



Comment l'informatique peut aider à la navigation intra-muros ?

BOLLE Emilien



Docteur PHILIPPE Laurent

Master 1 en Architecture des Systèmes Informatiques

Année académique 2023-2024

Table des matières

1.	Introduction	5
2.	Etat de l’art.....	5
2.1.	Contraintes.....	5
2.2.	Analyse de l’existant.....	5
2.3.	Objectif de recherche.....	9
2.4.	Possibilités technologiques	9
3.	Analyse de l’hypothèse retenue.....	10
3.1.	Scénario de navigation	10
3.2.	Fonctionnement.....	10
3.2.1.	Points de passage	10
3.2.2.	Liaisons	11
3.2.3.	Chemins.....	12
3.3.	Technologies	12
3.3.1.	Matérialisation des points de passage.....	12
3.3.2.	Navigation	14
3.3.3.	Calcul du chemin	14
3.3.4.	Informations.....	16
3.3.5.	Stockage des points de passage	17
3.4.	Utilisation	18
3.4.1.	Mise en place	19
3.4.2.	Navigation	20
3.5.	Résilience aux erreurs	21
3.5.1.	Correction des distances	22
3.5.2.	Correction des orientations	22
3.6.	Conclusion et perspectives.....	22
4.	Conception d’un prototype.....	23
4.1.	Cahier des charges	23
4.1.1.	Objectifs	23
4.1.2.	Limitations.....	23
4.1.3.	Fonctionnalités.....	23
4.1.4.	Contraintes.....	24

4.2.	Analyse fonctionnelle	24
4.2.1.	Cas d'utilisation	24
4.2.2.	Fonctionnement	24
4.2.3.	Maquettes d'écrans	27
4.3.	Analyse technique	38
4.3.1.	Choix technologiques	38
4.3.2.	Structure de l'application	40
5.	Validation	42
	Table des figures	43
	Table des tableaux	43
	Références.....	44
	Annexes	45

1. Introduction

Depuis quelques années, les systèmes de géolocalisation et de navigation font partie intégrante de la vie quotidienne. Par ces systèmes, nous pensons tous aux systèmes GPS qui nous permettent de se diriger au travers de territoires inconnus. Ainsi, il est possible de se rendre partout dans le monde aisément. Cependant, cela se limite au monde extérieur, les systèmes GPS se trouvent incompétents dans des environnements fermés et complexe. Par exemple, ils ne permettent pas de diriger la personne dans un hôpital ou un bâtiment à étages. Ce projet de recherche tentera de répondre à cette problématique en explorant diverses hypothèses que l'informatique peut apporter à la navigation intra-muros.

L'objectif de ce projet de recherche est de trouver des solutions pour réaliser une application de navigation en intérieur. Cette application doit permettre de diriger son utilisateur vers sa destination en lui indiquant le chemin au travers du bâtiment.

2. Etat de l'art

2.1. Contraintes

Comme évoqué dans l'introduction, les systèmes de navigation font penser aux systèmes GPS présents dans les véhicules et beaucoup d'applications. Cependant, ils sont destinés à de la navigation sur de grandes zones. Ils sont parfaits pour diriger l'utilisateur d'un point A à un point B avec plusieurs kilomètres de distance. Cependant, ils ne répondent pas aux contraintes présentes dans les endroits fermés tels que des bâtiments.

Tout d'abord, un bâtiment est généralement assez petit au niveau de sa superficie, le système de navigation doit donc avoir une certaine précision pour ne pas confondre les couloirs et les salles sur les quelques mètres qui les séparent. Ensuite, le bâtiment est un endroit fermé. Il se peut que la communication avec des satellites ou autre moyen de communication extérieur soit difficile. Par exemple, des murs épais peuvent ne pas laisser passer facilement les ondes électromagnétiques. Enfin, une contrainte à ne pas sous-estimer est que l'utilisation traditionnelle des systèmes comme le GPS ne permet pas de différencier un étage d'un autre dans un bâtiment. Les applications se limitent à positionner l'appareil sur un plan en deux dimensions comme une carte par exemple.

