



**CONCEPTION DE TABLEAUX DE BORD ADAPTATIFS POUR LES
HABITATS INTELLIGENTS**

PAR LUCAS GONZALEZ

**MÉMOIRE PRÉSENTÉ À L'UNIVERSITÉ DU QUÉBEC À CHICOUTIMI EN
VUE DE L'OBTENTION DU GRADE DE MAÎTRE ÈS SCIENCES (M.SC.) EN
INFORMATIQUE**

QUÉBEC, CANADA

© LUCAS GONZALEZ, 2022

RÉSUMÉ

Le vieillissement de la population apporte des transformations sociales qui affectent de plus en plus nos sociétés. Les évolutions technologiques soutiennent cette évolution sociale profonde et ce phénomène est destiné à croître dans les prochaines années. Un des effets de cette transformation est le nombre croissant de personnes en perte d'autonomie qui nécessite un soutien et une assistance au quotidien. Les habitats intelligents se présentent de plus en plus comme une solution intéressante pouvant répondre à une partie du problème. En effet, ils disposent d'une panoplie de capteurs et de fonctionnalités qui, combinés à l'intelligence artificielle, ont le potentiel d'assister leurs occupants au quotidien. Cependant, la complexité des systèmes et la nature brute des données ne sont pas abordables pour tous les publics. Au centre de cette solution, plusieurs sujets sont abordés : la visualisation de données, la conception d'applications et l'interfaçage humain-machine. Cette composition vise à proposer une conception d'applications modulables et flexibles afin d'assister de manière appropriée les utilisateurs du système. Parmi les solutions exploitables, les tableaux de bord ont généré beaucoup d'intérêt car ils offrent un équilibre entre adaptabilité et facilité d'utilisation. Les travaux couverts par ce mémoire visent à proposer des outils de visualisation adaptatifs pour les habitats intelligents en mettant l'accent sur la modularité des composants.

Pour ce faire, plusieurs étapes ont été implémentées. Dans un premier temps, une méthode de modularisation des applications a permis de diviser le problème en plusieurs modules et d'échanger leurs résultats de manière bidirectionnelle. Dans un second temps, nous avons développé une application qui intègre ces modules. Finalement, nous avons conçu une application de visualisation capable de fournir des données adaptées aux besoins. Enfin, nous avons testé l'application auprès de plusieurs participants dans le but d'évaluer l'efficacité de notre méthode. Nous avons obtenu un score de 85.3 en termes d'utilisabilité selon notre questionnaire SUS.

ABSTRACT

The aging of the population brings social transformations that increasingly affect our societies. Technological developments support this profound social evolution and this phenomenon will undoubtedly grow in the coming years. One of the effects of this transformation is the growing number of people with a loss of autonomy who require support and assistance on a daily basis. Smart homes are increasingly presenting themselves as an interesting solution that can solve part of the problem. Indeed, they have a range of sensors and functionalities which, combined with artificial intelligence, have the potential to assist their occupants on a daily basis. However, the complexity of the systems and the raw nature of the data are not affordable for all audiences. At the center of this solution, several topics are covered: data visualization, application design and human-machine interfacing. This composition aims to propose a design of modular and flexible applications in order to appropriately assist the users of the system. Among the actionable solutions, dashboards have generated a lot of interest because they offer a balance between adaptability and ease of use. The work covered by this master aims to provide adaptive visualization tools for smart homes with an emphasis on the modularity of components.

To do this, several steps have been implemented. At first, a method of modularization of the applications made it possible to divide the problem into several modules and to exchange their results in a bidirectional way. Secondly, we developed an application that integrates these modules. Finally, we designed a visualization application capable of providing data adapted to the needs. Finally, we tested the application with several participants in order to assess the effectiveness of our method. We obtained a score of 85.3 in terms of usability according to our SUS questionnaire.

TABLE DES MATIÈRES

RÉSUMÉ	ii
ABSTRACT	iii
LISTE DES TABLEAUX	vi
LISTE DES FIGURES	vii
LISTE DES ABREVIATIONS	ix
REMERCIEMENTS	x
CHAPITRE 1 - INTRODUCTION	1
1.1 - CONTEXTE DE LA RECHERCHE.....	1
1.1.1 - VIEILLISSEMENT DE LA POPULATION.....	1
1.1.2 - ENVIRONNEMENT INTELLIGENT	2
1.1.3 - VISUALISATION DES DONNÉES	3
1.2 - PROBLÉMATIQUE DE RECHERCHE.....	4
1.3 - MÉTHODOLOGIE DE RECHERCHE	6
1.4 - CONTENU DU MÉMOIRE.....	8
CHAPITRE 2 - REVUE DE LITTÉRATURE	11
2.1 - LA VISUALISATION DE DONNÉES	11
2.1.1 - IMPACT DE LA VISUALISATION DE DONNÉES	12
2.1.2 - EVALUATION ET UTILISABILITÉ.....	15
2.1.3 - ADAPTABILITÉ ET MODULARITÉ.....	18
2.2 - LES HABITATS INTELLIGENTS	21
2.2.1 – CASAS	22
2.2.2 - MOBILE AND PERVASIVE COMPUTING LAB	25
2.2.3 - DOMUS	28
2.3 - CONCLUSION.....	32
CHAPITRE 3 - DÉVELOPPEMENT D'APPLICATION POUR LES HABITATS INTELLIGENTS	34
3.1 - MÉTHODE DE CONCEPTION PAR MODULE	37
3.1.1 - WEBSOCKETS CLIENT SERVEUR.....	38
3.1.2 - MODULARISATION D'UN PROGRAMME.....	40
3.2 - APPLICATION D'INTÉGRATION DES HABITATS INTELLIGENTS	41

3.2.1 - CHOIX DES TECHNOLOGIES	42
3.2.2 - INTÉGRATION DES MODULES EXISTANTS	44
3.2.3 - COLLECTE DE DONNÉES	45
3.2.4 - PRÉSENTATION DE L'APPLICATION.....	47
3.3 - TABLEAU DE BORD POUR LES HABITATS INTELLIGENTS.....	49
3.3.1 - ARCHITECTURE DE DÉVELOPPEMENT (MVC).....	51
3.3.2 - CHOIX DES TECHNOLOGIES	53
3.3.2.1 - FRAMEWORK FRONT-END	54
3.3.2.2 - API.....	58
3.3.2.3 - BASE DE DONNÉES	62
3.3.3 - DÉVELOPPEMENT DES INTERFACES.....	66
3.3.4 - DÉFIS DE CONCEPTION	68
3.3.4.1 - MODULARITÉ DES GRAPHIQUES.....	69
3.3.4.2 - DIFFUSION DE DONNÉES EN TEMPS RÉEL.....	72
3.3.5 - PRÉSENTATION DE L'APPLICATION.....	76
CHAPITRE 4 - RÉCOLTE DE DONNÉES.....	82
4.1 - TEST D'UTILISABILITÉ (SUS)	82
4.2 - DÉROULEMENT DES ENTREVUES.....	84
4.3 - ANALYSE DES RÉSULTATS.....	86
CHAPITRE 5 - CONCLUSION.....	91
5.1 - REVUE DES CONTRIBUTIONS	91
5.2 - LIMITES.....	93
5.3 - TRAVAUX FUTURS.....	93
BIBLIOGRAPHIE.....	95
ANNEXE A - CERTIFICAT D'APPROBATION ÉTHIQUE.....	102

LISTE DES TABLEAUX

TABLEAU 4.1: SUS ORIGINALE VS NOTRE SUS.....	84
TABLEAU 4.2 : DOMAINE D'ÉTUDE DE NOTRE ÉCHANTILLON.....	87
TABLEAU 4.3 : RÉSULTAT DU QUESTIONNAIRE D'UTILISABILITÉ.....	89

LISTE DES FIGURES

FIGURE 3.1 - TECHNOLOGIES ET ARCHITECTURE COMPLÈTES DU PROJET	36
FIGURE 3.2 : REQUÊTES DE TYPE LONGUES	39
FIGURE 3.3 : ECHANGE DE DONNÉES WEBSOCKETS	39
FIGURE 3.4 : CODE D'INITIALISATION POUR UN WEBSOCKETS	40
FIGURE 3.5 : MODÈLE RÉPÉTEUR CLIENT-SERVEUR	41
FIGURE 3.6 : RASSEMBLEMENT DES DONNÉES VS MONGODB CLOUD	47
FIGURE 3.7 : APPLICATION DE GESTION DES MODULES (INTERFACE 1).....	48
FIGURE 3.8 : APPLICATION DE GESTION DES MODULES (INTERFACE 2).....	49
FIGURE 3.9 : STATE OF JS, UTILISATION DES FRAMEWORK FRONT END....	56
FIGURE 3.10 : INSTANCIATION D'UN COMPOSANT VUE.....	57
FIGURE 3.11 : COMPOSANT EN UN SEUL FICHIER (COMPOSANT.VUE).....	58
FIGURE 3.12 : INITIALISATION D'UN SERVEUR WEB API.....	59
FIGURE 3.13 : ROUTAGE CLASSIQUE ET ROUTAGE PAR ATTRIBUT	60
FIGURE 3.14 : REQUÊTE ET SESSION UTILISATEUR.....	61
FIGURE 3.15 : SCHÉMA DE VALIDATION DE DONNÉES MONGOOSE	63
FIGURE 3.16 : FILTRAGE DE DONNÉES MONGOOSE	64
FIGURE 3.17 : RÉCUPÉRATION ET SUPPRESSION EN CASCADE MONGOOSE	65
FIGURE 3.18 : PREMIÈRES ÉBAUCHES D'INTERFACES	67
FIGURE 3.19 : DEUXIÈME ITÉRATION D'INTERFACES	68
FIGURE 3.20 : GRAPHIQUE ET TYPE DE DONNÉES PLOTLY.JS.....	70
FIGURE 3.21 : FORMATAGE ET ADAPTABILITÉ DES DONNÉES POUR LA FLEXIBILITÉ DES GRAPHIQUES.....	71
FIGURE 3.22 : PLOTLY GRAPHIQUE FILTRAGE DE DONNÉES	72

FIGURE 3.23 : ARCHITECTURE STREAMING SALLE D'ÉCOUTE.....	74
FIGURE 3.24 : SOCKET.IO ET CHANGE STREAM	76
FIGURE 3.25 : INTERFACE CONNEXION ET INSCRIPTION	77
FIGURE 3.26 : RÉSIDENCES VUE GLOBALE.....	78
FIGURE 3.27 : RÉSIDENCES VUE GLOBALE.....	79
FIGURE 3.28 : DASHBOARD RÉSIDENTS PROFILE	80
FIGURE 3.29 : INTERFACE FORMULAIRE CRUD.....	81
FIGURE 4.1 : RÉPARTITION DU DOMAINE D'ÉTUDE.....	87
FIGURE 4.2 : CALCULE DU SCORE D'UTILISABILITÉ	90

LISTE DES ABREVIATIONS

ADL	Activité de la vie quotidienne
API	Application Program Interface
CNN	Convolutional Neural Network
CRUD	Create, Read, Update, Delete
DOM	Document Object Model
DOMUS	Domotique et informatique Mobile à l'Université de Sherbrooke
FTP	File Transfer Protocol
HCI	Interface humain-machine
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
JSX	JavaScript Syntax Extension
LAMP	Linux, Apache, MySQL, PHP
LIARA	Laboratoire d'Intelligence Ambiante pour la Reconnaissance d'Activités
LSTM	Long Short-Term Memory
MEAN	MongoDB, Express, Angular, NodeJs
MERN	MongoDB, Express, React, NodeJs
MEVN	MongoDB, Express, VueJs, NodeJs
MPCL	Mobile and Pervasive Computing Lab
MVC	Model View Controller
MVVM	Model-View-View-Model
ONU	Organisation des Nations unies
REST	Representational State Transfer
SEQ	Single Ease Question
SQL	Structured Query Language
SSH	Secure Shell
SUS	System Usability Scale
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UQAC	Université du Québec à Chicoutimi

REMERCIEMENTS

Tout d'abord, je tiens à remercier M. Kévin Bouchard, professeur au Département d'Informatique et de Mathématique (DIM) de l'Université du Québec à Chicoutimi (UQAC) et membre actif du Laboratoire d'Intelligence Ambiante pour la Reconnaissance d'Activités (LIARA), pour m'avoir soutenu durant mon parcours universitaire, pour m'avoir offert de formidables opportunités mais aussi la possibilité de participer à des projets de recherche sous sa direction.

J'aimerais également remercier M. Julien Maître, professeur au DIM de l'UQAC et membre actif du LIARA, et M. Sébastien Gaboury, professeur au DIM de l'UQAC et directeur du LIARA, pour leur aide et leur commentaire en tant que co-directeurs du projet.

Finalement, je voudrais remercier l'entièreté de l'équipe du LIARA pour leur accueil au laboratoire.

CHAPITRE 1

INTRODUCTION

1.1 - CONTEXTE DE LA RECHERCHE

1.1.1 - VIEILLISSEMENT DE LA POPULATION

En 1950, on comptait un total de 2,5 milliards d'habitants sur la planète pour une évolution jusqu'à 7,7 milliards en 2019 (Roser, 2013), soit une augmentation de 208% en moins de 70 ans. D'ici la fin du siècle, l'ONU (Organisation des Nations unies) s'attend à une population mondiale de 11,2 milliards. De plus, les progrès de notre société dans le domaine de la santé ont permis d'augmenter l'espérance de vie de nos populations. En effet, selon la révision du *World Population Prospects 2019* du département des affaires économiques et sociales de l'Organisation des Nations Unies (United Nations Department of Economic and Social Affairs, 2019), une personne sur six dans le monde aura plus de 65 ans (16% de la population) d'ici 2050, contre une personne sur onze en 2019 (9%).

Le vieillissement de la population est souvent caractérisé par l'apparition de problèmes de santé complexes, plus communément appelés *syndromes gériatriques* (accumulation de déficits et de maladies de plusieurs systèmes, menant à une perte des fonctions et d'indépendance fonctionnelle). Cette perte d'indépendance peut affecter la capacité d'une personne à effectuer des activités de la vie quotidienne comme : faire le ménage, se préparer à manger, prendre des médicaments et autres tâches liées à l'hygiène de vie personnelle. De ce fait, il n'est pas rare de constater que cette perte d'indépendance nuit considérablement à la capacité d'un individu à maintenir son domicile et sa personne (Naud et al., 2021).

En effet, le vieillissement de la population est une transformation sociale qui impacte de plus en plus différents niveaux de la société. Au niveau mondial, cette transformation a des répercussions dans presque tous les secteurs d'activités, notamment sur la demande de biens et services, ainsi que sur les logements et leur utilisation (Rheault & Poirier, 2012).

1.1.2 - ENVIRONNEMENT INTELLIGENT

Pour répondre à l'augmentation et à l'évolution de ces besoins, plusieurs chercheurs et autres acteurs privés tentent de proposer des solutions en installant des capteurs dans les lieux de résidence afin de répondre au besoin d'assistance et de suivi des personnes en perte d'autonomie (Chan et al., 2009).

La notion d'habitat intelligent (Marikyan & al., 2019) peut être définie comme étant une résidence standard équipée de diverses technologies de capteurs et effecteurs, visant à fournir des services adaptés à son ou ses résidents. Un habitat pourrait ainsi suivre, accompagner et supporter ses bénéficiaires, dans le but d'améliorer leur qualité de vie et de promouvoir leur autonomie. Cook et Das (2004) définissent également les environnements intelligents comme étant des espaces de vie qui ont la capacité d'extraire et exploiter des informations, dans le but d'améliorer l'expérience d'une personne dans ce même environnement.

En effet, au cours de ces dernières années, on a pu remarquer une augmentation importante de la quantité de capteurs et de micro-ordinateurs. Cela ouvre, à presque tous les objets de la vie courante, la capacité de déclencher un échange spontané

d'informations. Supportés par la diminution des coûts et de la taille des équipements numériques, les habitats intelligents peuvent maintenant regrouper une grande variété de capteurs destinés à récolter de l'information. Par exemple, on peut penser à des capteurs de température ambiante, du niveau de lumière, du niveau sonore ou encore à d'autres types d'interactions avec des objets communs. Ces capteurs sont souvent associés à des effecteurs, c'est-à-dire, des mécanismes informatisés qui interagissent avec l'environnement en fonction des informations qu'ils reçoivent. Par exemple, des systèmes de chauffage programmables où la régulation dépend de la température ambiante détectée par le thermostat (Vijayan et al., 2020).

Finalement, la modernisation des habitats répond en partie à ce besoin d'assistance. D'une part, elle facilite grandement les tâches de la vie quotidienne et d'autre part, c'est une source importante d'informations. Ainsi, des professionnels du domaine s'intéressent également aux données qui sont émises par ces capteurs dans le but d'analyser leur contenu et par la suite de mieux répondre aux potentiels besoins des usagers.

1.1.3 - VISUALISATION DES DONNÉES

En effet, dans le contexte des habitats intelligents, des sources potentielles de données sont présentes de manière constante. En effet, The National Human Activity Pattern Survey (Klepeis & al., 2001) souligne que le temps que nous passons en intérieur représenterait plus de 90% de notre journée. Les espaces intérieurs et les habitats sont des lieux qui concentrent une production phénoménale de données potentielles. Étant donné que ces informations sont des indicateurs pertinents pour les systèmes de reconnaissance d'activités, elles permettraient de répondre aux besoins des personnes ayant besoin d'une assistance et un suivi plus personnalisés.

Cependant, la quantité de données générée dans toutes sortes de domaines ne cesse d'augmenter de manière exponentielle. On note, par exemple, que 90 % du Big Data a été généré dans les deux dernières années (Rawat & Yadav, 2021). Toutefois, le volume et la quantité de données sont tels qu'on peut se questionner sur leur niveau de pertinence. Avant d'être utilisées à des fins d'information, de décision ou de recherche, celles-ci doivent être rassemblées, organisées puis analysées.

Ainsi la visualisation de données vise à rendre plus accessible la compréhension de l'information grâce à des représentations graphiques. Elle permet de mettre en lumière des informations complexes qui ne sont pas toujours possibles ou évidentes à déduire à partir du jeu de données brutes. Le problème que nous avons identifié, pour ce travail de maîtrise, est qu'il n'existe que très peu de travaux scientifiques sur la visualisation des données en provenance des habitats intelligents.

Malgré cette lacune, les tableaux de bord sont aujourd'hui une solution largement adoptée à travers secteurs d'activité incluant celui qui nous concerne. Ils nous permettent non seulement de visualiser les données mais aussi d'interagir avec celles-ci que ce soit dans la construction de graphiques personnalisés ou de visualisations en temps réel.

1.2 - PROBLÉMATIQUE DE RECHERCHE

Aujourd'hui, l'habitat du futur est une réalité plus proche qu'on ne le pense. En effet, il existe déjà de nombreuses habitations où le chauffage, la climatisation, l'éclairage et tout autre mécanisme sont en partie gérés par un système informatique dédié. La quantité de données numériques produite par jour ne cesse d'augmenter et c'est un phénomène qui n'échappe à aucun domaine d'activité.

Ainsi, cela implique de nouvelles problématiques concernant la diversité des données qui peuvent être récoltées et analysées selon les contextes (Chen et al., 2013). Pour aider à la compréhension et l'utilisation de ces masses d'informations brutes, les tableaux de bord demeurent une solution largement adoptée par les entreprises comme les particuliers (Pankaj, Hyde, & Rodger, 2006), et ce, que ce soit dans le domaine médical, financier ou encore dans le numérique de manière générale. Bien qu'ils représentent une solution efficace pour la généralisation et la vue d'ensemble qu'ils apportent sur les données, leur conception implique de nombreux défis (Alhamadi, 2020). En effet, ils visent principalement à répondre à la compréhension des données, mais peuvent aussi servir comme outils de comparaison ou de supervision. Ainsi, le principal défi auquel on peut être confronté est la rigidité de la manipulation des données. Les composants d'un tableau de bord sont souvent très spécifiques et peu modulaires. Par conséquent, les projets doivent généralement être repris de zéro d'un contexte à l'autre pour fournir des outils adaptés.

C'est pourquoi il serait intéressant de proposer un ensemble d'outils logiciels pour permettre la conception de tableaux de bord flexibles et personnalisables dans l'optique qu'il s'adapte aux besoins spécifiques de chaque utilisateur (Elshehaly et al., 2021). Un jeu de données brutes dans son ensemble, est difficilement utilisable sans outils d'analyse et d'interprétation. C'est une des problématiques que l'on retrouve dans le domaine du Big Data. De ce fait, ces outils sont très importants pour déduire de nouveaux résultats ou phénomènes provenant des données.

Au laboratoire LIARA, la notion de *smart home* et d'intelligence ambiante sont des éléments qui motivent fortement l'équipe de recherche. Les travaux récents de l'équipe sur les habitats intelligents (Maître et al., 2021) ouvrent une perspective

intéressante sur le potentiel de l'utilisation de la détection d'activités afin d'aider et supporter des résidents. Dans ces travaux, l'utilisation de radars UWB (Ultra Wideband) qualifiés de non intrusifs, permet de récupérer des données dans le but de déduire des activités de la vie courante, les déplacements utilisateurs, mais aussi les chutes potentielles. Aujourd'hui, les résultats de l'équipe de recherche sont d'autant plus optimistes qu'ils permettent d'identifier un panel de 15 activités de la vie quotidienne avec un taux de reconnaissance d'environ 80%. Conjointement à ces travaux précédents, l'étudiant-chercheur Alexandre Beaulieu, a tenté d'améliorer ce taux de reconnaissance. Pour ce faire, deux approches ont été testées. La première est une combinaison d'un système expert et d'un réseau de neurones. La seconde est un modèle d'apprentissage profond combinant deux architectures différentes : LSTM (Long Short-Term Memory) et CNN (réseau de neurones convolutifs). Ce modèle, appelé Tuned EfficientNetB0 avec LSTM, offre une amélioration de 18,63 % du taux de reconnaissance par rapport aux réseaux de neurones développés dans des travaux antérieurs tout en exploitant le même jeu de données (Beaulieu, 2021).

Cependant, la plupart des projets qui appartiennent à ce domaine ne se sont pas encore penchés sur l'aspect visualisation. Ce projet de recherche vise à combler cette lacune en proposant un ensemble d'outils logiciels pour permettre la conception de tableaux de bord qui offrent un équilibre entre adaptabilité et facilité d'utilisation, tout restant centré sur le contexte des habitats intelligents.

1.3 - MÉTHODOLOGIE DE RECHERCHE

Le mémoire présenté dans ce document a été réalisé en respectant une méthodologie scientifique divisée en 4 phases principales. La première partie de cette recherche consiste à recenser, de manière générale, les différentes méthodologies et

travaux existants concernant la visualisation de données et la construction des tableaux de bord. Ce travail a permis d'approfondir les critères importants à évaluer tels que : l'ergonomie, l'adaptabilité, l'utilisabilité, la flexibilité, *etc.* Cette étape s'est faite sous la forme d'une revue de la littérature qui apporte une meilleure compréhension du domaine en identifiant les limitations et les différentes solutions contenues dans l'état de l'art.

La deuxième phase de ce mémoire consiste à modéliser une première proposition de tableau de bord ainsi que ses composants. C'est une phase de préconception importante dans les projets de développement d'application car elle permet d'établir une vision d'ensemble plus large de la solution. Cela comprend les questions d'architecture, de pertinence, de faisabilité et de défis, sans avoir à se confronter directement au code.

La troisième phase consiste à développer l'application à partir des documents de conception, sans perdre de vue les concepts de flexibilité et de réutilisabilité pour le code qui devra être produit. Le développement de la solution est effectué avec le langage de programmation Typescript et le cadriciel Web VueJs pour l'application de tableau de bord mais aussi des bibliothèques ExpressJs, mongoose et React pour les modules accompagnant l'application. Ce sont, pour l'ensemble, des bibliothèques ou des cadriciels en accès libre, connus depuis plusieurs années et qui ont fait leurs preuves dans l'industrie du développement d'applications.

La dernière phase a consisté à valider l'utilisabilité de l'application. Pour ce faire, dans le cadre d'une collecte de données, plusieurs utilisateurs ont testé l'application et ont pu l'évaluer à l'aide d'un questionnaire d'utilisabilité. Les résultats ont été analysés, interprétés, apportant ainsi une meilleure idée des résultats de notre application.

1.4 - CONTENU DU MÉMOIRE

Dans un premier temps, le principe de modularisation a été exploré. L'objectif étant d'améliorer l'indépendance de nos programmes dans la résolution d'un problème complexe, un modèle basé sur l'architecture client-serveur a été incorporé à nos travaux de recherche. Une limitation fondamentale de cette approche reste qu'elle n'a pas été testée en dehors du contexte contrôlé du laboratoire de recherche. Cependant, cette méthode offre un compromis entre fiabilité, stabilité et facilité d'intégration. Cette approche nous a permis de faciliter la segmentation de nos programmes et de souligner l'importance d'un système de communication bidirectionnelle vis-à-vis de la propagation des résultats. Il ne s'agit pas d'une contribution majeure, mais d'une approche qui a été explorée dans le cadre du mémoire pour supporter les travaux qui le composent.

