, à des fins principalement énergétiques, empêche une rétention correcte des eaux dans les sols. Seuls 2% à 3% du territoire sont occupés par des forêts (venant de plus de 60% il y a un siècle). La sécheresse se trouve ainsi aggravée, et le ruissellement des eaux accélère l'érosion des sols [24]. Les terres sont ainsi progressivement emportées vers la mer. En 2008, un rapport des Nations Unies estimait cette perte à 37 millions de tonnes par an [15]. Les cours d'eaux ont donc un débit accéléré, parfois torrentiel, rendant difficile l'exploitation de l'écoulement naturel et nécessitant plus d'infrastructures pour obtenir de l'eau potable.

## **Problèmes politiques**

Pays colonisé français devenu indépendant en 1804, la République d'Haïti a connu plusieurs régimes politiques différents, dont une dictature qui s'est terminée en 1986. Depuis lors, le pays oscille politiquement jusqu'à une instabilité contemporaine relative (on dénombre 15 présidents et 22 premiers ministres depuis 1986 alors que le mandat est constitutionnellement valable 5 ans) et des tensions internes continuent de perturber le pays. Cela entraîne une difficulté à adopter des politiques de long terme, ou même une politique commune aux différents gouvernements.

La scène internationale s'est très largement mobilisée pour Haïti et tout particulièrement après le séisme de 2010. De nombreuses organisations humanitaires et

non-gouvernementales sont arrivées, mais les aides financières sont soupçonnées de détournement par ces mêmes organisations [7]. Ces aides sont parfois décriées dans la dépendance qu'elles créent et le manque de vision à long terme. La population s'est également déjà retournée contre les aides et notamment contre les casques bleus venu aider après le séisme, et toujours présents à l'heure actuelle. Des soldats népalais des Nations Unies, venus aider après le drame, auraient involontairement déclenché une grave épidémie de choléra en introduisant la maladie en tant que porteurs sains. Le *Center for Disease Control and Prevention (CDC)*, un service du département de la santé des États-Unis d'Amérique, a confirmé cette hypothèse et les Nations Unies ont reconnu leur responsabilité en décembre 2016. L'épidémie a causé 10.000 morts pour 800.000 cas dans les Caraïbes, majoritairement en Haïti. Cette maladie a été introduite par des rejets d'eaux usées dans un cours d'eau, par négligence et manque d'installations permettant d'évacuer les eaux usées dans un campement d'aide humanitaire.

La République d'Haïti entretient également des relations conflictuelles avec la République Dominicaine, avec laquelle elle partage l'île d'Hispaniola. Les raisons sont multiples (travailleurs étrangers, tensions ethniques, etc) et un lourd passé pèse sur la relation entre les deux pays.

Nous voyons donc que la situation politique intérieure et internationale haïtienne est un cas complexe. Les diverses tensions et soupçons de corruption ne permettent certainement pas de donner à la problématique de l'eau toute l'attention, ni tous les fonds nécessaires.

## Problèmes sociaux

De ces conditions environnementales et politiques émane une population pauvre, voire abandonnée dans les zones rurales où des conseils de village tentent de pallier les problèmes au quotidien. Là où les villes et principalement la capitale profitent des avantages de la mondialisation, la campagne haïtienne peine à bénéficier de l'aide nationale et son développement repose essentiellement sur les organisations locales et externes au pays. L'éducation y est difficilement accessible et la population est très majoritairement endettée (81% des foyers haïtiens), situation aggravée depuis 2015 lorsque la monnaie locale, la Gourde (HTG) a cessé d'être maintenue à un taux de change fixe avec le Dollar (USD). La Gourde continue depuis sa dépréciation (1 Euro vaut 93,5445 Gourdes<sup>1</sup>). La société haïtienne est donc pauvre et les inégalités nombreuses. La campagne en particulier, qui semble déjà oubliée financièrement, manque de techniciens et ingénieurs aptes à créer et maintenir un réseau de distribution d'eau potable avancé.

---

1. Taux de change du marché au 28 mars 2019

## Problèmes organisationnels

Le modèle politique et organisationnel d'Haïti se calque sur la théorie occidentale. Cependant les profondes différences résultent en une mise en pratique des concepts qui ne sont pas toujours efficaces. On constate notamment un manque de collaboration entre les entités haïtiennes ou même entre villages. Particulièrement dans les zones rurales, les croyances locales (religions, magie) sont parfois consultées pour des décisions importantes au détriment de la rationalité. La communication entre les différentes parties du pays n'est pas optimale. Les entités gouvernementales sont parfois nommées sans moyens financiers ni même bureaux et se retrouvent désœuvrées. Les fréquentes crises détournent l'attention des problèmes de fond tels que l'éducation, les infrastructures, ...

## 2.2 Gestion actuelle de la distribution d'eau en milieu rural

### Organisation générale

Comme exposé dans la section précédente, la situation environnementale, politique et sociale, cause une instabilité dans l'organisation des services publics d'Haïti. Les présidents changent souvent, et avec eux, les mesures mises en place. Les responsabilités des ministères et de leurs sous-entités sont assez peu claires, ou impossibles à mettre en pratique.

Par conséquent, nos lectures des différents documents fournis par l'ONG Protos nous a démontré qu'elle est sans doute l'une des entités les plus stables au niveau de l'organisation de la gestion de l'eau dans la région visée par ce travail. Elle supervise plusieurs organisations locales, en apportant son soutien (expérience, logistique, finances), tandis que ce sont les localités elles-mêmes qui tentent de solutionner les problèmes.

Chaque zone est traitée indépendamment (elle dispose de son ONG locale), mais on distingue tout de même trois groupes d'acteurs récurrents :

- Le triangle formé par les autorités locales, les fournisseurs et les usagers
- Le gouvernement national
- Les autres ONG (locales ou non)

Protos se repose beaucoup sur des entités locales, notamment les services étatiques, comme l'Université d'État d'Haïti (qui aura une place dans ce projet à l'avenir). Également, elle tend à être la plus inclusive possible, et à impliquer un maximum d'acteurs (locaux) dans ses projets. L'idée est que les citoyens instruits servent de bon exemple pour répandre les avancées sociales et/ou technologiques à travers le pays. Selon eux, pour avoir un système fonctionnel, la conscientisation sur la problématique de l'eau devrait partir des habitants eux-même avant de remonter vers l'État.

Actuellement, Protos agit dans deux zones du pays : Nord-Est (région dite "Moustiques") et Centre-Est (région dite "Onde Verte"). Celles-ci font partie des plus rurales, et ont été choisies car ce sont celles qui ont le plus besoin d'aide.

## **Structure organisationnelle**

En plus de l'aide fournie par l'ONG Protos, il existe en Haïti plusieurs structures dédiées à la gestion de l'eau. Premièrement, la DINEPA (Direction Nationale de l'Eau Potable et de l'Assainissement), qui a pour mission d'exécuter la politique de l'État dans le secteur de l'eau potable et de l'assainissement autour de trois grands axes : le développement du secteur au niveau national, la régulation du secteur et le contrôle des acteurs [1]. La DINEPA est sous contrôle direct du Ministère de Travaux Publics, Transports et Communication.

La DINEPA supervise des branches régionales appelées OREPA (Offices Régionaux d'Eau Potable et d'Assainissement), gérant chacune une zone du pays. Elles sont chargées de la prestation du service de distribution de l'eau potable en milieu urbain et rural.

Finalement, il existe des CAEPA (Comités d'Approvisionnement en Eau Potable et d'Assainissement), qui gèrent un réseau de distribution d'eau à un niveau local réduit. Ils sont moins formellement gérés par l'État et sont encore loin d'être en place sur tout le pays, mais ont prouvé leur efficacité. Surtout dans les régions rurales, les habitants se reposent bien plus sur les CAEPA pour leur approvisionnement en eau que sur d'autres entités privées ou publiques.

## Procédures actuelles

On peut remarquer qu'il n'y a pas vraiment de standardisation dans les procédures en place au niveau des CAEPA. En effet, un comité de gestion de l'eau existe à Poste-Métier depuis plus de 20 ans, mais il n'est présent que grâce à l'implication du Pasteur Chrisbon Fraïs et à l'aide de Protos. Cet effort s'étant montré efficace, le système établi à Poste-Métier a été appliqué dans d'autres zones, avec plus ou moins de succès.

Le principal problème rencontré par les comités de l'eau est le recouvrement des factures. En effet, on constate que les habitants d'Haïti n'ont pas l'habitude d'avoir à payer pour l'eau, et rechignent à le faire. Premièrement parce qu'ils ne l'ont jamais fait, mais également car la religion a encore une place forte dans la société haïtienne et l'eau y est considérée comme *don de Dieu*. Protos met en avant qu'une part essentielle du processus de stabilisation du réseau de distribution est l'éducation des habitants, pour qu'ils comprennent qu'ils ne payent pas l'eau elle-même, mais bien tout le service fourni par le réseau pour l'assainissement, le transport et la distribution.

Concrètement, si la zone de Poste Métier a un taux de recouvrement satisfaisant (89%), les autres zones comme Passe Catabois sont bien en deça (11%). Ceci est dû à plusieurs éléments : les habitants sont mieux informés sur la problématique de l'eau, mais les gestionnaires de fontaines ont aussi le pouvoir de couper l'accès à l'eau (à l'aide de cadenas) quand les factures ne sont pas payées. On remarque également que les modes de fonctionnements des différentes zones ne sont pas toujours les mêmes (pas toujours de bureau pour les réunions, règlements plus ou moins stricts).

En pratique, un comité de l'eau (*Komité Dlo* en Créole haïtien) est composé d'un président, un vice-président, un secrétaire, un trésorier et un conseiller, élus par les consommateurs locaux pour cinq ans. Ils se réunissent tous les mois pour parler des affaires courantes, et organisent une fois par an une assemblée générale où sont conviés les usagers du réseau. Ils sont en charge du bon fonctionnement du réseau, de la gestion du budget et des fonds amenés par les utilisateurs, de superviser les *comités fontaines* et de rendre des comptes à la population.

Les comités fontaines, eux, sont composés d'un responsable fontaine, un secrétaire et un trésorier. Ils sont en charge du bon fonctionnement et de la qualité du point d'eau, d'apporter les fonds au comité d'eau et de gérer les éventuels conflits entre usagers ou les réparations nécessaires.

Malgré la structure hiérarchique décrite ci-dessus, ces organismes ont encore du mal à effectuer la gestion du réseau au jour le jour. Ils utilisent encore les outils les plus basiques : un crayon et du papier. Cette situation rend l'accès aux informations antérieures difficile voire impossible. Il est également difficile de transmettre ces informations à la hiérarchie supérieure.

L'accès aux fontaines publiques est accordé après le règlement d'un paiement annuel. Les consommateurs se voient attribuer une carte spécifiant la somme, le lieu et la date. Cette carte est utilisée comme preuve de paiement pour accéder aux fontaines. Cependant, il n'est pas possible pour le responsable fontaine de se souvenir exactement qui a payé ou non et les informations se perdent. De plus, certains abus (distribution non-équitable, prise d'eau sans paiement) ont été constatés.

## **2.3 Comparaison avec d'autres pays**

La situation de l'eau dans d'autres pays, en particulier ceux aux économies fortes et établies, est différente. Le contexte d'Haïti est évidemment particulier, comme nous avons pu le voir dans les sections précédents. Ces circonstances nécessitent des solutions adaptées pour gérer la distribution d'eau. Néanmoins, une analyse de la situation d'autres pays a été utile afin de découvrir d'autres manières de faire, identifier les bonnes pratiques du milieu ainsi que les différences avec la situation haïtienne afin de bénéficier de l'expérience des organismes existants.

### **Gestion de l'eau en Belgique**

Nous avons d'abord étudié la situation de la distribution de l'eau en Belgique, choix naturel de par la localisation de nos études et du développement de ce mémoire. La première chose importante est que l'eau est une compétence régionale en Belgique, ce qui signifie que chaque région (Flandre, Wallonie et Bruxelles) développe sa propre politique à propos de l'eau.

Par exemple, en Wallonie, la gestion de l'eau est déléguée par le gouvernement wallon à la SPGE, Société Publique de la Gestion de l'Eau. Il s'agit d'une société publique, dont le fonctionnement est décrit par un contrat de gestion établi par le gouvernement. Elle travaille avec différentes intercommunales, sociétés également



publiques, pour s'assurer de la distribution d'eau potable, ainsi que de l'égouttage et l'assainissement des eaux usées. Ces différentes intercommunales sont mandatées par la SPGE via des contrats de gestions sur des parties spécifiques du territoire wallon.

On peut voir que contrairement à Haïti, la gestion de l'eau en Belgique est beaucoup plus liée au secteur public. Ceci n'est pas possible en Haïti, étant donné les problèmes politiques discutés plus tôt dans ce chapitre.

## **Visite d'un centre opérationnel en France**

Afin d'avoir une meilleure vision de ce qu'implique la gestion de l'eau sur le terrain, ainsi que du quotidien d'une société impliquée dans cette gestion, nous sommes allés visiter le groupe Saur<sup>2</sup>, une entreprise de gestion (assainissement, distribution, facturation, etc) de l'eau pour les particuliers et industriels, principalement en France.

Nous avons visité le Centre de Pilotage Opérationnel de Serris. Il s'agit du centre névralgique d'où sont dirigées les activités de Saur pour la région Île-de-France. Ce centre permet un suivi en temps réel des activités sur toute la région par des ordonnanceurs, employés se chargeant de la planification et du suivi du travail des opérateurs de terrain.

Les activités de ce centre sont principalement orientées autour de l'analyse des données reçues à travers tous les capteurs du réseau. Ces analyses sont étudiées par les ordonnanceurs qui déterminent les priorités du système et planifient les journées des opérateurs en conséquence.

Tout le centre fonctionne de manière informatisée. Le système informatique se base sur EPANET (voir section suivante) et propose un véritable ERP<sup>3</sup> intégrant toutes les activités du centre, afin d'en augmenter la productivité. Dans ce logiciel, l'eau n'est au final plus qu'une ressource de l'entreprise et l'important est la gestion au quotidien de celle-ci.

Ce système informatique ne détermine pas le réseau de distribution. Celui-ci s'est développé historiquement et le système informatique s'est greffé par dessus. Il permet une collecte des données de ce réseau et de déterminer les opérations à effectuer en fonction de ces données.

---

2. <https://www.saur.com/>

3. Enterprise Resource Planning, progiciel de gestion intégrée

## 2.4 Comparaison avec des outils existants

Afin d'avoir une meilleure idée de ce à quoi devrait ressembler un logiciel de gestion de distribution d'eau, nous avons recherché des logiciels existants afin d'identifier leurs similarités, forces et faiblesses.

**EPANET** [6] est un logiciel libre de droits, développé par l'*Environment Protection Agency* aux États-Unis. Ce logiciel est une véritable référence dans le milieu, et se retrouve souvent utilisé comme base d'autres logiciels. **WaterCAD** [19] s'inspire de suites logicielles connues afin d'apporter une interface moderne et des visuels avancés. **KY Pipe** [11] se base sur EPANET pour proposer une analyse hydraulique avancée et pour permettre de modéliser de nombreux contenus. Beaucoup d'autres logiciels existent, généralement basés sur EPANET ou AutoCAD (un logiciel plus générique).

On peut remarquer que tous ces logiciels proposent une modélisation géographique du réseau de distribution. Celle-ci permet un affichage du réseau sous forme de graphe avec de liens pour les conduites et des nœuds pour les différents points de traitement, ainsi qu'une superposition sur une carte géographique pour une meilleure visualisation. Un autre module commun à tous ces logiciels permet d'effectuer des analyses formelles sur le réseau. Les modèles mathématiques sont utilisés pour poser une évaluation (quantitative, relative, etc) du système. Ces analyses permettent d'étudier les flux, les dépôts, la pression, etc, que ce soit en un point ou sur toute une partie du réseau. Un dernier module commun permet d'obtenir des rapports sur l'état en temps réel du réseau de distribution.