En résumé, le système de navigation en intérieur doit pouvoir répondre à des contraintes qui ne sont pas présentes en extérieur comme la précision à quelques mètres près, la possibilité de perte de signal satellite ou encore la gestion des étages du bâtiment.

2.2. Analyse de l'existant

Avant de réfléchir à diverses hypothèses pour le système de navigation, il serait intéressant d'explorer ce qui est déjà utilisé dans des bâtiments ou des endroits fermés plus ou moins

complexes tels que des hôpitaux, des universités ou encore des centres commerciaux¹. Cette exploration permettra ensuite d'émettre des premières hypothèses.

a) Fléchage

Dans de nombreux bâtiments, on peut trouver un système de fléchage pour diriger les personnes. Ce système, illustré dans la figure suivante, consiste à disposer une flèche pour chaque destination aux changements de direction. Ainsi, si une personne désire se rendre à un service particulier, elle doit suivre les flèches dirigeant vers ce service. Elle va rencontrer une flèche à chaque intersection de couloirs dans le bâtiment. Ce système est comparable aux flèches directionnelles routières qui permettent de diriger les véhicules vers une ville ou un lieu de destination.



Figure 1 - Photos du fléchage dans le bâtiment IESN de l'Hénallux

b) Routes

Dans les hôpitaux Saint-Luc [1], les patients sont dirigés par le biais de routes. Une route peut représenter un couloir ou un service. Ainsi, pour se rendre dans un service particulier le patient recherche, à son arrivée dans le bâtiment, le service sur un tableau. Il y trouve la route correspondante. Ensuite, il peut suivre les panneaux de direction, représentés dans la figure suivante, le menant vers la bonne route.

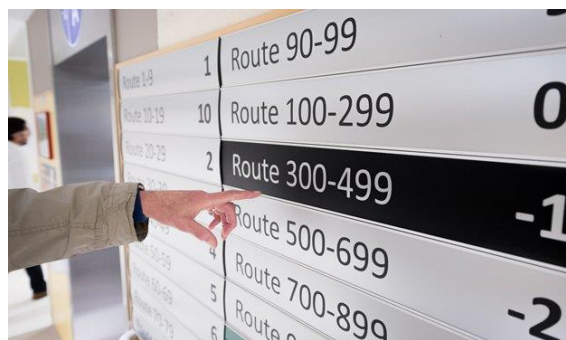


Figure 2 - Panneaux de direction [1]

Ce système de direction est comparable à la recherche d'une porte d'embarcation dans un aéroport. Dans ce cas, le voyageur consulte le tableau des départs pour trouver la porte

¹ Toutes les citations de marques ou d'enseignes n'ont aucun but commercial ou de mise en valeur.

attribuée à son vol. Il s'y rend ensuite en suivant les indications données par les divers panneaux.

Ce système reprend le concept de fléchage mis à part que les flèches ne correspondent pas à un service particulier. La personne doit consulter un tableau de correspondances avant de suivre les flèches. Cela permet de réduire le nombre de panneaux directionnels dans le bâtiment.

c) Plan

Le plan est peut-être le moyen le plus répandu pour diriger les personnes dans un endroit fermé. Il peut se trouver sous plusieurs formes. Par exemple, le plan peut être au format papier, les usagers peuvent en prendre un durant leur visite. Il peut être aussi affiché sur un support mural ou sur dispositif d'affichage, les visiteurs peuvent ainsi le consulter à différents emplacements du lieu.

Un plan peut être lu pour connaître sa position au sein du lieu ou encore pour se diriger vers une destination. Cependant, dans le deuxième cas, l'utilisateur doit faire preuve d'un certain sens de l'orientation pour pouvoir transcrire sa lecture du plan dans le bâtiment ou le lieu réel.

d) Application "plan"

Dans certains endroits, souvent de plus grande ampleur, les usagers peuvent se diriger à l'aide d'une application. Par exemple, ce système est utilisé dans des salons d'exposition ou dans des endroits clos comme des parcs d'attractions ou animaliers. Un exemple est donné dans la figure n°3 ci-après. Les applications proposent un plan du site avec des points d'intérêts disposés dessus correspondants aux attractions, aux stands ou aux sanitaires, par exemple. L'application peut, si cela est possible, aussi indiquer la position de l'utilisateur sur le plan. Ainsi, il sait où aller s'il veut se rendre à un point d'intérêt ou un endroit du site en lisant le plan de l'application et en suivant sa position sur ce plan.