Dans un second temps, le développement d'une application multiplateforme a été explorée. L'objectif était d'intégrer les modules secondaires que nous avons développés. Ainsi, nous avons exploré la technologie Electron¹ capable de déployer une application sur l'ensemble des principaux systèmes d'exploitation. Cette méthode nous a permis de fluidifier le développement de nos programmes et d'ouvrir l'horizon sur la combinaison des technologies que nous pourrions utiliser dans nos projets afin de réduire au maximum les limitations de choix technologiques lors de l'élaboration de solutions. De même, il ne s'agit pas ici d'une contribution majeure, mais d'une approche qui a été explorée dans le cadre du mémoire.

Pour terminer, nous avons développé une application de visualisation de données pour les habitats intelligents qui s'inscrit dans une méthode basée sur la modularité des

¹ <https://www.Electronjs.org/>

composants. Cette méthode nous a permis de définir une ligne directrice dans la conception des applications de visualisation modulaire et d'obtenir un prototype fonctionnel testé par un panel de 20 participants. Les résultats des tests ont démontré un score d'utilisabilité de 85,3 sur 100, ce qui qualifie l'application d'acceptable selon le *System Usability Scale*.

Ce mémoire est composé de 5 chapitres. Le premier est une introduction présentant le contexte de la recherche, la problématique et les différents sujets touchés par le mémoire.

Le second est une revue de littérature sur la visualisation de données et les sujets qui en découlent, afin de clarifier les différents concepts introduits par les travaux présentés.

Le troisième chapitre de ce mémoire concerne la méthode de développement de l'application utilisée dans le cadre de cette recherche, soit : les choix des technologies et de l'architecture, leurs différents avantages et particularités, les présentations de cas concrets et leurs défis de conception.

Le quatrième chapitre de ce mémoire est consacré à l'évaluation de l'utilisabilité de l'application. On y trouve la présentation des outils utilisés pour effectuer cette évaluation, tout comme une explication du déroulement des entrevues et les conditions dans lesquelles elles ont été effectuées. Enfin, une section est consacrée à l'analyse et l'interprétation des résultats.

Le dernier chapitre sert quant à lui à effectuer une synthèse générale sur les contributions du mémoire, d'identifier leurs limites et leur ouverture sur de potentiels travaux à venir

CHAPITRE 2

REVUE DE LITTÉRATURE

Les travaux présentés dans ce mémoire se concentrent principalement sur le domaine de visualisation de données et sur les habitats intelligents. De ce fait, il est important de faire un tour d'horizon des fondements de ces disciplines. Dans un premier temps, la visualisation de données sera abordée en détaillant trois grands concepts. Tout d'abord, une revue de littérature sur l'impact de la visualisation de données, suivie de son évaluation et son principe d'utilisabilité et, pour conclure, les travaux récents sur l'adaptabilité et la modularité de la visualisation. Par la suite, nous décrirons quelques infrastructures connues qui elles-aussi tentent de faire avancer la recherche dans le domaine des habitats intelligents. Nous porterons donc notre intérêt sur les derniers travaux de recherche publiés par ces laboratoires, mais aussi leur avancement dans la proposition d'application de visualisation destiné aux habitats intelligents.

2.1 - LA VISUALISATION DE DONNÉES

Le terme *visualisation de données* est issu d'une longue histoire. Dans les temps anciens, les premiers dessins et autres représentations visuelles étaient utilisés pour laisser des traces de notre histoire et transmettre de l'information aux générations futures. Ainsi, la visualisation des données a contribué de manière significative à l'invention et à la découverte tout au long de l'histoire humaine. L'invention de l'informatique est l'élément clef qui introduit un changement significatif du mode de représentation visuelle des données. La démocratisation des ordinateurs a conduit à des traitements automatisés de plus en plus rapides et précis. La visualisation de données fait désormais partie intégrante

de la recherche dans de nombreux domaines, permettant aux utilisateurs de manipuler des quantités substantielles d'informations brutes, tout en apportant des méthodes plus simples et abordables aux phases d'exploration. À cet égard, il existe aujourd'hui de nombreuses définitions de ce que recouvre le concept de visualisation de données, notamment, celle de Bikakis, N. (2018) qui la définit de la manière suivante :

La visualisation des données est la présentation des données dans un format pictural ou graphique. La visualisation des données offre aux utilisateurs des moyens intuitifs d'explorer et d'analyser les données de manière interactive, leur permettant d'identifier efficacement des modèles intéressants, de déduire des corrélations et des causalités, et de prendre en charge les activités de création de sens.

2.1.1 - IMPACT DE LA VISUALISATION DE DONNÉES

Ainsi, au cœur de la visualisation de données, se trouve un lien entre représentation visuelle et déduction d'information. La visualisation de données est aussi largement utilisée pour informer les utilisateurs. Cependant, dans la limite de nos connaissances, peu de recherches ont été effectuées permettant d'examiner l'influence de la visualisation sur les utilisateurs et les moyens de rendre le message plus persuasif. Dans l'étude *The persuasive power of data visualization* (Pandey et al., 2014), les auteurs présentent des expérimentations de recherche pour combler cette lacune à travers une analyse factuelle de la visualisation persuasive. Pour en explorer les effets, leur méthodologie est la suivante : ils s'appuient sur la recherche de persuasion en psychologie, ainsi que sur la littérature sur les interfaces utilisateurs. Du fait de l'étendue du sujet, du sujet, les auteurs décident de concentrer leur recherche sur l'hypothèse suivante : « *la*

représentation graphique d'informations statistiques conduit à une persuasion accrue par rapport aux représentations non graphiques des mêmes données » [Traduction libre]. De plus, ils suggèrent aussi que les représentations de données sous forme de graphiques ont une plus grande probabilité de persuader que les représentations tabulaires de la même donnée. Afin de vérifier cette hypothèse, ils ont construit une expérimentation basée sur la mesure du changement d'attitude suite à l'exposition d'utilisateurs à des messages persuasifs. Pour ce faire, des informations sous forme factuelle sont présentées aux participants. Suite à cela, une série de questions est posée aux participants. Puis, les mêmes informations sous forme visuelle (graphique ou tabulaire) sont présentées, suivies d'un nouveau questionnaire. Au final, sur 720 participants confrontés à 3 sujets de données différents (L'impôt sur le revenu des sociétés, l'incarcération, les jeux vidéo), les résultats tendent à montrer que les graphiques conduisent à une plus grande persuasion lorsque les participants ont un avis initial faible à propos du sujet. Les tableaux, en revanche, semblent surpasser les graphiques lorsque les participants ont un avis initial fort à l'encontre du message persuasif.

Dans la continuité de ces recherches, on peut déduire que de simples détails dans la présentation visuelle des données peuvent changer la façon dont elles sont interprétées. En effet, il existe aujourd'hui de multiples manières de représenter de l'information. Malheureusement, dans un grand nombre de disciplines, de nombreuses représentations expriment des informations de manière incorrecte ou bien, lorsqu'elles ne sont pas incorrectes, elles utilisent des données qui ne sont pas évidentes à visualiser. C'est pourquoi, des études comme *Principles of Effective Data Visualization* (Midway, 2020) ont établi dix principes qui servent de guide aux auteurs cherchant à améliorer leurs messages visuels. Certains principes sont des conseils de base qui peuvent sembler évidents, comme définir le message avant d'établir un visuel. D'autres sont plus

complexes, comme l'impact des différentes combinaisons de couleurs sur les différents types d'informations. Alors que la conception et la normalisation des figures n'est souvent formellement enseignée aux scientifiques, ainsi peu appliquée, il leur incombe de prendre connaissance des meilleures pratiques afin de restituer le plus clairement possible leur interprétation des données utilisées. L'étude évoquée ci-dessus ne propose pas de résultat numérique à proprement parler, mais une liste de lignes directrices, dont voici quelques principes qui semblent être importants à retenir. Le principe no. 3 : choisir la bonne géométrie pour le bon type de données. La plupart des formes géométriques se retrouvent dans les représentations suivantes : quantité, composition, distribution et relationnel. De fait, le message est plus pertinent si le format est plus adapté. Le principe no. 6 : inclure des petits multiples d'une même figure. C'est une approche visuelle particulièrement efficace qui consiste à répéter une figure pour souligner les différences. L'objectif est de rendre les données inévitablement comparables. Principe no. 8 : un visuel simple et une légende détaillée. En effet, une étude des chiffres dans le *Journal of American Medicine* (Cooper, 2002) a révélé que plus d'un tiers des graphiques n'étaient pas explicites. Évidemment, tous les chiffres ne peuvent pas être complètement autonomes, certains modèles statistiques et d'autres procédures nécessitant plus qu'une légende pour devenir explicites. Cependant, le principe demeure que les légendes doivent permettre de comprendre en peu de mots la visualisation et les représentations utilisées. Principe no. 10 : obtenir des opinions. En effet, bien qu'il puisse exister des principes et des théories sur l'efficacité de la visualisation de données, la mise en situation concrète démontre que les visuels les plus efficaces sont ceux avec lesquels les lecteurs se connectent le plus. Par conséquent, il est souvent recommandé de tester les visuels auprès de plusieurs profils différents afin de sélectionner les plus appréciés.

Le sujet de la visualisation et son impact sur la compréhension des données est si vaste qu'il ouvre aux nombreux chercheurs la possibilité de l'aborder sous des angles différents. On peut notamment citer *Ten guidelines for effective data visualization in scientific publications* (Kelleher & Wagener, 2011) qui fournissent eux aussi des lignes directrices pour la visualisation de données mais, cette fois, adaptées aux publications scientifiques. Par ailleurs, des études comme *Design principles for data visualization in evaluation* (Evergreen & Metzner, 2013) explorent comment la visualisation de données peut être utilisée en tant qu'outil de communication pour maintenir l'attention des auditoires.

2.1.2 - EVALUATION ET UTILISABILITÉ

Comme on a pu le constater, de nombreuses techniques de visualisation de données et de bonnes pratiques sont proposées. En réponse aux problématiques modernes de ces dernières années et de notre besoin d'obtenir de l'information en temps réel, la demande d'applications logicielles qui intègrent les conceptions de visualisation de données a fortement augmenté. Cependant, la visualisation logicielle n'est pas aussi avancée dans tous les domaines. Un groupe de chercheurs (*Improving usability through enhanced visualization in healthcare*, Khan & al., 2017) a exploré cette hypothèse en utilisant le domaine d'applications logicielles liées aux soins de santé. Ils émettent le constat que les professionnels de la santé disposent d'un soutien limité vis-à-vis des applications logicielles. En outre, la mauvaise convivialité des applications de soins de santé et le faible degré d'utilisabilité sont des obstacles majeurs à l'adoption des systèmes de santé électroniques plutôt que les méthodes traditionnelles à base de papier. Dans l'article, l'équipe propose l'élaboration d'un dossier de santé électronique pour intégrer différentes données techniques de visualisation. Lors des premières phases de ce travail,

l'accent est mis sur une revue de littérature qui se concentre sur deux domaines principaux liés à l'expérience utilisateur, c'est-à-dire, l'utilisabilité et la visualisation. À travers celle-ci, une définition plus précise de l'utilisabilité de la visualisation est donnée ainsi que les différentes étapes allant de la récolte des données à la visualisation. La méthodologie de recherche de ce travail se base sur l'étude des limites des systèmes existants en suggérant des améliorations sur la base d'avis d'experts et de notions de visualisation. Les chercheurs ont alors contacté des hôpitaux afin d'avoir l'avis d'experts et d'améliorer la connaissance des besoins concernant le flux de travail personnels soignants. Suite à cela, ils ont implémenté une application qui utilise les différentes techniques de visualisation pour surmonter les inconvénients identifiés lors de l'analyse des systèmes de santé existants. Par exemple, dans la plupart des cas, l'historique des patients est maintenu sur une période maximum de neuf mois, rendant difficile le suivi des patients à plus long terme. La visualisation peut jouer un rôle essentiel pour surmonter ce problème en implémentant un système d'aperçus. Lors de chaque visite, le médecin a la possibilité d'accéder à une vue d'ensemble des visites passées ce qui améliore la fiabilité et la rapidité des diagnostics (Stachelek et al., 2020). Dans le but de mesurer l'impact et l'utilisabilité de ce système, deux méthodes standards d'évaluation ont été utilisées. La première, le *Single Ease Question* (SEQ) fournit un moyen d'estimer l'aisance ou la difficulté lors de l'exécution d'une tâche. Il est basé sur une seule question qui mesure la facilité d'utilisation (ou tâche utilisateur) et se base sur une échelle numérique avec des valeurs allant de 1 (très difficile) à 7 (très facile). La seconde méthode, *System Usability Scale* (SUS), est un autre outil fiable pour évaluer l'utilisabilité des systèmes logiciels (Lewis, 2018). Il produit des résultats fiables sur des échantillons de petite taille et utilise une norme de dix questions pouvant être notées de 1 (fortement d'accord) à 5 (fortement en désaccord). Suite à l'évaluation du système proposé par l'équipe de recherche, les deux méthodes affichent

un score deux fois plus élevé pour les applications intégrant les nouvelles approches de visualisation, si on les compare aux anciennes versions. L'analyse des résultats indique que l'utilisation appropriée d'une visualisation de données dans les systèmes de santé se traduit par une meilleure utilisabilité, contribuant ainsi à une amélioration de la qualité des soins.

Suite à cela, le résultat des recherches montre qu'il existe bien des corrélations entre visualisation et utilisabilité (Saket, et al., 2016). Ces corrélations se déterminent et se mesurent à l'aide d'outils d'évaluation. Dans l'article *Measuring effective data visualization* (Zhu, 2007), l'auteur examine deux questions fondamentales : la première, comment définir une visualisation efficace ? La seconde, comment peut-on la mesurer ? À travers une revue de littérature, l'auteur souligne que les définitions de l'efficacité sont souvent incomplètes et incohérentes, ce qui a profondément affecté la conception et l'évaluation du concept de visualisation. Il souligne également l'absence de normes et de procédures standardisées pour en mesurer l'efficacité. Par ailleurs, l'auteur définit l'efficacité de la visualisation des données en fonction de trois principes : la précision, l'utilité et l'efficacité. Concernant la précision, pour qu'une visualisation soit dite efficace, les attributs des éléments visuels doivent correspondre aux attributs des éléments de données. Ainsi, la structure de visualisation doit correspondre à celle de l'ensemble des données. Concernant l'utilité, une visualisation efficace doit aider les utilisateurs à atteindre l'objectif de tâches spécifiques. Concernant l'efficacité, la visualisation doit réduire la charge cognitive au profit d'une tâche spécifique sur des représentations qui ne sont pas visuelles. Les travaux présentés dans cet article tentent de contribuer à la recherche fondamentale de la visualisation de l'information en proposant une présentation plus complète de la définition d'une visualisation efficace, ainsi qu'une discussion critique sur un ensemble de mesures quantitatives et qualitatives.

En effet, d'autres équipes participent à l'avancement de la recherche dans ce domaine. Par exemple, dans la publication *Patterns for visualization evaluation* (Elmqvist & Yi, 2015), un ensemble de solutions générales réutilisables pour résoudre différents problèmes récurrents d'évaluation des outils, des techniques et des systèmes de visualisation est proposé. L'article *Data visualization and evaluation* (Azzam et al., 2013) retrace quant à lui un historique de la visualisation de données pour mettre en évidence les tendances futures et leurs influences potentielles sur la pratique d'évaluation.

2.1.3 - ADAPTABILITÉ ET MODULARITÉ

Parmi les tendances populaires, l'apparition des tableaux de bord, issue des *Executive Information Systems* dans les années 1970-80, est devenue une solution largement adoptée pour répondre à la problématique de la visualisation de données (Few, 2006). Ils permettent aux utilisateurs techniques et non techniques de comprendre et d'exploiter de l'information pour prendre des décisions plus éclairées. Ils accèdent aux données compilées via l'intermédiaire de tableaux, de jauges et autres propositions graphiques pour visualiser les tendances, les événements, ou encore les mesures de performance donnant lieu à une compréhension immédiate. De plus, une des fonctionnalités les plus importantes des tableaux de bord est leur capacité à transmettre de l'information en temps réel. Quelle que soit la nature des données ou le domaine d'application, cette fonction bénéficie à notre capacité décisionnelle sur l'information. Il n'est pas nécessaire d'attendre le prochain rapport ou d'attendre des mises à jour journalières. L'information est disponible sur le moment et on peut agir en conséquence.

Les tableaux de bord sont des outils particulièrement intéressants qui trouvent leur succès dans leur simplicité et leur efficacité. Ils offrent à ceux qui les utilisent un condensé

d'informations qui facilite la prise de décision face à l'arrivée de nouvelles données (Stadler et al., 2016). Cependant, malgré les nombreux avantages offerts par les tableaux de bord, certains chercheurs se sont penchés sur leur manque d'adaptabilité et de modularité. Adapter leur conception aux différents contextes d'utilisation reste une question ouverte dans la recherche en visualisation. Les concepteurs cherchent souvent à trouver un équilibre entre l'adaptabilité et la facilité d'utilisation. Ainsi, dans l'article *QualDash: adaptable generation of visualisation dashboards for healthcare quality improvement* (Elshehaly et al., 2020), l'équipe de recherche introduit cette problématique à travers le domaine médical où les défis de visualisation découlent de la diversité des mesures clés, des modèles de données et des utilisateurs impliqués à différents niveaux organisationnels. Dans cette étude, les auteurs présentent un moteur de génération de tableaux de bord qui permet une configuration dynamique et une visualisation personnalisée pour les soins de santé. L'élaboration de ces tableaux est basée sur des entretiens avec des professionnels de santé et des ateliers de co-conception avec des analystes de première ligne. Le point fort de cette étude est la mise en place d'une structure de spécification métrique *JavaScript Object Notation* permettant aux auteurs de tableaux de bord de configurer de manière concise des variantes spécifiques concernant les données. Suite au déploiement de QualDash, les résultats d'évaluation ont démontré que l'adaptabilité et la personnalisation de la visualisation ont été très bien reçues et cela a conduit à des collaborations naissantes pour ce type de génération de tableaux de bord adaptables.

Cependant, suite à certaines limitations clairement identifiées concernant la flexibilité des tableaux de bord actuels et de leur mise en page, les flux traditionnels de présentation de données n'affichent pas nécessairement tous les attributs et variables disponibles. Ils ont également tendance à s'encombrer trop rapidement. Dans *Principles*

of dashboard adaptability to get insights into origin-destination data (Dobraja et al., 2020), les chercheurs tentent de surmonter ces limites afin d'introduire une méthode générale concernant l'adaptabilité des tableaux de bord. Ils introduisent l'adaptabilité comme la capacité d'un système à s'adapter aux situations d'utilisations courantes et celui-ci comprend trois composantes. La première permet d'identifier le besoin utilisateur : elle est composée de questions spatio-temporelles qui permettent d'établir une relation entre les besoins de visualisation et les données qui vont être présentées. La seconde représente la composante information : elle comprend les données et le tableau de bord qui contient la représentation graphique des données issues des besoins. La troisième est la composante environnement : elle se rapporte à l'apparence physique de l'environnement d'utilisation, notamment les appareils sur lesquels les informations sont affichées. Le besoin d'adaptabilité est souvent requis car le tableau de bord ne répond pas aux besoins des utilisateurs. Les raisons sont souvent qu'il y a trop d'informations à représenter. Les données étant composées de plusieurs niveaux d'agrégation, plusieurs paramètres ne peuvent pas être représentés par une seule vue ou encore le besoin de plus de détails et la nécessité de remplacer les représentations graphiques lorsque les utilisateurs requièrent une représentation différente de celle présente par défaut. Pour illustrer leur recherche par un cas concret, les auteurs ont alors développé un tableau de bord où la stratégie de visualisation se base sur l'approche *aperçue d'abord, zoom et filtre, puis détails* à la demande. Cette méthode a permis aux utilisateurs de jouer un rôle actif dans le processus d'obtention des informations. De plus, elle a introduit des sélections en fonction de leurs besoins. Le déploiement d'une visualisation adaptable garantit que les besoins de l'utilisateur soient pris en compte dans le contenu et l'environnement. Avec l'adaptabilité, les utilisateurs peuvent explorer les données en mettant l'accent sur l'une des composantes ou sur une combinaison de celles-ci. Ils peuvent aussi explorer des informations qui ne

sont pas disponibles en mode aperçu et obtenir plus de détails sur le sous-ensemble et l'élément d'intérêt. Ils peuvent remplacer ou compléter une représentation graphique par une autre et modifier la disposition du tableau de bord en fonction des préférences de l'utilisateur.

Pour conclure sur ce point, on peut constater que les tableaux de bord sont devenus des éléments essentiels dans le domaine de la visualisation puisqu'ils représentent un condensé de bonnes pratiques de visualisation. Cependant, de plus en plus de personnes ont des difficultés avec leur conception lorsqu'ils doivent intégrer plusieurs systèmes de visualisation, c'est-à-dire, montrer les mesures appropriées aux personnes ayant des besoins différents. C'est pourquoi le nombre d'études soutenant le principe de modularité ne cesse d'augmenter. Des travaux comme *Complex systems: on design and architecture of adaptable dashboards* (Strugar, 2019) continuent de baser leur travaux sur l'architecture de tableaux de bord adaptables. D'autres comme *Data visualization: enhancing big data more adaptable and valuable* (Fiaz et al., 2016) se concentrent principalement sur les techniques de visualisation de données qui rendent l'utilisation de leur analyse plus efficace en s'inspirant des problématiques du Big Data.

2.2 - LES HABITATS INTELLIGENTS

En concordance avec les problématiques de la visualisation de données, nos travaux s'intéressent également à la visualisation dans le contexte spécifique des habitats intelligents (Calvaresi et al., 2017). Il nous a donc semblé opportun de décrire quelques infrastructures connues qui elles-aussi tentent de faire avancer la recherche dans le domaine des habitats intelligents. Nous porterons donc notre intérêt sur les derniers travaux de recherche publiés par des laboratoires, mais aussi leur avancement dans la proposition d'application de visualisation destiné aux habitats intelligents.

2.2.1 – CASAS

CASAS (Center of Advanced Studies in Adaptive System) (Cook et al., 2013) est un laboratoire de recherche de Washington State University qui centre sa recherche autour des technologies produisant des données utilisables dans un environnement d'habitat intelligent. Ce laboratoire a ouvert ses portes en 2007, suite à une convergence des technologies d'apprentissage automatique et fort d'une volonté de contribuer à l'évolution de l'intervention à distance dans le domaine de la santé. Depuis, le laboratoire définit les habitats intelligents comme un environnement capable d'acquérir de l'information et d'appliquer des connaissances afin d'améliorer l'expérience des habitants (Cook & Das, 2007). L'objectif étant de donner à l'habitat la capacité de raisonner et de s'adapter aux informations fournies par les usagers pour les assister au quotidien. Le CASAS explique également que le besoin d'une telle technologie est accentué par les problématiques actuelles du vieillissement de la population, du coût des soins de santé et de l'importance que les individus accordent au fait de rester autonomes dans leur propre maison². Ainsi, en plus de faciliter la vie de tous au quotidien, les habitats intelligents pourraient révolutionner la prise en charge des personnes en besoins d'assistance en permettant d'améliorer l'autonomie à domicile.

Le CASAS, peut être considéré comme un contributeur assez actif dans la recherche, puisque depuis sa création jusqu'à aujourd'hui, environ 8 à 20 articles sont publiés chaque année³. Cependant, ce n'est pas avant 2011 qu'un article concernant un outil de visualisation de données commence à être publié. C'est l'article *A demonstration of PyViz, a flexible smart home visualization tool* (Thomas & Crandall, 2011) qui ouvre

² <http://casas.wsu.edu/>

³ <http://casas.wsu.edu/publications/>

la marche et explique que l'avancement de la recherche et les nouvelles technologies pour les maisons intelligentes continuent d'être déployées mais qu'il y a un besoin de visualisation de qualité pour les données domestiques. En effet, construire des outils facilement configurables, avec un temps de développement court et capable de communiquer avec des historiques de données en temps réel n'est pas toujours facile. Cet article présente PyViz, un logiciel axé sur l'interfacage interactif pour les maisons intelligentes. Au cours de son développement, des questions de configuration, de facilité d'utilisation et d'adaptabilité ont été abordées. L'objectif est de donner une idée de comment un bon logiciel fait une différence dans les résultats d'un projet. Finalement, on peut retenir des observations de cette publication que la capacité pour un logiciel d'interagir avec les capteurs d'un environnement et la possibilité de se connecter à une base de données ou d'utiliser des fichiers texte comme sources de données sont des fonctionnalités essentielles pour le déploiement d'une application. A ce stade, mise à part la présentation de quelques fonctionnalités, l'équipe de recherche n'aboutit pas à de réelles conclusions. Cependant, à travers la démonstration de cet outil, ils espèrent pouvoir mettre en évidence et comparer ses capacités en répondant à plusieurs questions telles que : à qui sont destinés les outils de visualisation ? A quel moment un outil de visualisation peut-il faire la différence dans un projet de recherche ? Ou encore, pourquoi la visualisation est-elle importante plutôt que de simples statistiques ?