La majeure différence entre ces solutions est le modèle tarifaire. On passe d'un logiciel gratuit à d'autres dont le prix peut s'élever jusqu'à 14.000 \$. Les autres variations sont plus d'ordre visuel, avec des interfaces plus ou moins stylisées ainsi que des représentations du réseau et des données plus ou moins modernes.

# Chapitre 3

## Organisation

Tout au long de l'année académique, la réalisation du mémoire a été influencée par la composition particulière du groupe. Trois mémorants, deux promoteurs, un client, Protos, et des acteurs de terrain. Autant de points de vue, horaires, désirs et apports à intégrer dans ce travail. Afin de conserver un cap commun et de pouvoir concilier les objectifs de chacun, il a été nécessaire de définir, planifier et attribuer des tâches.

### 3.1 Approche de travail

Comprendre les problématiques liées au mémoire a été notre première tâche. En tant qu'étudiants en sciences informatiques, notre formation n'inclut pas de cours sur la gestion de l'eau et le contexte haïtien. L'idéal aurait été d'intégrer dans l'équipe des consultants sur les domaines méconnus et d'envisager une visite en Haïti pour rencontrer les acteurs locaux et appréhender la situation.

Pour des raisons évidentes de budget et de temps, l'option de se rendre sur place et engager plus de partenaires a immédiatement été écartée. Pour comprendre la situation et relever les défis posés par le contexte, nous avons pu compter sur l'expertise de nos promoteurs et les 700 pages de documentation qui nous ont été fournies en préambule. Cette première documentation, orientée sur le contexte, a apporté des éclaircissements sur les défis de la gestion de l'eau potable en Haïti (ci-avant présentés en section 2.1).

Tandis que les besoins non-fonctionnels de l'application étaient établis par ce contexte et les vues de Protos, le contenu fonctionnel s'est basé sur les outils actuellement en place. Nous avons vu en section 2.2 que des solutions locales ont été mises en place. En essayant de tirer le meilleur de toutes ces solutions, nous avons établi un cahier des charges (voir chapitre 4).

L'arrivée en Belgique de plusieurs acteurs haïtiens impliqués dans la gestion de l'eau potable a modifié notre approche. A partir de janvier 2019, des réunions régulières avec des utilisateurs potentiels de l'application ont été mises en place. Ce changement d'approche a permis d'orienter le développement vers un but bien plus précis qu'auparavant, nous permettant de bénéficier de nouvelles expertises et nouveaux points de vue.

## Planification

L'organisation d'un groupe, même réduit à trois personnes, est nécessaire pour orienter le travail dans un but commun. La planification de la réalisation du mémoire, qui est constitué en vaste majorité de l'application web produite, s'est effectuée sur trois plans :

**Mensuel** Au départ du mémoire, un plan général a été dressé. Au début, les inconnues (e.g., étendue du projet, fonctionnalités à implémenter) ont empêché de définir avec précision les productions et les dates butoires. Avec l'avancée du mémoire, nous avons pu estimer l'état et notre vitesse de travail. Les réunions avec Protos pour présenter nos deux plus grandes échéances, à savoir les versions alpha et beta de l'application web, ont été l'occasion de refaire le point et reformuler les objectifs.

**Hebdomadaire** Chaque semaine en alternance, nous rencontrions un de nos promoteurs. Dans la continuité de la planification mensuelle, ces réunions permettaient de présenter les productions, rapporter les tâches en cours et leur avancement, et enfin de projeter le futur du travail. Ces présentations étaient également l'occasion de rassembler les productions et d'avoir, chaque semaine, des documents ou démonstrations pouvant être comprises par des acteurs étrangers à l'équipe de développement. C'était donc autant un médium d'information que l'occasion de nous recentrer sur le travail, sans compter les précieux retours de nos promoteurs.

**Quotidien** Après une période d’expérimentation avec plusieurs outils de suivi, la gestion au quotidien des tâches à accomplir s’est ancrée sur Trello<sup>1</sup>, un tableau en ligne composé de cartes amovibles réparties dans des colonnes. Dans notre organisation (exemple simplifié en figure 3.1), une carte représente une tâche plus ou moins atomique semblable à un *récit utilisateur*<sup>2</sup>. Nous utilisons les colonnes pour classer les cartes selon l’état de la tâche associée. Nous avons ainsi défini les colonnes : *à faire*, *à faire urgemment*, *en cours*, *problématique* et *fini*. Nous avons également utilisé des messageries instantanées (e.g. Slack<sup>3</sup>) et canaux vocaux pour organiser les tâches de manière plus informelle.

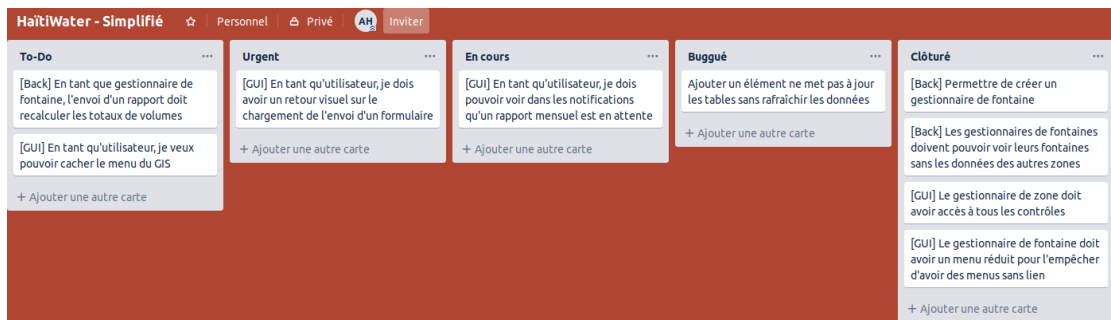


FIGURE 3.1 – Exemple simplifié d’interface Trello

Etablir un plan et s’y tenir n’était cependant pas suffisant. Le projet a connu de nombreuses modifications dans ses objectifs et contraintes. Différents facteurs viennent expliquer ces modifications :

- La présentation de prototypes aux partenaires et aux promoteurs ont souvent amené de nouvelles idées pour l’application, ou suggéré la modification de fonctionnalités existantes.
- L’arrivée de nouveaux acteurs, principalement aux mois de février et mars. Plusieurs Haïtiens ont pu apporter des éclaircissements sur la situation qui ont modifié l’application et l’orientation de son développement.
- Certaines fonctionnalités se sont révélées plus complexes à implémenter que prévu.

1. <https://trello.com>

2. Un récit utilisateur est une phrase ou texte court décrivant une fonctionnalité à développer

3. <https://slack.com/intl/fr-be/>

La figure 3.2 présente une vue simplifiée du diagramme de Gantt<sup>4</sup> final sur lequel les modules implémentés (voir section 4.3) sont placés sur une ligne temporelle en dessous de laquelle les grandes phases du mémoire ont été placées, ainsi que les deux versions majeures, *alpha* et *beta*, présentées à tous les acteurs du projet.

Les phases d'*analyse* et *architecture* se sont également clôturées par des réunions de taille plus importante afin qu'un maximum de personnes puissent donner leur avis sur les artefacts (cahier des charges, ébauches d'interface graphique, etc). C'est notamment après ces réunions que les changements de planification étaient décidés pour répondre aux nouveaux besoins du client.

Le diagramme en figure 3.2 est donc la version finale. Comparé au plan initial, on constate une augmentation de la phase d'implémentation en raison de l'augmentation de la charge de travail, les modules *historique* et *finances* étaient en effet imprévus. De nombreuses autres fonctionnalités ajoutées a posteriori ont allongé cette phase. Il était également prévu de passer plus de temps sur l'analyse et la conception du système. Or le cahier des charges et les ébauches n'étaient pas assez concrets pour avoir des retours précis sur l'outil. Devant ce fait et l'ampleur du projet, nous avons démarré l'implémentation plus tôt que prévu, et elle s'est également allongée sur le mois d'avril (pour un total d'environ 6 semaines supplémentaires).

Ces imprévus n'ont heureusement pas eu d'impact sur la production. Le démarrage anticipé de l'analyse en août, l'ajout d'une semaine de travail imprévue durant la session d'examen de janvier ainsi qu'une zone tampon de deux semaines prévue dans l'horaire initial ont permis d'accomplir la totalité des tâches initialement prévues, ainsi qu'une part appréciable des suggestions apportées au cours de l'année.

---

4. Un diagramme de Gantt présente l'avancement et les prévisions des tâches d'un projet en fonction du temps.

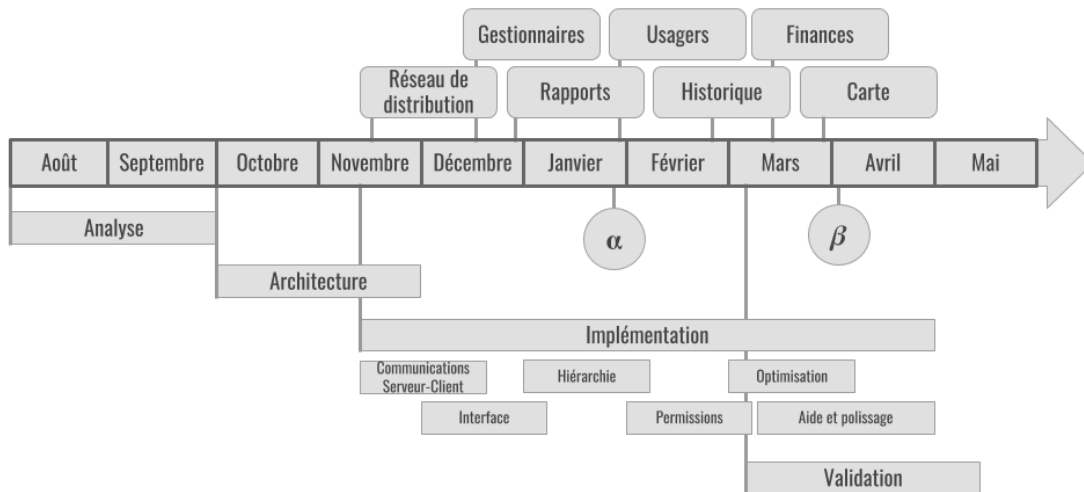


FIGURE 3.2 – Représentation simplifiée du diagramme de Gantt final

## 3.2 Méthodologie

Garantir l'avancement sans variation indésirable de tout projet se fait par le biais d'une méthodologie [10]. Il est important que cette méthode de travail soit connue de chacun pour garantir un rôle fixe et un suivi efficace. Cette notion de méthode est une caractéristique bien connue de la gestion de projet, le but étant de partir d'un besoin et d'apporter une solution informatisée. La méthode lie ces deux états et définit un système de communication, échéances, tâches, etc. De nombreuses méthodes déjà établies existent et sont prêtes à l'emploi par n'importe quelle équipe.

### Agile

Dans la section précédente, nous avons évoqué les changements constants dans la planification et les tâches. C'est donc tout naturellement que nous avons opté

pour une *méthode agile*, en opposition aux méthodes traditionnelles dites *waterfall* qui se montrent plus rigides et réfractaires au changement.

Nous n'avons pas suivi une méthode agile fixe, préférant une version plus relaxée, adaptée aux variations de disponibilité inhérentes à notre statut d'étudiant. On peut cependant relever des grands principes inspirés de méthodes existantes, notamment Scrum [16].

### **Développement dirigé par les fonctionnalités** *Feature Driven Development*

en anglais, ce type de développement focalise les ressources sur la création. Tout d'abord création d'un prototype général et d'une liste de fonctionnalités. Ensuite création de celles-ci, ajoutées à l'application en impliquant le client à chaque fois qu'une fonctionnalité est créée. C'est également le client qui définit la priorité des fonctionnalités. Ce dernier point fut très utile dans notre mémoire où les acteurs haïtiens ont souhaité orienter le développement vers l'aspect financier de l'application, en opposition à Protos qui accentuait principalement le réseau de distribution.

**Itérations courtes** Une itération est une étape indépendante du développement. Chaque itération a sa propre planification et rétrospective. En méthode agile, les itérations sont courtes. Puisque nous voyions nos promoteurs toutes les semaines en alternance, nos itérations se sont naturellement calquées sur ce rythme. Nous avons évoqué l'intérêt de recentrer le développement en section 3.1. C'est également une excellente manière de détecter rapidement les risques (tant au niveau de la planification que de l'application produite) et d'agir en conséquence.

**Livraison continue** A partir de février, un acteur haïtien arrivé en Belgique a utilisé intensivement l'application. Grâce à ses retours, nous avons pu modifier l'application. Durant trois mois, nous avons déployé de nouvelles versions de l'application toutes les deux ou trois itérations afin qu'il puisse essayer les fonctionnalités ajoutées. Cela a nécessité un peu plus de travail, pour déployer l'application et garantir l'intégrité des données entre les différentes versions. Effort récompensé par des retours fréquents de la part d'un utilisateur impliqué dont le point de vue extérieur s'est avéré extrêmement utile.

## **Phases du mémoire**

Si le développement (implémentation) de l'application a suivi ces principes agiles, on notera que le déroulement d'ensemble du mémoire est divisé en grandes



phases. Ce principe est mis en avant dans la figure 3.2. Cette caractéristique est traditionnellement indicatrice d’un développement en cascade où les différentes étapes du développement logiciel sont délimitées en temps et responsabilités (analyse, conception, implémentation, test, déploiement).

Les raisons sont multiples. On citera notamment notre méconnaissance des contextes de la gestion de l’eau potable et d’Haïti pour lesquels il a fallu se documenter pendant une période assez longue ainsi que la quantité de documents initiaux qui ont forcé à travailler durant ces premiers mois sur la compréhension et l’ébauche d’une solution.

Les disponibilités des acteurs externes au développement (membres de Protos, Haïtiens, chercheurs externes) ont aussi été un facteur. Chacun ayant une mission déterminée, nous ne pouvions avoir de contacts trop fréquents (aussi dûs à la distance et au décalage rendant difficile l’organisation de réunions). Nous avons donc dû aussi adapter les itérations aux disponibilités, variables, de chacun. Ceci explique aussi partiellement pourquoi le développement s’est autant accéléré lorsque des acteurs haïtiens sont venus en Belgique et nous ont offert des retours fréquents, ont participé à différentes réunions. La disponibilité du client est très importante dans le développement agile afin de pouvoir évaluer le travail accompli. En l’absence de membres extérieurs, ce sont nos promoteurs qui ont parfois joué ce rôle de client.

### 3.3 Répartition des tâches

Dès le début, nous avons cherché à tirer avantage de nos centres d’intérêts et capacités différents. Nous avons eu la chance d’être une équipe complémentaire et d’avoir un mémoire intéressant au point que toutes les tâches étaient attribuées sans problème et rapidement. Dans cette section, nous discutons de cette attribution systématique et de ce qu’elle implique pour ce travail.

#### Analyse

La phase d’analyse principale, d’août à octobre, avait pour but de comprendre la situation et le rôle de l’application. Afin que toute l’équipe avance vers le même objectif, il a été nécessaire que chacun soit impliqué dans l’analyse. En débutant par une recherche libre, nous avons cherché à nous documenter sur le contexte haïtien,

les solutions existantes et les entreprises qui appliquent ces solutions. L'apport de plusieurs centaines de pages de documentation par Protos et nos promoteurs a nécessité une certaine organisation. Demander à chacun de nous, mémorants, de tout lire nous a semblé improductif, mais il était nécessaire que les informations circulent. Nous avons donc adopté un système de file pour les documents. Chaque nouveau document [17] est placé dans cette file et pris en charge par un de nous trois (selon disponibilité et affinité). Au fil de la lecture, un document résumant le contenu et relevant les informations utiles pour le mémoire est produit. Le résumé terminé et mis à disposition, il est lu et discuté en groupe. Un autre objectif de ce système est de pouvoir bénéficier de documents raccourcis pour rechercher des informations, notamment dans la rédaction de ce travail.