Figure 3 - Capture d'écran de l'application "plan" de Walibi Belgium

Ces applications sont un compromis entre l'utilisation d'un plan physique et une application de navigation. En effet, elles ne permettent pas de diriger la personne en lui indiquant le chemin à suivre.

e) Application de navigation

Enfin, il est intéressant de rechercher des solutions déjà existantes qui correspondent à l'objet de cette recherche. Dès les premiers résultats, il s'avère que quelques solutions existent déjà. Par exemple, la personne sous le pseudonyme « VAISHAK » répertorie dans une publication [2] sur le blog <https://medium.com/>, selon son propre avis, les 10 meilleures applications de navigation en intérieur : Becomap, Cisco systems, Here technologies...

La plupart des applications de navigation existantes base la localisation de l'utilisateur sur un réseau de bornes disposées dans le bâtiment. Ces bornes sont généralement des points d'accès wifi ou des balise BLE (de l'anglais « Bluetooth Low Energy »). Ces bornes sont généralement placées sur les murs à 2,5 mètres du sol et elles sont espacées de maximum 8 mètres. Cette navigation utilise deux techniques de localisation : si l'appareil de l'utilisateur détecte deux balises, il se localise en utilisant l'indication des signaux reçus (en anglais « Received Signal Strength Indication », abrégé en « RSSI »). S'il détecte trois balises ou plus, il utilise la multilatération [3].

La localisation avec les RSSI est une technique qui consiste à mesurer la puissance du signal reçu par l'appareil de l'utilisateur. Elle se base sur le principe que les RSSI reçus des balises sont différents à chaque position. La mise en place de cette technique se fait en deux étapes. Tout d'abord, il faut calibrer le système en mesurant les RSSI à différentes positions du bâtiment. La collecte des RSSI se fait en se déplaçant dans le bâtiment. Ces RSSI collectés sont traités et stockés dans une base de données. Ensuite, lors de l'utilisation de l'application, l'appareil de l'utilisateur mesure les RSSI des balises qu'il détecte. Il compare ces RSSI avec ceux stockés dans la base de données pour estimer sa position [4].

La localisation par multilatération se base aussi sur la mesure de la puissance du signal reçu par l'appareil de l'utilisateur. Cependant, cette technique utilise au minimum trois balises pour estimer la position de l'utilisateur. La position est calculée en utilisant la distance entre l'utilisateur et les balises. Cette distance est déduite de la puissance du signal reçu en se basant sur le principe que la puissance du signal diminue exponentiellement en fonction de l'éloignement de la balise émettrice. La position de l'utilisateur est ensuite estimée en utilisant une formule mathématique qui prend en compte les distances entre l'utilisateur et les balises [5].

Ces deux techniques de localisation nécessitent d'avoir une couverture complète du bâtiment avec les rayons d'émission des balises. Cela implique de mettre un nombre assez conséquent de balises pour que le système de navigation fonctionne correctement.

2.3. Objectif de recherche

L'analyse des solutions existantes a pu montrer qu'il existe déjà des applications de navigation. Cependant, celles-ci se basent, pour la plupart, sur une architecture physique pour localiser l'utilisateur dans le bâtiment. Cette approche offre une grande précision de localisation et une expérience utilisateur fluide. Cependant, une architecture physique implique un investissement financier pour le matériel, l'installation et la maintenance.

Certains clients pourraient être réticents à adopter une telle solution en raison de contraintes financières ou de préoccupations liées à l'installation de matériel dans leur bâtiment. Pour ces clients, il serait intéressant de concevoir une solution sans infrastructure ou, si cela est compliqué à mettre en œuvre, au moins avec une infrastructure minimale.