Suite à ces travaux, plusieurs études touchant au domaine de la visualisation ont été publiées par le laboratoire. Cependant, elles ne font pas état d'un logiciel permettant de visualiser les données d'un habitat intelligent. Par exemple, l'article *Application of cognitive rehabilitation theory to the development of smart prompting technologies* (Seelye et al., 2012) introduit les technologies d'incitation à l'action (application de rappel) comme outil potentiel pour aider les personnes atteintes de troubles cognitifs à

effectuer les activités instrumentales de la vie quotidienne. Ils expliquent que même si ces techniques sont couramment utilisées par les soignants et les prestataires de soins de santé pour aider les personnes en besoin d'assistance, il n'y a pas de consensus clair vis-à-vis du contenu affiché ou d'une méthode d'instruction dans la littérature. Ainsi, l'objectif de cet article est de démontrer comment les principes de la réadaptation cognitive à travers un logiciel d'incitation à l'action, peuvent informer et faire progresser le développement de systèmes d'assistance des habitats intelligents. De même, avec l'article *PUCK: an automated prompting system for smart environments: toward achieving automated prompting* (Das et al., 2012), un nouveau système d'affichage est introduit par l'équipe de recherche. Il s'agit également d'un outil d'incitation à l'action. Cependant, il a la particularité de ne pas être basé sur des règles prédéfinies. Contrairement à d'autres approches, les interventions du système sont automatisées et définies en fonction de données réelles collectées par des capteurs. La finalité de ce projet est d'apprendre au système les étapes auxquelles les invitations à l'action sont requises.

Toujours sur cette intention de supporter les personnes en besoin d'assistance et d'améliorer les systèmes présents dans les habitats intelligents, l'application la plus récente présentée par le laboratoire est introduite dans l'article *Creating a digital memory notebook application for individuals with mild cognitive impairment to support everyday functioning* (Raghunath et al., 2019). Celui-ci fait état d'une application d'annotation d'activités soutenant le fonctionnement quotidien des personnes atteintes de troubles cognitifs légers. Malgré le fait que ce ne soit pas une véritable application de visualisation de données, les travaux indiquent que de nombreux efforts ont été mis sur le degré de satisfaction et d'utilisabilité. En effet, vingt participants âgés de 54 ans et plus et de différents niveaux de capacités cognitives, ont été recrutés pour quatre itérations de tests

d'utilisabilité. Chaque participant a effectué une série de tâches à l'aide du carnet de mémoire numérique DMN, suivies de questionnaires évaluant la satisfaction des interfaces utilisateurs et de même pour l'utilisabilité pour l'application. Ainsi, les résultats indiquent qu'entre les itérations 2 et 5, les participants ont démontré des réductions marquées du temps nécessaire pour effectuer plusieurs types de tâches (par exemple, ajouter un événement, naviguer dans l'interface) et qu'à l'itération 5 l'ensemble des sous-échelles des questionnaires ont également évalué très positivement. Des temps d'exécution des tâches plus rapides étaient corrélés à des évaluations du système plus favorables. Cependant, ni les temps d'exécution des tâches ni les évaluations du système n'étaient corrélées aux capacités cognitives, à l'utilisation des outils de planification ou à l'aisance avec la technologie.

Finalement, lorsqu'on regarde le nombre total d'articles publié par le laboratoire et le nombre d'articles abordant la visualisation de données, on remarque rapidement qu'il s'agit encore un sujet peu représenté dans le domaine des habitats intelligents. Même si des efforts sont faits comme on a pu le voir dans la présentation des articles ci-dessus, il reste question d'outils applicatifs ne mettant pas en application les principes de visualisation de données destinés à clarification et à la déduction d'information.

2.2.2 - MOBILE AND PERVASIVE COMPUTING LAB

Parmi les autres laboratoires qui centrent leur recherche sur les habitats intelligents, citons également le Mobile and Pervasive Computing Lab (MPCL) situé à l'Université de Floride aux Etats-Unis. Il est dirigé par le professeur Sumi Helal en charge de plusieurs projets de recherche dans le domaine des espaces intelligents, de l'Internet des objets, de la santé numérique et des technologies d'assistance. L'équipe de recherche dispose également d'un habitat intelligent, Gator-Tech (Helal et al., 2005), dans lequel

sont installés de nombreux capteurs destinés aux expérimentations mais aussi à l'élaboration de jeux de données disponibles en open source⁴. De plus, si on se fie à l'historique de publication du laboratoire⁵, environ 7 à 20 articles sont publiés chaque année depuis 1999, ce qui le situe comme contributeur assez actif dans le domaine de la recherche. Même si son implication et ses publications sur le sujet des visualisations reste limité, il reste intéressant d'explorer les quelques articles qui en parlent, notamment pour l'approche particulière proposée.

Tout d'abord, il semblerait que la première application du laboratoire disposant d'interface graphique remonte à février 2002. Même si elle n'aborde pas réellement le concept d'habitat intelligent, il s'agit d'une application produite par le laboratoire qui intègre de la visualisation de données. En effet, dans la publication *CAD-HOC: a CAD-Like tool for generating mobility benchmarks in ad-hoc networks* (Shah, 2002), le problème de l'utilisation de scénarios de mobilité irréalistes pour la simulation de protocoles de routage est abordé. L'équipe de recherche présente alors CAD-HOC, un outil qui permet d'effectuer des expériences de mise en réseau dans des scénarios visuellement réalistes tels qu'un aéroport, un terminal de bus, des bâtiments et d'autres installations. De manière générale, on remarque bien que la fonction primaire de cet outil n'est pas en rapport avec la visualisation de données. Cependant il dispose d'une fonctionnalité de génération de benchmark qui permet d'analyser et de questionner la fiabilité des résultats. Finalement, cette application posera une ligne directrice pour la conception d'outils de simulation et de visualisation du laboratoire du laboratoire.

⁴ <https://www.cise.ufl.edu/~helal/opensource.htm>

⁵ <https://www.cise.ufl.edu/~helal/publications.htm>

Par la suite, dans le prolongement de ces travaux, l'étude *Persim-Simulator for human activities in pervasive spaces* (Helal et al., 2011) introduit un simulateur d'activités humaines capable de capturer des éléments physiques d'un espace, comprenant les capteurs et le comportement des résidents. Cet outil permet aux chercheurs de concevoir un espace et de modifier la conception de manière progressive sur plusieurs sessions, de simuler et modifier l'espace à plusieurs reprises jusqu'à ce qu'il réponde aux exigences de recherche. Cela permet également de remonter dans le temps et d'explorer de légères variations dans le système sans réellement répéter les expériences, ce qui permet de gagner un temps considérable pour la collecte des données. Par la suite, dans le but de valider les capacités de l'outil, une comparaison entre un jeu de données réelles et un jeu de données synthétiques produit par l'application a été effectuée. L'analyse des résultats montre en moyenne une ressemblance de 76,6% entre les deux jeux de données, ce qui semble très prometteur. On remarquera que la présence d'une interface web développée en Java, JavaScript, and Java Server Pages (JSP) est mentionnée, mais très peu d'informations supplémentaires sont communiquées sur le sujet. En revanche, la section discussion mentionne des limitations concernant les interfaces et la représentation actuelle des simulations. Ainsi, ils adressent ces limitations en annonçant une refonte 3D des interfaces permettant aux utilisateurs une manipulation de haute précision sur les éléments.

C'est seulement cinq ans plus tard, à travers l'article *Automatic agent generation for IoT-based smart house simulator* (Lee et al., 2016) que l'équipe de recherche répond à ces limitations et propose un générateur d'agent autonome qui simule l'activité humaine dans les maisons intelligentes. Celui-ci est fourni avec une interface utilisateur graphique en 3D permettant la configuration spatiale, ainsi que des capteurs virtuels simulant la réalité. De plus, le simulateur est fourni avec un agent d'intelligence artificielle, qui

interagit avec l'environnement à l'aide d'une méthode de comportement axée sur la motivation. L'implémentation des capteurs virtuels est conçue pour détecter les différents états de la maison, ses agents mais aussi l'ensemble des données détectées qui résultent de ces interactions. A long terme, cette méthode a pour but de réduire le coût des tests et de faciliter la réalisation de nouvelles expérimentations (Jahromi et al., 2011). En effet, évaluer la qualité des environnements dans les maisons intelligentes nécessite de grands ensembles de données contenant des interactions entre les personnes et les environnements, ainsi ces données sont extrêmement importantes pour les tests matériels et logiciels. De ce fait, il n'est pas exagéré de rappeler que les phases de tests et de simulation nécessitent une quantité importante de temps et de ressources et que la capacité de pouvoir simuler ces environnements avec une application informatique en spécifiant des paramètres adaptés aux expériences serait un gain de temps considérable.

Comme dans le cas précédent, l'examen de l'ensemble des travaux publiés par le laboratoire montre qu'il n'existe qu'un faible nombre d'articles traitant de la visualisation et des interfaces d'applications. Mais à travers les travaux présentés ci-dessus, on peut remarquer un intérêt croissant vis-à-vis de ces sujets. Les équipes de recherche prennent le temps d'adresser les limitations d'interfaces et les effets positifs de la visualisation prennent peu à peu leur place dans le monde de la recherche.

2.2.3 - DOMUS

Depuis sa fondation en 2002, le laboratoire DOMUS (Domotique et informatique mobile à l'Université de Sherbrooke) (Giroux et al., 2009) se donne pour mission de pallier à la perte d'autonomie cognitive des personnes à l'aide de solutions

technologiques⁶. En effet, au même titre que les limitations cognitives, le laboratoire désire offrir des produits d'adaptation du milieu de vie en cas de limitation physique ou de mobilité. Pour la réalisation des projets de recherche, l'infrastructure est composée d'un appartement intelligent, d'un logement alternatif pour les personnes ayant un traumatisme crânio-cérébral et de résidences pour personnes âgées, permettant d'effectuer des études à plus long terme. Le DOMUS regroupe une équipe interdisciplinaire d'une vingtaine de chercheurs couvrant de nombreux sujets, allant de l'informatique et la neuropsychologie à la l'ergothérapie et au design industriel. Il a également mis sur pied un maillage national, à travers des projets de recherche conjoints avec d'autres universités comme l'Université de Montréal ou l'UQAC, et le milieu clinique de la santé⁷. Ainsi, son implication dans le domaine des habitats intelligents et sa fréquence de publication en fait un laboratoire reconnu par de nombreuses équipes de recherche.

Le DOMUS est donc un laboratoire de recherche bien établi dans le domaine des habitats intelligents et de l'aide à la personne. Cependant, ces travaux font souvent état de sujets techniques et laissent peu de place à l'intervention de la visualisation de données. Ainsi, *Pervasive behavior tracking for cognitive assistance* (Giroux et al., 2008) est l'un des premiers travaux du laboratoire qui introduit ce sujet. Le sujet principal de cet article porte sur l'aide aux personnes en déficience cognitive. L'équipe de recherche explique que les petits problèmes de mémoire et d'attention n'ont pas de conséquences graves, mais les personnes souffrant de déficits plus complexes (Alzheimer, schizophrénie, déficiences intellectuelles) doivent y faire face au quotidien et elle sait à quel point cela peut changer la vie de ces personnes. L'objectif de cet article est de montrer comment un suivi des

⁶ <https://domus.recherche.usherbrooke.ca/domus/mission/>

⁷ <https://domus.recherche.usherbrooke.ca/recherches/>

comportements peut aider à transformer toute un habitat en prothèse cognitive à l'aide d'outils informatiques omniprésents, d'interfaces tangibles et de surveillance à distance. Archipel (Pigot et al., 2008) est l'application présentée qui remplit le rôle d'assistant cognitif. Elle se consacre à une tâche bien précise : la préparation des repas. L'interface utilisateur a été améliorée et conçue pour des activités, étape par étape. Une interface simple s'affiche sur un écran tactile dans la cuisine. Chaque étape est illustrée par du texte, des images et une vidéo. Elle propose des instructions procédurales pour l'achèvement de l'activité basée et une sensibilisation au contexte. Archipel est connecté aux capteurs de l'appartement via un serveur et peut déterminer si l'étape a été effectuée correctement et complètement. Si c'est le cas, il peut passer automatiquement à l'étape suivante. Malheureusement, lorsque les personnes en déficience cognitives se retrouvent coincées à un niveau, elles ont tendance à abandonner la tâche. L'assistance contextuelle exploite les dispositifs disponibles dans l'appartement pour les stimuler, par exemple : la fonction « trouver un objet » où le résident n'a qu'à cliquer sur la photo de l'objet recherché et l'assistance utilise des appareils dans l'appartement pour attirer l'attention de l'utilisateur pour signaler son emplacement. Finalement, une expérimentation concernant la réalisation d'une recette de cuisine a été réalisée dans l'appartement intelligent du DOMUS. L'expérience impliquait 12 personnes atteintes de déficience intellectuelle. Tous les participants ont pu compléter la recette avec Archipel. L'analyse préliminaire indique qu'Archipel a réduit de moitié l'assistance humaine. On notera que la partie de cette étude de visualisation de données n'est pas le sujet principal mais qu'elle soutient fortement le projet de recherche. De plus, on peut ajouter que très récemment, le DOMUS a sorti COOK, une application plus récente du même ordre d'idées (Gagnon-Roy et al., 2022).

Dans le prolongement des études sur l'aide à la personne, l'article *From conception to evaluation of mobile services for people with head injury: a participatory design perspective* (Groussard et al., 2018) présente un assistant cognitif mobile capable d'améliorer l'autonomie des personnes vivant avec un traumatisme crânien. Cet article porte autant sur la méthode mise en place que les résultats de l'application, mettant en évidence l'intérêt du laboratoire à l'égard des principes de visualisation et de l'utilisabilité des applications. La méthodologie de conception participative présentée est particulièrement intéressante. En effet, pour être utile, une technologie doit être conçue en fonction des besoins et formulée, directement ou non par la population cible, de manière à pouvoir assurer une adéquation entre la technologie et les caractéristiques des utilisateurs (Scherer et al., 2005). La population de l'étude est composée de quatre hommes adultes présentant des troubles cognitifs et de trois soignants. La première phase de cette recherche a consisté à concevoir l'application Services d'Assistance Mobiles et Intelligents (SAMI) en fonction des besoins exprimés par les participants. Au cours de la deuxième phase, les participants ont évalué l'assistant mobile à domicile pendant huit semaines. Les résultats montrent que les participants ont pu participer activement à la conception de l'application et l'ont utilisé avec succès. Les personnes atteintes de troubles ont montré une légère amélioration de leur satisfaction à l'égard de la vie. Les résultats semblent prometteurs, mais des limitations concernant le petit nombre de participants sont adressées, soulignant que ces résultats prometteurs devront être confirmés par une étude à plus grande échelle.

Jusqu'à présent, nous avons principalement eu l'occasion d'étudier des applications ou des supports visuels qui supportent l'axe de recherche principal. Nous avons constaté que peu d'articles font état du concept de visualisation de données et de l'analyse d'information. Cependant, la crise sanitaire récente a montré la nécessité et l'intérêt de mettre en place des outils techniques permettant une prise de décision rapide et facilitant le maintien à domicile

des personnes âgées (Dixit et al., 2020). Cette évolution impliquerait la gestion et le traitement de données résultant de l'exercice des activités de la vie quotidienne. Le dernier article présente le cadre NEARS-Hub, initié par le laboratoire DOMUS (Ngankam et al., 2022). L'objectif principal de cette application est d'être capable d'absorber des données provenant simultanément d'un grand nombre de capteurs, puis d'effectuer une analyse de ces données en temps réel, alors qu'elles sont réactualisées en fonction de la situation. De ce fait, l'architecture de l'application est structurée en trois couches de base : la couche d'interface du système, la couche d'analyse intergiciel autonome et la couche cloud. Le but de cette composition est de fournir une solution complète : l'acquisition de données, le prétraitement, le traitement et l'analyse de données volumineuses, la conservation de ces données. Pour conclure, l'analyse des performances et des expériences basées sur les données réelles démontrent que le NEARS-Hub peut être considéré comme performant. Ces résultats se basent sur des mesures de temps du couplage des appareils au contrôleur, l'étiquetage et l'identification des capteurs dans le système et l'efficacité grâce à laquelle il est possible d'effectuer des calculs locaux lorsque certaines activités sont détectées.

2.3 - CONCLUSION

Dans la section 2.1, la visualisation occupe une place importante dans les travaux de recherche de nombreuses équipes. Certaines problématiques tel que : l'adaptabilité, la modularité, l'évaluation ou encore l'impact de la visualisation sont abordés. Ce sont pour l'ensemble des éléments qui amènent à faire avancer la recherche et à présenter des résultats plus pertinents.

Dans la section 2.2, on a vu que les travaux de recherche sur la visualisation de données dans les laboratoires de recherche sur les habitats intelligents sont encore très peu représentés. Certains articles qui abordent ce sujet ont été présentés. Malgré le fait

qu'il reste encore de nombreux efforts à déployer, leur l'intérêt grandissant vis-à-vis de ce sujet, nous laisse présager une évolution positive dans ce domaine.

Pour conclure, le fait que la proposition d'un système de gestion de l'environnement et de visualisation de données, capable de présenter l'information de façon pertinente soit de plus en plus important pour la construction des habitats intelligents. Le manque de travaux inscrit les travaux présentés par ce mémoire, au cœur des problématiques actuelles.

CHAPITRE 3

DÉVELOPPEMENT D'APPLICATION POUR LES HABITATS INTELLIGENTS

La quantité de projets informatiques produits chaque jour est en constante augmentation. Les développeurs sont en permanence à la recherche de nouvelles pratiques de développement pour assurer la qualité et la réutilisabilité de leur code (Padhy et al., 2018). Une des pratiques les plus connues est la séparation du code. Que ce soit au niveau de l'architecture logicielle ou de la programmation en elle-même, cela permet d'obtenir une structuration ainsi que des fonctionnalités bien plus modulaires pour l'application ou le projet en question. Certainement inspiré de la phrase *diviser pour mieux régner*, il est souvent conseillé de subdiviser le problème initial en sous-problèmes pour les résoudre afin d'en combiner les résultats finaux et répondre à la problématique de départ.

Au LIARA, dans le cadre du projet de détection d'activités de la vie quotidienne pour les habitats intelligents, l'équipe de recherche a su décomposer le problème en plusieurs parties, par exemple : le traitement des données de radars à bande ultra-large (Bertuglia, 2021), l'entraînement de modèles d'apprentissage machine pour la détection d'activités (Maître, 2021), ou encore, la conception d'applications pour la visualisation des données. En effet, indépendamment des autres, ces modules réalisent efficacement la tâche pour laquelle ils ont été conçus, mais ce n'est qu'une fois connecté entre eux, qu'ils permettent de résoudre la problématique initiale de détection d'activités de la vie quotidienne. Une partie du travail effectué lors de la réalisation de ce mémoire a été de chercher à conserver l'indépendance des programmes réalisés dans les travaux

précédents, tout en permettant la communication de leurs résultats afin de rejoindre la solution finale.

Pour avoir une vue d'ensemble plus complète des modules et de l'architecture du travail de recherche que l'on souhaite effectuer, la Figure (3.1) représente l'architecture complète de cette idée. On y retrouve les trois couches majeures du projet, soit : la section détection d'activités dans les appartements et les différentes applications utilisées pour diffuser l'information, la section stockage des données hébergée dans le cloud pour le déploiement final (en local pendant le développement), enfin la partie visualisation de données qui décrit l'architecture et les technologies utilisées.

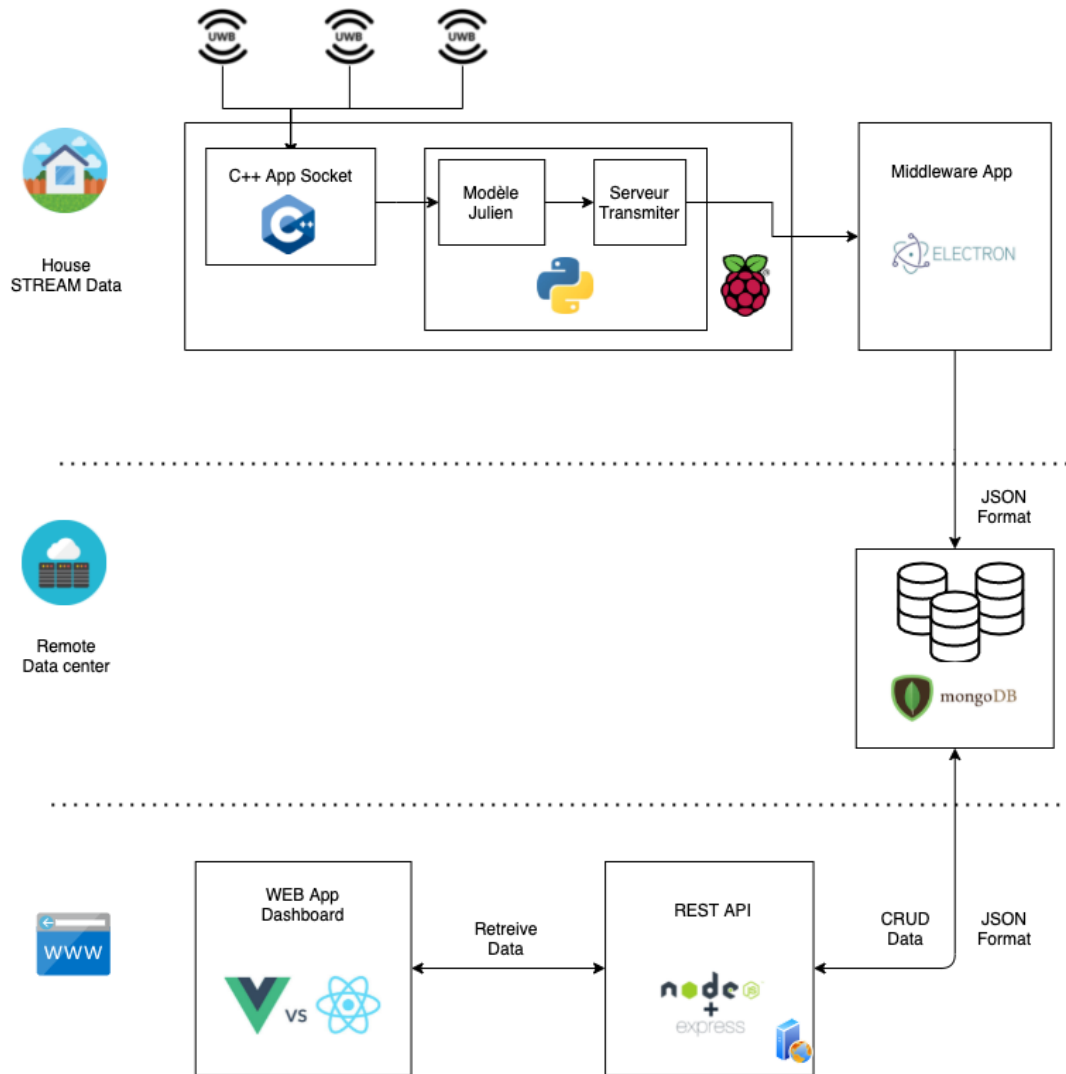


Figure 3.1 - Technologies et architecture complètes du projet

© Lucas Gonzalez

Dans ce chapitre, nous voyons comment s'accordent les différents modules qui ont été développés dans le cadre du projet de détection d'activités. Le premier module correspond à celui des radars à bande ultra-large, il est chargé de récupérer et d'envoyer les données. Le second a pour fonction de traiter les données afin d'effectuer des prédictions d'activités et le dernier est en charge de l'enregistrement pour mener à bien la collecte de ces données. Nous présentons leurs fonctionnalités, mais aussi les

technologies qui nous ont permis d'obtenir des modules à la fois indépendants et connectés.

3.1 - MÉTHODE DE CONCEPTION PAR MODULE

Comme introduit précédemment, diviser pour mieux régner est une méthode efficace de résolution de problème. L'objectif est ici de réduire la taille du problème initial à résoudre pour rendre la tâche plus abordable. C'est pourquoi, suite aux travaux de recherche (Bertuglia, 2020) du LIARA, nous avons développé un programme de prédiction d'activités. À l'origine, ce programme ne lisait que des données brutes provenant d'un fichier JSON (JavaScript Object Notation) pour ensuite afficher une activité déduite par le modèle de prédiction. De manière indépendante, celui-ci fonctionne très bien, mais dans les conditions réelles d'un habitat intelligent, il n'est pas totalement adapté à la situation. D'une part, les données ne proviennent pas toujours d'un fichier brut, elles pourraient provenir d'une API (Application Program Interface) ou bien dans notre cas, de radars qui envoient des données de manière constante. D'autre part, afficher le résultat de la prédiction n'est pas suffisant, il faut que le programme soit en mesure de transmettre l'information pour la rendre utilisable. D'autre part, lorsqu'on travaille avec plusieurs modules qui fonctionnent de manière isolée, l'un des principaux défis rencontrés est de rendre accessible le résultat aux programmes informatiques qui en ont besoin afin qu'il soit utilisé de manière pertinente.

Poursuivant l'idée d'établir cette communication entre les programmes, nous avons choisi de nous inspirer de l'architecture d'un système client-serveur. Très fréquemment utilisé pour tout type d'échange entre deux entités, celui-ci désigne un mode de transaction entre plusieurs programmes ou processus. L'un est qualifié de client et envoie des requêtes, l'autre est qualifié de serveur et reçoit les requêtes pour y répondre.