Précisons cependant que même si ce que nous considérons être comme la phase d'analyse principale s'est déroulée de août à octobre, les changements d'exigences et l'apport de nouvelles informations au cours de l'année ont induit des phases d'analyse complémentaires, principalement au démarrage l'implémentation de chaque module. A nouveau, nous nous sommes tous trois impliqués dans ces phases afin que le développement aille dans un sens unique.

## Conception et implémentation

Contrairement à l'analyse, la conception et l'implémentation ont connu une très forte séparation des tâches entre nous trois, chacun ayant un champ de travail défini.

**Adrien** responsable de la gestion du projet (planification, présentations intermédiaires, communication), de l'analyse fonctionnelle et du *front-end* (le client) : interface graphique, fonctionnalités utilisateur.

**Céline** responsable du *back-end* (le serveur) : base de données, traitement des informations et calculs, gestion des performances.

**Sébastien** responsable du *service worker* (l'accès hors-ligne à l'application), des tests et renfort au *back-end* sur le module financier et la vérification des permissions.

Pour chacune de ces préoccupations, le responsable désigné s'est chargé de la conception, des schémas, de la réalisation (implémentation) et de la documentation. Une communication constante et présentation de l'avancement des tâches à

chaque membre de l'équipe se sont implicitement imposées, toujours dans l'optique d'orienter le développement dans le même sens.

L'inconvénient principal de ce type d'organisation est la compartementalisation qui nuit à l'apprentissage individuel puisque tout le monde ne travaille pas avec la totalité des technologies utilisées. Cela pose un risque d'arrêt, même temporaire, du développement si un des membres se trouve dans l'incapacité d'assumer sa charge de travail. En entreprise, ce risque serait à corriger au plus vite en impliquant plus d'acteurs dans chaque branche du développement, ou en impliquant tous les acteurs dans la totalité du code (par exemple via un système de rotation).

Nous avons tout de même effectué ce choix pour la puissance de travail qu'il permet et les facilités de développement et d'organisation apportées. Chacun de nous a développé une connaissance accrue de la technologie utilisée dans sa partie, permettant une augmentation drastique de l'efficacité qui s'est particulièrement ressentie dans la seconde partie (second quadrimestre) du travail. Organisationnellement parlant, savoir directement à qui s'adresser en cas de problème ou demande de fonctionnalité a été également particulièrement pratique.

## Rédaction

La rédaction du présent texte a à nouveau rassemblé l'équipe. En tant que document résumant et justifiant le travail accompli, il était naturel d'être tous inclus dans son élaboration. Nous avons considéré les différentes sections en tant qu'entités individuelles, permettant un déroulement de la rédaction simplifié, étape par étape. A des granularités différentes (parfois allant jusqu'à établir le plan pour chaque sous-section), nous avons dressé une liste du contenu à intégrer. La section était ensuite attribuée au mémorant responsable du sujet discuté.

Une fois le premier jet établi, une relecture permettait à chacun d'apporter les modifications désirées, parfois discutées, pour obtenir un texte dans lequel chacun de nous peut s'identifier. Ce processus était répété pour chaque section et chaque jet, encouragés et orientés par les commentaires de nos promoteurs.

## Au quotidien

La planification générale avait une précision au maximum hebdomadaire. Pour le travail effectué au jour le jour, nous nous sommes entièrement reposé sur Trello avec un système de colonnes décrivant l'état et la priorité de chaque tâche comme exposé ci-avant. Autre héritage des méthodologies agiles, ces cartes sont des tâches à réaliser pour avancer le développement. Dans notre cas, les tâches étaient souvent atomiques ou ne reflétaient qu'une seule unité de développement (e.g., ajouter un champ dans un formulaire).

Un système de description de la tâche, assignation de membre(s) à une carte et étiquetage, ajouté au principe des différentes colonnes indiquant la priorité et l'état de chaque carte, a permis de gérer efficacement le travail.

Quant à l'ajout des cartes, on relève deux possibilités :

- Au début des itérations, lorsqu'on planifiait le sprint à venir.
- Durant une itération, lorsqu'un membre de l'équipe avait besoin d'une fonctionnalité qu'un autre membre devait implémenter, le premier membre ajoutait une carte en assignant le second. Par exemple, si le développement du client nécessitait l'implémentation d'une fonctionnalité du serveur, la carte était ajoutée en assignant directement la personne responsable.

La colonne de travail prioritaire s'est avérée une très bonne idée. Puisque nos fonctionnalités sont dépendantes à la fois du client et du serveur, cela permettait d'identifier clairement les besoins. Nous pouvions ainsi débiter une séance de développement à n'importe quel moment en sachant ce dont les autres avaient besoin.

# Chapitre 4

## Analyse des besoins

La phase d'analyse des besoins est, dans tout projet, l'une des plus importantes. Elle se doit d'être méticuleuse, afin que tous les partis impliqués soient d'accord sur ce qu'ils souhaitent. Cependant, nous devons rester conscients que le temps accordé à un mémoire est assez court par rapport à ce qui se fait dans l'industrie pour un projet de cette ampleur. Nous ne pouvions donc pas passer trop de temps sur cette phase, sous peine de ne plus en avoir assez pour implémenter les fonctionnalités établies.

Nous pensons avoir trouvé un bon entre-deux, prenant avantage des conseils glanés au cours de nos études [13] : nous avons présenté au plus vite des visuels de la future application, et avons organisé une réunion Skype pour pouvoir échanger en direct avec nos clients malgré le décalage horaire.

Le cahier des charges résultant de ces interactions nous semble complet, et a été assez peu modifié au cours du projet. Deux éléments s'y sont cependant ajoutés, l'un pour répondre aux besoins non-fonctionnels et l'autre suggéré par Louis Fritzno, un acteur haïtien venu pour apporter un point de vue concret.

### 4.1 Besoins fonctionnels

Le terme *besoins fonctionnels* décrit les besoins précis d'un client, les non-négociables dans le cadre d'un projet. Ce sont les points qui ont été mis en avant par Protos et les Haïtiens comme étant essentiels, nécessaires afin que notre application leur soit utile.

## Gestion des données

La chose la plus évidente dont ont besoin les acteurs sur place est une aide à la gestion de leurs données. La technique papier-crayon décrite plus tôt s'est montrée inefficace, il faut donc un système permettant à la fois de centraliser les données et d'éviter les pertes, tout en ne s'éloignant pas trop de ce que les Haïtiens connaissent et font aujourd'hui, afin que la transition vers le nouveau système informatisé puisse être facilitée par une certaine familiarité.

Pour mieux comprendre la nature exacte des données en question, nous avons étudié des fichiers Excel [17], qui servent à ce jour de premier pas vers une informatisation de la gestion du réseau de distribution d'eau.

Ces documents nous ont été d'une très grande utilité, puisqu'ils nous ont permis de lister toutes les données nécessaires à notre application, nous donnant un solide point de départ pour notre base de données. Nous en avons retiré une structure de données présentée dans la figure 4.1. Ils nous ont également permis de mieux comprendre quelles étaient les responsabilités des différents acteurs, ce qui nous a donné la structure de l'application elle-même.

En nous inspirant directement des *Komité Dlo* décrit dans le chapitre 2.2, nous avons décidé de créer deux types de gestionnaires : ceux dits *de zone*, avec peu d'interaction avec les usagers du réseau, qui ont une vision plus globale de celui-ci. Ils collaborent avec des gestionnaires dits *de fontaine(s)*, qui gèrent directement un ou plusieurs points d'eau, et sont en contact direct avec les consommateurs.

Cette division permet à chacun d'avoir un rôle clair et précis, les gestionnaires de fontaines apportant les données au jour le jour (paiements, nouveaux usagers, etc.), tandis que les gestionnaires de zone s'occupent du réseau en lui-même.

## Simplification des procédures

La seconde observation que nous avons pu mettre en avant en lisant les documents fournis par Protos et en parlant aux Haïtiens est que les procédures en place sont à la fois trop compliquées et peu standardisées. Il est souvent nécessaire de se déplacer physiquement jusqu'à un point d'eau ou chez la personne qui en

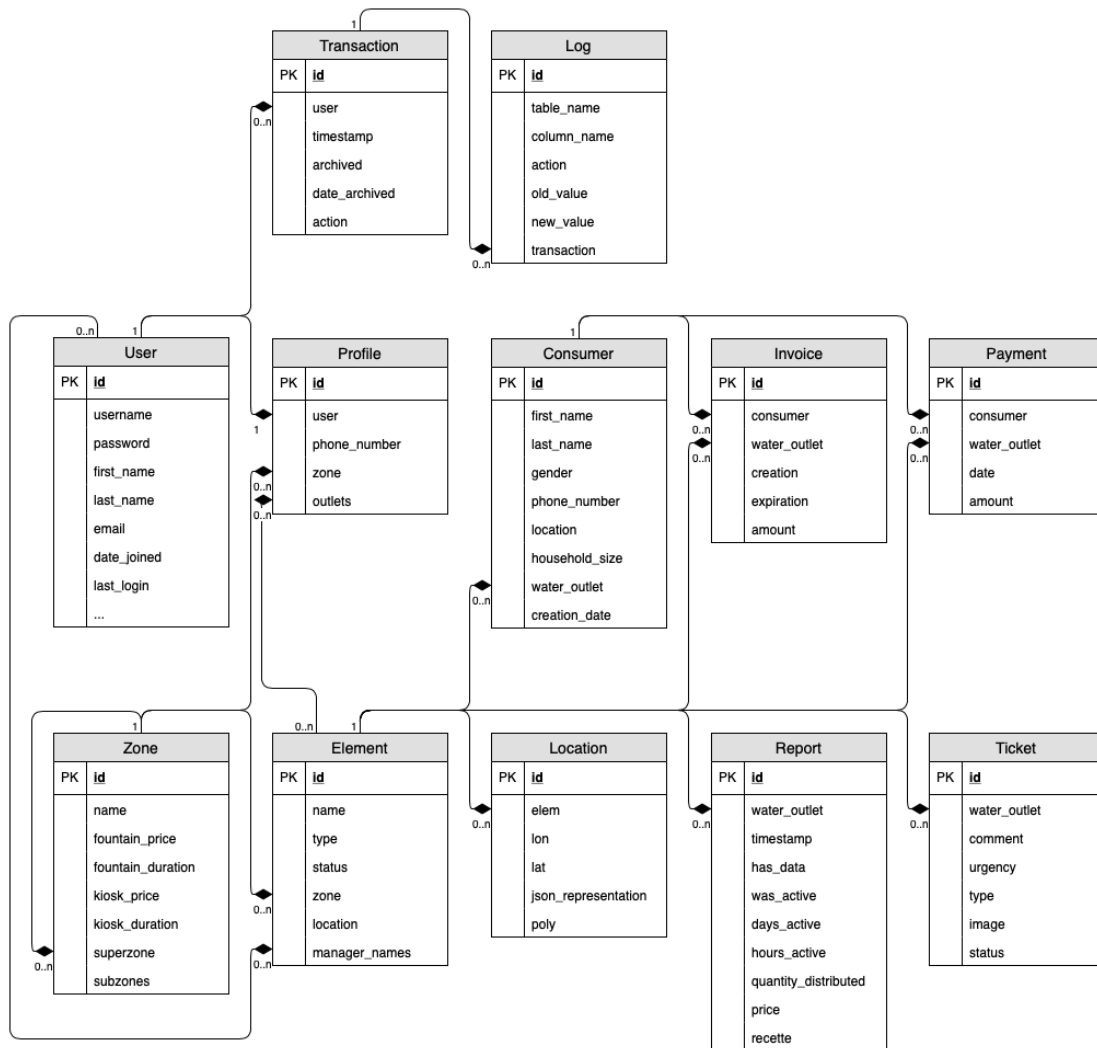


FIGURE 4.1 – Schéma des données du système

est responsable pour avoir l'information nécessaire, ce qui est à la fois une cause directe et un résultat de l'absence de centralisation.

La division en différents rôles que nous avons établie permet aux utilisateurs de l'application d'avoir toujours une vue sur les éléments du réseau de distribution dont ils sont responsables, et de faire circuler les informations beaucoup plus vite.

Nous souhaitons également aider les gestionnaires de zone à prendre des décisions plus rapidement. En effet, grâce à la numérisation des données, il leur est possible

de visualiser leur réseau de distribution avec des filtres. Ceux-ci peuvent mettre en avant les points d’eaux nécessitant une réparation, ou les villages dans lesquels les factures sont en retard.

## 4.2 Besoins non-fonctionnels

Le terme de *besoin non-fonctionnel* désigne un besoin caractérisant le système, une contrainte liée à l’implémentation, aux performances ou à une situation spécifique au projet en lui-même. Si ces buts ne sont pas atteints, l’application est quand même utilisable, leur réalisation est cependant nécessaire pour obtenir un système fiable et qualitatif.

### Sécurité des données

Dans le monde d’aujourd’hui, il est évident qu’une certaine attention doit être accordée à la sécurité d’une application web. En Haïti peut-être plus qu’ailleurs, une attention particulière se doit d’être portée à la possibilité qu’un utilisateur malveillant ou non se connecte au système.

Par conséquent, nous avons non seulement implémenté un principe de connexion avec privilèges (certaines données sont inaccessibles sans un niveau de privilège particulier), mais nous avons également ajouté un système de *logs*.

Ce module, appelé *module historique* n’était pas prévu initialement, nous l’avons ajouté pour remédier aux erreurs des utilisateurs, ou aux comportements malicieux. Ce module permet aux supérieurs hiérarchiques de consulter les modifications faites par leurs subordonnés. Si ces modifications leur semblent incorrectes, un seul clic leur permet de faire revenir les données à leur état précédent. Si une modification est acceptée, elle disparaît de la liste en laissant la modification, jugée valide, dans la base de données.



## Connexions lentes et peu fiables

L'utilisation de notre application en Haïti amène un besoin particulier : dans ce pays, la connexion Internet est bien plus lente que nos standards européens et y est souvent peu fiable. Pour pallier à cela, nous avons d'abord utilisé les techniques courantes : nous envoyons aussi peu de données que possible, faisons usage de la cache 5.4 et envoyons les informations des différents éléments de la page séparément.

Cependant, une demande de nos clients a été de pouvoir utiliser certaines fonctionnalités de l'application lorsqu'il n'y a pas de connexion. Pour cela, nous avons développé un *service worker*, détaillé en section 5.4.

## Technologies simples et populaires

Par rapport à un projet classique, une contrainte de ce travail est qu'il se déroule dans le cadre d'un mémoire universitaire. À la fin de nos études, ce projet sera repris par d'autres étudiants pour un deuxième mémoire et sera à terme déployé en Haïti pour être maintenu par les acteurs locaux. Comme ces futurs développeurs ne sont pas connus, une attention particulière doit être apportée à la facilité de la maintenance de notre application.

Pour cela, nous avons décidé d'utiliser des technologies relativement simples et populaires, de sorte que même si nos successeurs ne les connaissent pas, ils puissent assez facilement les apprendre par eux-mêmes, en utilisant des tutoriels disponibles gratuitement sur Internet par exemple. Une description plus détaillée des choix technologiques que nous avons faits se trouve à la section 5.1.