C'est pourquoi l'objectif de ce projet de recherche est de trouver une solution de navigation en intérieur sans infrastructure physique. Cette solution doit être capable de localiser l'utilisateur dans le bâtiment et de lui indiquer le chemin à suivre pour se rendre à sa destination.

2.4. Possibilités technologiques

a) GPS

La technologie utilisée pour bon nombre de systèmes de navigation en extérieur est le GPS (de l'anglais « Global Positioning System »). Par exemple, il est utilisé dans les voitures et dans les téléphones. Ce système est apparu dans les années 1970 et est composé d'une constellation de satellites en orbite autour de la Terre. Ces satellites émettent des signaux que les récepteurs GPS captent pour déterminer leur position au moyen de la multilatération. La position déterminée a une précision inférieure à 20 mètres. Pour améliorer cela, il est possible de combiner le GPS avec le système européen EGNOS (de l'anglais « European Geostationary Navigation Overlay Service ») qui corrige les erreurs du GPS pour obtenir une précision de l'ordre du mètre et demi. Cependant, ces systèmes ne sont pas adaptés à une utilisation en intérieur. En effet, les signaux émis par les satellites peuvent être bloqués par l'infrastructure du bâtiment (murs épais, structure métallique, etc.). De plus, les récepteurs GPS ne sont pas tous compatibles avec la réception des signaux EGNOS. Ils ne peuvent donc pas bénéficier d'une précision suffisante. C'est pourquoi il est peu probable que le GPS soit une solution viable pour la navigation en intérieur [6].

b) Wifi

L'hypothèse du wifi peut être évoquée pour l'objectif de recherche. En effet, il se peut que le bâtiment soit déjà équipé d'une infrastructure de points d'accès wifi. Ces points d'accès peuvent être utilisés pour localiser l'utilisateur dans le bâtiment comme expliqué dans le point 2.2.e. L'infrastructure wifi étant déjà en place, il n'y a pas besoin d'investir dans du matériel supplémentaire si elle est suffisamment développée pour l'utilisation du système de navigation.

c) Points de passage

Une autre hypothèse est de baser la navigation sur un système de points de passage. Un point de passage est un point dans le bâtiment par lequel l'utilisateur doit passer pour se rendre à sa destination. Cela peut être une entrée/sortie, une intersection de couloirs, un escalier, etc.

L'utilisateur doit passer par une suite de points de passage pour atteindre sa destination. Pour cela, il doit être informé du prochain point de passage à atteindre. Cela permet de guider l'utilisateur dans le bâtiment sans avoir besoin de le localiser précisément.

C'est sur cette dernière hypothèse que la suite de ce projet de recherche va se concentrer. La suite de ce document va explorer les différentes manières de mettre en place un système de navigation basé sur des points de passage.

3. Analyse de l'hypothèse retenue

Cette partie va tenter d'analyser l'hypothèse retenue pour le projet de recherche. Comme expliqué dans le point précédent (2.4.c), il va se concentrer sur un système de navigation basé sur des points de passage. Dans cette analyse, le fonctionnement de ce système sera expliqué. L'analyse permettra de construire une méthodologie pour le mettre en place et de décrire au mieux son utilisation pour la navigation.

3.1. Scénario de navigation

Pour commencer, il est intéressant de décrire un scénario de navigation pour le système. Imaginons un bâtiment avec plusieurs étages. L'utilisateur se trouve à une entrée du bâtiment. Il souhaite se rendre à un service situé au premier étage. Il ouvre l'application de navigation sur son smartphone et entre sa destination. L'application lui indique le premier point de passage à atteindre pour se rendre à sa destination. L'utilisateur suit les indications de l'application pour se rendre à ce point de passage. Une fois arrivé, l'application lui indique le prochain point de passage à atteindre. L'utilisateur suit les indications de l'application pour se rendre à ce point de passage. Il répète cette opération jusqu'à atteindre sa destination.