Il existe de nombreux protocoles de communication tels que : FTP (File Transfer Protocol), SSH (Secure Shell), HTTP (Hypertext Transfer Protocol), etc. Cependant, face au phénomène d'augmentation des applications web (navigateur et bureau) et au vu de la nature de notre projet, nous avons choisi de nous orienter vers une architecture client-serveur simple d'intégration basée sur les *websockets*.

3.1.1 - WEBSOCKETS CLIENT SERVEUR

WebSockets un standard désignant un protocole réseau permettant d'ouvrir un canal de communication bidirectionnelle sur un Socket TCP (Transmission Control Protocol). Principalement utilisé pour les navigateurs et les serveurs web, ce système permet aux clients d'envoyer des messages à un serveur en temps réel et de recevoir des réponses de manière événementielle, sans avoir à le consulter à nouveau. Les limites de la technologie basée sur HTTP ont inspiré une réflexion qui a conduit au développement de WebSockets. Un client demandera une ressource en utilisant HTTP tandis que le serveur répondra avec les détails. HTTP est un protocole unidirectionnel, ce qui signifie que toutes les données envoyées depuis le serveur vers le client doivent d'abord faire l'objet d'une demande.

Les requêtes dites longues (voir figure 3.2) ont beaucoup été utilisées pour contourner cette restriction. L'interrogation longue se produit lorsqu'un client envoie une requête HTTP avec un long délai d'attente et que le serveur utilise le délai d'attente pour envoyer des données au client. C'est une méthode qui fonctionne très bien, mais qui monopolise les ressources du serveur pendant la durée de l'interrogation même si aucune donnée n'est disponible à soumettre.

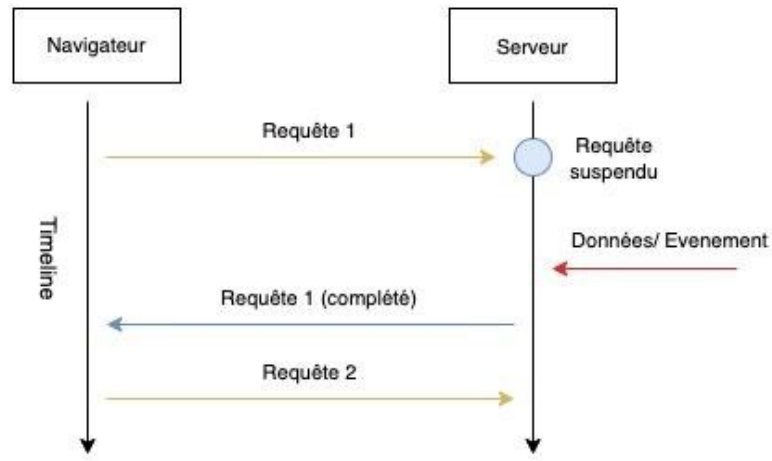


Figure 3.2 : Requêtes de type longues

© Lucas Gonzalez

Pour pouvoir passer outre la limite de l'échange unidirectionnel, les *WebSockets* ont permis la transmission de données basées sur des messages similaires à UDP, intégrant toutefois la sécurité du système TCP. *WebSockets* utilise HTTP comme méthode de transport initiale, tout en laissant le lien TCP ouvert après la réception de la réponse HTTP, afin que les messages puissent être envoyés entre le client et le serveur. Cet échange est illustré par la Figure (3.3). Cela nous permet alors de créer des applications de type *temps réel* sans nécessiter l'utilisation du système d'interrogation longue.

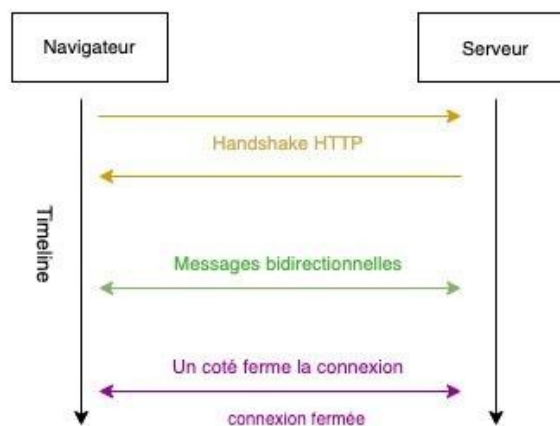


Figure 3.3 : Echange de données Websockets

En plus d'être disponible dans la majorité des langages les plus populaires tels que : Java, C, C++, JavaScript, Python, etc., la facilité d'intégration et d'utilisation des bibliothèques *Websockets* est l'un des facteurs de sélection qu'il ne faut pas négliger. La quantité de code nécessaire pour les faire fonctionner est minimale et le système de programmation événementielle rend la gestion du programme extrêmement adaptable. Ainsi en quelques lignes de code, tel que montré en Figure (3.4), on est capable d'encapsuler toutes sortes de programmes et de leur donner la capacité d'échanger de l'information en temps réel avec le monde extérieur de manière bidirectionnelle.

```
1 let socket = new WebSocket("wss://127.0.0.1");
2
3 socket.onopen = function(e) {
4   socket.send("[open] Connection established");
5 };
6
7 socket.onmessage = function(event) {
8   console.log('[message] Data : ${event.data}');
9 };
10
11 socket.onclose = function(event) {
12   console.log('[close] Connection died');
13 };
14
15 socket.onerror = function(error) {
16   console.log('[error] ${error.message}');
17 };
```

Figure 3.4 : Code d'initialisation pour un Websockets

3.1.2 - MODULARISATION D'UN PROGRAMME

Dans le cadre de notre recherche, un grand nombre de nos modules sont équipés de *Websockets* pour bénéficier de cette capacité d'échange bidirectionnel. En effet, des fonctionnalités complexes peuvent être implémentées avec leur l'aide. Par exemple, l'un des défis de cette recherche était d'implémenter des salles de *streaming* pour desservir de l'information en temps réel et de manière simultanée. Pour illustrer le concept de modularisation rapide, nous avons choisi de présenter le modèle qui a été le plus implémenté sur nos modules. Notre besoin primaire étant de rendre accessible le résultat d'un programme indépendant, c'est le modèle du *répéteur* (voir Figure 3.5) qui a été implémenté. Plus concrètement, chaque module est composé d'un client pour aller chercher de l'information et d'un serveur pour la transmettre à tous les modules qui se connectent.

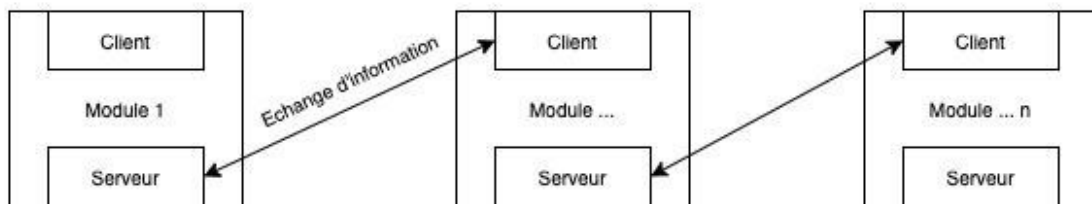


Figure 3.5 : Modèle répéteur Client-Serveur

© Lucas Gonzalez

Comme expliqué ci-dessus, en prenant l'exemple de notre projet, le programme de prédiction d'activités est composé d'un client pour récupérer l'information. Une fois le module connecté à l'émetteur des radars, l'information est traitée et une prédiction d'activités est émise. Finalement, le récepteur du module de prédiction se charge de d'enregistrer l'information et de la transmettre à tous les modules qui se connectent.

3.2 - APPLICATION D'INTÉGRATION DES HABITATS INTELLIGENTS

Au LIARA, la détection d'activités pour les habitats intelligents est un sujet sur lequel plusieurs professeurs et étudiants travaillent conjointement. Chacun d'entre eux participe à l'avancement de la recherche en développant des programmes qui font avancer la solution ou améliorent celles déjà existantes (Cousyn et al., 2021) (Beaulieu et al., 2022). Pour faciliter la suite du projet, il nous est apparu pertinent de développer une application qui rende plus accessible l'utilisation et l'intégration de ces programmes. Dans le cas d'un potentiel déploiement chez des usagers, l'application aurait pour but de centraliser les modules qui composent la solution et d'interagir avec eux, ce qui permettrait de vérifier l'état de leurs activités.

3.2.1 - CHOIX DES TECHNOLOGIES

Dans le domaine du développement d'application, le choix des technologies est une étape importante qui détermine la direction que va prendre le projet. Qu'il s'agisse des fonctionnalités qui vont pouvoir être implémentées ou du type d'utilisateur qui y aura accès, ces questions sont importantes à aborder car elles permettent d'aligner le processus de développement avec les besoins finaux. L'application dont il est question dans cette recherche regroupant des modules développés sur Windows, Linux ou macOS, c'est la question de la portabilité qui s'est posée. Notre intérêt s'est alors orienté vers les outils de développement multiplateformes susceptibles de permettre la portabilité.

En effet, de nombreux langages de programmation sont considérés comme multiplateformes. Pourtant dans la réalité, un logiciel conçu sous Linux ne fonctionnera pas tel quel sous Windows. Même si un langage comme le C est similaire d'une plateforme à l'autre, le fonctionnement interne et la gestion de l'affichage seront différents. Il faut donc utiliser un compilateur destiné à générer un code exécutable et

compréhensible pour chacun des systèmes d'exploitation que l'on désire utiliser. On aurait donc un seul langage de programmation et un compilateur par type de systèmes d'exploitation.

De ce fait, l'arrivée d'Electron, un cadriciel Web en accès libre, a permis aux programmeurs de développer des applications bureau multiplateformes avec les technologies de base : JavaScript, HTML et CSS. Les applications basées sur ce cadriciel fonctionnent comme des applications Web, la seule différence étant qu'elles peuvent interpréter et enregistrer des informations dans le fichier du système de traitement de données. Electron a été ajouté à la boîte à outils des programmeurs en raison de sa capacité à augmenter la vitesse de développement. Il a comblé la lacune repérée entre un besoin d'applications de bureau modernes et un outil accessible. Même si aujourd'hui Electron comporte certains inconvénients comme une utilisation élevée de la mémoire et des téléchargements volumineux, il reste une solution vastement adoptée. De grandes entreprises comme Discord, WhatsApp, Slack, Visual Studio Code et autres, utilisent ce cadriciel pour le développement d'application bureau et les avantages qu'il apporte : sa flexibilité en termes de fonctionnalités, c'est-à-dire, l'ensemble de l'écosystème JavaScript et de ses millions de bibliothèques en accès libre, mais aussi en termes de réutilisabilité. Le fait de n'utiliser qu'une seule base de code uniformise le développement et facilite la migration d'une application web vers une application bureau. Concrètement, dans le cadre de notre recherche, Electron a permis le développement d'une application d'intégration où la majorité des tests d'utilisation ont été effectués en laboratoire sur les systèmes d'exploitation Windows et macOS. Une Raspberry Pi⁸ tournant sur Linux a

⁸ <https://www.raspberrypi.com/>

également permis de créer des conditions proches des conditions finales de déploiement de l'habitat intelligent.

3.2.2 - INTÉGRATION DES MODULES EXISTANTS

Comme expliqué précédemment, l'application d'intégration permet de centraliser et gérer l'utilisation des modules existants. La motivation principale de celle-ci est de faciliter le déploiement des programmes et du matériel nécessaire à la détection d'activités dans les habitats intelligents. Même si le déploiement final n'a pas encore eu lieu, l'application a pu être testée dans l'appartement laboratoire du LIARA. Aucune mesure concernant le degré d'appréciation ou d'utilisabilité de l'application n'a été effectuée à ce jour. Cependant, l'utilisation en mode expérimental a clairement facilité les étapes de récolte d'activités et de synchronisation des modules.

A l'origine, le projet est composé de deux modules principaux. Le premier est chargé de diffuser un flux de données émis par des radars placés dans un appartement. Les fondamentaux et la théorie sur les radars à bande ultra-large et leur fonctionnement ont été traités dans le mémoire *Reconnaissance d'activités dans un habitat intelligent à l'aide de radars Ultra-WideBand* (Bertuglia, 2020). L'objectif principal de ce programme est de diffuser les données émises par les radars aux formats JSON.

Le second module est chargé de la détection d'activités. Une première version développée par l'équipe du LIARA permet d'identifier 15 activités de la vie quotidienne avec un taux de reconnaissance d'environ 80% (Bouchard et al., 2020). En réponse à cette recherche, les travaux couverts par l'étudiant-chercheur Alexandre Beaulieu tentent d'améliorer le taux de reconnaissance et pour ce faire, deux approches ont été testées. La

première est une combinaison de systèmes experts et de réseaux de neurones et la seconde, un modèle d'apprentissage profond combinant différentes architectures (Long Short-Term Memory (LSTM) et réseau de neurones convolutifs (CNN). Ce modèle, appelé Tuned EfficientNetB0 avec LSTM, offre une amélioration de 18,63 % du taux de reconnaissance par rapport aux réseaux de neurones développés dans des travaux antérieurs tout en exploitant le même jeu de données (Beaulieu, 2021).

Finalement, pour répondre à un besoin de flexibilité ainsi qu'au fait que les modules sont développés dans des technologies différentes pouvant évoluer dans le temps, il a été préférable d'intégrer le système de *Websockets* client-serveur que nous avons développé. Les modules représentés dans l'application sont donc de simples connecteurs bidirectionnels capables de récupérer et de transmettre de l'information. Ainsi, comme expliqué dans la Section 3.1, lorsqu'un programme est équipé d'un système *Websockets*, il devient possible de dupliquer la méthode d'intégration pour tous les programmes qui suivent. Il suffit simplement d'ajouter un nouveau *répéteur* et par la suite, les programmes peuvent choisir de s'échanger de l'information par l'intermédiaire de leur connecteur ou d'utiliser l'information récupérée dans l'application. Pour le troisième module qui peut être considéré comme légèrement différent des autres, nous avons choisi d'utiliser l'information que l'application récupère, de manière à centraliser l'échange et la circulation des données. Les détails de celui-ci sont abordés dans la section suivante.

3.2.3 - COLLECTE DE DONNÉES

Disposer d'un ensemble de données organisées est nécessaire pour la plupart des applications d'intelligence artificielle/apprentissage automatique. L'acquisition de ces

données est souvent un défi de taille. Non seulement on doit collecter des données provenant du monde réel, mais on doit aussi les annoter et les préparer pour notre modèle. Pour les petites équipes de recherche, la formation d'un ensemble de données est un obstacle important à surmonter. En effet, dans le domaine de la détection d'activités, il est intéressant de disposer de données réelles nous permettant de valider les résultats de nos prédictions. De ce fait, un module capable de faire la transition entre détection d'activités et collecte de données est important pour l'amélioration de nos modèles, mais aussi pour nos potentielles applications de visualisation de données.

Le troisième module de notre application n'est donc pas destiné à répéter un résultat produit. Sa fonction principale est d'enregistrer de l'information dans une base de données. Pour diverses raisons qui sont détaillées dans la Section 3.3.2, c'est le système MongoDB⁹, un outil de gestion de base de données orienté documents qui a été sélectionné. Parmi les nombreux avantages qu'offre la technologie NoSQL, c'est aussi la disponibilité de bases de données distantes à faible coût qui rend la centralisation d'une récolte plus simple à organiser. Par exemple, dans le cas d'une résidence de n appartements (Figure (3.6)), dans laquelle on décide d'organiser une collecte de données, selon deux mises en situation différentes. Dans le premier cas, on aurait une installation où chaque appartement enregistre des données sur appareil local. Ainsi, pour y accéder, il faudrait récupérer la totalité des données de chaque appartement manuellement, les fusionner et les rendre disponibles sur un serveur. Dans le deuxième cas, pour éviter d'avoir à effectuer ces tâches manuellement sur l'ensemble de nos installations ou de développer un système de cloud automatisé, MongoDB nous met à disposition un système de base de données distante qui règle cette problématique. Les modules disposés dans

⁹ <https://www.mongodb.com/>

chaque appartement utilisent une seule et même adresse de connexion et les données sont envoyées dans la même base de données. Ainsi, on règle le problème de centralisation et de disponibilité des données puisqu'elles sont toutes enregistrées au même endroit et disponibles en ligne de manière constante.

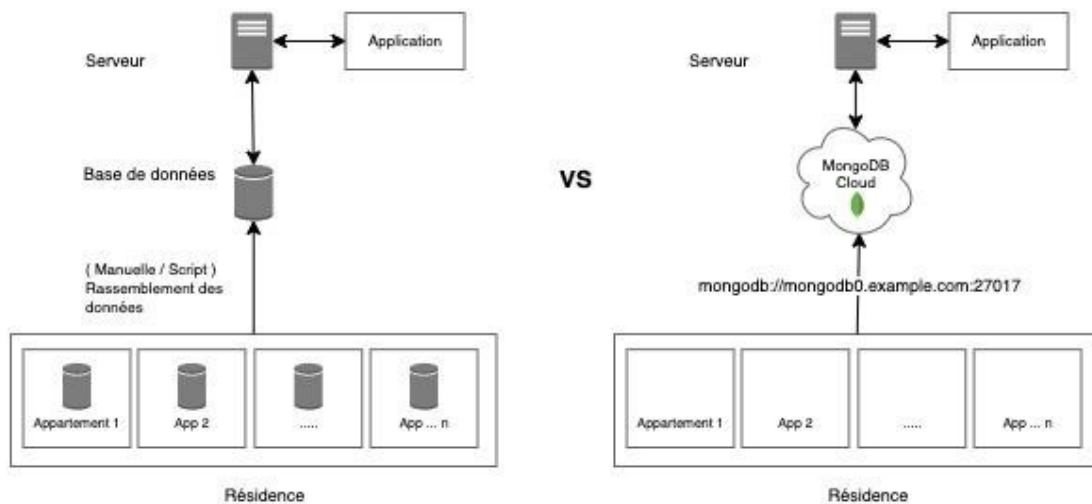


Figure 3.6 : Rassemblement des données vs MongoDB Cloud

© Lucas Gonzalez

3.2.4 - PRÉSENTATION DE L'APPLICATION

Pour mieux comprendre les apports de l'application en termes de fonctionnalités et d'informations, il convient de la présenter plus en détail. Tout d'abord, on peut distinguer, sur les Figures 3.7 et 3.8, deux interfaces accessibles à tout moment depuis la barre de navigation. La première interface offre une vue d'ensemble sur les modules. Depuis celle-ci, il est possible d'être informé de l'état actif ou inactif des modules Figure 3.7, numéro 3, mais aussi si on le souhaite, d'obtenir une rétroaction plus détaillée sur les tâches en cours, ou encore les erreurs et déconnexions qui peuvent survenir, Figure 3.7, numéro 4. De plus, les modules disposent d'un champ *text*, Figure 3.7, numéro 5, utilisé

pour recevoir de l'information sous forme de paramètres à leur initialisation. Plus concrètement, ceux-ci sont utilisés pour fournir l'adresse de connexion d'un *Websockets*, d'une base de données ou bien d'un *tag* utilisateur pour identifier les usagers lors de la récolte. Pour conclure sur cette interface, soulignons que chaque module est activable par l'intermédiaire d'un bouton, Figure 3.7, numéro 6, qui réagit en fonction de l'état du module en question.

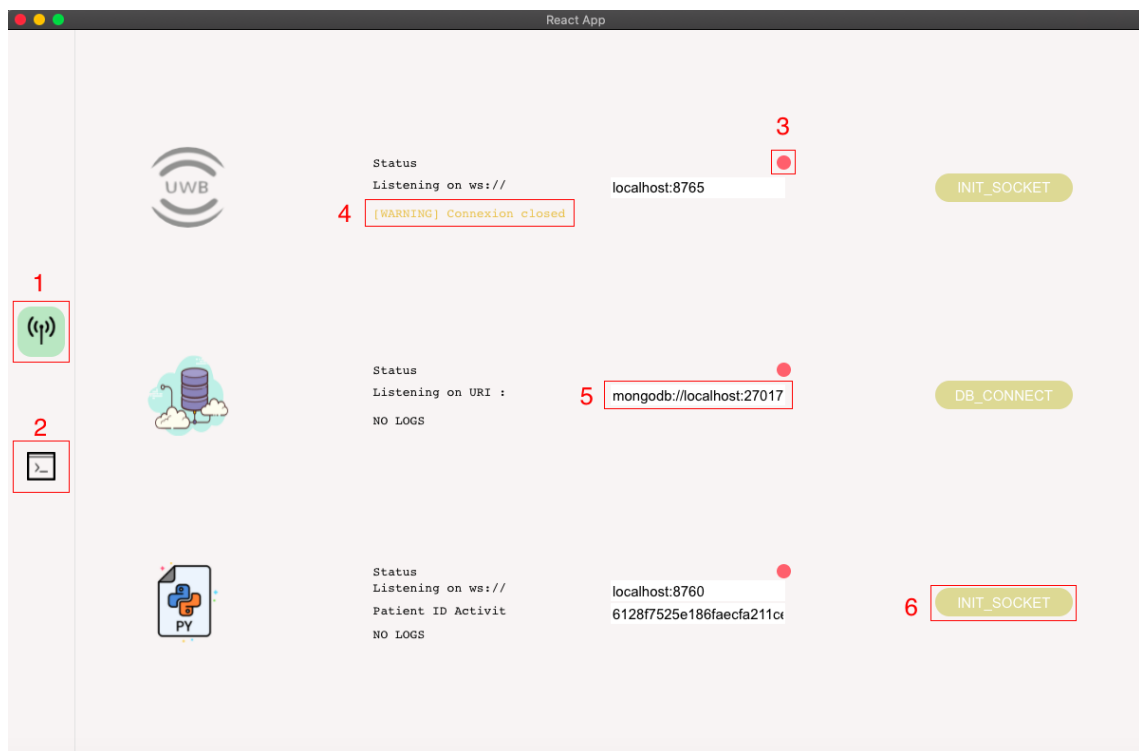


Figure 3.7 : Application de gestion des modules (interface 1)

© Lucas Gonzalez

L'objectif de la seconde interface est d'apporter de l'information un peu plus poussée sur l'activité des modules. Celle-ci est présentée sous la forme de console où l'information est journalisée en direct. Deux interfaces consoles sont mises à disposition, Figure 3.8, numéro 1 : une première où il est possible de voir les activités qui sont prédites et l'utilisateur concerné en direct. Une seconde où l'on peut apercevoir l'activité de la

collecte de données, autrement dit, les enregistrements qui sont faits dans la base de données. Afin de fournir de l'information pertinente, celle-ci est journalisée dans les interfaces et alimentée en direct depuis les modules.

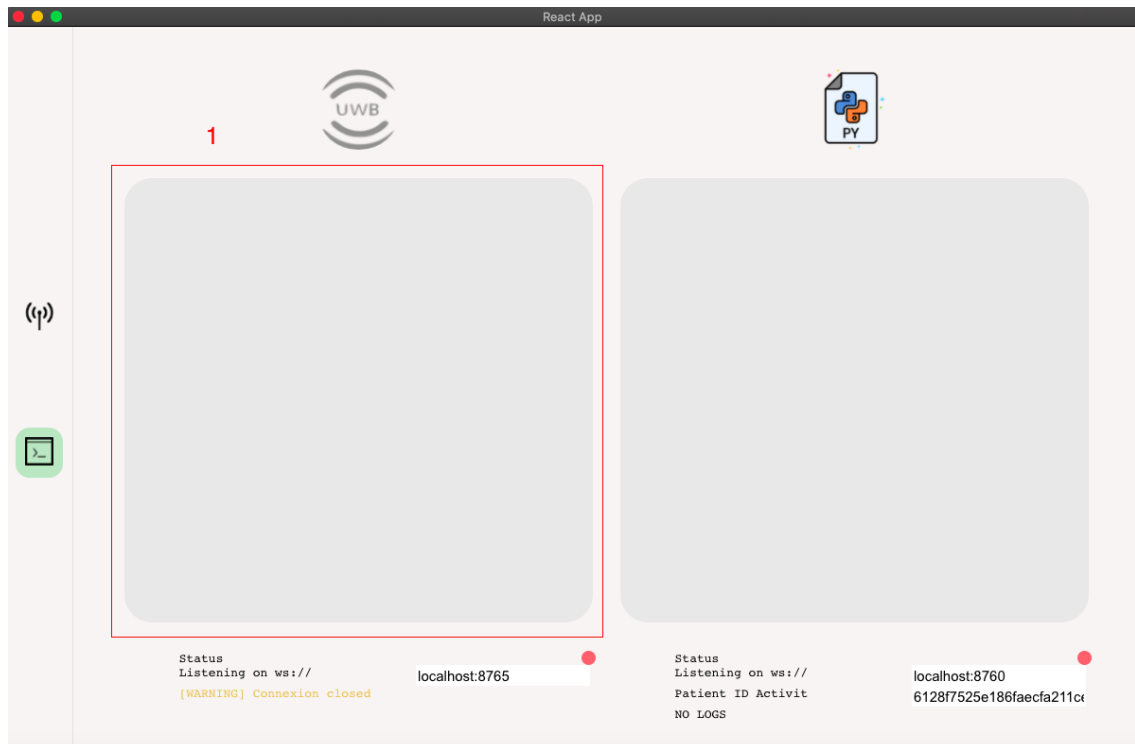


Figure 3.8 : Application de gestion des modules (interface 2)

© Lucas Gonzalez

3.3 - TABLEAU DE BORD POUR LES HABITATS INTELLIGENTS

La popularité grandissante des habitats intelligents rapproche peu à peu notre société du rêve d'avoir des milieux de vie autonomes, capables d'offrir du soutien à la vie de leurs résidents. Ainsi, la présentation de l'information est d'une importance majeure, surtout en matière de compréhension. L'information visuelle semble être l'une des plus précieuses pour notre cerveau. En effet, selon le *Massachusetts Institute of Technology*, le cerveau humain est capable de traiter une image en seulement 13 millisecondes (Potter et al., 2013). Un surplus d'information ou bien de données présentées de façon inadéquate

auront de grandes probabilités d'impacter négativement leur compréhension. Il s'agit alors de trouver un équilibre entre représentation visuelle, pertinence et quantité de données.