## 4.3 Cahier des charges

Après discussion avec Protos et les Haïtiens, nous avons décidé d'organiser notre application en différents modules indépendants interagissant avec notre base de données. Ce choix de conception nous a permis dans un premier temps d'appliquer la méthode Agile plus facilement, déployant notre travail module par module. Dans un second temps, il permet à notre application d'évoluer très rapidement : un nouveau module vient se greffer à la base de donnée sans changer le fonctionnement

des autres, un ancien module qui aurait perdu son utilité peut être enlevé sans rien endommager. De plus, ce choix devrait aider les futurs développeurs à ajouter à la base applicative existante plus rapidement.

Chaque module représente un type d'activité pratiqué par les gestionnaires du réseau de distribution. Après cette première année de développement, nous en avons créé huit :

- Le module **réseau**, permet de créer de nouveaux éléments du réseau (source, conduite, réservoir, kiosque, fontaine, prise individuelle) et de consulter des informations à leur propos (nom, emplacement, état de fonctionnement)
- Le module **carte interactive**, fortement lié avec le précédent, permet de placer et visualiser les éléments du réseau de distribution sur une carte. Le module permet de placer des éléments au jugé avec la souris, ou directement avec trois systèmes de coordonnées : le degré décimal, le degré-minute-seconde et la transverse universelle de Mercator (cette dernière méthode demandée par notre contact haïtien).
- Le module **consommateurs** contient toutes les informations à propos des usagers du réseau (nom, adresse, etc), groupés par ménages.
- Le module **financier**, fortement lié avec le module précédent, permet de consulter et d'ajouter les informations concernant les paiements réalisés par les usagers du réseau.
- Le module **rapports** permet aux gestionnaires de fontaine(s) d'entrer des informations sur le fonctionnement du réseau (s'il a été actif, quelle quantité d'eau a été distribuée) et de créer des tickets de problèmes en cas de panne d'un élément.
- Le module **gestion de zone**, plus administratif, permet de créer de nouveaux gestionnaires et de les assigner à une zone ou des éléments du réseau en fonction de leur rôle.
- Le module **historique** permet aux supérieurs hiérarchiques de visualiser tous les changements, ajouts ou suppressions effectués par leurs subordonnés. Ils peuvent valider ou annuler ces changements. Pendant trois semaines, un choix (validation ou annulation) est affiché dans une seconde table pour informer les autres supérieurs qu'une décision a été prise, afin de permettre de récupérer des informations supprimées en cas d'annulation erronée.
- Le module **offline** permet d'assurer la disponibilité du site même lorsque la connexion internet n'est pas stable, donnant la possibilité de compléter le rapport mensuel.

# Chapitre 5

## Implémentation

Total des pages : 16 à 22

### 5.1 Choix technologiques

Pages : 3 à 4

#### Web

Établie dès le premier jour, l'exigence du support web pour ce travail a influencé la totalité du mémoire. Nous n'avons pas eu de réel choix à ce niveau, il nous est néanmoins possible de trouver des avantages à ce type de développement :

- L'application ne nécessite aucune installation, aucun prérequis et la seule nécessité est de disposer d'un navigateur internet (préférentiellement majeur et relativement récent). C'est une facilité pour l'utilisateur, puisqu'il suffit d'accéder au site internet. L'avantage est également étendu au développement qui ne doit pas se préoccuper du système d'exploitation de l'utilisateur.

- Le déploiement des mises à jour et la manière dont l'application fonctionne en général est uniforme à travers les utilisateurs et les conflits de version, bien que devant toujours être gérés notamment par le biais du cache (voir section 5.4), sont moindres que sur une application locale. D'éventuels conflits de données peuvent être réglés directement via le serveur et les outils de développement disponibles à l'équipe, ce qui est plus rapide et moins complexe que de créer des automatisations pour gérer ces mises à jour individuellement pour chaque utilisateur.

— Ne présentant pas que des avantages, le développement web apporte son lot de problématiques nouvelles :

- Utiliser une application en ligne nécessite une connexion internet, et même si cette dernière n'est pas requise en permanence (voir section 5.4), il est impératif qu'elle soit présente ponctuellement pour charger l'application et envoyer/recevoir les données lors de son utilisation.
- La diversité des navigateurs internet (Firefox, Chrome, Edge, etc) entraîne la nécessité de gérer ces environnements différents pour que l'application fonctionne sur un maximum de navigateurs et de manière similaire.

## Choix du framework - Django

Un framework est une infrastructure logicielle dans laquelle l'application est développée. On peut argumenter, avec raison, qu'un framework ajoute une certaine quantité d'outils qui ont pour effet de complexifier l'application et d'en augmenter la taille. Dans une application aussi large que la nôtre cependant, l'apport bénéfique qu'ont ces outils supplante largement ces inconvénients. Parmi ces apports bénéfiques on citera surtout :

- L'environnement de travail d'un framework impose certaines contraintes qui répandent des bonnes pratiques. Par exemple une structure des fichiers du code source qui permet à long terme une meilleure maintenance, surtout dans de larges applications.
- Les outils fournis sont testés, utilisés et éprouvés. Il n'est pas nécessaire de créer nous-même des fonctionnalités déjà présentes dans un framework, et même si recréer ces fonctionnalités est une expérience enrichissante, elle reste moins efficace. De plus, les fonctionnalités existantes sont de meilleure qualité, offrent de meilleures performances et surtout ne sont pas à maintenir par l'équipe de développement de l'application, mais par l'organisme en charge de la publication du framework. Ces avantages seront d'autant plus importants que le framework est populaire et maintenu à jour.

- Un framework aura beaucoup de fonctionnalités par défaut qui rendront le développement plus aisé. Messages d'erreur, outils de mesures, etc.
- La documentation et l'aide sont de meilleure qualité. Les framework ont généralement une communauté et une équipe de développement dont les capacités et l'expérience permettent une production de documentation rendant très facile la compréhension du projet et la recherche d'aide. Avantages qu'il est impossible d'égaliser dans une équipe de trois personnes sur dix mois.

La table 5.1 reprend une liste non-exhaustive de frameworks qui ont été envisagés pour ce projet. Afin de pouvoir développer une application compatible, nous avons évalué les possibilités et arrêté un choix durant la phase d'analyse du développement. Ci-après sont exposées les motivations de ce choix.

Langage	Framework	C	M	Pe	Po	T	Total
<b>JavaScript</b>	Express	M	M	B	B	M	3.5
	MeteorJS	B	B	M	N	B	3.5
	NodeJS	M	M	M	B	N	3.5
<b>Python</b>	Django	M	M	B	B	M	3.5
	Flask	B	B	B	B	B	5
	Pyramid	B	B	B	N	B	4
<b>Ruby</b>	Rails	N	M	B	B	B	3

TABLE 5.1 – Comparatif de quelques frameworks connus

## Critères - [B]on, [M]oyen, [N]égatif

**C** Complexité - Difficulté d'utilisation / apprentissage.

Évaluer la complexité est hautement subjectif en fonction des préférences et habitudes du développeur. Nous pouvons néanmoins comparer les langages utilisés et la structure du framework.

Une mesure de la complexité se base sur l'imbrication des blocs de code. Un code comportant beaucoup de structures imbriquées sera moins lisible, et donc plus complexe. Les résultats<sup>1</sup> en figure 5.1 montrent que cette mesure favorise le langage Python dans nos options.

**M** Maintenabilité - Aisance et possibilités de maintenance.

L'aisance avec laquelle un code est maintenu à jour est inversement lié à la complexité. À cela, nous pouvons ajouter les fréquences de mise à jour des frameworks. La structure du framework impacte également la maintenabilité ; moins on modifie de fichiers pour implémenter une fonctionnalité, plus facile sera la maintenance à moyen et long terme.

---

1. Étude de Seerene ©, 2015

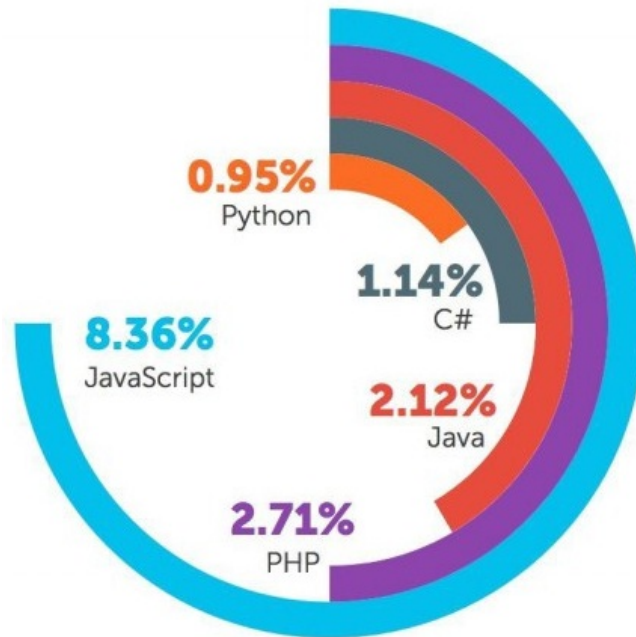


FIGURE 5.1 – Pourcentage de code imbriqué

**Pe** Performance - Capacités du framework.

Bien que moins importantes que dans une application critique (e.g. : aérospatial, banque), les performances sont mesurées en termes de temps de réponse pour une requête du client au serveur.

**Po** Popularité - Popularité du langage à travers le monde.

La popularité est sans doute la plus importante des mesures de ce comparatif. Un langage populaire dispose de plus de tutoriels et aide en ligne que d'obscurs langages / frameworks peu ou pas utilisés.

La figure 5.2<sup>2</sup> montre une évolution de la popularité des frameworks les plus courants.

**T** Taille - Espace de stockage requis

Un framework léger sera plus intéressant pour ce projet. D'une part pour la complexité, car moins il est nécessaire de modifier/ajouter des fichiers, plus l'utilisation est aisée. D'autre part car le serveur de destination nous est inconnu et nous ne savons donc pas quel espace de stockage sera mis à disposition.

---

2. <https://hotframeworks.com/>

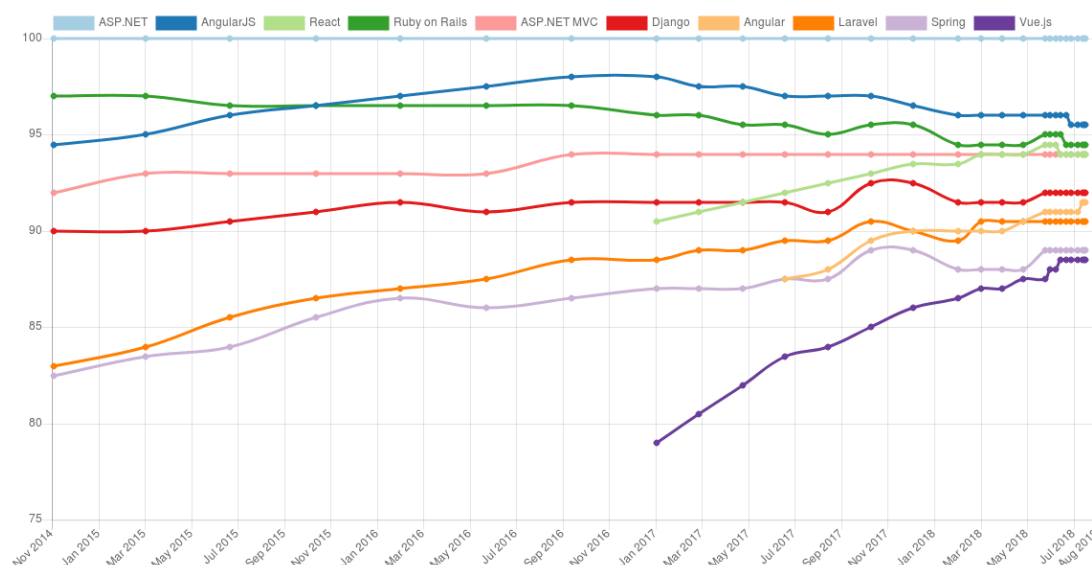


FIGURE 5.2 – Popularité des frameworks de novembre 2014 à août 2018

## Django

Le choix d'un framework repose sur de nombreux critères dont plusieurs sont subjectifs. En l'absence d'une ligne directrice, le choix se fait souvent selon l'expérience et les désirs de l'équipe. Notre comparatif terminé, il est devenu clair que nous favorisons un framework utilisant le langage de programmation Python.

Flask a été envisagé, cependant les facilités offertes par Django dans la gestion de la base de données, particulièrement pour les données géographiques, ont été le facteur décisif. Parmi les sites populaires utilisant Django, on citera Instagram, DropBox, Disqus<sup>3</sup>.

## Python

Le langage Python est un langage de programmation populaire principalement grâce à sa syntaxe légère permettant un code compréhensible et généralement plus court (voir figure 5.1). Il n'est pas le plus utilisé, ce sont PHP (79% d'utilisation),

3. <https://djangostars.com/blog/10-popular-sites-made-on-django/>

ASP.NET (11.3%) et Java (4%) qui constituent le trio de tête [18]. Toutefois, nous pensons que Python s'adapte parfaitement aux besoins de l'application demandée. C'est un langage relativement simple et compréhensible dont la popularité croît ces dernières années. Les bibliothèques logicielles et la documentation, toutes deux disponibles à foison sur internet, en font un langage particulièrement ouvert aux débutants. Un dernier argument, et non des moindres, est l'expérience déjà acquise dans la programmation en Python durant nos années d'études.

## PostgreSQL et PostGIS

Si l'utilisation du langage Python était imposée dès lors que Django fut choisi en tant qu'environnement logiciel, le système de gestion de base de données, dont le rôle est de lier l'application web et la base de données, a pu être choisi un peu plus librement.

Nous avons tout de suite écarté les systèmes non-relationnels. Bien que présentant des avantages, principalement dans l'agilité avec laquelle la structure des données peut évoluer, ces derniers ne sont pas adaptés à une application comme celle développée dans le cadre de ce mémoire. En effet, les données ont de fortes dépendances, les éléments du réseau étant rattachés à des gestionnaires faisant parties de zones géographiques elle-même gérées, etc. Les outils offerts par les bases de données relationnelles pour joindre différents groupes de données (par exemple pour croiser les consommateurs avec leurs paiements afin d'obtenir un détail financier), sont tout simplement mieux adaptés au fonctionnement de l'application.

Restent donc les systèmes de gestion de base de données (SGBD) relationnels, dans lesquels nous avons opté pour PostgreSQL, classé quatrième système de gestion de bases de données le plus utilisé au monde<sup>4</sup>. La première motivation de ce choix est la recommandation du framework Django, lequel s'adapte mieux à ce SGBD via son système de *mapping objet-relationnel* (voire section 5.5). La seconde motivation est l'existence de l'extension PostGIS [4], une extension du SGBD permettant de traiter aisément les données géographiques.

---

4. <https://db-engines.com/en/ranking>, 1 mai 2019



## Bootstrap

Bien que l'utilisation de frameworks *front-end* soit de plus en plus répandue, nous n'en avons pas intégré dans le client de l'application. Les choix sont pourtant nombreux et tant Angular, React que Vue ont été envisagés. Nous avons décidé de n'en utiliser aucun pour ne pas ajouter de technologie supplémentaire et pour limiter le nombre de dépendances du projet afin d'en garantir la perrenité sur le long terme.

Néanmoins, la version 3 (plus documentée, éprouvée et ayant de meilleures compatibilités avec les autres frameworks et bibliothèques que la version 4) de Bootstrap est présente dans l'application. Bootstrap est une bibliothèque implémentant une large quantité de fonctionnalités que nous utilisons majoritairement pour agencer visuellement les blocs de l'interface graphique en fonction de la taille de l'affichage disponible, afficher des composants avancés (multisélection, sélection de date, menus) et styliser les composants de l'interface.