3.2. Fonctionnement

3.2.1. Points de passage

Un point de passage est un point dans le bâtiment par lequel l'utilisateur doit passer pour se rendre à sa destination. Chaque point de passage correspond à un élément du bâtiment qui est facilement identifiable par l'utilisateur. Cela peut être une porte (entrée/sortie, une salle, un service, etc.), une intersection de couloirs, un escalier, un ascenseur, etc. Ainsi, le bâtiment est traduit en un ensemble de points de passage dans l'application de navigation.

Pour illustrer le concept, voici un schéma simplifié représentant un étage d'un bâtiment avec des points de passage :

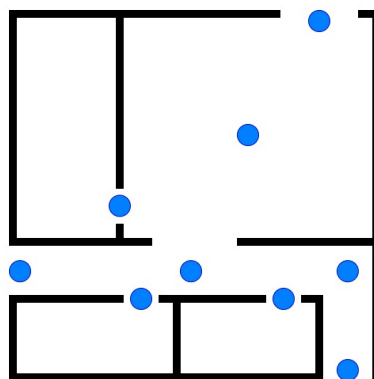


Figure 4 - Schéma des points de passage

Dans la figure ci-dessus, les points de passage sont illustrés par les points bleus. Ils sont disposés à chaque entrée/sortie, à chaque porte de salle et aux intersections de couloirs.

3.2.2. Liaisons

L'objectif étant de faire un chemin composé de points de passage, il est nécessaire de déterminer comment les relier entre eux. Cela permettra de transformer l'ensemble des points de passage en un réseau de points reliés entre eux.

Pour cela, il faut définir des liaisons entre les points de passage. Une liaison est une connexion entre deux points qui permet de diriger l'utilisateur d'un point à un autre. Cependant, il ne serait pas pertinent de relier tous les points de passage entre eux. En effet, un point de passage ne doit pas être relié à un autre situé à l'opposé du bâtiment. Il est donc nécessaire de déterminer quelles liaisons sont pertinentes pour guider l'utilisateur de manière efficace. Cela peut être fait en définissant des règles pour établir les liaisons entre les points de passage.

Tout d'abord, un point de passage peut être relié à un ou plusieurs autres points de passage. Il ne doit pas être isolé, c'est-à-dire qu'il doit être relié à au moins un autre point de passage. Il ne peut également pas être relié à lui-même. Ensuite, deux points de passage reliés doivent être voisins dans le bâtiment. Cela signifie qu'ils doivent être proches et accessibles l'un de l'autre. Par exemple, dans le cas d'un couloir, le point de passage doit être visible depuis le point de passage voisin. Un autre exemple, dans un escalier, le point de passage doit être à l'étage suivant ou précédent, mais pas à un étage plus éloigné. Enfin, les liaisons doivent être uniques. Cela signifie qu'il ne peut pas y avoir deux liaisons entre les mêmes points de passage. Ces règles permettent de définir un ensemble de liaisons pertinentes pour guider l'utilisateur de manière simple et efficace.

En reprenant le schéma précédent, voici un exemple de liaisons entre les points de passage, représentées par des lignes :

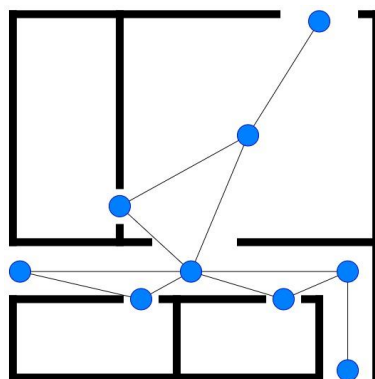


Figure 5 - Schéma des liaisons entre les points de passage

Ces liaisons créent un réseau de points de passage reliés entre eux. Cela permet dorénavant de déterminer un chemin entre deux points de passage en suivant les liaisons.

3.2.3. Chemins

Maintenant qu'un réseau de points de passage a été défini, il est possible de déterminer un chemin entre deux endroits du bâtiment. Un chemin est une suite de points de passage qui permet de guider l'utilisateur de sa position actuelle vers une destination.

Par exemple, dans l'illustration ci-après, si l'utilisateur se trouve au point de passage A et souhaite se rendre au point de passage B, l'application le mènera à travers les points de passage en suivant les liaisons. Le chemin à suivre est illustré en vert.