De fait, les tableaux de bord se présentent comme des outils de visualisation décisionnelle capables de satisfaire les utilisateurs avec de l'information pertinente. Cependant leur conception s'avère plus délicate qu'on peut le penser. Lorsque les données observées ne changent jamais, il est assez simple de concevoir un modèle de visualisation. Les besoins en conception et en design sont alors plutôt faibles et il n'est pas nécessaire de concevoir un système complexe. Cependant, dès lors que l'on souhaite un système un peu plus flexible adapté à nos besoins de visualisation, les étapes de conception se compliquent. Dans le domaine des habitats intelligents, disposer d'une interface de visualisation où les données sont adaptables à chacun des appartements est une manière de maximiser la pertinence des informations. De plus, avoir la possibilité de choisir le type de visualisation ou de cibler certaines données pour un résident en particulier met en avant les besoins utilisateurs. Que ce soit pour ceux qui les utilisent ou y vivent, les tableaux de bord présentent de l'information plus pertinente et disposent d'une supervision plus personnalisée. Ainsi, un système de tableaux de bord adaptable semble être plus bénéfique pour l'ensemble des parties prenantes.

Dans le cadre de cette étude, notre objectif a donc été de développer une méthode de conception pour les tableaux de bord adaptatifs, plus particulièrement dans les conditions d'un habitat intelligent équipé pour détecter des chutes de personnes et des activités de la vie quotidienne. Pour cela, plusieurs éléments sont nécessaires. Tout d'abord, un système capable de chercher et stocker des données de manière flexible et efficace. Ensuite, un système capable de faire circuler l'information entre les ressources

et une application. Enfin, une application capable de traiter ces données et de les présenter de manière simple et concise. Dans les sections qui suivent, nous abordons, plus en détails, les choix de conception et les raisons pour lesquelles ils sont importants.

3.3.1 - ARCHITECTURE DE DÉVELOPPEMENT (MVC)

Lorsqu'on développe un modèle d'architecture logiciel, il est important de mettre l'accent sur la séparation entre la gestion, l'acheminement des données et l'affichage du logiciel. L'objectif est de permettre une meilleure répartition du travail et une maintenance du projet améliorée. Le motif d'architecture logiciel *Modèle-Vue-Contrôleur* (MVC) lancé en 1978 est très populaire pour les applications web, car il offre un équilibre en évolutivité et indépendance. C'est un modèle d'architecture assez souple qui permet à chacune des couches d'évoluer sans induire de changements dans les autres couches. Ainsi, il se compose de trois niveaux ayant chacun une responsabilité différente.

La première couche, aussi appelée *Modèle*, est l'élément qui contient les données et la logique qui leur est associée. En d'autres termes, cela correspond à la lecture, l'enregistrement et la validation des données. Elle est connue comme le niveau le plus bas de l'architecture MVC, ce qui signifie qu'elle est responsable de la gestion et de la maintenance des données. Il est intéressant de spécifier que la couche *Modèle* ne communique jamais directement avec la couche de présentation graphique. Elle répond essentiellement aux requêtes du contrôleur afin de transmettre les données nécessaires et ceci dans le but de minimiser toute forme de dépendance entre le stockage des données et leur présentation. La seconde couche qui est la partie visible d'une interface graphique est appelée *Vue*. Elle contient l'ensemble des éléments visuels ainsi que la logique nécessaire pour afficher les données provenant du *Modèle*. Pour les applications web, lorsqu'on fait référence au composant de la *Vue*, cela correspond principalement à la

partie HTML/CSS de l'application, mais ce n'est pas la seule façon de définir la composante *Vue* du modèle MVC. Finalement, elle représente toute forme d'information qui est présentée telle que les graphiques, les diagrammes mais aussi chaque élément d'une interface graphique. La dernière couche correspond à celle du *Contrôleur*. Elle est relativement connue pour être l'élément principal de l'architecture car c'est le composant qui agit en tant qu'intermédiaire entre les deux autres couches. Elle permet l'interconnexion entre les *Vue* et le *Modèle*. Le *Contrôleur* n'a pas à se soucier de la gestion de la logique des données, il indique simplement au *Modèle* ce qu'il doit faire. Après avoir reçu les données du *Modèle*, il les traite, puis transmet toutes ces informations à la *Vue* qui pourra alors gérer son affichage.

Encore aujourd'hui, l'architecture MVC reste un modèle très répandu dans le domaine du développement d'applications et voici plusieurs raisons pour lesquelles nous avons fait le choix de l'adopter. Tout d'abord, comme il existe une ségrégation du code entre trois niveaux, il devient extrêmement facile de diviser et d'organiser la logique des applications à grande échelle. Le principal avantage de l'utilisation de ces pratiques de code est qu'elles aident à diminuer la dépendance du code et facilitent l'intégration de nouvelles fonctionnalités en toute simplicité. Ainsi, tout changement dans une certaine section de l'application n'affectera jamais l'ensemble de l'architecture. Un autre avantage que nous apporte ce modèle est la facilité à diversifier nos *Vues*. En effet, dans le cas où on souhaiterait présenter de l'information adaptée à un type d'utilisateur, la séparation entre la couche graphique et la couche de données offre une flexibilité et une agilité de développement particulièrement intéressante. En concordance avec les méthodes de développement *Agile* de plus en plus populaires dans le monde de l'informatique, il devient alors facile de créer, modifier, ajouter ou éliminer de nouvelles interfaces. De même, dans le cadre de notre recherche, il était important de disposer de ce niveau de

flexibilité, car le projet était sensible à des changements de technologie durant les premiers mois de développement. Dans le cas d'un changement imprévu, nous aurions pu assurer une adaptation rapide aux modifications qui n'impactent pas l'ensemble du projet.

3.3.2 - CHOIX DES TECHNOLOGIES

L'une des étapes les plus importantes à suivre lorsqu'il s'agit de développer un produit logiciel est de choisir la bonne pile technologique. La pile technologique est aux projets informatiques ce que les matériaux de construction sont à une maison. Il s'agit de la liste des langages de programmation, des outils et des cadres que les développeurs peuvent combiner pour créer des applications Web ou mobiles. Le terme *pile* est utilisé, car lors du développement d'une application, plusieurs couches sont construites les unes sur les autres. La structure d'une *pile* technologique web se compose généralement en trois parties. Le côté client, le serveur et la base de données. Depuis quelques années, certains cadres ou bibliothèques se spécialisent dans l'une de ces parties pour offrir aux développeurs l'expérience de développement la plus appropriée possible. Par exemple, parmi les cadres spécialisés dans la partie interface, les plus populaires comme *Vuejs*, *Angular* ou *React*, proposent chacun un système de création d'applications Web monopage via la création de composants, mais la façon dont sont gérés les composants et leurs états varie d'un cadre à l'autre. Ainsi, on comprend déjà qu'il va falloir faire des choix dans le mode de conception de notre application.

Un autre facteur à prendre en compte est que certains cadres s'associent mieux avec d'autres lors l'assemblage d'une pile technologique. Si on prend l'exemple de la composition MERN (MongoDB, Express, React, NodeJs), que cela soit pour la circulation des données, la facilité d'intégration, ou le potentiel d'évolution de

l'application, les technologies disposent de code constamment mis à jour par les auteurs ou la communauté, qui leur permettent de s'accorder parfaitement entre elles. En effet, même si certaines piles technologiques semblent très similaires (MEAN (MongoDB, Express, Angular, NodeJs), MEVN (MongoDB, Express, VueJs, NodeJs) et tout aussi efficaces, d'autres alternatives telles que la pile LAMP (Linux, Apache, MySQL, PHP) sont aussi reconnues dans le monde de l'industrie. Au final, chaque pile technologique permet d'aboutir au même résultat, c'est-à-dire, au déploiement d'une application, mais les particularités technologiques peuvent spécialiser une pile dans un domaine plus qu'un autre.

Sans entrer dans une étude comparative entre chacune des différentes piles technologiques, dans les sections qui suivent nous abordons les choix technologiques pour l'application de tableaux de bord.

3.3.2.1 - FRAMEWORK FRONT-END

L'apparition des cadres et bibliothèques d'interface utilisateur *JavaScript* fonctionne par cycles. À peu près tous les six mois, un nouveau cycle apparaît, affirmant qu'il vient de révolutionner le développement de l'interface utilisateur (Kumar & Reinartz, 2018). Dans la foulée, plusieurs milliers de développeurs choisissent de l'adopter, des articles de blog sont écrits, des centaines de questions sur la populaire plateforme d'entraide *Stack Overflow* sont posées, puis six mois plus tard, un nouveau encore plus révolutionnaire fait son apparition pour prendre sa place. Les cycles sont parfois tellement courts que le manque d'une définition distincte complique la détermination du nombre exact de cadres *JavaScript* qui existent au total. Ainsi, selon Wohlgethan, E. (2018), les cadres offrent une pile complète de fonctions utilitaires et assument la responsabilité des décisions au lieu que le développeur ait à le faire avant d'entrer

réellement dans l'écriture du code de l'application. Ceci comprend des stratégies pour le routage dans l'application, la gestion des états, le regroupement et autres fonctionnalités. En outre, ce sont des cadres fournissant des améliorations du flux de travail, qui incluent les meilleures pratiques pour les aspects de développement de base comme la structure globale d'une application ou la génération code passe-partout. Il est important de noter que la plupart des cadres sont basés sur le modèle des composants. La logique derrière le concept de composants est importante car ils décrivent à eux seuls l'interface utilisateur. Lorsque des changements surviennent dans les données, le cadre met à jour automatiquement le composant d'interface utilisateur et c'est dans la gestion de l'état des composants que l'on trouvera le plus de différences entre les multiples cadres qui existent.

Selon *State of JS 2020*, Figure 3.9 et le *Stack Overflow Survey 2021* où plus de 80 000 développeurs donnent leur avis, *React*, *VueJs* et *Angular* sont les cadres web les plus populaires depuis quelques années. *React* et *Angular* ont probablement tiré leur popularité du fait qu'ils sont développés par des géants de l'industrie, tels Facebook et Google. Outre le fait que cela apporte une certaine garantie sur la stabilité et la fiabilité du développement, ce sont des cadres qui ont su faire leurs preuves sur le marché. Que se soit McDonald's, Apple ou Nike pour *Angular* ou bien Netflix, PayPal et Uber pour *React*, il ne fait aucun doute que ce sont aujourd'hui des outils vastement adoptés par la communauté des développeurs. Cependant, même si ces deux cadres ont pour même objectif la création d'applications web, les outils qu'ils mettent à disposition ne sont pas les mêmes. Quand l'un propose l'avantage d'un *virtual DOM* (concept de programmation où une représentation virtuelle d'une interface utilisateur est conservée en mémoire et synchronisée avec le vrai DOM (Document object model), l'autre ne le fait pas. Et inversement, quand l'un propose un système de liaison des données bidirectionnel, l'autre

ne le fait pas (cela fait référence au partage de données entre une classe de composants et son modèle). Si des données sont modifiées à un endroit, elles seront automatiquement calculées à l'autre extrémité. Par exemple, si on change la valeur d'un champ texte, elle mettra également à jour la valeur de la propriété jointe dans la classe du composant.

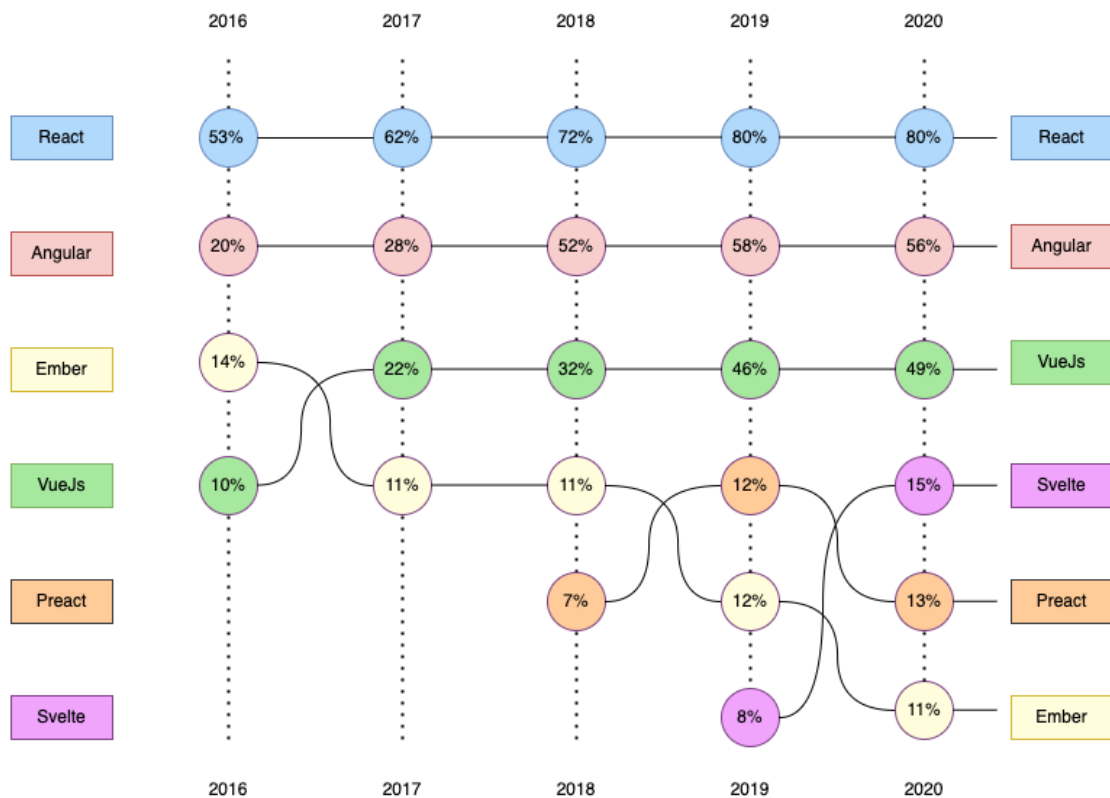


Figure 3.9 : State of JS, utilisation des framework front end

© Lucas Gonzalez

Vue.js est peut-être considéré comme le cadriciel le plus récent dans ce domaine. Bien que sa première sortie se soit produite environ deux ans avant *Angular 2*, ce dernier a été construit sur divers concepts de la première itération (c'est-à-dire *AngularJS*). C'est Evan You, un ancien employé de Google où il a beaucoup travaillé avec *AngularJS* qui a publié le cadriciel fin 2014. Bien qu'il ne soit pas strictement associé au modèle MVVM (Model-View-Viewmodel), les principes de conception de *Vue* ont été en partie inspirés

par ce dernier. D'après leur site officiel¹⁰ de documentation pour *Vue*, le cadriceil peut être utilisé à la fois pour de petits projets où la bibliothèque principale est utilisée parmi d'autres technologies et pour des projets complets à part entière. C'est pourquoi considérant l'envergure de notre projet, la courbe d'apprentissage et la facilité d'intégration, nous avons choisi *Vue.js* comme cadriceil de développement front-end.

Ce qui a séduit la communauté, c'est la simplicité d'utilisation du cadriceil. En effet, la Figure 3.10 présente comment instancier un composant qui écoute les événements d'un bouton sur une page web et met à jour les données d'une variable de manière bidirectionnelle (dans le composant et la page web). Peu de lignes de code, une syntaxe claire et limpide, capable de mettre en œuvre de nombreuses fonctionnalités.

```
1 <div>
2   <p> J' ai {{ age }} ans </p>
3
4   <button v-on:click=" age+=1 "> Add </button>
5
6 </div>
7
8 export default defineComponent({
9
10   name: "exemple",
11   data() {
12     return { age : 0 };
13   },
14 });
```

Figure 3.10 : Instanciation d'un composant vue

© Lucas Gonzalez

Un autre facteur qui permet à *Vue* de se démarquer, est le développement des composants en un seul fichier, Figure 3.11. En d'autres termes, cela permet aux

¹⁰ <https://vuejs.org/>

développeurs de créer de nouveaux composants en utilisant seulement les standards HTML, CSS et *vanilla JavaScript*, tout cela dans un seul fichier qui prend l'extension *.vue*. Cela fait une grande différence à la fois dans les opérations de maintien et d'organisation de code mais aussi vis-à-vis de la courbe d'apprentissage des technologies, contrairement à *React* qui nécessite d'apprendre la syntaxe JSX (*JavaScript Syntax Extension*) pour fonctionner. De plus, on ajoutera à la liste des points forts de *Vue*, l'ensemble des fonctions mises à disposition et faciles à prendre en main qui nous permettent d'intervenir dans le cycle de vie des composants notamment lorsque l'on souhaite récupérer des données avant l'affichage du composant ou bien en envoyer avant qu'il soit détruit. L'ensemble de ces fonctions (*onMounted*, *onUpdated*, *onBeforeMount*, *etc.*) est présenté dans la documentation officielle¹¹.

```
1
2 <template>
3   [...] // Code HTML
4 </template>
5
6 <script>
7   [...] // Code JavaScript/TypeScript
8 </script>
9
10 </style>
11 [...] // Code CSS
12 </style>
```

Figure 3.11 : Composant en un seul fichier (composant.vue)

© Lucas Gonzalez

3.3.2.2 - API

Concernant la couche *contrôleur*, lorsqu'on souhaite fournir de l'information dans nos applications, il nous faut disposer d'un système capable de manipuler et desservir de

¹¹ <https://vuejs.org/api/options-lifecycle.html>

l'information. REST (*Representational State Transfer*) API (*Application Program Interface*) est une architecture standard pour créer et communiquer avec des services Web. Elle impose généralement que les ressources sur le Web soient représentées dans un format texte (comme JSON, HTML ou XML) et puissent être consultées ou modifiées par un ensemble prédéterminé d'opérations. Étant donné que les API REST sont souvent exploitées avec HTTP au lieu d'autres protocoles, ces opérations correspondent à des méthodes HTTP telles que GET, POST ou PUT.

Pour le développement de notre API, nous avons donc choisi d'utiliser *Express* qui selon *Stack Overflow Survey 2021*¹² fait partie du top 3 des cadres les plus populaires. En effet, *Express* est un cadre d'application *Web Node.js* minimal et flexible qui fournit un ensemble robuste de fonctionnalités pour les applications Web et mobiles¹³. Il offre de nombreuses méthodes utilitaires HTTP qui permettent la création d'API, simple et rapide. Comme on peut le voir dans la Figure 3.12, le code pour faire fonctionner le serveur est intuitif et nous permet de fournir de l'information à toutes les requêtes HTTP de type GET à destination du port 3000.

```
1  const express = require('express')
2  const app = express()
3  const port = 3000
4
5  app.get('/', (req, res) => {
6    res.send('Hello World!')
7  })
8
9  app.listen(port, () => {
10   console.log(`Example app listening on port ${port}`)
11 })
```

Figure 3.12 : Initialisation d'un serveur web API

© Lucas Gonzalez

¹² <https://insights.stackoverflow.com/survey/2021#technology-most-popular-technologies>

¹³ <https://expressjs.com/>

De plus, parmi l'éventail de fonctionnalités que nous propose *Express* dans la programmation du serveur, les ressources principales que nous avons utilisées sont: le routage via attributs et la gestion de session utilisateur. Le routage par attribut nous a permis de définir des points d'accès spécifiques aux ressources que l'on souhaite utiliser. Dans le cadre d'un tableau de bord pour les habitats intelligents, cela fait référence à l'enregistrement des résidents, des activités de la vie quotidienne, la restitution des données sur les interfaces graphiques, en d'autres termes, l'ensemble des ressources qui nécessite un échange client-serveur. Dans la Figure 3.13, il est possible de faire la différence entre deux moyens différents que nous avons utilisés pour accéder à nos ressources. Le premier est le routage classique où l'on accède à l'ensemble des données sur nos résidents si notre requête correspond à la chaîne de caractères */résidents*. Le second correspond à l'accès à la mise à jour d'un résident en spécifiant un identifiant unique dans la requête. Ce sont les deux méthodes que nous avons utilisées pour desservir de l'information ou fournir un service d'actions.

```
1 app.get('/residents', (req, res) => {
2   ...
3   res.send(data)
4 })
5
6 app.post('/residents/update/:id', (req, res) => {
7   ...
8   res.send(dataUpdated)
9 })
10
```

Figure 3.13 : Routage classique et routage par attribut

© Lucas Gonzalez

La seconde utilité de notre API est la mise en place de sessions utilisateurs. En effet, elle permet de rapprocher l'application des conditions réelles d'utilisation, c'est-à-dire que chaque utilisateur dispose d'un *session cookie* qui lui permet à la fois d'être

reconnu par le serveur de manière unique mais aussi de disposer de données personnalisées auxquelles il a droit. Ainsi, comme on peut le voir dans la Figure 3.14, lorsqu'une requête HTTP est effectuée depuis un navigateur vers notre serveur web, une fonction de validation intercepte l'ensemble des requêtes de type GET et POST et vérifie si le *cookie* attribué lors de l'authentification est toujours valide. Si c'est le cas, la requête peut continuer et chercher le chemin de routage qui lui correspond, sinon un message d'erreur est retourné à l'utilisateur. Avec du recul sur le développement, c'est une fonctionnalité plutôt simple à implémenter qui se révèle essentielle dans la gestion des données utilisateur. Cela permet d'identifier de qui proviennent les requêtes et quelles données lui sont associées dans la base de données. L'objectif est de fournir un accès aux données fidèles aux conditions réelles d'utilisation.

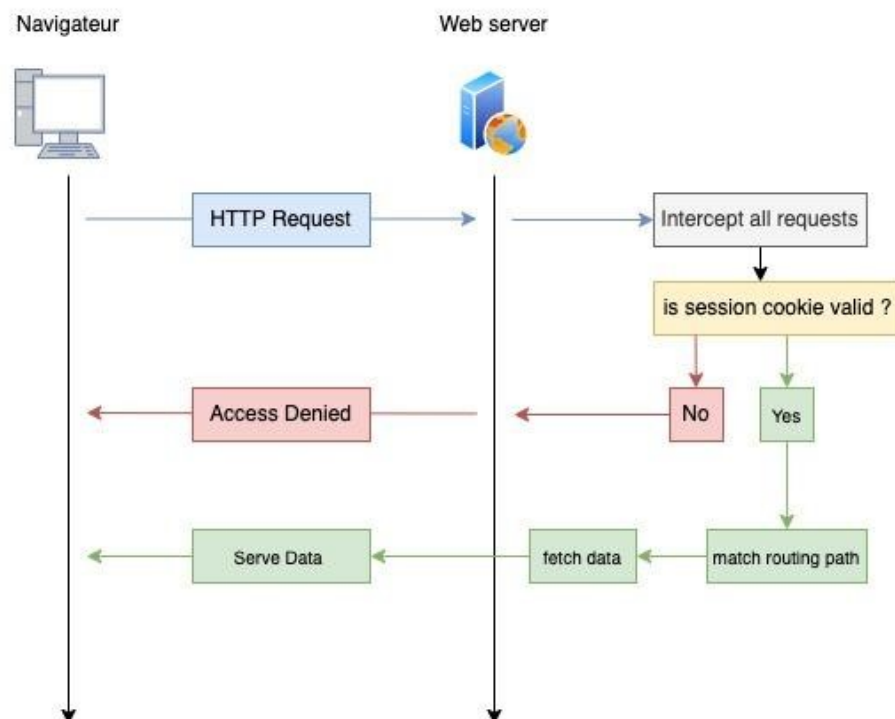


Figure 3.14 : Requête et session utilisateur

© Lucas Gonzalez

3.3.2.3 - BASE DE DONNÉES

Il est difficile de penser à un système qui ne possède pas d'un moyen de persistance des données. Dans le contexte du stockage de données d'un système informatique, cela signifie que les données survivent après la fin du processus avec lequel elles ont été créées. En d'autres termes, les données sont écrites dans un stockage non volatile. Dans le contexte d'une application, qu'il s'agisse d'habitats intelligents ou de tout autre type de domaine, cela permet de garder une trace des données accumulées et d'effectuer toutes sortes de manipulations de données pour en déduire ou fournir de l'information personnalisée.