Bootstrap ajoute l'avantage non-négligeable d'implémenter un bon nombre de fonctionnalités nous permettant d'accélérer le développement, tout en étant une bibliothèque assez connue que pour être extensivement testée et documentée, la rendant une meilleure alternative à une création personnalisée qui aurait sans doute été moins performante.

Le désavantage majeur est son poids. Accompagnée de nombreuses fonctionnalités qui n'ont pas été utilisées dans ce projet, Bootstrap est lourde (approximativement 400 KB), d'autant plus en considérant la lenteur des connexions haïtiennes. Il est cependant tout à fait possible de nettoyer la bibliothèque (voir section 7.1) pour n'en conserver que les fonctionnalités utilisées, et ainsi optimiser les téléchargements.

## DataTables

DataTables [12] est la seconde plus grosse bibliothèque du projet, gérant l'affichage des données sous forme de tables. A nouveau sélectionnée de par son ancienneté permettant une documentation et des tutoriels efficaces, DataTables était très loin d'être la seule option. Nous avons envisagé, et débuté la première version avec, la bibliothèque *Django-Tables2* qui causait un ralentissement du chargement des pages contenant les données, et une perte de contrôle de l'information

au niveau du serveur car les tables étaient directement rattachées aux modèles de la base de données.

Dans notre volonté d’offrir une API pour communiquer avec le serveur (voir section 5.5), nous avons migré vers DataTables. En utilisant la technologie AJAX (Asynchronous Javascript And XML) de DataTables, nous pouvons exploiter pleinement l’asynchronicité du navigateur et charger une page de l’application en totale indépendance des données, ce qui augmente la vitesse de chargement des contrôles, permettant d’utiliser directement une fonctionnalité de l’application sans attendre le téléchargement des données (par exemple pour ajouter un élément).

Les capacités offertes par DataTables sur le contrôle des données nous permettent également de gérer les données à l’envoi et/ou à la réception et de contrôler leur affichage, tout en bénéficiant des fonctionnalités implémentées par la bibliothèque (champ de recherche, tri dynamique, pagination).

Outre son poids conséquent auquel nous pouvons à nouveau opposer la solution évoquée dans la bibliothèque Bootstrap, DataTables requiert d’implémenter les fonctionnalités de l’API et de gérer le transfert et traitement des requêtes en envoi et réponse du client et du serveur, ce qui représente une charge de travail non-négligeable.

## **Chart.JS**

La bibliothèque Chart.JS est utilisée pour générer les représentations visuelles de l’application. Les bibliothèques permettant de créer des graphiques sont nombreuses et Chart.JS a de nombreux concurrents qui offrent parfois plus de fonctionnalités. Chart.JS a une bonne réputation dans ce domaine et est réputée pour ne pas être trop complexe à utiliser, toujours dans le but de respecter les exigences technologiques. De plus, Chart.JS génère des canvas là où un de ses concurrents majeurs, la bibliothèque D3.JS, génère des images vectorielles ou des éléments du domaine, plus gourmands en performances en particulier avec de larges ensembles de données.

## 5.2 La hiérarchie dans l'application

Le bon fonctionnement de l'application repose sur le principe hiérarchique de la récolte des informations venant de la saisie de données par les utilisateurs de l'application puisque le manque de financement et l'état actuel du réseau ne permet pas une récolte en temps réel des données via des capteurs. Une représentation de la structure hiérarchique vous est présentée en figure 5.3.

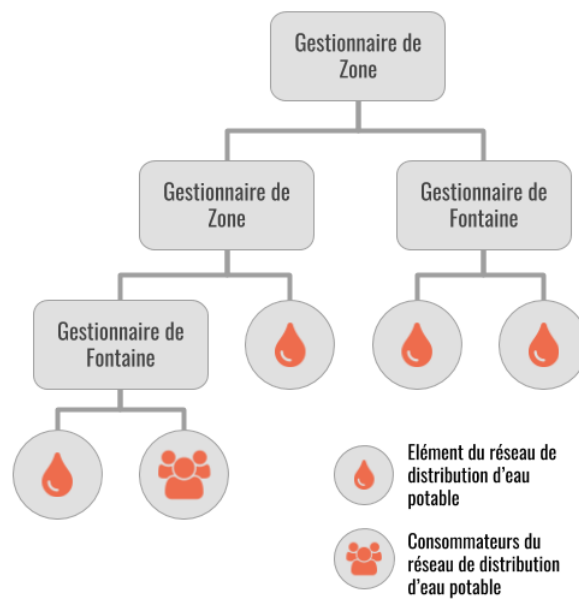


FIGURE 5.3 – Représentation du principe hiérarchique des utilisateurs, éléments et consommateurs

### Structure

Rappelons que l'application a trois concepts clefs :

#### Zone

Une zone est un partitionnement des données de l'application. A la base de l'application se situe la zone racine dans laquelle toutes les données sont enregistrées. Une zone peut être composée d'autres zones. Une zone parente rassemble la totalité des données des zones filles. En revanche, une zone fille ne rassemble que les données qui lui sont rattachées et ne voit pas le contenu des zones soeurs. On peut ainsi se servir des zones pour

diviser géographiquement le réseau de distribution d'eau selon le principe de la hiérarchie haïtienne existante (DINEPA, OREPA, CAEPA) exposée en section 2.2. Chacun a donc ses propres responsabilités et ne vient pas empiéter sur le territoire de l'autre.

### **Elément du réseau**

Un élément du réseau peut être une conduite, un réservoir, une source, une fontaine, un kiosque ou une prise individuelle. Ces six éléments constituent le réseau de distribution d'eau potable haïtien et sont enregistrés dans une zone.

### **Consommateur**

Un consommateur est un habitant qui utilise le réseau de distribution d'eau potable. Dès son enregistrement dans l'application, le consommateur est rattaché à un élément de sortie (fontaine, kiosque ou prise individuelle) du réseau, ce qui permet de gérer sa facturation et, de manière plus générale, de consulter la densité des consommateurs.

Deux types d'utilisateurs sont amenés à se connecter à l'application :

### **Gestionnaire de Fontaine**

Ce gestionnaire est un noeud terminal dans un chemin de la hiérarchie des utilisateurs. Son rôle est simple ; au plus proche du consommateur, il est chargé de gérer (et donc assigné à) un ou plusieurs éléments de sortie du réseau, c'est-à-dire qu'il s'occupe du bon fonctionnement de la distribution aux consommateurs. Il a quatre charges relatives aux éléments du réseau de distribution d'eau qui lui sont assignés :

1. Rapporter l'utilisation, sous forme des volumes distribués.
2. Signaler les problèmes avec la pompe ou avec la qualité de l'eau.
3. Ajouter, modifier ou supprimer les consommateurs.
4. Rapporter les paiements des consommateurs.

Le gestionnaire de fontaine est donc une entité locale similaires aux *comités fontaines* vus en section 2.2.

### **Gestionnaire de Zone**

Le gestionnaire de zone est un rôle plus complexe. Il est un administrateur local ou global, dépendamment de sa position dans la hiérarchie. Un gestionnaire de zone peut superviser d'autres gestionnaires, de zone ou de fontaine, et ce en nombre illimité. Un gestionnaire de zone peut effectuer les mêmes actions qu'un gestionnaire de fontaine, mais ses charges sont plutôt orientées dans l'administration de sa zone :

1. Ajouter, modifier ou supprimer des zones.
2. Ajouter, modifier ou supprimer des gestionnaires.

3. Assigner des gestionnaires de zone à une zone, et des gestionnaires de fontaine à un ou plusieurs éléments de sortie du réseau.
4. Gérer ses gestionnaires et leurs actions via l'historique.
5. Utiliser les données récoltées par ses gestionnaires pour gérer le réseau de distribution.

## Permissions

Bien entendu, une hiérarchie implique un système de permissions. Dans notre application, ce dernier se calque exactement sur la structure exposée ci-avant. En toute normalité, les utilisateurs ne peuvent effectuer que les actions qui sont assignées à leur rôle. L'interface propose des contrôles réduits aux gestionnaires de fontaine, en opposition aux gestionnaires de zone qui doivent assumer leurs responsabilités supplémentaires.

De plus, les données sont fortement liées à la hiérarchie et empêchent une quelconque utilisation, ajout, modification ou suppression par un utilisateur non-autorisé. Lorsqu'une donnée, peu importe son type (une zone, un consommateur, un rapport mensuel, etc), est ajoutée dans l'application, elle n'est visible que par l'utilisateur qui l'a créée et les utilisateurs qui lui sont supérieurs dans la hiérarchie (utilisateurs parents).

En somme, plus un utilisateur est haut dans la hiérarchie, plus il a de pouvoir sur les données de l'application, la zone racine étant donc la somme de toutes les données, faisant des gestionnaires de cette zone les utilisateurs ayant le plus d'impact sur le système.

## 5.3 Interface utilisateur

Partie visible de notre application, l'interface graphique, permet d'utiliser les fonctionnalités d'HaïtiWater. En réalité, l'interface graphique n'est qu'un moyen de faciliter l'utilisation de l'application et de rendre les données accessibles et lisibles par un humain. L'application est entièrement contrôlée par l'API qui requiert les données à afficher ou modifier. Somme toute, n'importe quelle autre interface (application locale, mobile, etc) pourrait envoyer les mêmes requêtes. Cela permet

donc de décentraliser la logique de l'application dans le serveur tandis que l'interface permet de présenter l'information à un humain (tableaux, graphiques), le guider dans son utilisation et faciliter la saisie de données via les formulaires. Non content de permettre, dans le futur, la création de nouvelles interfaces, cette séparation des rôles nous a permis d'améliorer le développement et la maintenance de l'application, nous y reviendrons en section 5.5.

Se voulant simple, l'interface est composée de blocs de contrôle permettant de gérer les données dans une zone de travail commune à toute les pages permettant de naviguer à travers l'application.

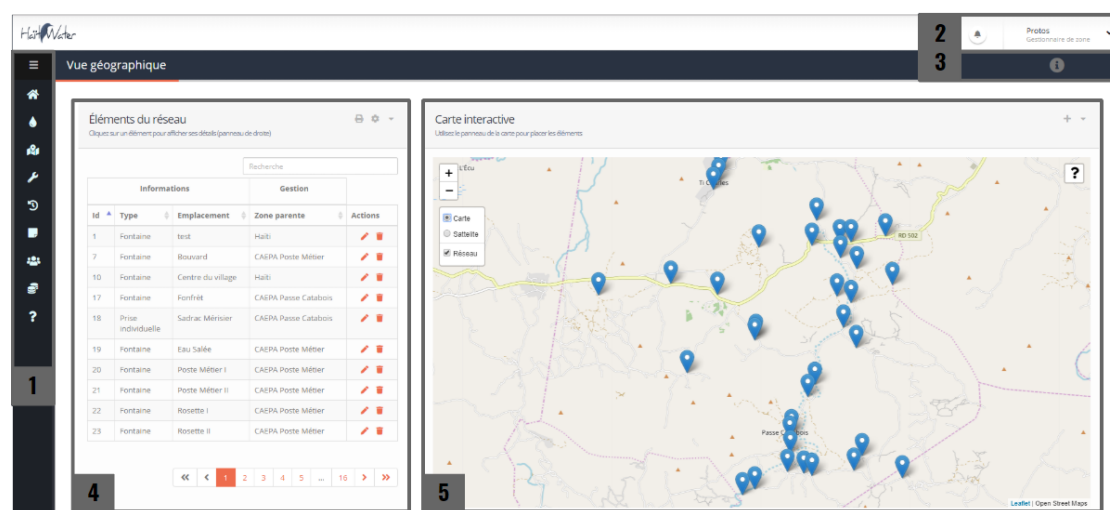


FIGURE 5.4 – Capture d'écran du module géographique d'HaïtiWater

Une capture d'écran de l'application vous est présentée en figure 5.4. Les contrôles communs incluent :

1. Le menu déroulant permettant d'accéder aux modules de l'application.
2. La zone utilisateur qui affiche ses notifications et le nom de compte qui lie à la déconnexion ou modification des informations personnelles dans son menu déroulant.
3. L'aide rapide, permettant de déclencher une visite virtuelle de la page ouverte, visite textuelle qui présente brièvement les contrôles et leur utilité.

Les contrôles spécifiques à cette page sont :

4. La table est un bloc de contrôle récurrent dans notre application. Ici listant les différents éléments du réseau de distribution, chaque table possède un champ de recherche, un système de tri et des contrôles permettant d'ajouter, supprimer et modifier les éléments contenus. Remarquez que pour augmenter la lisibilité, les tables disposent d'une pagination permettant de limiter l'affichage à 10 (valeur par défaut), 25, 50 éléments ou de tout afficher.
5. Un bloc de contrôle plus complexe, ici la carte interactive. Sur celle-ci, il est possible de placer des marqueurs (fontaines, kiosques, prises individuelles, sources et réservoirs) ou des lignes polygonales (conduites). Supportant les habituels contrôles de zoom (molette ou boutons) et déplacement (en cliquer/glisser), il est également possible de basculer entre une vue cartographique d'OpenStreetMap<sup>5</sup> et une vue satellite de Google<sup>6</sup>

Tant pour accélérer le développement que pour permettre à l'utilisateur de s'habituer à l'utilisation de l'interface en se familiarisant avec les contrôles, les blocs de l'interface graphique sont réutilisés à travers les différentes pages de l'application.

La saisie des données occupe également une grande importance dans l'application. Pour ajouter et modifier les données, nous utilisons des fenêtres modales contenant des formulaires

FIGURE 5.5 – Capture d'écran du formulaire de rapport mensuel d'HaïtiWater

5. <https://www.openstreetmap.org/>
6. <https://www.google.com/maps>

Ces formulaires ont pour objectif d'accompagner l'utilisateur dans ses tâches. En figure 5.5, vous observez que pour atteindre ce but, nous mettons en place un formulaire qui se consulte de haut en bas et de droite à gauche, où chaque information entrée est une information qui ne nécessite aucun retour en arrière. Sur ce formulaire en particulier, l'utilisateur peut entrer des données dans l'unité de volume de son choix (mètres cubes, gallons), la conversion est faite automatiquement. On trouve un autre exemple de ce genre d'accompagnement dans ce même formulaire, qui est séparé en quatre parties distinctes. Celles-ci qui évitent un bloc formulaire trop grand et permettent une transition entre différentes parties logiques du rapport mensuel.

## 5.4 Client

L'interface et sa facilité d'utilisation n'a pas été le seul axe du développement du client. C'est au niveau technologique que nous trouvons deux contraintes importantes : l'allégement de l'utilisation de la connexion et la simplification du code source derrière l'application.

### Mise en cache

La mise en cache est un système natif commun à la vaste majorité des navigateurs modernes. Le concept est simple ; conserver l'ancienne version d'un fichier pour la réutiliser la prochaine fois si elle n'a pas changé. HaïtiWater utilise pleinement ce concept pour alléger le poids des téléchargements. Nous avons poursuivi un objectif de généricité dans le code source afin d'éliminer la redondance et de permettre une forte réutilisation des composants. Grâce à cette mise en cache, les pages de l'application qui font en moyenne 1,2MB lors de la première visite ne requièrent qu'approximativement 10KB de téléchargement pour les suivantes. Une exception peut être notée ; le module d'information géographique est plus gourmand en ressources réseau en raison du téléchargement des images composant la carte.



## Gabarits, modularité et responsiveness

Nous avons vu plus tôt que l'application est constituée d'une zone de travail dans laquelle des blocs de contrôles sont placés pour former les différents modules de l'application. Ces blocs de contrôle reflètent l'architecture logicielle de l'interface.