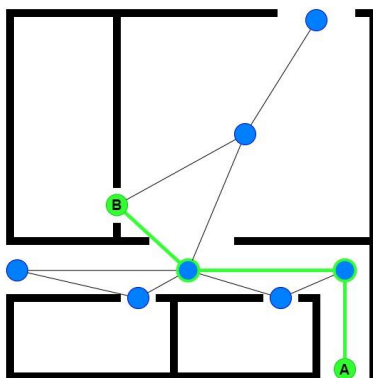


Figure 6 - Schéma du tracé entre un point A et un point B

3.3. Technologies

Après avoir défini le fonctionnement du système de navigation basé sur des points de passage, il est intéressant d'explorer les techniques et technologies qui peuvent être utilisées pour sa mise en place.

3.3.1. Matérialisation des points de passage

Pour que le système de navigation fonctionne, il est nécessaire de matérialiser les points de passage dans le bâtiment. Cela peut être fait de différentes manières.

a) Balises BLE

Les balises BLE (de l'anglais « Bluetooth Low Energy ») sont des petits appareils qui émettent un signal Bluetooth basse consommation. Les smartphones peuvent détecter les signaux émis par ces balises et recevoir des informations associées. Les balises BLE peuvent être utilisées pour matérialiser les points de passage dans le bâtiment. Chaque point de passage est représenté par une balise BLE. Lorsque l'utilisateur se trouve à proximité d'une balise, l'application de navigation détecte le signal émis et identifie le point de passage correspondant.

Ces balises impliquent un coût d'achat et d'installation. Cependant, le système par points de passage nécessite moins de balises que les systèmes de navigation existants. En effet, il n'est pas nécessaire de couvrir l'ensemble du bâtiment avec le signal des balises. Ce coût est généralement déterminé par la portée de la balise, la qualité de la batterie et d'autres caractéristiques. Dans le cas du système de navigation, la portée de la balise doit être adaptée à la largeur d'un couloir. Une portée de 3 mètres serait suffisante. La qualité de la batterie doit permettre une autonomie d'au moins un an pour éviter des remplacements fréquents. Enfin, il n'y a pas besoin que les balises soient résistantes à l'eau car elles seront installées à l'intérieur du bâtiment.

b) QR codes

Les QR codes sont des codes-barres en deux dimensions qui peuvent être lus par un smartphone. Chaque point de passage est représenté par un QR code. L'utilisateur scanne le QR code avec l'application de navigation pour identifier le point de passage correspondant.

Pour diriger le choix de la technologie à utiliser, il est nécessaire de prendre en compte plusieurs critères. Voici une matrice de décision reprenant les critères à considérer. Chaque critère est pondéré selon une échelle de 1 à 3, 1 étant le moins important et 3 le plus important. Les critères sont également notés selon une échelle de 1 à 5, 1 étant le moins adapté et 5 le plus adapté. La somme des produits des pondérations et des évaluations donne un score pour chaque technologie. La technologie avec le score le plus élevé est la plus adaptée pour le système de navigation.

Les critères à considérer sont les suivants :

- Facilité d'installation : la facilité d'installation des points de passage dans le bâtiment.
- Faible coût : le coût d'achat et d'installation des points de passage. Un coût faible est préférable pour correspondre à l'objectif de recherche.
- Fluidité d'utilisation : la fluidité de l'expérience utilisateur pendant la navigation.

Tableau 1 - Matrice de décision - Matérialisation des points de passage

Critère	Pondération	Balises BLE	QR codes
Facilité d'installation	2	4	4
Faible coût	3	3 ($\approx 20\text{€} / \text{u}$)	5 ($\approx 0,01\text{€} / \text{u}$)
Fluidité d'utilisation	2	4	2
	Total	21	27

D'après la matrice de décision, les QR codes sont la technologie la plus adaptée pour le système de navigation. Ils obtiennent un score total de 27, contre 21 pour les balises BLE. Leur principal avantage est leur coût presque nul, cela entre parfaitement dans l'objectif de recherche de trouver une solution sans coût d'infrastructure physique.