En effet, il existe aujourd'hui deux grands types de bases de données. La première est la base de données relationnelle, accessible avec SQL (*Structured Query Language*). Développée dans les années 1970 en mettant l'accent sur la réduction de la duplication des données car le stockage était beaucoup plus coûteux que le temps de développement. Elles fonctionnent avec des schémas tabulaires qui ont tendance à être rigides, complexes et nécessitent généralement une mise à l'échelle verticale coûteuse en ressources. La seconde est la base de données NoSQL (non SQL ou pas seulement SQL) qui a été développée à la fin des années 2000 en mettant l'accent sur la mise à l'échelle, les requêtes rapides, permettant des changements d'applications fréquents et simplifiant la programmation pour les développeurs. Certaines études, comme *A performance of SQL and NoSQL databases* (Li & Manoharan, 2013), démontrent que ce n'est pas l'ensemble des bases de données NoSQL qui sont les plus rapides. Certaines bases de données SQL restent toutes aussi performantes, cela dépend principalement du type d'opérations effectuées sur les données. Finalement, c'est notre besoin de flexibilité sur les opérations et

la construction des données qui a orienté notre choix sur les bases de données non relationnelles.

Notre choix s'est donc porté sur MongoDB, un système de gestion de base de données orienté *documents*. Comme son nom l'indique, les données sont enregistrées sous forme de *documents*, qui sont eux-mêmes dans des *collections*. Du point de vue des bases de données relationnelles, les collections sont comparables aux tables, et les documents à des enregistrements. Les données sont stockées dans des documents flexibles de type JSON, ainsi les champs peuvent varier d'un document à l'autre et la structure des données peut être modifiée au fil du temps. Dans notre pratique de développement, nous avons utilisé MongoDB via la librairie mongoose¹⁴, cela permet aux développeurs de définir des schémas de validation de données fortement typés pour éviter tout enregistrement non désiré (voir Figure 3.15).

```
1
2 var activitySchema = new Schema({
3
4   _id:{ type: Schema.Types.ObjectId, auto: true },
5   user:{ type: Schema.Types.ObjectId, ref: "User" },
6   activity:{ type: String, required: true },
7   date: { type: Date, default: Date.now, required: true },
8   fall: { type: Boolean, default: false, required: true },
9 });
```

Figure 3.15 : Schéma de validation de données mongoose

© Lucas Gonzalez

Elle fournit également toutes sortes de fonctionnalités additionnelles facilitant la manipulation, le filtrage et la gestion des données. Dans la Figure 3.16 présentée ci-

¹⁴ <https://mongoosejs.com/>

dessous, il est possible de voir une des fonctions disponibles nous permettant de filtrer les données et un exemple de résultat. On utilise le schéma d'une activité comme base de recherche et on impose le filtre *fall = true*, qui signifie que l'on recherche tous les documents d'activités qui ont détecté une chute.

```
1
2 const activities = await Activity.find({ fall: true });
3 ...
4 ...
5 {
6   "_id": "616d7b7eff67a787b0ff862",
7   "fall": true,
8   "user": { "id": "6168775f39e8cb3743edfda6" },
9   "activity": "fall",
10  "date": { "$date": "2021-10-18T13:49:50.943Z" },
11  "--v": 0
12 }
```

Figure 3.16 : Filtrage de données mongoose

© Lucas Gonzalez

De plus, lorsque l'on choisit un système de base de données, il arrive souvent que des données soient liées entre elles. La librairie mongoose met donc à disposition un système de propagation extrêmement puissant lorsque l'on souhaite récupérer des données. Pour une base de données relationnelle, cela serait l'équivalent de la récupération en cascade. Il est important de noter que les bases de données relationnelles ne font généralement pas la différence entre les différents types d'opérations, que ce soit pour récupérer, mettre à jour ou supprimer, il n'est pas nécessaire de gérer le chaînage de données. Ainsi dans le cas de MongoDB, il est important de noter que les références entre les documents s'il y a lieu, ne disparaissent pas en chaîne à leur suppression ou à leur mise à jour. Il faut alors spécifier un intergiciel qui sera responsable de cette tâche. Pour mieux illustrer la différence entre la récupération et la suppression en cascade, un exemple

est donné à la Figure 3.17. Pour la récupération, le schéma *Activity* contient une référence à un utilisateur, Figure 3.15, puis c'est la fonction *populate* qui permet de récupérer les données. Dans les paramètres de la fonction, on spécifie la référence *user*, nos *attributs* et lorsqu'on reçoit les données, le document utilisateur est ajouté au document de l'activité. Pour le cas de la suppression, voici un autre exemple. Une résidence contient plusieurs appartements et lorsqu'on supprime un appartement, il faut le retirer de la résidence. Voici donc comment mongoose nous propose de le faire, c'est avec un intergiciel qui écoute l'action *deleteOne* sur le schéma appartement et lorsque cela arrive, les données contenues dans cet appartement sont mises à jour.

```
1 // Recuperation Cascade
2
3 const activity = await Activity.find({ fall: true }).
4   populate(
5     {
6       path: "user",
7       select: "nom adresse"
8     }
9   )
10
11 // Suppression Cascade
12
13 apartmentSchema.pre("deleteOne", async function () {
14   ...
15   Residence.updateMany(
16     { _id: doc.residence },
17     { $pull: { apartments: doc._id } }),
18   ...
19 });
```

Figure 3.17 : Récupération et suppression en cascade mongoose

© Lucas Gonzalez

Un dernier point concernant MongoDB explique pourquoi nous avons choisi de l'utiliser dans notre projet. Mis à part qu'il fasse partie du top 4 des systèmes de base de

données les plus utilisés¹⁵, c'est son potentiel de d'évolution horizontale qui nous a confortés dans notre choix. Cela fait référence à l'ajout de nœuds supplémentaires pour partager la charge. Ceci est difficile avec les bases de données relationnelles en raison de la difficulté à répartir les données associées entre les nœuds. Avec les bases de données non relationnelles, les collections sont autonomes et ne sont pas couplées entre elles. Cela permet de les répartir plus simplement entre les nœuds, car les requêtes n'ont pas à être jointes entre les nœuds¹⁶. Compte tenu de notre développement et de la taille du projet actuel, ce n'est pas une fonctionnalité qui a été intégrée, mais la détection d'ADL (activité de la vie quotidienne) est un secteur qui génère une quantité importante de données, ce qui peut rapidement créer une situation de manque de stockage. Il paraissait donc judicieux de s'assurer que des solutions existent et éviter de développer à nouveau notre système de gestion des données.

3.3.3 - DÉVELOPPEMENT DES INTERFACES

En tant que développeur, lorsqu'on entreprend les phases de conception, on a tendance à ne pas s'attarder sur l'aspect visuel. Or, c'est un élément fondamental pour tout type d'application. En effet, selon Justin Mifsud dans le rapport *The Trillion Dollar UX Problem*, 88 % des consommateurs en ligne sont moins susceptibles de revenir sur un site après une mauvaise expérience. Autrement dit, la conception d'interfaces est importante car elle peut construire ou défaire une base d'utilisateurs. De bon visuels augmentent l'implication utilisateur, le perfectionnent les fonctionnalités et créent un lien fort entre les usagers et l'application. Afin de poursuivre l'objectif de ces bonnes

¹⁵ <https://insights.stackoverflow.com/survey/2021#technology-most-popular-technologies>

¹⁶ <https://www.mongodb.com/databases/scaling>

pratiques de design, notre équipe de recherche étant plus familière avec le développement de fonctionnalités, nous avons fait appel à une graphiste professionnelle.

Ainsi, le travail entre Morgane Cabrol, notre graphiste et notre équipe s'est déroulé en plusieurs étapes. Tout d'abord, la familiarisation et la compréhension du projet. Pour lui permettre de mieux cerner les enjeux du projet, nous avons présenté à cette dernière, les objectifs que nous souhaitons atteindre et le type d'utilisateurs que nous visons. Suite à cela, nous avons établi la liste des fonctionnalités que nous souhaitons voir apparaître dans l'application et présenté quelques ébauches d'interfaces auxquelles nous avons pensé (figure 3.18).

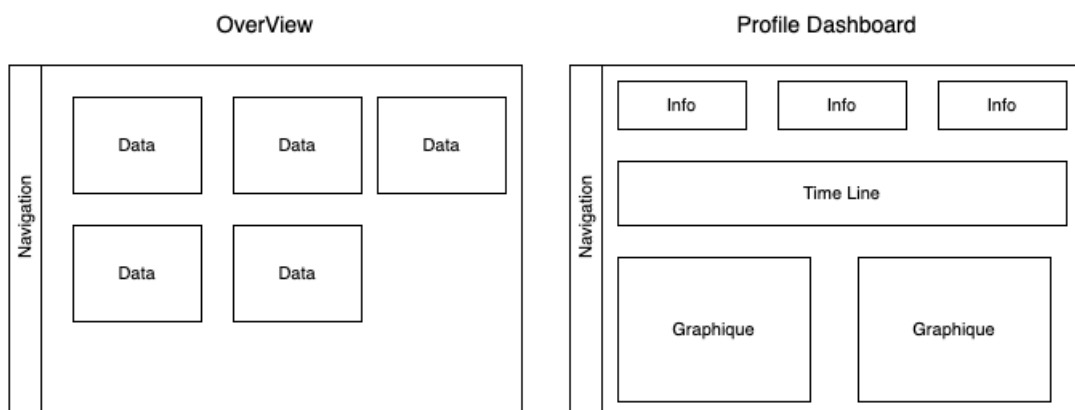


Figure 3.18 : Premières ébauches d'interfaces

© Lucas Gonzalez

A l'issue de nos premières rencontres, une première version des interfaces visible dans la Figure 3.19, nous a été remise. Celle-ci nous a permis de définir notre palette de couleurs ainsi que certains composants clés de l'application tel que : la navigation, les types formulaires et la présentation d'informations sous forme de table de données.

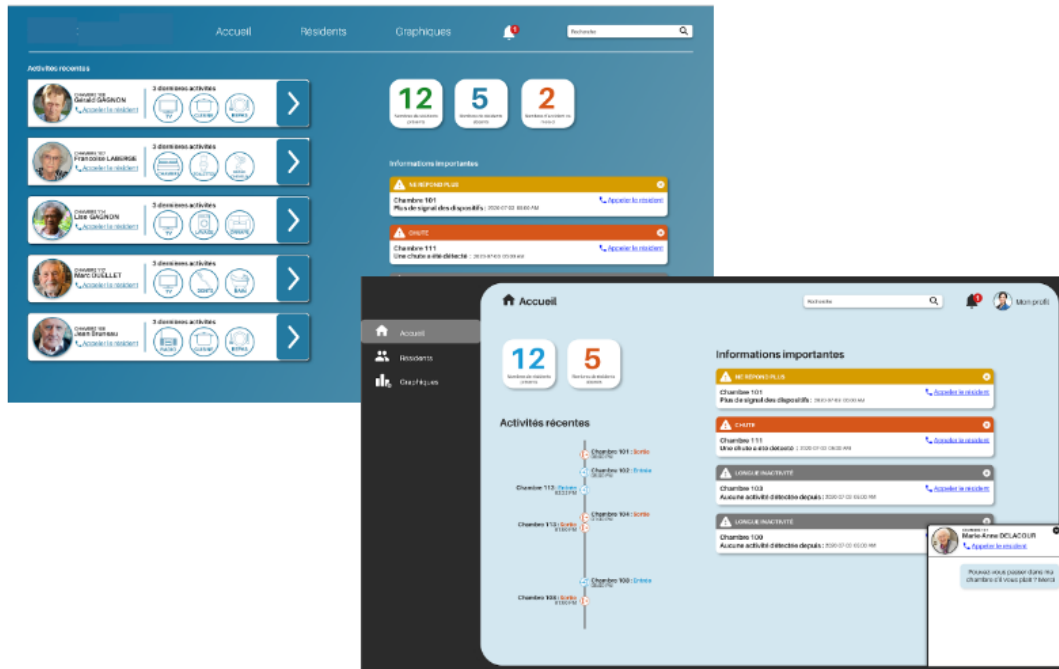


Figure 3.19 : Deuxième itération d’interfaces

© Lucas Gonzalez

Au vu des premiers résultats de ces interfaces, nous n’étions pas totalement satisfaits et avons pris la décision d’effectuer une dernière itération pour à améliorer les visuels. Cela comprend une reprise des couleurs, une réorganisation dans la disposition des composants ainsi que leur affichage, le but étant, d’accentuer le côté *user friendly* des interfaces et de réduire l’écart entre des visuels qui fonctionnent et des visuels agréables à utiliser. La dernière itération d’interfaces effectuée est présentée dans la Section 3.3.5.

3.3.4 - DÉFIS DE CONCEPTION

À travers les différentes sections de ce mémoire, plusieurs références évoquent notre besoin de flexibilité. En effet, encore aujourd’hui de nombreuses applications intègrent des fonctionnalités et du code que l’on peut qualifier de rigide : il s’agit de *legacy systems*. En d’autres termes, ce sont des logiciels continuant d’être utilisés alors

qu'il en existe de bien plus modernes. L'obsolescence de ces systèmes et leur rigidité les rendent difficilement remplaçables, sans engendrer des projets coûteux et risqués. L'un des défis majeurs lorsqu'on programme une solution généralisable est qu'elle couvre un maximum de cas d'utilisation possible. En effet, il est plus simple de coder un programme de simulation d'ascenseur plutôt qu'un système capable d'effectuer n'importe quel type de simulation. Ainsi, lorsqu'on commence à travailler sur la généralisation d'un système, il est naturel que l'on rencontre des défis lors de la conception.

3.3.4.1 - MODULARITÉ DES GRAPHIQUES

Dans les premières phases du projet, notre objectif était de proposer un tableau de bord prédéterminé pour les habitats intelligents où l'ajustement des données n'était pas possible. Puis, au cours de nos recherches, nous avons émis l'idée de pouvoir laisser la possibilité aux utilisateurs d'adapter les différentes composantes du tableau de bord en fonction de leurs besoins. C'est pour cela, que comme présenté dans la Section 3.3.5, nous avons intégré une fonctionnalité permettant aux utilisateurs de choisir le type de graphique adapté à leur besoin, au lieu d'avoir à tout le temps utiliser une composition rigide.

Cependant, il est clair que les différents types de graphiques ne sont pas tous identiques. C'est-à-dire qu'ils ne demandent pas le même format ni le même type de données pour fonctionner. Par exemple, un graphique peut demander en entrée un tableau x correspondant aux abscisses et un y correspondant aux ordonnées, ne contenant que des entiers. À l'inverse, un diagramme circulaire, un tableau rempli de nombres pour les valeurs et un rempli de chaînes de caractères pour les étiquettes. Voici un exemple, en Figure 3.20, parmi plusieurs formats de données possibles.


```

1
2 var graphique = [{
3
4   x: [1, 2, 3, 4],
5   y: [10, 15, 13, 17],
6   mode: 'lines'
7 }];
8
9
10 var diagrammeCirculaire = [{
11
12   values: [19, 26, 55],
13   labels: ['Residential', 'Non-Residential', 'Utility'],
14   type: 'pie'
15 }];

```

Figure 3.20 : Graphique et type de données Plotly.js

© Lucas Gonzalez

Pour tenter de résoudre ce problème et se rapprocher d'une utilisation des graphiques plus flexible, il a fallu concevoir une méthode pour accorder à la fois configuration et formatage de données. Ainsi, pour mieux comprendre comment les données sont traitées et adaptées aux graphiques, la Figure 3.21 présente le fonctionnement d'initialisation d'un graphique jusqu'à la présentation des données. Tout d'abord, les composants sont instanciés dans l'application et leurs configurations sont récupérées dans la base de données (Figure 3.21 numéro 1). Lorsque la configuration est bien reçue, chaque graphique devient un composant responsable de récupérer ses propres données. Ils émettent alors une requête, spécifiant les paramètres de filtrage présents dans la configuration (date de début, date de fin, résident) (Figure 3.21 numéro 2). Une fois les données filtrées et retournées depuis la base de données (3), c'est le formateur qui intervient. Ainsi, en fonction du type de graphique spécifié dans la configuration, les données sont à nouveau filtrées et formatées dans le format requis (Figure 3.21 numéro

4). Finalement, les données sont transmises à chaque graphique (Figure 3.21 numéro 5) et affichées respectivement dans l'application (Figure 3.21 numéro 6).

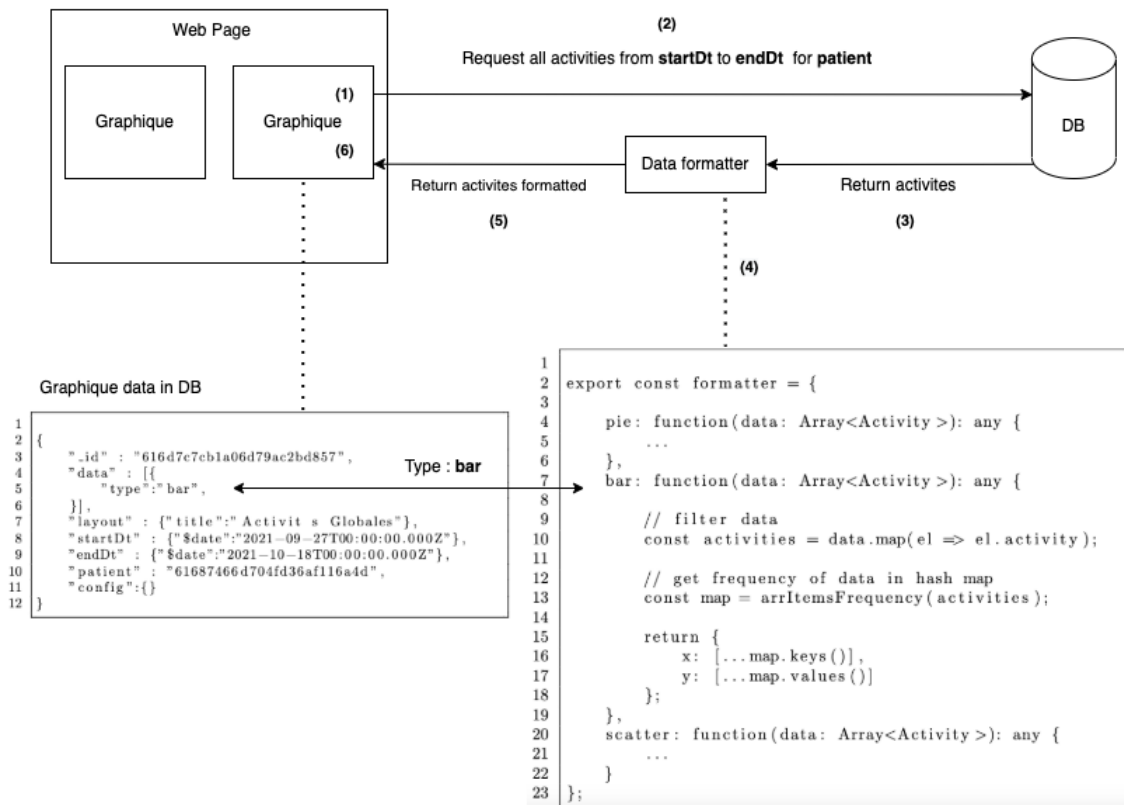


Figure 3.21 : Formatage et adaptabilité des données pour la flexibilité des graphiques

© Lucas Gonzalez

On notera, que nous avons été plutôt limités dans le type de graphique possible à présenter. Cette limite ne provient pas de la méthode employée ci-dessus, ni de la difficulté de formatage, mais du jeu de données en lui-même. En effet, *Plotly* étant la librairie graphique que nous avons utilisée, elle propose de nombreuses façons de représenter les données notamment des *heatmaps*, de *ternary plots* ou encore des figures en 3D. Cependant, aucune n'est réellement compatible ou pertinente avec la nature de nos données sur les activités de la vie quotidienne. Malgré cela, les graphiques disposent de

fonctions plutôt intéressantes, comme du *scaling*, du *zoom* ou bien du filtrage, très utile lorsqu'on souhaite faire de l'analyse de données.

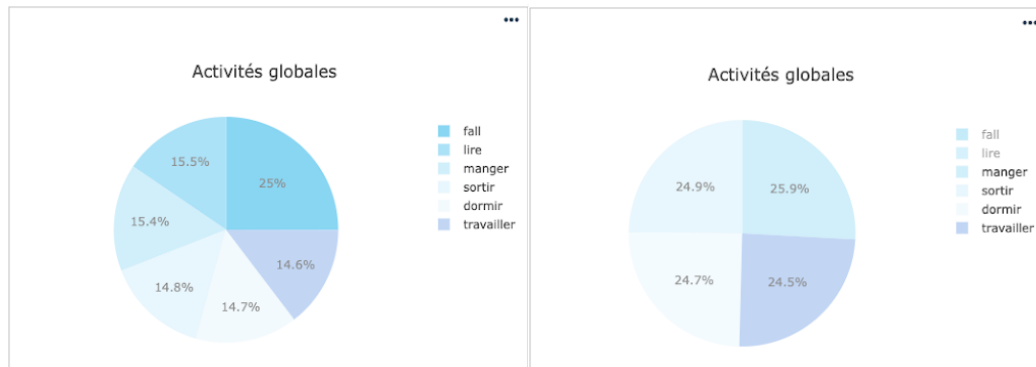


Figure 3.22 : Plotly graphique filtrage de données

© Lucas Gonzalez

3.3.4.2 - DIFFUSION DE DONNÉES EN TEMPS RÉEL

Comme expliqué dans les sections précédentes, l'objectif principal des tableaux de bord est de fournir un résumé des informations dans un format concis, clairement visible et facile à lire. Ajouté à une présentation efficace de l'information, le point fort des tableaux de bord reste la mise à jour en temps réel des données. En effet, grâce à cette fonctionnalité, les utilisateurs ont la possibilité d'analyser l'information de manière optimale et prendre des décisions cohérentes à tout moment.

Pour permettre à notre tableau de bord d'implémenter cette fonctionnalité, il a fallu mettre en place des flux de données continus. Sans début ni fin, ils établissent un lien constant qui peut être utilisé ou modifié sans avoir besoin d'être téléchargé au préalable. Dans le cadre de notre recherche pour les habitats intelligents, cette fonctionnalité est utilisée pour transmettre en direct les données d'activités de la vie quotidienne qui ont été détectées au tableau de bord. A ce jour, le recul sur le

développement permet de constater que cela reste une des fonctionnalités qui a été la plus complexe à concevoir. Du fait que la technologie *streaming* établit un lien persistant entre le serveur, la base de données et le client, il faut être prudent sur plusieurs éléments. Tout d'abord, les conditions d'ouverture et de fermeture des streams sont à prendre en considération. L'idée est d'éviter de laisser des flux de données ouverts indéfiniment, mais aussi de les regrouper pour éviter de créer des centaines de *streams* utilisateurs qui écoutent les mêmes données et gaspillent des ressources. C'est pourquoi pour réduire la charge sur le serveur, nous avons mis en place un système de salle d'écoute illustré à la Figure 3.23. Le système fonctionne de la manière suivante : l'identifiant unique d'un résident est en quelque sorte l'identifiant de la *room*. Ainsi, seuls ceux qui y possèdent la référence peuvent y accéder. Lorsqu'un client veut écouter un flux de données, on vérifie si un flux identique existe. Si oui, il rejoint la *room* avec les autres utilisateurs pour écouter les données, sinon une *room* est créée et des utilisateurs pourront la rejoindre par la suite. De ce fait, on passe d'un système de flux multiple pour la même donnée à un système de flux multiple pour une quantité illimitée d'utilisateurs.

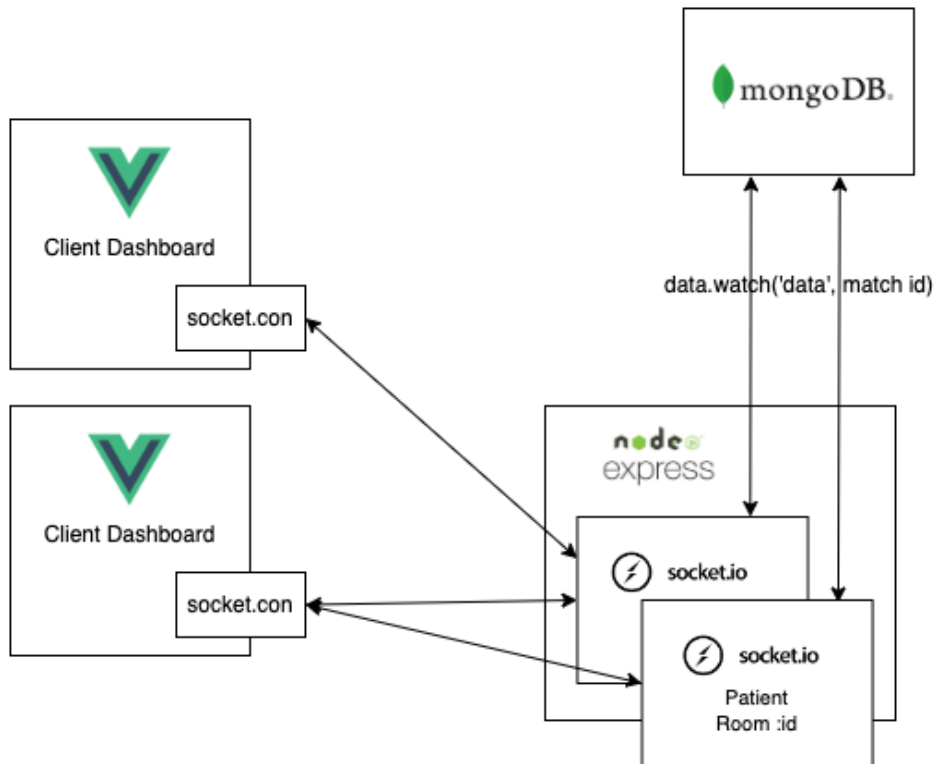


Figure 3.23 : Architecture streaming salle d'écoute

© Lucas Gonzalez

Pour diffuser de l'information actualisée en temps réel, le second problème que nous avons dû résoudre est de créer à nouveau un lien persistant entre notre base de données et le serveur. En effet, lorsque l'on fait référence à des méthodes de récupération de données, on est habitué à des requêtes qui filtrent et transmettent leur résultat sur un ensemble déjà existant. Ainsi, lorsque la requête prend fin, le lien est coupé et aucune autre donnée n'est transmise sans une seconde requête. En substitution, nous cherchons à créer un lien qui filtre et transmet le résultat sur de nouvelles données entrantes. Par exemple, à chaque fois qu'un document *Activité* est enregistré dans la base de données avec l'identifiant unique d'un résident, nous avons défini l'objectif de diffuser l'information sur le flux de données.