Pour implémenter ces contrôles, nous utilisons les gabarits Django qui sont des blocs de code HTML établissant la structure de la page web. Ces différents blocs de structure sont ensuite utilisés dans l'application pour créer les différentes pages :

- En *extension*, c'est-à-dire que nous créons une nouvelle page à partir d'un gabarit existant. Les contrôles communs (figure 5.4.1-3) de l'application sont ainsi définis une seule fois et réutilisés à travers les modules.
- En *inclusion*, dans laquelle un bloc de contrôle est défini indépendamment d'une page, lui permettant de conserver le même fonctionnement en une seule implémentation à travers plusieurs pages web. Les tables et formulaires sont des exemples typiques.

## Accessibilité hors-ligne

Une contrainte importante en Haïti est la faible disponibilité de la connexion internet. Dès lors, il est nécessaire pour les futurs utilisateurs de notre application de pouvoir y accéder hors-ligne. Cela n'est d'habitude pas possible pour un site internet, cependant la fondation Mozilla a récemment développé les Services Workers [14]. Il s'agit d'un module qui s'ajoute sur la machine de l'utilisateur et vient intercepter les requêtes effectuées par le client avant qu'elles ne soient transmises au serveur, afin d'y effectuer différents traitements.

L'une des capacités du Service Worker est de coordonner la mise en cache des documents renvoyés par le serveur sur le navigateur internet de l'utilisateur. Initialement, le cache est vide. Lorsqu'une requête pour un document est reçue, ce dernier est stocké dans ce cache par le Service Worker. Par la suite, les prochaines requêtes pour ce document seront interceptées par ce module, qui ira chercher les informations directement dans le cache du navigateur au lieu de transférer la requête au serveur.

Il est également possible d'intercepter les requêtes faites au serveur lorsque celui-ci n'est pas accessible, comme lorsque la connexion internet n'est pas stable.

De cette manière, nous pouvons quand même offrir certaines fonctionnalités hors-connexion. Cependant, il est encore nécessaire de choisir quelles fonctionnalités proposer dans ce cas. Le Service Worker se trouvant sur la machine de l'utilisateur ne peut utiliser que les données qui lui sont accessibles puisque le serveur n'est pas disponible. Pour cela, nous avons imaginé trois choix possibles :

- Toutes les fonctionnalités. Cela implique d'avoir une version locale de la base de données sur la machine de l'utilisateur. Il est aussi nécessaire de synchroniser les différentes versions de celle-ci, y compris en cas d'inconsistance, si deux utilisateurs déconnectés modifient le même élément par exemple.
- Version statique des données. Cela implique également d'avoir une version locale de la base de donnée, mais en supprimant la possibilité de modifier ces données sans connexion. Cela simplifie la synchronisation et règle les problèmes d'inconsistance.
- Quelques formulaires d'ajout. En retirant la possibilité de visualiser des données, il n'est dès lors pas nécessaire de garder une base de donnée sur la machine de l'utilisateur. Reste la possibilité de remplir des formulaires d'ajout pour différentes tables, ce qui nous semble être la tâche la plus importante à avoir lorsqu'il n'y a pas de connexion, afin de ne pas perdre d'informations.

Nous nous sommes finalement décidés pour cette dernière option, qui nous semblait plus réaliste et plus en accord avec la faible connexion d'Haïti. Il est dès lors possible pour un utilisateur de compléter le rapport mensuel hors connexion. Ce module est facilement extensible à la possibilité de compléter d'autres formulaires dans le futur, voire à la possibilité d'accéder aux données s'il apparaît que cette fonctionnalité est importante pour les utilisateurs finaux de l'application.

Au final, les Services Workers nous permettent d'avoir des fonctionnalités autrement impossible sur un site web. Ils ont aussi l'avantage d'être un standard maintenant disponible sur tous les navigateurs majeurs. Cependant, cette compatibilité est relativement récente et des utilisateurs utilisant de vieilles versions de ces navigateurs risquent de ne pas avoir accès à cette technologie. Heureusement, cette fonctionnalité est entièrement optionnelle et le site fonctionne encore sans cette compatibilité. Il est également possible pour tout utilisateur de simplement mettre à jour ou changer de navigateur, s'il venait à avoir besoin des Service Workers.

## 5.5 Serveur

### 5.5.1 Requêtes

La fonction principale du serveur est de répondre aux requêtes que lui envoie le client, on en compte trois types différent. Les deux premiers types sont responsable de gérer la représentation de l'information à l'utilisateur tandis que le dernier est en charge de la gestion de l'information en elle même.

- Les requêtes de fichier, comme un script JS ou une feuille de style CSS.
- Les requêtes de document, donnant les pages HTML à afficher à l'utilisateur.
- Les requêtes de données, interagissant avec la base de données.

Les documents ne contiennent pas les données, le client effectue des requêtes après qu'ils soient rendus pour télécharger ces dernières et ainsi compléter les tables. Ceci nous permet de cacher plus efficacement les documents et fichiers (voir section 5.4) et de les rendre plus léger pour accélérer le rendu des pages HTML même en cas de faible connexion.

Nous avons décidé de séparer la logique des requêtes de données du reste de l'application en le rassemblant dans un module spécifique API. Cela nous a permis de mieux compartementaliser notre code et la logique de l'application.

Une autre possibilité est de rajouter une autre interface pour notre application. Il suffit d'interagir avec l'API pour pouvoir gérer toutes les données, sans s'embarrasser de la manière de la représenter. De cette manière, nous pourrions créer une autre application même sur une toute autre plateforme comme Android par exemple. Elle n'aurait qu'à gérer sa manière de représentation sur cette plateforme et pourrait interagir de notre première application au moyen de requêtes de données.

### 5.5.2 Mapping Objet-Relationnel

Dans la section 4.1, nous avons décrit comment nous avons créé la base de données et sa structure. Il nous reste à faire interagir notre serveur avec cette dernière. Pour cela, nous avons utilisé le Mapping Objet-Relationnel proposé par

Django. Il effectue la connexion avec le serveur de base de donnée, et se charge d'effectuer toutes les requêtes à ce dernier.

Il nous suffit de décrire la structure des données en utilisant des classes Python et l'ORM saura créer les tables et les modifier au fur et à mesure des changements que nous introduisons. Ensuite, nous pouvons facilement interagir avec ces classes pour créer, modifier, supprimer et récupérer des données sans toucher au SQL.

Cela nous permet de simplifier l'interaction avec la base de données, et cela nous permet aussi de nous détacher de quel serveur voire quel système de gestion de base de données nous utilisons. Il serait possible de passer de PostgreSQL que nous utilisons pour un autre comme MySQL en changeant simplement la configuration de l'ORM.

# Chapitre 6

## Validation

La validation rassemble les étapes du projet qui ont pour but de vérifier le fonctionnement et la qualité des productions. Les premières validations se sont essentiellement basées sur des productions écrites (cahier des charges, schémas) présentées aux acteurs d'Haïti, à Protos et à nos promoteurs pour exposer notre avancement et l'orientation du travail futur.

### 6.1 Vérifications automatiques

#### Tests unitaires

Le but d'un test unitaire est de vérifier le fonctionnement d'une partie atomique de l'application. Souvent, on utilise une fonctionnalité suivant une utilisation précise, un récit utilisateur. Dans le cas de notre application, toute interaction se fait au moyen de requêtes au serveur. Nous avons dès lors décidé d'effectuer nos tests sur le fonctionnement de ces requêtes suivant différent cas d'utilisation. Nous n'avons pas créé de tests pour les requêtes de fichier.

D'abord pour les requêtes de document, nous vérifions que ceux-ci sont correctement chargés dans plusieurs cas, comme lorsque l'utilisateur n'est pas connecté, ou avec des utilisateurs possédant différents niveaux d'autorisation.

Pour les requêtes de données, nous vérifions sur la base de donnée que son état est cohérent après que la requête se soit effectuée, ou que les données que nous avons récupéré sont correctes. Ici aussi, nous vérifions plusieurs cas, comme lorsque l'utilisateur n'a pas les droits d'utilisation ou lorsque les requêtes sont mal formées.

## Tests fonctionnels

### 6.2 Vérifications utilisateurs réels

Cette étape, qui s’est déroulée de mars à avril, a mis des utilisateurs étrangers à l’équipe de développement face à l’application. Derrière cette vérification, deux points de l’application étaient mis à l’épreuve : l’interface graphique et la procédure d’utilisation. Après plusieurs mois de développement, le point de vue d’utilisateurs externes avec des connaissances en informatique et en gestion de l’eau fortement variables était nécessaire.

Les séances de validation ont eu lieu selon les disponibilités des utilisateurs d’essai, répartis en deux groupes. Un premier groupe de 9 utilisateurs a testé l’application dans la version *Alpha 3*, un second groupe de 11 utilisateurs a testé l’application dans la version *Beta 1*, dans laquelle les modules financier et historique ont été rajoutés ainsi que plusieurs changements suggérés par les utilisateurs d’essai du premier groupe de validation.

## Méthodologie

Afin d’introduire un minimum d’erreurs statistiques, la mesure de la satisfaction des utilisateurs a suivi un protocole (protocole complet disponible en annexe E) établi avec l’expérience de nos promoteurs sur base d’un ancien travail de fin d’études [2], de cours [21, 22] et recherches sur la satisfaction utilisateur [9, 20].

Autant que possible, les séances se sont déroulées en présence physique de l’utilisateur d’essai. Certaines contraintes horaires et logistiques ont imposé la réalisation de séances à distance dans lesquelles nous avons utilisé Skype ou Discord, logiciels permettant de transmettre la voix et partager l’écran de l’utilisateur.

Après une brève introduction au concept de l’application et aux objectifs de la séance, les utilisateurs obtiennent un scénario composé d’un contexte et de tâches à réaliser qu’ils peuvent lire à leur aise. Un utilisateur travaille seul et est chronométré durant la séance. S’il a une question ou s’il reste bloqué sur une tâche, un indice lui est donné et noté dans un document avec le temps nécessité pour réaliser la tâche.

Après trois scénarios accomplis, l’utilisateur d’essai remplit un formulaire (Google Form©) anonymisé composé de 24 questions fermées réparties en quatre sections :

**Général** est une première section cherchant à identifier si l'utilisateur a compris les tâches qu'il vient d'accomplir avec l'application.

**Fonctionnalités** cherche à identifier la compréhension des contrôles de l'application par l'utilisateur.

**Usabilité** est une section particulière. Composée de dix questions alternant des assertions positives et négatives en alternance, cette section est une transcription en français de la *System Usability Scale (SUS)* [3]. Se décrivant comme une mesure rapide et sâle, la SUS présente l'avantage de pouvoir être complétée rapidement par nos utilisateurs d'essai durant ces séances de maximum deux heures non-rémunérées.

**Esthétique** est une dernière section, sans doute la plus subjective, composée de questions sur l'apparence de l'interface graphique.

L'utilisateur sélectionne la réponse sur une échelle de Likert [23] à 5 valeurs allant de *Pas du tout d'accord* à *Tout à fait d'accord*. Si l'utilisateur ne comprend pas la question ou n'a pas d'avis, il sélectionne le *neutre*.

## Résultats obtenus

Les contraintes inhérentes au cadre du mémoire, principalement en temps et budget, nous empêchent de pouvoir réaliser une étude qui donnerait une mesure plutôt précise de la satisfaction utilisateur. Néanmoins, de nos vingt réponses des utilisateurs d'essai nous pouvons tout de même extraire une certaine indication de l'expérience utilisateur d'HaïtiWater.

Les résultats des 14 questions évaluant la satisfaction des utilisateurs sur le concept, les fonctionnalités et l'esthétique de l'application sont représentés en figure 6.1.

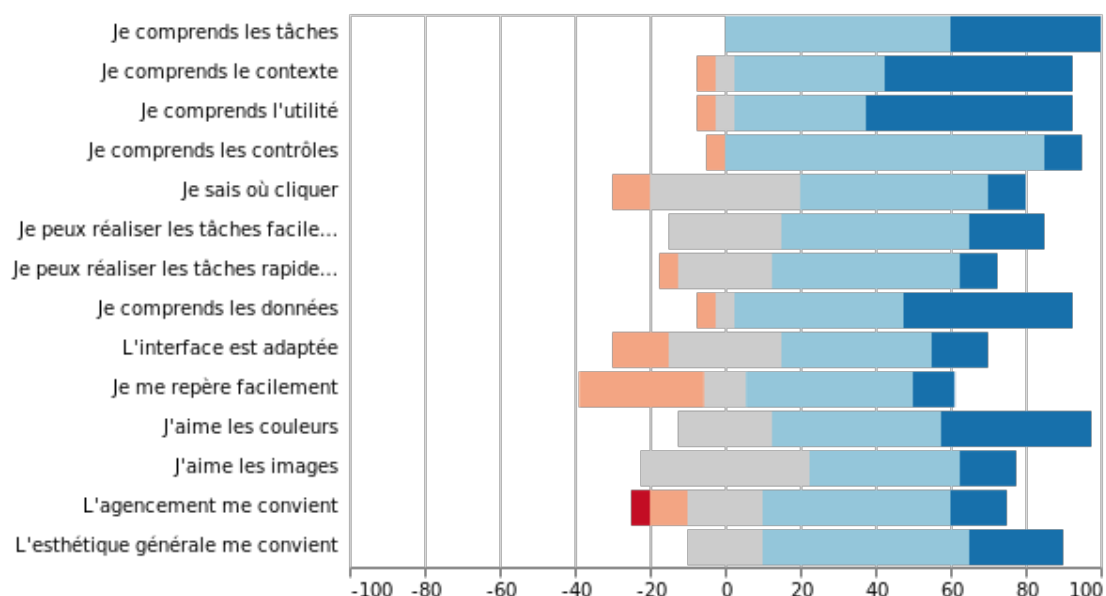


FIGURE 6.1 – Répartition des réponses aux questions du formulaire de validation

Nous pouvons constater une tendance positive dans les réponses aux questions ainsi qu'une cohérence dans les groupements : les quatre questions ayant reçu le score le plus bas sont toutes liées à une certaine complexité de l'interface.

Les dix questions de la SUS s'évaluent différemment. L'échelle a sa propre formule de calcul [3] qui permet de normaliser les résultats à une valeur de 0 à 100. Notez cependant qu'il ne s'agit pas d'un pourcentage de qualité mais d'une valeur suivant une distribution gaussienne dont la moyenne empirique se situe à 68. Le premier groupe de validation nous plaçait à 67,5, le second à 77,5 pour une moyenne à 73. Nous devons rester précautionneux quant à ces résultats, mais ils montrent que l'interface est plutôt bien reçue. L'augmentation des résultats entre les deux groupes est positive et montre que les changements déjà apportés à l'application après le premier groupe ont eu un impact positif sur l'expérience utilisateur.

## Modifications apportées

Sur base des commentaires apportés durant les séances de validation, plusieurs modifications ont été apportées à l'application. Certaines prévues mais retravaillées sur les conseils des utilisateurs, d'autres imprévues.



La figure 6.2 présente un des multiples ajouts post-validation apportés à l'application. Nous avons vu dans la section précédente que la complexité de l'interface pouvait être sa faiblesse, alors nous avons implémenté une fonctionnalité de *visite*, concept existant déjà sur de nombreuses applications web. En appuyant sur le bouton d'aide rapide dans le coin supérieur droit de l'interface, l'utilisateur déclenche une série d'astuces qui présentent brièvement les contrôles disponibles sur la page.