3.3.2. Navigation

Lors de la navigation d'un utilisateur, l'application doit être capable de lui indiquer où se trouve le prochain point de passage à atteindre. Pour cela, il faut déterminer comment situer un point de passage par rapport à un autre.

a) Orientation

L'utilisateur doit savoir dans quelle direction se trouve le prochain point de passage pour s'y rendre. Pour cela, il est nécessaire de déterminer l'orientation des points de passage par rapport aux autres. Cela peut être fait en définissant des directions pour chaque liaison entre les points de passage. Par exemple, une liaison entre les points A et B a une certaine direction pour aller de A vers B et une autre pour aller de B vers A. Ainsi, l'application peut indiquer à l'utilisateur la direction à suivre au moyen d'une flèche, par exemple.

b) Distance

La distance entre les points de passage n'est pas indispensable pour le système. En effet, les points de passage doivent être proches et visibles les uns des autres pour respecter les règles de liaison. L'utilisateur peut donc se rendre au prochain point de passage en suivant les indications de direction sans avoir besoin de connaître la distance à parcourir.

Cependant, il peut être intéressant d'indiquer une distance approximative pour donner une idée de la distance à parcourir ou encore pour choisir le chemin le plus court (voir le point suivant 3.3.3.). Cette distance peut être calculée en comptant le nombre de pas à faire entre les points de passage.

c) Gestion des étages

Pour gérer les différents étages du bâtiment, l'utilisateur doit être informé qu'il doit monter ou descendre pour poursuivre son chemin. Pour savoir cela, l'application pourrait retenir "+1" ou "-1" dans l'orientation des points de passage positionnés aux escaliers et aux ascenseurs. Ainsi, si l'utilisateur se trouve au point de passage A et que la liaison avec le point de passage B est orientée "+1", l'application sait qu'il doit monter pour se rendre au point de passage B.

3.3.3. Calcul du chemin

Pour déterminer le chemin à suivre entre deux endroits du bâtiment, il est nécessaire de calculer une suite de points de passage qui mènent de l'un à l'autre. Cela peut être fait en utilisant un algorithme de recherche de chemin. Un algorithme de recherche de chemin est un algorithme qui détermine le chemin le plus court entre deux points dans un graphe. Dans le cas du système de navigation, le graphe est le réseau de points de passage reliés entre eux par des liaisons.

a) Algorithme en largeur (BFS)

L'algorithme en largeur ("BFS" de l'anglais "Breadth-First Search") est un algorithme de recherche de chemin qui explore un graphe de type arbre en largeur. Il commence par le point de départ et explore tous les points de passage voisins. Ensuite, il explore les points de passage voisins des points de passage déjà explorés et ainsi de suite jusqu'à atteindre le point de destination. Cet algorithme permet de déterminer le chemin le plus court entre deux points dans un graphe non pondéré. Cet algorithme a une complexité en temps de $O(N + A)$ où N est le nombre de nœuds dans le graphe et A est le nombre d'arêtes [7].

b) Algorithme en profondeur (DFS)

L'algorithme en profondeur ("DFS" de l'anglais "Depth-First Search") est un algorithme de recherche de chemin qui explore un graphe de type arbre en profondeur. Il commence par le point de départ et explore un chemin jusqu'à atteindre le point de destination. S'il n'y a pas de chemin, il revient en arrière et explore un autre chemin. Cet algorithme permet de déterminer le chemin le plus court entre deux points dans un graphe non pondéré. Cet algorithme a une complexité en temps de $O(N + A)$ où N est le nombre de nœuds dans le graphe et A est le nombre d'arêtes [8].