De ce fait, MongoDB a introduit dans sa panoplie de fonctionnalité, les *Change Streams*. Ce sont des flux qui permettent d'accéder aux modifications de données en temps réel sans complexité et ni risque. Les applications peuvent utiliser les flux pour s'abonner à toutes les modifications de données sur une seule collection, une base de données ou un déploiement complet et y réagir immédiatement¹⁷. On peut retrouver dans la Figure 3.24, un exemple de notre implémentation des *Change Stream* couplé à *socket.io*¹⁸ pour les salles de streaming. Tout d'abord, le gestionnaire de *socket* reçoit une requête pour écouter un flux d'activités. Ensuite, on vérifie si la salle de *streaming* existe déjà (3.24 section 1). Si oui on la rejoint, sinon on spécifie les paramètres du système de données que l'on souhaite observer (3.24 section 2). On spécifie le format que l'on souhaite obtenir pour chaque nouveau document et l'identifiant qui déclenche le transfert de données. Finalement, on ouvre notre *stream* en utilisant la fonction *openChangeStream*. Celle-ci prend en paramètre le type de schéma de données à observer (*Activity*), la configuration de notre recherche et l'identifiant de la *room* où émettre les données. La fermeture d'un *stream* est plus ou moins identique, on vérifie que plus personne n'utilise le flux et on le ferme, le cas échéant.

¹⁷ <https://www.mongodb.com/docs/manual/changeStreams/>

¹⁸ <https://socket.io/>

```

1 socket.on("join.Activity", (params: any) => {
2
3
4   // Get size of room
5   1 const roomSize =
6     io.
7     sockets.
8     adapter.rooms.get(params.roomId)?.size || 0;
9
10  // Check size so we don't duplicate stream
11  if (roomSize < 1) {
12
13  2    const pipelineActiv = [
14    {
15      $match: { "fullDocument.user":
16        mongoose.
17        Types.
18        ObjectId(params.roomId)
19    }
20  }
21  3    potentialStream = openChangeStream(
22    Activity,
23    pipelineActiv,
24    params.roomId
25  );
26  }
27
28  4    socket.join(params.roomId);
29  });

```

Figure 3.24 : Socket.io et Change Stream

© Lucas Gonzalez

3.3.5 - PRÉSENTATION DE L'APPLICATION

Suite à la présentation technique effectuée ci-dessus, cette section est consacrée à une description visuelle de la composition de l'application et aux fonctionnalités qui y sont implémentées. Cette description peut être comparée à celle d'un parcours utilisateur dans une application. Sont décrites : les différentes interfaces, les données observables et les différentes façons d'y accéder.

Tout d'abord, comme dans la majorité des applications qui contiennent des données utilisateurs, on retrouve une interface de connexion et d'inscription présentée dans la Figure 3.25. L'identification utilisateur permet, d'une part, de faciliter l'association de données entre les utilisateurs et le système. D'autre part, elle permet de garantir une meilleure sécurité de celles-ci.

Figure 3.25 : Interface connexion et inscription. L'image présente deux écrans d'interface utilisateur. Le premier écran, intitulé 'Connexion', propose un formulaire avec un champ 'Adresse Email' (Email address), un champ 'Mot de passe' (Password), un bouton vert 'Connexion', un lien 'Vous n'avez pas de compte ?' et un bouton 'Créer un compte'. Le second écran, intitulé 'Inscription', propose un formulaire avec un champ 'Adresse Email' (Email address), deux champs 'Prénom' (Nom) et 'Nom', un champ 'Telephone' (phone), un champ 'Mot de passe' (Password), un bouton vert 'Envoyer', un lien 'Vous avez déjà un compte ?' et un bouton 'Connexion'.

Figure 3.25 : Interface connexion et inscription

© Lucas Gonzalez

Suite à cela, lorsque l'on est connecté, l'utilisateur est directement redirigé sur la page d'accueil. D'ici, il est capable de visualiser l'ensemble des résidences auxquelles il est associé (Figure 3.26 numéro 2), d'en créer de nouvelles (Figure 3.26 numéro 3) et d'appliquer des filtres de tri par ordre croissant, décroissant ou bien des filtres secondaires (nom, adresses, etc.) (Figure 3.26 numéro 5). Un bouton est également disponible pour modifier la présentation des données. Celui-ci permet d'alterner un affichage en carte ou en table de donnée. Finalement, à gauche de l'interface, on retrouve la barre de navigation fixe accessible depuis l'ensemble des interfaces (Figure 3.26 numéro 1). Elle nous permet

d'accéder à la section : *résidences* qui contient l'ensemble des résidences, *appartements*, pour l'ensemble des appartements et comme son nom l'indique, *résident* pour l'ensemble des résidents.

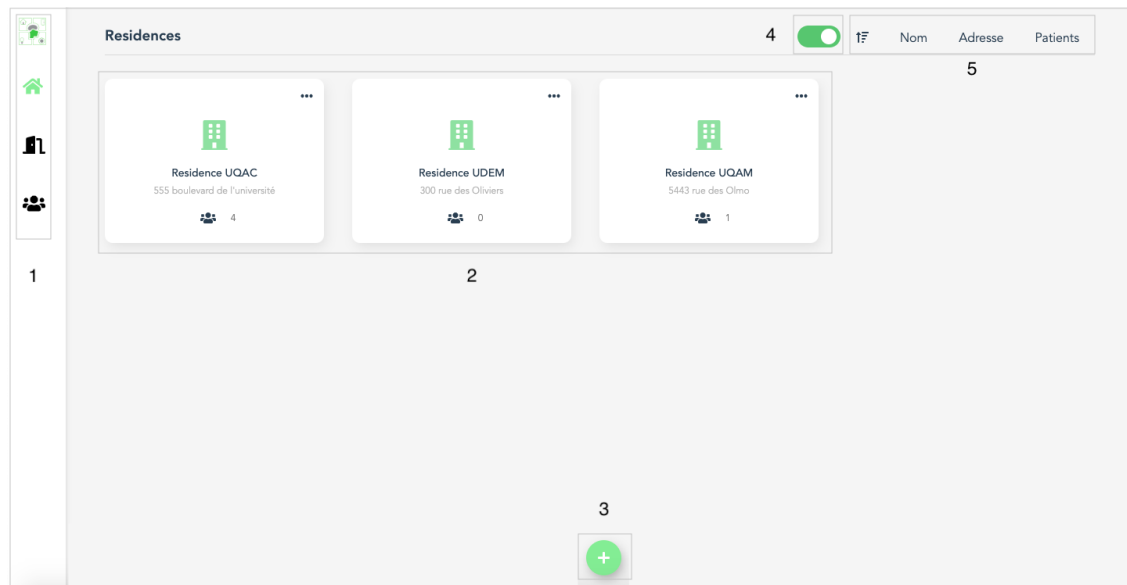


Figure 3.26 : Résidences vue globale

© Lucas Gonzalez

De manière générale, on notera que l'ensemble des interfaces ont tendance à se ressembler et sont construites sur le même principe. D'une part, la section *Overview* (vue globale) présente un résumé des données dans leur ensemble (Figure 3.26). D'autre part, la section *Profil* (Figure 3.26 numéro 2) s'affiche lorsqu'on clique sur une carte pour avoir plus d'informations sur l'objet en question. Comme on peut le voir dans la Figure 3.27, on retrouve plus ou moins les mêmes fonctionnalités. Néanmoins, quelques éléments ont été adaptés pour la section *Résident*. Dans cette section, on est capable d'observer en temps réel les activités que les résidents sont en train d'effectuer (ADL ou chute) (Figure 3.27 numéro 1), mais aussi le nombre de notifications de chute éventuelle pour chacun d'entre eux. (Figure 3.27 numéro 2).

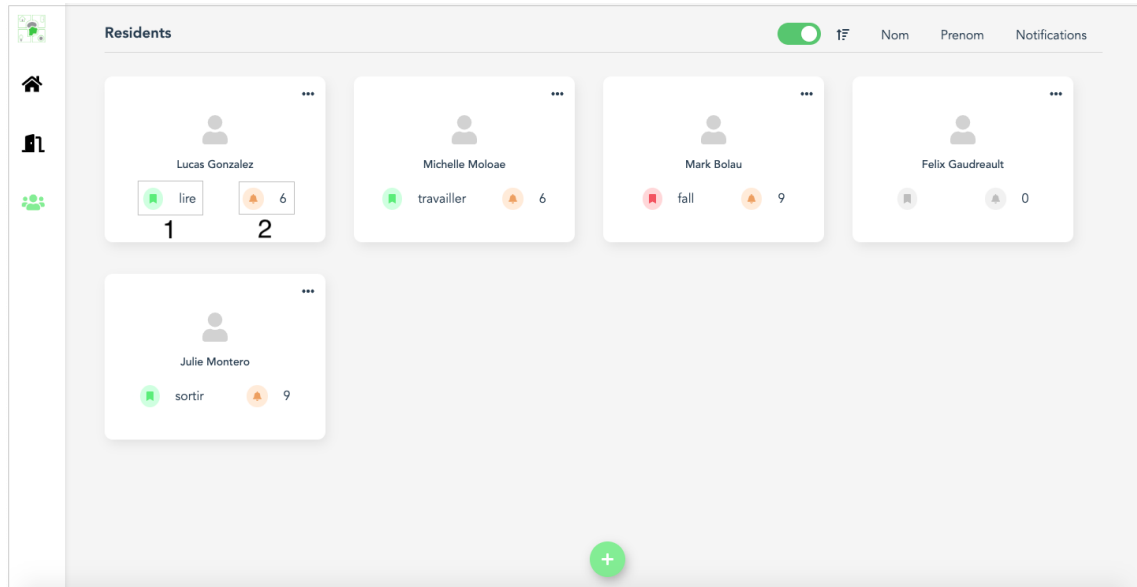


Figure 3.27 : Résidences vue globale

© Lucas Gonzalez

L'interface *Resident Profile* (Figure 3.28) est celle qui concentre le plus de fonctionnalités vis-à-vis de la visualisation de données. En effet, elle nous permet d'obtenir des informations personnelles sur le résident (Figure 3.28 numéro 1), sur son activité en cours (Figure 3.28 numéro 2) et de traiter les notifications de chute (Figure 3.28 numéro 3). On a également la possibilité de visualiser les cinquante dernières activités qui ont été effectuées à l'aide d'une ligne de temps (Figure 3.27 numéro 4) et d'un système de conception de graphique, qui compile des données sur les activités des patients (Figure 3.27 numéro 5). Ainsi, l'interface nous permet de générer autant de graphiques qu'on le souhaite pour un même patient et de spécifier un intervalle de temps pour la pertinence des données. L'idée est de fournir une manière flexible de sélectionner les données tout en s'adaptant aux besoins de visualisations.

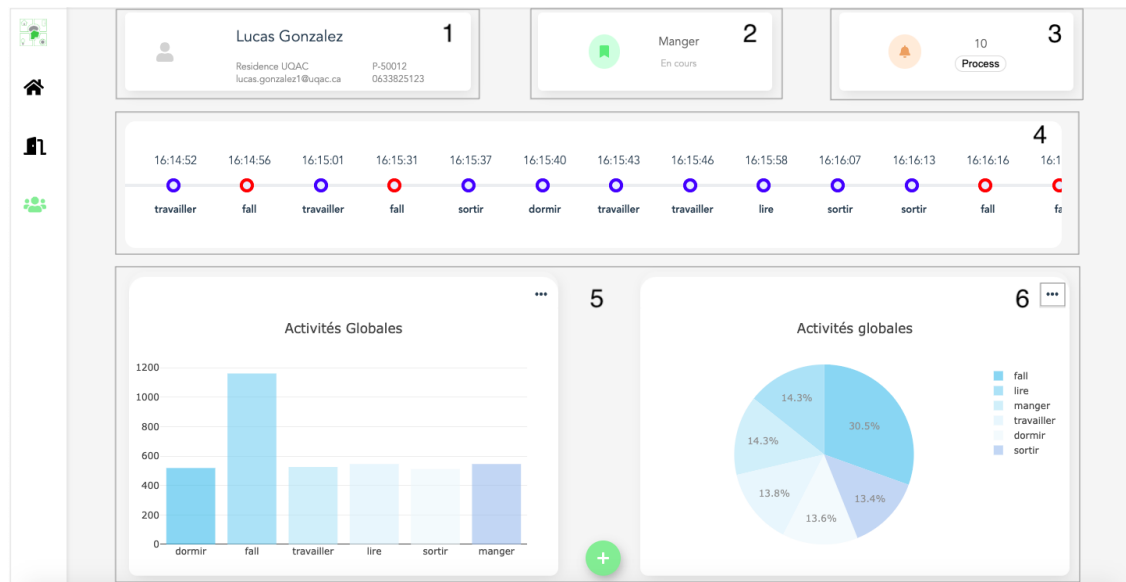


Figure 3.28 : Dashboard résidents profile

© Lucas Gonzalez

Pour conclure sur la présentation des fonctionnalités, il est important de souligner que lorsque l'on met à jour ou que l'on crée de nouvelles données avec les protocoles d'échange web, on fait souvent référence à l'utilisation de formulaires. Ainsi, comme on peut le voir dans la Figure 3.28 (numéro 6), mais aussi dans plusieurs des autres interfaces, la plupart des composants disposent d'une option de modification et de suppression (Figure 3.29). Que ce soit pour les résidences, les résidents, les appartements ou les graphiques, ils disposent tous des outils permettant de répondre aux quatre opérations de base pour la persistance des données : le CRUD (Create, Read, Update, Delete).

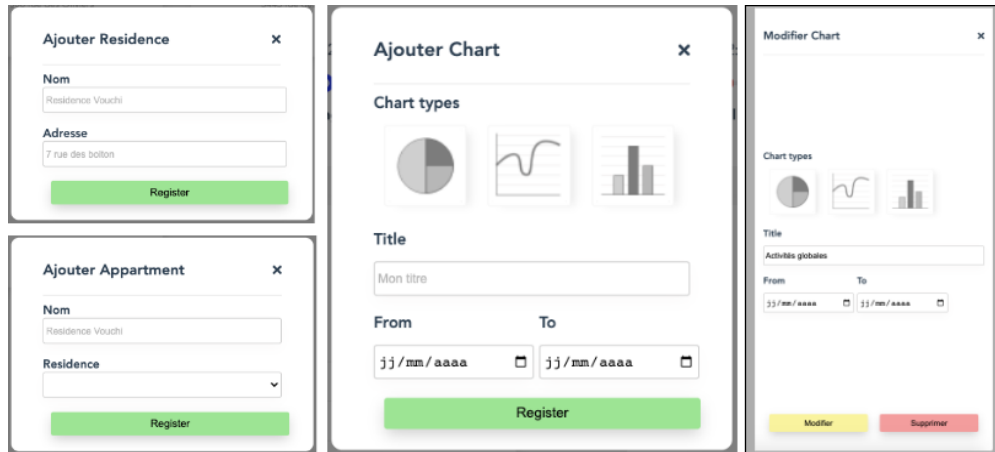


Figure 3.29 : Interface formulaire CRUD

© Lucas Gonzalez

CHAPITRE 4

RÉCOLTE DE DONNÉES

La plupart des sites applications web ne cessent de croître et évoluent de manière constante. Que ce soit dans les services qu'ils proposent ou dans leur méthode de conception, ils continuent d'offrir des fonctionnalités et des services de plus en plus riches aux utilisateurs. Même si cela paraît évident pour certains, tout le monde n'est pas en capacité de s'adapter aux interfaces et aux fonctionnalités qui sont proposées. Ainsi, suite à la première version d'une application, il nous a semblé intéressant de récolter l'avis de plusieurs utilisateurs vis-à-vis de l'utilisabilité de celle-ci. Les phases test jouent un rôle particulièrement important pour les applications car elles peuvent remettre en question l'ensemble du développement et la façon dont les fonctionnalités sont implémentées. Les expérimentations présentées dans ce chapitre ont fait l'objet d'une évaluation par le Comité d'éthique de la recherche de l'Université du Québec à Chicoutimi (CER-UQAC) et cette étude a été approuvée avec le certificat 2022-967.

4.1 - TEST D'UTILISABILITÉ (SUS)

Dans les premières mesures de l'utilisabilité, on parlait surtout HCI (interaction humain-machine). L'objectif principal est de répondre au mieux aux besoins et aux attentes des utilisateurs pour d'améliorer l'utilisabilité des systèmes logiciels. Ainsi, pour évaluer l'utilisabilité, il a fallu être capable de la définir. Selon la norme internationale ISO 9241-11, on définit l'utilisabilité comme suit : l'efficacité, l'efficience et la satisfaction procurées par des produits dans un environnement particulier pour un utilisateur spécifique et à des fins spécifiques. L'évaluation de l'utilisabilité requiert

diverses méthodes qui recueillent les commentaires des utilisateurs. Cependant ce type d'évaluation reste parfois long et fastidieux. Il faut rassembler un nombre important de participants, définir des questions pertinentes pour chaque projet et enfin traiter les résultats.

C'est pourquoi a été créée l'échelle d'utilisabilité du système (SUS). Elle met à disposition un outil fiable et rapide, permettant de mesurer l'utilisabilité. Elle se compose d'un questionnaire de dix éléments, avec cinq options de réponse pour les répondants, allant de fortement d'accord à fortement en désaccord. Développé par John Brooke en 1986 dans l'article *System usability scale (SUS): a quick-and-dirty method of system evaluation user information* (Brooke, 1986), elle permet d'évaluer une grande variété de produits et services, notamment du matériel, des logiciels, des appareils mobiles, des sites Web et des applications¹⁹. Aujourd'hui, le SUS est considéré comme un standard de l'industrie, mais est aussi référencé dans plus de 1300 publications scientifiques. Dans les avantages de ce système, on peut dire qu'il comporte un questionnaire très facile à utiliser pour les participants, il peut être utilisé sur de petites tailles d'échantillons avec des résultats fiables et peut très efficacement différencier les systèmes utilisables et inutilisables. Suite à l'étude du SUS, nous avons décidé de légèrement modifier et d'ajouter quelques questions au questionnaire établi et montré en Figure 4.1. L'objectif était de personnaliser les questions pour qu'elles soient plus adaptées à l'évaluation de notre projet. Cependant, nous avons quand même veillé à conserver la construction des questions originales. Ainsi, le calcul final du score sera adapté à ces modifications.

¹⁹ <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>

Originale SUS	Notre SUS
1. Je pense que j'aimerais utiliser ce système fréquemment	<i>Identique</i>
2. J'ai trouvé le système inutilement complexe	Je n'ai pas trouvé ce système inutilement complexe
3. Je pensais que le système était facile à utiliser.	<i>Identique</i>
4. Je pense que j'aurais besoin du soutien d'une personne technique pour pouvoir utiliser ce système.	Je ne pense pas avoir besoin d'un support technique pour pouvoir utiliser ce système
5. J'ai trouvé que les différentes fonctions de ce système étaient bien intégrées.	<i>Identique</i>
6. Je pensais qu'il y avait trop d'incohérence dans ce système	J'ai trouvé qu'il y n'y avait pas trop d'incohérence dans ce système
7. J'imagine que la plupart des gens apprendraient à utiliser ce système très rapidement	<i>Identique</i>
8. J'ai trouvé le système très lourd à utiliser	Je n'ai pas trouvé que ce système était très lourd à utiliser
9. Je me sentais très confiant en utilisant le système	<i>Identique</i>
10. J'avais besoin d'apprendre beaucoup de choses avant de pouvoir utiliser ce système	Je n'ai pas eu a apprendre beaucoup de choses avant de pouvoir commencer à utiliser ce système.
<i>Null</i>	11. J'ai apprécié les interfaces de l'application.
<i>Null</i>	12. Les informations dans l'application étaient bien organisées, je n'ai pas eu de difficultés à trouver ce dont j'avais besoin
<i>Null</i>	13. Globalement, je suis satisfait du système

Tableau 4.1: SUS originale vs notre SUS

© Lucas Gonzalez

4.2 - DÉROULEMENT DES ENTREVUES

Lorsque l'on effectue une récolte de données, il est important de détailler le déroulement, mais aussi les conditions dans lesquelles celle-ci a été effectuée. L'objectif de ce procédé est d'énoncer l'ensemble des facteurs qui auraient pu influencer les résultats. Ainsi, concernant la phase de recrutement, nous avons tout d'abord diffusé une annonce de recrutement dans l'infolettre Électronique de l'UQAC (Université du Québec à Chicoutimi) pour toucher un maximum de personnes. Suite à cela, les participants intéressés par la recherche ont eu la possibilité de communiquer avec l'étudiant-chercheur responsable du projet pour fixer une date de rencontre. Le jour de la rencontre, les

modalités sont à nouveau transmises au participant. Il lui est expliqué que l'entrevue se déroulera en trois parties.

La première partie consiste en l'explication du contexte de la recherche et des objectifs du LIARA vis-à-vis de celle-ci. Ainsi, il est clairement présenté au participant que le LIARA base ses travaux de recherche sur la problématique de l'informatique ambiante, notamment sur le concept d'habitat intelligent pour l'assistance et le maintien à domicile des personnes âgées²⁰. Il est aussi présenté que les derniers travaux de recherche en collaboration avec ce projet se basent sur la détection d'activités de la vie quotidienne de manière non intrusive afin d'assister les personnes en perte d'autonomie partielle ou totale.

En deuxième partie, nous avons procédé au test de l'application de visualisation. Il est expliqué au participant que l'objectif de cette application est de fournir un outil de gestion capable d'assister la prise en charge de personnes en perte d'autonomie, qu'elle permet de visualiser en direct les activités effectuées mais aussi de formuler des graphiques qui compilent et comparent les données. Lors du test, le participant parcourt l'application de manière individuelle pour éviter toute sorte d'influences extérieures mais il est quand même expliqué que l'étudiant-chercheur est disponible en cas d'inconvénient.

Finalement, il est expliqué au participant qu'il devra remplir un questionnaire visant à évaluer le degré d'utilisabilité de l'application. De même, ce questionnaire ne demande aucun renseignement personnel et sera rempli à l'abri des regards pour éviter toute sorte d'influence. Une fois le questionnaire remis, l'accès à l'identité du participant

²⁰ <https://liara.uqac.ca/index.htm>

est impossible pour qui que ce soit, même pour les membres de l'équipe de recherche puisque les résultats sont mélangés sans aucun facteur d'identification.

Ainsi, le participant à conscience que la participation à ce projet de recherche est volontaire et que les résultats obtenus contribueront à l'avancement des connaissances scientifiques dans ce domaine. Il est libre de refuser d'y participer en ne répondant pas au questionnaire envoyé et qu'il n'y a pas de possibilité de se retirer de l'étude une fois que les réponses ont été soumises, car il est impossible de retracer le questionnaire en question.

4.3 - ANALYSE DES RÉSULTATS

Suite à la réalisation des entrevues, nous avons eu l'opportunité d'interroger un total de 20 personnes. La majorité étant une de nos conditions de recrutement, l'âge de l'ensemble des participants se situe entre 18 et 27 ans. Dans le formulaire de consentement de participation à la recherche, il est clairement mentionné que nous ne récoltons aucune information personnelle permettant l'identification des participants. Cependant, nous avons récolté le domaine et le niveau d'études afin de pouvoir observer la diversité de nos participants. Étant donnée la grande variété de participants présentés dans le Tableau 4.2, on peut considérer la diversité de nos participants comme un point fort de notre échantillon. On peut ainsi dire que les résultats de l'évaluation représentent un public assez diversifié même si concernant le niveau d'études plus de 50% détiennent une maîtrise.