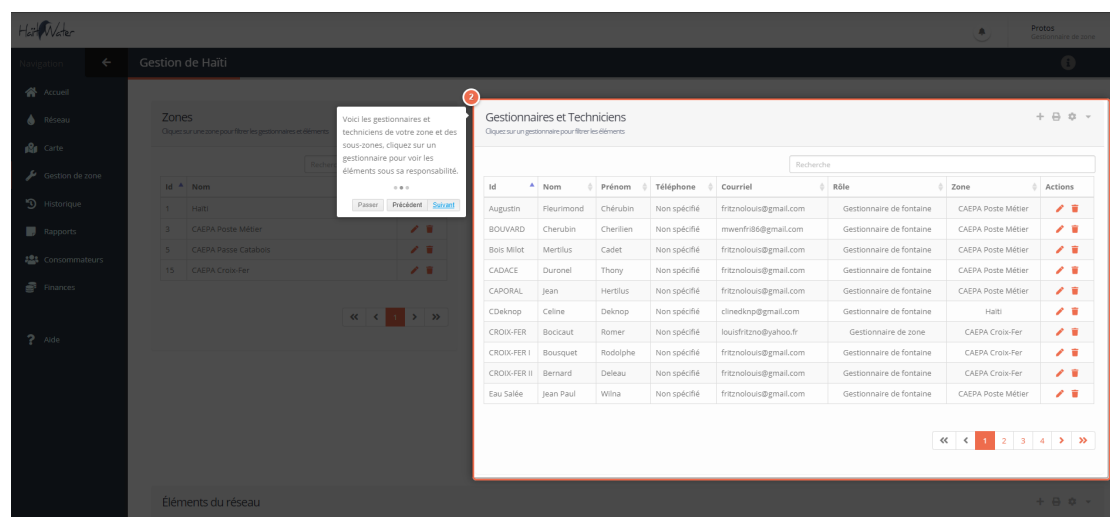


FIGURE 6.2 – Capture d'écran du module de gestion de zone avec l'aide rapide activée

D'autres modifications ont également été apportées à l'application et nous pouvons citer les plus importantes :

- Une table *bac à sable* dans la page d'aide qui permet d'utiliser une table qui n'a aucun impact sur l'application pour comprendre les contrôles.
- Des boutons de pagination pour les tables permettant d'accéder directement à la première ou dernière page d'une table.
- Un bouton permettant d'afficher ou masquer le mot de passe sur la fenêtre de connexion, particulièrement utile sur smartphones et tablettes.
- Refonte des contrôles du module géographique qui n'étaient auparavant pas bien compris. A présent les boutons sont plus explicites, certains ont changé de place pour être plus proches de la fenêtre sur laquelle ils ont un effet, et enfin l'ajout de textes d'aide dont le contenu est dynamiquement modifié selon l'action en cours.
- Une icône de chargement lors de l'envoi d'un formulaire.
- La totalité des contrôles dispose désormais de *tooltips* affichés au survol de la souris.

# Chapitre 7

## Améliorations futures

La soumission de ce travail de fin d'études ne marque pas la fin du développement de l'application Haïtiwater ni des travaux connexes. Comme tout projet de développement, un certain suivi minimum doit être assuré afin de maintenir l'outil à jour et parer aux problèmes du quotidien. De plus, HaïtiWater est considéré comme étant toujours en développement au vu des ambitions des différents acteurs souhaitant utiliser l'application.

### 7.1 Suite du projet

Poursuivre le développement d'HaïtiWater est donc la priorité des futures équipes. L'application dans son état actuel fournit une base logicielle assez large implémentant de nombreux principes (gestion de données, visuels, accès hors-connexion, etc), lesquels devraient être poursuivis pour augmenter leur qualité et quantité.

A présent que l'application va être essayée en Haïti, les futures équipes de développement devront se montrer à l'écoute des acteurs du projet et modifier l'application selon leurs demandes pour, par exemple, prendre en charge plus de données. Une autre tâche importante sera la visualisation et aggrégation des données pour fournir plus de rapports automatiques. Les graphiques déjà présents en sont un bon exemple, permettant de présenter l'information rapidement et de

manière compréhensible pour ainsi identifier directement les problèmes potentiels et améliorations possibles du réseau de distribution d'eau potable.

Le déploiement de l'application sur un serveur haïtien est un autre objectif à court terme du projet. La transition ne devrait pas être un problème puisque le serveur prêté par l'UCLouvain est un serveur de deux coeurs à 2GHz avec 1Go de mémoire vive, donc n'étant pas particulièrement performant. Cette migration du serveur de l'application devrait améliorer le temps de réponse de l'application en Haïti qui doit en ce moment systématiquement communiquer avec la Belgique.

Une autre amélioration du temps de réponse serait d'alléger les pages de l'application. Bien que ces dernières soient largement mises en cache (voir section 5.4), leur téléchargement initial et à chaque fois que le cache est vidé représente entre 1 et 1.6Mo de données. En passant l'outil *coverage* de *DevTools (Chrome)*, nous observons qu'en moyenne l'application utilise 40% des fichiers téléchargés ce qui est dû à l'importation des librairies comme Bootstrap. En supprimant les fonctionnalités inutilisées, l'application pourrait donc être allégée. Cela n'a pas été fait jusqu'à présent puisque l'application est en développement et que des fonctionnalités inutilisées pourraient le devenir. Nous avons tout de même créé un script<sup>1</sup> qui permet de raccourcir les fichiers en enlevant les fonctions inutilisées, script qui pourrait être réutilisé dans les travaux futurs sur HaïtiWater.

## 7.2 Défis rencontrés

Pages : 1 à 2

## 7.3 Propositions

Pages : 1 à 2

---

1. <https://github.com/AdrienHalletUCL/chromeCoverageParser>

## Chapitre 8

## Conclusion

# Bibliographie

- [1] Mission de la DINEPA. Site internet du gouvernement haïtien.
- [2] Richard Bastien. ReflectOn : un outil logiciel de coaching à la réflexion et l'apprentissage d'employés. Travail de Fin d'Etudes, 2014-2015.
- [3] John Brooke. SUS - a quick and dirty usability scale. *Redhatch Consulting Ltd*, 1995. Université de Genève.
- [4] PostGIS Steering Committee. Postgis.
- [5] Andrew Powell et Oscar Becerra Eduardo A. Cavallo. Estimating the direct economic damage of the earthquake in Haiti. *IDB Working Paper Series*, (IDB-WP-163), February 2010.
- [6] EPANET. Main website, 2019. Consulté le 10 mai 2019.
- [7] Greeet Schaumans et Katrien Van Hooydonk. Analyse commune de contexte Haïti, Octobre 2015.
- [8] Evens Emmanuel et Per Lindskog. Regards sur la situation des ressources en eau de la république d'haïti, Octobre 2015.
- [9] Software Sustainability Institute. Resources, feb 2019.
- [10] Manuel Kolp. Management des systèmes d'information - it management. Cours universitaire, 2016-2017.
- [11] Ky PIPE. Ky PIPE main website, 2019. Consulté le 10 mai 2019.
- [12] SpryMedia Ltd. Datatables.
- [13] Kim Mens. Conception orientée objet et gestion de données. Cours universitaire, 2016-2017.
- [14] Fondation Mozilla. Service worker api.
- [15] Comission Économique pour l'Amérique Latine et les Caraïbes. Impact socioéconomique de la dégradation des terres en Haïti et interventions pour la réhabilitation du milieu cultivé, Janvier 2008.
- [16] Ken Schwaber. Scrum website. Consulté le 10 mai 2019.

- [17] C. Deknop A. Hallet S. Strebelle. Synthèses des documents de contexte.
- [18] Web Technology Surveys. Historical yearly trends in the usage of server-side programming languages for websites, 2019. Consulté le 30 avril 2019.
- [19] Bentley Systems. WaterCAD main website, 2019. Consulté le 10 mai 2019.
- [20] Usability.gov. Usability evaluation methods, 2019. Consulté pour la dernière fois le 11 mai 2019.
- [21] Jean Vanderdonckt. Human-computer interaction. Cours universitaire, 2016-2017.
- [22] Jean Vanderdonckt. Computer-supported collaborative work. Cours universitaire, 2017-2018.
- [23] Wikipedia. Echelle de Likert, feb 2019.
- [24] Vereda Williams. A case study of the desertification of Haiti. *Journal of Sustainable Development*, 4, Juin 2011.

# Annexe A

## Cahier des charges complet

Pages : beaucoup

# Annexe B

## Base de données

Pages : beaucoup



## Annexe C

### Wireframes

Connexion

Utilisateur

Nom d'utilisateur

Mot de passe

Connexion

HaitiWater

Licence

Mentions Légales | Contact

[Logo titre]

[Utilisateur] Déconnexion

Accueil

Réseau

Techniciens

Comptabilité

Rapports

Consommateurs

Aide

La zone :  
Quartier Morin  
Grand-Pré

Centralise  
40 points d'eau

Répartition des types de fontaines

Légende :

[Couleur] : [Valeur]

• 10 Gestionnaires ont rendu leur rapport de septembre

• 2 Gestionnaires n'ont pas encore rendu leur rapport de septembre

Il y a 2 problèmes non-résolus :

• Fontaine Boniface Nord a une panne matérielle

• Kiosque Moustique Sud 3 ne distribue plus d'eau

Vous avez :

600 consommateurs

répartis sur :

475 foyers

Consommateurs n'ayant pas encore payé :

12% (72)

Recouvrement période actuelle :

60.000 HTG

HaitiWater

Licence

Mentions Légales | Contact

65

[Logo titre]

[Utilisateur]

Déconnexion

Accueil

Réseau

Techniciens

Comptabilité

Rapports

Consommateurs

Aide

Utilisateurs par Fontaine

La zone :

Groupement Nord

Approvisionnement :

6700 consommateurs

Via :

42 points d'eau

Ce mois de (septembre) nous avons distribué :

200 mètres cube d'eau

Il y a 2 problèmes non-résolus :

- Fontaine Boniface Nord a une panne matérielle
- Kiosque Moustique Sud 3 ne distribue plus d'eau

Recherche ...

Ajouter

Supprimer

Type	Emplacement	Utilisateurs	État	Quantité d'eau distribuée (mois)
Fontaine	Bois Milot	600	En service	40 m³
Kiosque	Bouvard	40	En panne	0 m³
Prise individuelle	Village, numéro, rue	6	En service	2 m³

HaitiWater

Licence

Mentions Légales | Contact

[Logo titre]

[Utilisateur]

Déconnexion

Accueil

Réseau

Techniciens

Comptabilité

Rapports

Consommateurs

Aide

Ajouter un élément du réseau

Type\* :

Kiosque

Emplacement\* :

...

Placer sur la carte

Gestionnaire :

-- vide --

\* = champ obligatoire

Annuler

Ajouter

HaitiWater

Licence

Mentions Légales | Contact

[Logo titre]

[Utilisateur]  
Déconnexion

Accueil

Réseau

Techniciens

Comptabilité

Rapports

Consommateurs

Aide

Recherche ...

Ajouter

Supprimer

Filtres ...

Élément	Adresse	État	Responsable
Fontaine	Village machin, rue du truc	En fonctionnement	Bertrand Boniface
Kiosque	Village machin, rue du truc	En fonctionnement	José Jasmin
Conduite	Lie Fontaine A à Fontaine B	En fonctionnement	Bertrand Boniface

Point sélectionné :

- Fontaine Sainte-Sophie

Consommateurs enregistrés :

- 1337

Volume fourni le mois dernier (septembre) :

- 42 m³

HaitiWater
Licence
Mentions Légales | Contact

[Logo titre]

[Utilisateur]  
Déconnexion

Accueil

Réseau

Techniciens

Comptabilité

Rapports

Consommateurs

Aide

Gestionnaire(s) de fontaines

Recherche ...

Ajouter

Supprimer

Nom	Prénom
Jean	Dubois

Plombier(s)

Recherche ...

Ajouter

Supprimer

Nom	Prénom
Marc	Manoeuvre

Points d'eau

Recherche ...

Ajouter

Supprimer

Type	Zone	Adresse	Consommateurs attribués
Fontaine	Centre-Ouest	[adresse du point d'eau]	700
Kiosque	Centre-Ouest	[adresse du point d'eau]	1.200
Fontaine	Sud-Est	[adresse du point d'eau]	2.000
Fontaine	Sud-Est	[adresse du point d'eau]	2.000

HaitiWater
Licence
Mentions Légales | Contact

[Logo titre]

[Utilisateur]  
Déconnexion

Accueil

Réseau

Techniciens

Comptabilité

Rapports

Consommateurs

Aide

Utilisateurs par Fontaine

Recherche ...

Ajouter

Supprimer

Nom et Prénom (chef de ménage)	Sexe	Adresse	Nombre de consommateurs	Type d'adduction	Nom adduction	Téléphone	Dernier paiement

Nombre de foyers consommant de l'eau :  
25.000

Nombre total de consommateurs individuels :  
50.000

Consommateurs n'ayant pas payé :  
500

HaitiWater

Licence

Mentions Légales | Contact

[Logo titre]

[Utilisateur]  
Déconnexion

Accueil

Réseau

Techniciens

Comptabilité

Rapports

Consommateurs

Aide

Vous avez :  
1  
ticket en attente de réponse

Signaler un problème

Déposer le rapport mensuel

Mes tickets

Recherche ...

Tickets actifs uniquement

# Ticket	Ordre	Emplacement	Type	Urgence	Commentaire	Statut
127	1	Fontaine Nord	Problème matériel	Urgent	La pompe ne fonctionne plus	Non-résolu (prioritaire)
128	2	Fontaine Centre 4	Qualité de l'eau	Moyen	L'eau a parfois des dépôts boueux	Non-résolu

HaitiWater

Licence

Mentions Légales | Contact

[Logo titre]

[Utilisateur]

Déconnexion

Accueil

Réseau

Techniciens

Comptabilité

Rapports

Consommateurs

Aide

Rapport Mensuel

<< Septembre 2018 >>

Points d'eau concernés

Tous mes points d'eau

Etat général

Service partiel

Entité réparatrice

Autre Technicien

Le SAEP a été en service

Heures de services

24h par jour

Jours de service

30 jours

Nombre d'usagers

2000

Recettes

Calculées

des kiosques

2000

des fontaines

2000

des branchements

2000

totales

2000

Réelles

des kiosques

2000

des fontaines

2400

des branchements

1000

totales

5400

Fontaine centre du village

Volume distribué

20 m³

=

5283 gal

Coût unitaire au volume

x /m³

=

y /gal

Branchement maison famille Boniface

Volume distribué

20 m³

=

5283 gal

Coût unitaire au volume

x /m³

=

y /gal

Kiosque sortie Nord du village

Volume distribué

20 m³

=

5283 gal

Coût unitaire au volume

x /m³

=

y /gal

Commentaire éventuel ...

Annuler Envoyer

HaitiWater

Licence

Mentions Légales | Contact

[Logo titre]

[Utilisateur]

Déconnexion

Accueil

Réseau

Techniciens

Comptabilité

Rapports

Consommateurs

Aide

Signaler un problème

SAEP Concerné

...

Type de problème

Panne matérielle

Niveau d'urgence

Urgent

Commentaire

La pompe ne fonctionne plus depuis ce matin

Annuler Envoyer

HaitiWater

Licence

Mentions Légales | Contact

[Logo titre]

[Utilisateur]  
Déconnexion

Accueil

Réseau

Techniciens

Comptabilité

Rapports

Consommateurs

Aide

Année(s) 2015 - 2018

▼

Evolution des dépenses / Revenus

Total des entrées :  
**1.000.000 HTG**

Total des sorties :  
**600.000 HTG**

Balance :  
**400.000 HTG**

Catégorie	Total (HTG)
Salaire	- 500.000
Entretien matériel	- 100.000
Revenu consommateur	+ 1.000.000

Nouvelle Catégorie

Recherche ...

Ajouter

Supprimer

Catégorie	Description	Quantité	Prix Unitaire	Total
Salaire	Employés SAEF Nord 2	10	10.000	100.000
Salaire	Employés SAEF Centre 3	10	10.000	100.000
Entretien matériel	Nouveau tuyau pour fuite	1	500	500
Revenu consommateur	Paiements mensuels zone 2	100.000	150	15.000.000

HaitiWater
Licence
Mentions Légales | Contact

[Logo titre]

[Utilisateur]  
Déconnexion

Accueil

Réseau

Techniciens

Comptabilité

Rapports

Consommateurs

Aide

Recherche ...

Questions Fréquemment Posées (type FAQ)

Manuel Utilisateur

Contact aide :  
xxx@yyy.zzz

HaitiWater
Licence
Mentions Légales | Contact

## Annexe D

# Diagrammes d'activité

Pages : beaucoup



# Annexe E

## Documents de validation

### Abstract

Ce document établit le protocole et les questions de l'expérience de validation de l'interface et des fonctionnalités de l'application, principalement du point de vue de la facilité d'utilisation et de compréhension, avec des utilisateurs hors de l'équipe de développement. En fin de document, vous trouvez les scénarios tels que les utilisateurs les verront imprimés lors des essais.

---

### E.1 Objectifs

Les expériences de validation cherchent à mettre en avant les points forts/faibles de l'interface graphique en utilisant un protocole identique, pour plusieurs participants, visant à normaliser les résultats et tenter de dégager une tendance positive ou négative. Les résultats de l'expérimentation permettront de confirmer ou infirmer certains choix techniques, esthétiques et fonctionnels. L'exactitude des résultats n'est pas évaluée lors de cette validation.

## E.2 Protocole

### Résumé

Chaque participant est introduit à l'application de manière générale sans présentation de l'interface. Chaque participant reçoit un numéro. L'expérimentation commence avec un scénario et un type d'interface (mobile ou desktop) défini. L'application est déjà ouverte sur le périphérique utilisé. Le participant doit ensuite compléter un scénario, divisé en plusieurs tâches, dans le temps imparti. Si le participant ne parvient pas à résoudre une tâche, il peut être aidé de manière brève et orale par l'expérimentateur qui notera chaque intervention de sa part. Une fois le scénario accompli, le participant réitère sur d'autres scénarios pour une durée maximale d'une heure trente au total. La dernière demie-heure est réservée à la complétion d'un formulaire en ligne composé de 24 questions utilisant une échelle de Likert [23] et à une éventuelle discussion avec les participants.

### Présentation

L'objectif de l'application est de fournir un appui logiciel aux gestionnaires du réseau de distribution d'eau potable d'Haïti. On considère qu'il y a deux groupes d'utilisateurs principaux :

- Le *gestionnaire de zone* qui coordonne l'activité dans une certaine zone géographique. Il est un responsable administratif.
- Le *gestionnaire de fontaine* qui participe au quotidien à la gestion physique du réseau de distribution. Il est au plus proche contact des consommateurs.

En tant que sujet d'expérimentation, le participant va être présenté à différentes situations qui vont lui demander d'utiliser l'application à divers degrés de responsabilité et à travers plusieurs écrans. Le participant reçoit un numéro permettant de l'identifier à travers les différents scénarios.

## Scénarios d'utilisation

L'utilisateur est placé devant l'application, ouverte sur le portail de connexion, et peut lire le scénario qui lui est proposé. Une fois prêt, l'utilisateur démarre la tâche et l'expérimentateur lance le chronomètre. Chaque participant est seul devant l'application ouverte en mode écran complet, écran simulant une tablette ou un smartphone. Le scénario (comprenant les instructions) lui est laissé à disposition. L'expérimentateur peut venir en aide au participant de manière brève et sans interagir avec le périphérique. Toute aide apportée sera notée. Une fois le scénario accompli, le temps écoulé est noté par l'expérimentateur. Si le participant ne parvient pas à compléter la tâche dans le temps imparti, l'expérimentateur interrompt l'expérience.

Après un scénario accompli, le participant peut en commencer un nouveau. Les scénarios sont choisis aléatoirement.

## Questionnaire

Après accomplissement des tâches, le participant complète le questionnaire annexe visant à évaluer sa satisfaction quant à l'utilisation du logiciel et composé des 24 questions ci-après et d'un commentaire libre. Les questions sont regroupées en catégories. Chaque question dispose de quatre choix à sélection unique basés sur une échelle de Likert [23] : --, -, Neutre, + et ++. La section usabilité se base sur la *Software Usability Scale* [3, 20], qui est une technique d'évaluation simple et rapide de l'usabilité, traduite et adaptée au contexte de l'application. Les participants doivent répondre sans trop réfléchir aux questions et de manière individuelle. Si un participant ne peut répondre à une question, il doit sélectionner le neutre comme demandé par la software usability scale. Les autres sections se basent sur des gabarits d'évaluation logicielle [9] et expérimentations précédentes [2]. En plus d'un commentaire libre final, chacune des quatre sections de questions dispose d'un champ de commentaire libre.

## Général

**G1** Je comprends les tâches demandées.

**G2** Je comprends l'utilité des tâches demandées dans le contexte de la gestion de l'eau.

**G3** Je comprends l'utilité de l'application.

## **Fonctionnalités**

- F1** Je comprends l'utilité des contrôles de l'interface, je sais à quoi m'attendre en utilisant un contrôle (bouton, champ de texte, ...).
- F2** Je sais où cliquer pour réaliser la tâche demandée.
- F4** Je peux réaliser les tâches facilement.
- F5** Je peux réaliser les tâches rapidement.
- F6** Je comprends les données qui me sont présentées à l'écran.
- F7** L'interface de l'application est adaptée aux tâches demandées.
- F8** Je sais me repérer facilement dans les contrôles et menus de l'application.

## **Usabilité**

- U1** Je pense que j'aimerais utiliser l'application fréquemment.
- U2** Je trouve l'application inutilement complexe.
- U3** Je trouve l'application facile à utiliser.
- U4** J'aurais besoin de l'aide d'une personne qualifiée pour utiliser ce système.
- U5** J'ai trouvé les différentes fonctionnalités du système bien intégrées.
- U6** J'ai trouvé l'application trop inconsistante.
- U7** Je pense que la plupart des utilisateurs apprendraient à utiliser l'application rapidement.
- U8** J'ai trouvé l'application très lourde à utiliser.
- U9** Je suis confiant en utilisant l'application.
- U10** J'ai besoin d'apprendre beaucoup de choses avant de pouvoir utiliser l'application.

## **Esthétique**

- E1** J'aime les couleurs de l'application.
- E2** J'aime les images de l'application.
- E3** L'agencement visuel (position des éléments) de l'application me convient.
- E4** L'esthétique générale de l'application me satisfait.

## E.3 Attentes

Avec cette expérimentation, nous cherchons à obtenir des vues externes sur l'application et ses fonctionnalités, récoltées de manière rigoureuse afin d'être critiques et objectifs quant aux réalisations logicielles. La phase de validation arrivant avant la phase de documentation finale, nous espérons non seulement récolter des informations sur la qualité de l'application sans documentation, mais aussi sur les points importants de la documentation et sur lesquels il faudra insister. Les objectifs sont donc de déceler les problèmes et d'envisager leurs solutions, dans l'objectif de les mettre en place dans la documentation utilisateur ou dans le logiciel (selon des critères d'utilité, expérience utilisateur et pertinence).

## Scenario 1 - Gestion

**Rôle** Administrateur principal

**Objectif** Créer une nouvelle zone et y assigner un gestionnaire

**Prérequis du système** /

**Contexte** Vous êtes Claude, membre de l'ONG Protos. L'application HaïtiWater est déjà utilisée dans plusieurs départements d'Haïti et le département de l'Artibonite souhaite pouvoir l'utiliser également. En tant que responsable de l'application, vous devez permettre au responsable de la gestion de l'eau en Artibonite de se connecter à l'application et de gérer son réseau. Pour que le réseau de l'Artibonite soit indépendant des autres réseaux, vous devez créer une nouvelle zone et y assigner le responsable de la gestion de l'eau en Artibonite.

### Informations nécessaires

Vos informations :

**Utilisateur** Protos

**Mot de passe** Protos

Gestionnaire en Artibonite :

**Nom** Registre

**Prénom** Jean

**Courriel** haitiwater.test@gmail.com

### Tâches (30 minutes)

1. Connectez-vous à l'application avec votre compte de gestionnaire : Protos.
2. L'Artibonite n'existe pas encore dans l'application, vous devez l'ajouter en tant que nouvelle zone. Pour cela, il faut aller dans l'onglet de gestion de zone et créer une nouvelle zone. Vous pouvez choisir son nom.
3. Maintenant que l'Artibonite existe en tant que zone, vous devez y assigner un gestionnaire pour que le réseau puisse s'y développer. Ajoutez donc Jean Registre en tant que gestionnaire de cette nouvelle zone afin qu'il puisse se connecter à l'application.
4. Quittez l'application en vous déconnectant pour revenir à l'écran d'accueil.

## Scenario 2 - Utiliser les tables

**Rôle** Gestionnaire de zone

**Objectif** Imprimer la liste des tous les éléments (fontaines, kiosques, ...) du réseau de la zone *CAEPA Passe Catabois* classées par volume de sortie total

**Prérequis du système** Plusieurs avec plus de dix fontaines par zone, ou la version live du serveur.

**Contexte** Vous êtes Dominique, gestionnaire de zone et supervisez les opérations au niveau national. Cette semaine, une réunion a lieu pour décider des budgets alloués à la maintenance des fontaines de la zone de l'Ouest. Pour vous aider à allouer les fonds de manière équitable, vous avez besoin de la liste des installations de cette zone, classées par nombre d'utilisateurs décroissant.

### Informations nécessaires

Vos informations :

**Utilisateur** Protos

**Mot de passe** Protos

### Tâches (30 minutes)

1. Connectez-vous à l'application avec votre compte de gestionnaire : Protos.
2. Rendez-vous dans votre page de gestion et trouvez un moyen de filtrer les éléments pour ne conserver que ceux de la zone *CAEPA Passe Catabois*.
3. Triez les éléments par nombre d'utilisateurs décroissant.
4. Imprimez les éléments en vous assurant d'avoir la liste complète affichée. Par défaut, les tables n'affichent qu'un nombre limité d'éléments (10). Vous pouvez vous arrêter une fois que le navigateur vous propose d'imprimer le bon document, il n'est pas nécessaire de l'imprimer physiquement.

## Scenario 3 - Gestionnaire de fontaine

**Rôle** Gestionnaire de fontaine

**Objectif** Ajouter des consommateurs et envoyer un rapport mensuel

**Prérequis du système** Un gestionnaire de fontaines et ses éléments du réseau : une fontaine, un kiosque et une prise individuelle.

**Contexte** Vous êtes Frédérique, gestionnaire de fontaine. Vous avez la responsabilité de trois points d'eau et d'une communauté d'utilisateurs. Votre objectif de la journée est d'enregistrer un nouveau consommateur ayant récemment déménagé dans la région, puis d'envoyer votre rapport mensuel.

### Informations nécessaires

Vos informations :

**Utilisateur** frederique

**Mot de passe** gestionfontaine

Le nouveau consommateur est Monsieur Alain Proviste. Il vit seul au 32, Rue du foin, Belleville. Nous n'avons pas son numéro de téléphone et il habite juste à côté de la seule fontaine du village.

### Tâches (30 minutes)

1. Connectez-vous sur l'application avec votre compte.
2. Ajoutez le nouveau consommateur récemment arrivé
3. Envoyez le rapport du mois en cours Informations mensuelles :

**Activité** 22 jours, 14 heures par jour.

**Kiosque** 400 gallons de volume total à 10 gourdes le gallon

**Fontaine** 2 mètres cubes de volume total à 500 gourdes le mètre cube

**Prise individuelle** un demi mètre cube à 500 gourdes le mètre cube

**Recettes** 0 gourdes pour le kiosque et la fontaine, 150 gourdes pour la prise individuelle

4. Déconnectez-vous de l'application



## Scenario 4 - Système d'Informations Géographiques

**Rôle** Gestionnaire de zone

**Objectif** Dessiner une partie du réseau sur la carte

**Prérequis du système** Connecté et sur la page d'accueil. Aucun élément dans le système.

**Contexte** Vous êtes le gestionnaire d'une zone nouvellement établie. Avant de créer les gestionnaires de fontaine qui vont vous aider dans votre travail, vous souhaitez avoir votre réseau encodé dans l'application. Vous allez donc prendre un peu de temps pour ajouter les éléments un par un dans le système et sur la carte.

### Informations nécessaires

Votre réseau de distribution de l'eau potable est composé de :

**Une source** qui pompe l'eau dans le circuit. Elle est située dans le lieu-dit *Cabaret*. Nous avons les coordonnées précises : 19°33'56.6"N, 72°00'32.5"W.

**Une fontaine** dont nous n'avons pas les coordonnées mais qui se situe en aval de la rivière, sur la route au premier carrefour après la pompe.

**Une conduite** qui relie les deux.

### Tâches (30 minutes)

1. Vous êtes sur la page d'accueil, rendez-vous sur la page de gestion géographique du réseau.
2. Créez la source *Cabaret* dans les éléments du réseau et ajoutez sa position via ses coordonnées géographiques.
3. Créez la fontaine du carrefour dans la base de données, et placez-la sur la carte, utilisez la source déjà placée pour trouver le prochain carrefour au nord.
4. Créez une conduite dans les éléments, et placez la sur la carte pour relier les deux autres éléments que vous venez de créer.
5. Une fois que les trois éléments sont encodés dans la base de données et placés sur la carte, déconnectez-vous.

## Scenario 5 - Première connexion et problème

**Rôle** Gestionnaire de fontaine

**Objectif** Se connecter pour la première fois, changer de mot de passe et signaler un problème

**Prérequis du système** Un utilisateur avec un mot de passe autogénéré et un élément

**Contexte** Vous êtes Jean, un gestionnaire de fontaine nouvellement nommé, félicitations ! Vous venez à l'instant de recevoir un courriel vous donnant vos informations de connexion. Le mot de passe généré est impossible à retenir alors vous allez le modifier. Vous allez également en profiter pour rapporter un problème mineur : votre fontaine produit des sons étranges quand on arrête de pomper.

### Informations nécessaires

Vos informations :

**Utilisateur** GestionnaireFontaine

**Mot de passe** [Donné par l'examineur]

### Tâches (30 minutes)

1. Connectez-vous à l'application à l'aide de votre mot de passe autogénéré.
2. Modifiez votre mot de passe. Vous adorez le mot de passe "123456".
3. On dirait qu'il y a un problème avec votre mot de passe. Trouvez une solution.
4. Reconnectez-vous sur l'application avec votre nouveau mot de passe.
5. Rapportez le problème décrit dans le contexte afin que votre hiérarchie en soit informée.

## Scenario 6 - Transaction financière

**Rôle** Gestionnaire de zone

**Objectif** Se connecter et ajouter des paiements pour deux consommateurs.

**Prérequis du système** Deux consommateurs dans la zone utilisateur (noms ci-dessous).

**Contexte** Vous êtes John, un membre de l'administration du système Haïti-Water. En visite dans un village, vous en profitez pour aider un gestionnaire de fontaine local dans son encodage des paiements. Deux consommateurs viennent vous voir, billets en main, et viennent payer leur redevance.

### Informations nécessaires

Vos informations :

**Utilisateur** Protos

**Mot de passe** Protos

### Tâches (30 minutes)

1. Connectez-vous à l'application à l'aide de vos informations.
2. Rendez-vous dans la gestion financière de l'application.
3. *Monsieur Sadrac Mérisier* vous donne 150 Gourdes, entrez ce paiement dans l'application.
4. *Madame Roselène Monestime* vous donne 100 Gourdes, faites de même.
5. A sa demande, informez *Madame Roselène Monestime* de l'état général de ses paiements (dites à votre examinateur à combien de Gourdes s'élève son retard ou son avance sur les paiements).
6. Déconnectez-vous de l'application.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN  
École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | [www.uclouvain.be/epl](http://www.uclouvain.be/epl)