Domaine	Totale
DEC en technologie forestière	1
DEC en Diététiques	1
Licence science et technologie pour l'ingénieur	1
Licence en technologie de l'information	1
Baccalauréat en physiologie	2
Baccalauréat interdisciplinaire en art cinéma vidéo	1
Baccalauréat en administration	2
Maîtrise en science clinique et biomédicale	1
Maîtrise en ingénierie - spécialité informatique	2
Maîtrise programmation jeux video	1
Maîtrise en gestion de projet	2
Maîtrise informatique cybersécurité	2
Maîtrise informatique Big data IA	1
Maîtrise en biologie médicale	1
Maîtrise études cinématographiques	1

Tableau 4.2 : Domaine d'étude de notre échantillon

© Lucas Gonzalez

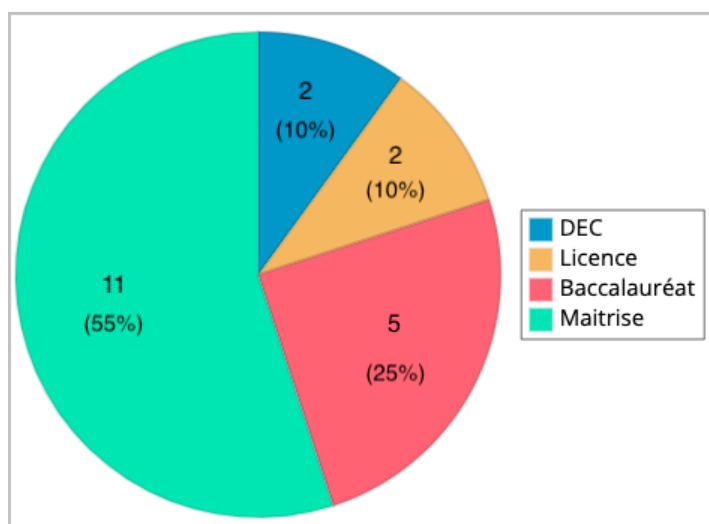


Figure 4.1 : Répartition du domaine d'étude

© Lucas Gonzalez

L'analyse des résultats de notre test d'utilisabilité dans le tableau 4.3 met plusieurs points en évidence. Tout d'abord, la question qui soulève l'avis le plus diversifié est la numéro 4 (*Je ne pense pas avoir besoin d'un support technique pour pouvoir utiliser ce système*). En effet, malgré nos efforts pour rendre l'application la plus abordable à tous, la grande diversité de notre échantillon semble confirmer que certains domaines d'activité, peu familiers avec les interfaces logicielles, ont encore des difficultés à interagir avec notre application. Dans un second temps, la question 6 (*J'ai trouvé qu'il n'y avait pas trop d'incohérences dans ce système*) reçoit le consensus le plus large : 80% de notre échantillon se dit d'accord avec cette affirmation. La question numéro 10 (*Je n'ai pas eu à apprendre beaucoup de choses avant de pouvoir commencer à utiliser ce système*) a obtenu le meilleur score. Nos participants se disent satisfaits avec un score à hauteur de 4,6/5. En revanche, la question numéro 12 (*Les informations dans l'application étaient bien organisées, je n'ai pas eu de difficultés à trouver ce dont j'avais besoin*) a obtenu le score le moins performant (3,95/5) nous orientant vers l'idée qu'une amélioration de la disposition des informations serait probablement bénéfique à l'application.

Questions	Score					Moyenne /5
	Pas du tout d'accords (1)	Pas d'accords (2)	Ni en désaccord ni d'accord (3)	D'accord (4)	Tout à fait d'accord (5)	
Q1	0	0	2	12	6	4,2
Q2	0	0	2	11	7	4,45
Q3	0	0	1	8	11	4,5
Q4	1	1	2	9	7	4
Q5	0	0	3	12	5	4,1
Q6	0	0	1	16	3	4,1
Q7	0	0	3	7	10	4,35
Q8	0	0	0	10	10	4,5
Q9	0	0	2	10	8	4,3
Q10	0	0	1	6	13	4,6
Q11	0	1	2	10	7	4,15
Q12	0	0	5	11	4	3,95
Q13	0	0	0	15	5	4,25

Tableau 4.3 : Résultat du questionnaire d'utilisabilité

© Lucas Gonzalez

Si on souhaite interpréter nos résultats en se basant sur la notation SUS, il est important de souligner ici que les scores des participants pour chaque question sont additionnés puis convertis sur une échelle de 0-100. D'après les consignes d'utilisation, l'obtention d'un score SUS supérieur à 68 serait considéré comme *utilisable et supérieur à la moyenne*. A l'opposé, tout résultat inférieur à 68 serait considéré comme *inutilisable et inférieur à la moyenne*. Ainsi, si on se fie au calcul de la moyenne (Figure 4.2) du score final que nous avons obtenu (85,3), l'application peut être qualifiée d'*acceptable* en termes d'utilisabilité.

$$\left(\frac{\text{sum}(\text{Score})}{\text{sum}(\text{ScoreMax})}\right) * 100 = \left(\frac{55,45}{65}\right) * 100 = 85,3$$

Figure 4.2 : Calcul du score d'utilisabilité

© Lucas Gonzalez

CHAPITRE 5

CONCLUSION

Parmi les enjeux sociaux à surveiller dans les prochaines années, on rappelle le vieillissement de la population. La problématique posée par ce phénomène est la quantité de ressources nécessaires pour répondre à une population croissante et en besoins d'assistance au quotidien. Les habitats intelligents ont généré beaucoup d'intérêt pour répondre à ce problème, cependant la nature complexe des systèmes entraîne le besoin de nouveaux outils intelligents pour assister avec plus de facilité un maximum de personnes. L'objectif de ce mémoire vise à proposer des outils de visualisation adaptatifs pour les habitats intelligents en mettant l'accent sur la modularité des composants. Pour ce faire, plusieurs approches ont été explorées : la modularisation de nos programmes de détection d'activités, le développement d'une application visant à intégrer ces modules, ainsi que le développement d'une application de visualisation de données sous la forme d'un tableau de bord. De plus, une expérimentation supplémentaire a été réalisée dans le but d'évaluer le degré d'utilisabilité de notre application de visualisation.

5.1 - REVUE DES CONTRIBUTIONS

Dans un premier temps, le principe de modularisation a été exploré dans le but d'améliorer l'indépendance de nos programmes dans la résolution d'un problème complexe et afin de faciliter la transmission des résultats. Pour ce faire, un modèle basé sur l'architecture client-serveur a été incorporé à nos travaux de recherche déjà existants. La limite de cette approche reste qu'elle n'a pas été testée en dehors des travaux du laboratoire. Cependant, cette méthode offre tout de même un compromis entre fiabilité,

stabilité et facilité d'intégration. La recherche développée a permis de souligner l'importance d'un système de communication bidirectionnelle pour faciliter la segmentation de nos programmes, ainsi que la propagation de nos résultats. En ce sens, il s'agit d'une contribution relative au domaine, explorée dans le cadre d'un mémoire de Master de recherche.

Par ailleurs, le développement d'application multiplateforme a été étudié dans le contexte d'une application d'intégration visant à gérer les modules que nous avons développés. Pour ce faire, nous avons exploré la technologie Electron capable de déployer une application sur l'ensemble des principaux systèmes d'exploitation. Cette méthode nous a permis de fluidifier le développement de nos programmes et d'ouvrir l'horizon sur la combinaison des technologies que nous pouvons utiliser dans nos projets. En d'autres termes, cela nous a permis de réduire au maximum les limitations liées aux choix technologiques et de concentrer nos efforts sur l'élaboration de solutions. De même, cette deuxième contribution correspond à l'approche explorée dans le cadre du mémoire.

Pour finir, rappelons que nous avons développé une application de visualisation de données pour les habitats intelligents qui s'inscrit dans une méthode basée sur la modularité des composants. Dans ce but, nous avons expliqué nos choix architecturaux et technologiques ainsi que les différents défis de conception rencontrés, c'est-à-dire, la modularité des graphiques et l'émission de données en direct. Cette méthode a permis de définir une ligne directrice dans la conception des applications de visualisation modulaire et d'obtenir un prototype fonctionnel testé par un ensemble de 20 participants. Cette phase test doit être soulignée comme une étape importante de l'ensemble du processus de recherche dans le sens où elle a permis de prendre du recul sur la production de l'application de visualisation et d'introduire une dimension critique dans l'analyse des

objectifs poursuivis. Les résultats semblent plutôt satisfaisants suite à l'obtention d'un score d'utilisabilité de 85,3, ce qui la qualifie d'acceptable selon le *System Usability Scale*.

5.2 - LIMITES

Dans ce mémoire, certaines limites ont été identifiées. En ce qui concerne les applications développées, aucune d'entre elles n'a été déployée dans les conditions réelles d'un habitat intelligent. Même si de nombreuses expérimentations ont été effectuées en laboratoire dans le but de se rapprocher de ces conditions, il reste tout de même une part d'inconnu vis-à-vis d'un déploiement complet. Par exemple, il se peut que certaines interactions entre des humains, des capteurs ou encore l'environnement en lui-même n'aient été pas prévues lors de la conception du système et qu'il en résulte une série de fonctionnement imprévus.

Puis, en ce qui concerne l'application de visualisation aucun test d'utilisabilité n'a pu être réalisé avec des professionnels du domaine de la santé. En effet, si la diversité des participants au test reste un point fort de notre recherche, une partie du public que nous souhaitons soutenir à travers elle fait partie du domaine de la santé. Nous aurions souhaité pouvoir intégrer le retour de spécialistes concernés par nos travaux, mais nous n'avons malheureusement pas pu développer la phase test jusqu'à ce degré plus pointu de consultation.

5.3 - TRAVAUX FUTURS

Plusieurs travaux futurs peuvent découler de cette recherche. Dans un premier temps, il est possible d'explorer directement les limitations énumérées plus haut. Ainsi, il serait intéressant de déployer l'ensemble des applications dans un environnement autre

que notre laboratoire de recherche pour étudier leur comportement et démontrer leur viabilité à long terme.

En ce qui concerne les participants qui ont testé notre application de visualisation, il serait également approprié d'organiser une session de test où parmi l'ensemble des volontaires, certains feraient partie du domaine de la santé, d'autres du public senior. Cela nous permettrait de raffiner nos observations et d'améliorer notre application suite aux commentaires de professionnels du domaine et d'éventuels futurs usagers.

Enfin, concernant la modularité des graphiques que nous avons implémentés, il serait intéressant d'intégrer une fonctionnalité capable de comparer les données de deux individus distincts. Lors des tests d'utilisabilité, quelques participants nous ont spécifié que la comparaison de données entre patients serait une fonctionnalité très intéressante pour déduire de nouvelles informations. En effet, dans la version actuelle de notre application, nous ne sommes pas en mesure de traiter les données de patients de manière simultanée et comparative. Comblers cette lacune apporterait un véritable plus à notre projet d'application de visualisation de tableaux de bords.

BIBLIOGRAPHIE

- Alhamadi, M. (2020). Challenges, Strategies and Adaptations on Interactive Dashboards. Dans *UMAP '20: Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization* (pp. 368–371). Genoa, Italy: ACM. <https://doi.org/10.1145/3340631.3398678>
- Azzam, T., Evergreen, S., Germuth, A. A., & Kistler, S. J. (2013). Data visualization and evaluation. *New Directions for Evaluation*, 2013(139), 7-32.
- Beaulieu, A. (2021). *Combinaison de CNN et LSTM pour la reconnaissance d'activités dans un environnement intelligent avec radars Ultra-Wideband*. Mémoire de maîtrise, Université du Québec à Chicoutimi.
- Beaulieu, A., Thullier, F., Bouchard, K., Maître, J., & Gaboury, S. (2022). Ultra-wideband data as input of a combined EfficientNet and LSTM architecture for human activity recognition. *Journal of Ambient Intelligence and Smart Environments*
- Bertuglia, C. (2020). *Reconnaissance d'activités dans un habitat intelligent à l'aide de radars Ultra-WideBand* (Doctoral dissertation, Université du Québec à Chicoutimi).
- Bikakis, N. (2018). Big data visualization tools. *arXiv preprint arXiv:1801.08336*.
- Bouchard, K., Maitre, J., Bertuglia, C., & Gaboury, S. (2020). Activity recognition in smart homes using UWB radars. *Procedia Computer Science*, 170, 10-17
- Brooke, J. (1986). System usability scale (SUS): a quick-and-dirty method of system evaluation user information. *Reading, UK: Digital equipment co ltd*, 43, 1-7.
- Calvaresi, D., Cesarini, D., Sernani, P., Marinoni, M., Dragoni, A. F., & Sturm, A. (2017). Exploring the ambient assisted living domain: a systematic review. *Journal of Ambient Intelligence and Humanized Computing*, 8(2), 239-257.
- Chan, M., Campo, E., Estève, D. et Fourniols, J.-Y. (2009). Smart homes — current features and future perspectives. *Maturitas*, 64(2), 90 – 97.
- Chen, J., Chen, Y., Du, X., Li, C., Lu, J., Zhao, S., & Zhou, X. (2013). Big data challenge: a data management perspective. *Front. Comput. Sci*, 7, 157-164. <https://doi.org/10.1007/s11704-013-3903-7>

Cook, D. J., Crandall, A. S., Thomas, B. L. et Krishnan, N. C. (2013). CASAS : A smart home in a box. *Computer*, 46(7), 62–69. <http://dx.doi.org/10.1109/MC.2012.328>

Cook, D. J. et Das, S. K. (2004). Overview, Dans *Smart Environments*, (p. 1–10). John Wiley & Sons, Ltd

Cook, D. J., & Das, S. K. (2007). How smart are our environments? An updated look at the state of the art. *Pervasive and mobile computing*, 3(2), 53-73.

Cooper, R. J., Schriger, D. L., & Close, R. J. (2002). Graphical literacy: the quality of graphs in a large-circulation journal. *Annals of emergency medicine*, 40(3), 317-322.

Cousyn, C., Bouchard, K., & Gaboury, S. (2021). Web-based objects detection to discover key objects in human activities. *Journal of Ambient Intelligence and Humanized Computing*, 1-16.

Das, B., Cook, D. J., Schmitter-Edgecombe, M., & Seelye, A. M. (2012). PUCK: an automated prompting system for smart environments: toward achieving automated prompting—challenges involved. *Personal and ubiquitous computing*, 16(7), 859-873.

Dixit, R. A., Hurst, S., Adams, K. T., Boxley, C., Lysen-Hendershot, K., Bennett, S. S., ... & Ratwani, R. M. (2020). Rapid development of visualization dashboards to enhance situation awareness of COVID-19 telehealth initiatives at a multihospital healthcare system. *Journal of the American Medical Informatics Association*, 27(9), 1456-1461.

Dobraja, I., & Kraak, M. J. (2020). Principles of dashboard adaptability to get insights into origin-destination data. *Journal of location based services*, 14(1), 28-48.

Elmqvist, N., & Yi, J. S. (2015). Patterns for visualization evaluation. *Information Visualization*, 14(3), 250-269.

Elshehaly, M., Randell, R., Brehmer, M., Mcvey, L., Alvarado, N., Gale, C. P., & Ruddle, R. A. (2021). QualDash: Adaptable Generation of Visualization Dashboards for Healthcare Quality Improvement. *IEEE Transactions on Visualization and Computer Graphics*, 27(2), 689-699. <https://doi.org/10.1109/TVCG.2020.3030424>

Evergreen, S., & Metzner, C. (2013). Design principles for data visualization in evaluation. *New directions for evaluation*, 2013(140), 5-20.

Few, S. (2006). *Information dashboard design: The effective visual communication of*

data (Vol. 2). Sebastopol, CA: O'reilly.

Fiaz, A. S., Asha, N., Sumathi, D., & Navaz, A. S. (2016). Data visualization: enhancing big data more adaptable and valuable. *International Journal of Applied Engineering Research*, 11(4), 2801-2804.

Gagnon-Roy, M., Pinard, S., Bottari, C., Le Morellec, F., Laliberté, C., Lagha, R. B., ... & Bier, N. (2022). Smart Assistive Technology for Cooking for People With Cognitive Impairments Following a Traumatic Brain Injury: User Experience Study. *JMIR Rehabilitation and Assistive Technologies*, 9(1), e28701.

Giroux, S., Bauchet, J., Pigot, H., Lussier-Desrochers, D., & Lachappelle, Y. (2008, July). Pervasive behavior tracking for cognitive assistance. In *Proceedings of the 1st international conference on Pervasive Technologies Related to Assistive Environments* (pp. 1-7).

Giroux, S., Leblanc, T., Bouzouane, A., Bouchard, B., Pigot, H. et Bauchet, J. (2009). The Praxis of Cognitive Assistance in Smart Homes. BMI Book, 183–211. <http://dx.doi.org/10.3233/978-1-60750-048-3-183>

Groussard, P. Y., Pigot, H., & Giroux, S. (2018). From conception to evaluation of mobile services for people with head injury: a participatory design perspective. *Neuropsychological rehabilitation*, 28(5), 667-688.

Helal, S., Lee, J. W., Hossain, S., Kim, E., Hagaras, H., & Cook, D. (2011, July). Persim-Simulator for human activities in pervasive spaces. In *2011 Seventh International Conference on Intelligent Environments* (pp. 192-199). IEEE.

Helal, S., Mann, W., El-Zabadani, H., King, J., Kaddoura, Y. et Jansen, E. (2005). The Gator tech smart house : A programmable pervasive space. *Computer*, 38(3), 50–60. <http://dx.doi.org/10.1109/MC.2005.107>

ISO 9241, Ergonomics Requirements for Office with Visual Display Terminals(VDTs), International Organization for Standardization, Geneva, 2001.

Jahromi, Z. F., Rajabzadeh, A., & Manashty, A. R. (2011). A multi-purpose scenario-based simulator for smart house environments. *arXiv preprint arXiv:1105.2902*.

Kelleher, C., & Wagener, T. (2011). Ten guidelines for effective data visualization in scientific publications. *Environmental Modeling & Software*, 26(6), 822-827.

Khan, A., Mukhtar, H., Ahmad, H. F., Gondal, M. A., & Ilyas, Q. M. (2017, March). Improving usability through enhanced visualization in healthcare. In *2017 IEEE 13th International Symposium on Autonomous Decentralized System (ISADS)* (pp. 39-44). IEEE.

Klepeis, N. E., Nelson, W. C., Ott, W. R., Robinson, J. P., Tsang, A. M., Switzer, P., ... & Engelmann, W. H. (2001). The National Human Activity Pattern Survey (NHAPS): a resource for assessing exposure to environmental pollutants. *Journal of Exposure Science & Environmental Epidemiology*, *11*(3), 231-252.

Kumar, V., & Reinartz, W. (2018). Software Tools and Dashboards. In *Customer Relationship Management* (pp. 171-178). Springer, Berlin, Heidelberg.

Lee, W., Cho, S., Chu, P., Vu, H., Helal, S., Song, W., ... & Cho, K. (2016). Automatic agent generation for IoT-based smart house simulator. *Neurocomputing*, *209*, 14-24.

Lewis, J. R. (2018). The system usability scale: past, present, and future. *International Journal of Human-Computer Interaction*, *34*(7), 577-590.

Li, Y., & Manoharan, S. (2013, August). A performance comparison of SQL and NoSQL databases. In *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)* (pp. 15-19). IEEE.

Maitre, J., Bouchard, K., Bertuglia, C., & Gaboury, S. (2021). Recognizing activities of daily living from UWB radars and deep learning. *Expert Systems with Applications*, *164*, 113994. <https://doi.org/10.1016/j.eswa.2020.113994>

Marikyan, D., Papagiannidis, S., & Alamanos, E. (2019). A systematic review of the smart home literature: A user perspective. *Technological Forecasting and Social Change*, *138*, 139-154.

Midway, S. R. (2020). Principles of effective data visualization. *Patterns*, *1*(9), 100141.

Naud, D., Généreux, M., Alauzet, A., Bruneau, J. F., Cohen, A., & Levasseur, M. (2021). Social participation and barriers to community activities among middle-aged and older Canadians: Differences and similarities according to gender and age. *Geriatrics & Gerontology International*, *21*(1), 77-84.

- Nielsen, J., & Loranger, H. (2006). *Prioritizing web usability*. Pearson Education.
- Pankaj, P., Hyde, M., & Rodger, J. (2006). Business Dashboards-Challenges and Recommendations. Dans *Proceedings of the Twelfth Americas Conference on Information Systems* (pp. 1423-1431). Acapulco, Mexico: AMCIS.
- Ngankam H., Lussier M., Aboujaoudé A., Pigot H., Gaboury S., Bouchard K., Couture M., Bier N., Giroux S. (2022): *NEARS-Hub, a Lightweight Edge Computing for Real-Time Monitoring in Smart Environments*, 8th EAI International Conference on Smart Objects and Technologies for Social Good (GOODTECHS'22), Springer
- Padhy, N., Satapathy, S., & Singh, R. P. (2018). State-of-the-art object-oriented metrics and its reusability: a decade review. *Smart Computing and Informatics*, 431-441.
- Pandey, A. V., Manivannan, A., Nov, O., Satterthwaite, M., & Bertini, E. (2014). The persuasive power of data visualization. *IEEE transactions on visualization and computer graphics*, 20(12), 2211-2220.
- Pigot, H., Lussier-Desrochers, D., Bauchet, J., Giroux, S., & Lachapelle, Y. (2008). A smart home to assist in recipe completion. In *Technology and Aging* (pp. 35-42). IOS Press.
- Potter, M. C., Wyble, B., Haggmann, C. E., & McCourt, E. S. (2014). Detecting meaning in RSVP at 13 ms per picture. *Attention, Perception, & Psychophysics*, 76(2), 270-279.
- Raghunath, N., Dahmen, J., Brown, K., Cook, D., & Schmitter-Edgecombe, M. (2019). Creating a digital memory notebook application for individuals with mild cognitive impairment to support everyday functioning. *Disability and Rehabilitation: Assistive Technology*.
- Rheault, S., & Poirier, J. (2012). *Le vieillissement démographique : de nombreux enjeux à déchiffrer*. Institut de la statistique du Québec.
- Roser, M. (2013). Future population growth. *Our world in data*.
- Sacha, G., Raphaël, B., (2021.), 2020.stateofjs, Saisie le 13 mars 2022 de <https://2020.stateofjs.com/en-US/technologies/front-end-frameworks/>
- Saket, B., Endert, A., & Stasko, J. (2016, October). Beyond usability and performance: A review of user experience-focused evaluations in visualization. In *Proceedings of the*

Sixth Workshop on Beyond Time and Errors on Novel Evaluation Methods for Visualization (pp. 133-142).

Scherer, Marcia J., et al. "Assistive technologies for cognitive disabilities." *Critical Reviews™ in Physical and Rehabilitation Medicine* 17.3 (2005).

Seelye, A. M., Schmitter-Edgecombe, M., Das, B., & Cook, D. J. (2012). Application of cognitive rehabilitation theory to the development of smart prompting technologies. *IEEE Reviews in biomedical engineering*, 5, 29-44.

Shah, S., Hernandez, E. A., & Helal, A. S. (2002, January). CAD-HOC: a CAD-Like tool for generating mobility benchmarks in ad-hoc networks. In *Proceedings 2002 Symposium on Applications and the Internet (SAINT 2002)* (pp. 270-279). IEEE.

Stachelek, G. C., McNutt, T., Thompson, C. B., Smith, K., DeWeese, T. L., & Song, D. Y. (2020). Improvements in Physician Clinical Workflow Measures After Implementation of a Dashboard Program. *Practical Radiation Oncology*, 10(3), 151-157.

Stack Overflow, (2021), 2021 Developer Survey, Saisie le 14 mars 2022 de <https://insights.stackoverflow.com/survey/2021>

Stadler, J. G., Donlon, K., Siewert, J. D., Franken, T., & Lewis, N. E. (2016). Improving the efficiency and ease of healthcare analysis through use of data visualization dashboards. *Big data*, 4(2), 129-135.

Strugar, D. (2019, October). Complex systems: on design and architecture of adaptable dashboards. In *International conference on objects, components, models and patterns* (pp. 176-186). Springer, Cham.

Thomas, B. L., & Crandall, A. S. (2011, March). A demonstration of PyViz, a flexible smart home visualization tool. In *2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)* (pp. 304-306). IEEE.

United Nations Department of Economic and Social Affairs (2019). World population prospect 2019 : Highlights. Récupéré le 2022-02-03 de https://population.un.org/wpp/Publications/Files/WPP2019_Highlights.pdf

Vijayan, D. S., Rose, A. L., Arvindan, S., Revathy, J., & Amuthadevi, C. (2020). Automation systems in smart buildings: a review. *Journal of Ambient Intelligence and Humanized Computing*, 1-13.

Wohlgethan, E. (2018). *Supporting Web Development Decisions by Comparing Three Major JavaScript Frameworks: Angular, React and Vue.js* (Doctoral dissertation, Hochschule für Angewandte Wissenschaften Hamburg).

Zhu, Y. (2007, November). Measuring effective data visualization. In *International symposium on visual computing* (pp. 652-661). Springer, Berlin, Heidelberg.

ANNEXE A

CERTIFICAT D'APPROBATION ÉTHIQUE

Ce mémoire a fait l'objet d'une certification éthique. Le numéro du certificat 2022-967.