c) Algorithme de Dijkstra

L'algorithme de Dijkstra a été publié en 1959 par Edsger W. Dijkstra [9]. C'est un algorithme qui permet de déterminer le chemin le plus court entre un nœud de départ et tous les autres nœuds d'un graphe pondéré. Un graphe pondéré est un graphe dans lequel chaque arête a un poids associé. Dans le cas du système de navigation, le poids des arêtes pourrait être la distance entre les deux nœuds. Cet algorithme se base sur une approche itérative pour déterminer le chemin le plus court. Il commence par initialiser tous les nœuds, excepté le nœud de départ, avec une distance infinie et un statut non visité. Ensuite, il met à jour les distances des nœuds liés au nœud de départ avec le poids des arêtes les reliant et il met le nœud de départ comme visité. Il choisit ensuite le nœud avec le poids le plus faible parmi les nœuds mis à jour et répète la même opération pour ce nœud. Il continue ainsi jusqu'à ce que tous les nœuds soient visités. L'algorithme de Dijkstra a ainsi une complexité en temps de $O(N^2)$ où N est le nombre de nœuds dans le graphe. Toutefois, il existe des variantes de cet algorithme qui permettent d'améliorer cette complexité. Une de ces variantes est d'utiliser un tas de Fibonacci dans la gestion des nœuds. Cela permet de réduire la complexité en temps à $O(A + N \log N)$ où A est le nombre d'arêtes dans le graphe [10]. Cet algorithme est utilisé dans de nombreux domaines, notamment dans le protocole de routage OSPF (de l'anglais « Open Shortest Path First »).

d) Algorithme de Bellman-Ford

L'algorithme de Bellman-Ford est un algorithme de recherche de chemin similaire à l'algorithme de Dijkstra. Cependant, il a l'avantage qu'il est capable de gérer les graphes avec des arêtes de poids négatif. Cet algorithme fonctionne de la même manière que l'algorithme de Dijkstra. Il a une complexité en temps de $O(N * A)$ où N est le nombre de nœuds dans le

graphe et A est le nombre d'arêtes. Il est utilisé, par exemple, dans le protocole de routage RIP (de l'anglais « Routing Information Protocol ») [11].

Pour déterminer l'algorithme le plus adapté au système de navigation, il est nécessaire de prendre en compte plusieurs critères. Voici une matrice de décision reprenant les critères à considérer. Comme auparavant, les critères sont pondérés de 1 à 3 selon leur importance et ils sont évalués de 1 à 5.

Les critères à considérer sont les suivants :

- Complexité en temps : la complexité en temps de l'algorithme. La valeur sera le classement de l'algorithme par rapport aux autres (5 étant le meilleur classement).
- Pondération : la capacité de l'algorithme à gérer les graphes pondérés.
- Pondération négative : la capacité de l'algorithme à gérer les arêtes de poids négatif.

Tableau 2 - Matrice de décision - Algorithme de recherche de chemin dans un graphe

Critères	Pondération	BFS	DFS	Dijkstra	Bellman-Ford
Complexité en temps	3	3	3	4	2
Pondération	2	0 (Non)	0 (Non)	1 (Oui)	1 (Oui)
Pondération négative	1	0 (Non)	0 (Non)	0 (Non)	1 (Oui)
	Total	9	9	14	9

D'après la matrice de décision, l'algorithme de Dijkstra est le plus adapté pour le système de navigation. Il obtient un score total de 14, contre 9 pour les autres algorithmes. Son principal avantage est sa complexité en temps optimale.

3.3.4. Informations

3.3.4.1. Points de passage

Pour que l'application de navigation puisse guider l'utilisateur, il est nécessaire de stocker des informations à propos des points de passage. Voici une liste des informations à stocker pour chaque point de passage :

- Identifiant : un identifiant unique pour le point de passage.
- Nom : un nom pour le point de passage.
- Type : le type du point de passage (entrée/sortie, intersection, escalier, ascenseur, etc.).

L'identifiant permet d'identifier de manière unique le point de passage dans le bâtiment. Le nom permet de donner une indication à l'utilisateur sur le lieu où il se trouve. Le type permet de déterminer le comportement de l'application en fonction du point de passage.

3.3.4.2. Liaisons

Il est également nécessaire de stocker des informations à propos des liaisons entre les points de passage. Voici une liste des informations à stocker pour chaque liaison :

- L'orientation du point de passage A par rapport au point de passage B.
- L'orientation du point de passage B par rapport au point de passage A.
- La distance entre les points de passage.

Il est nécessaire de stocker l'orientation de chaque point de passage par rapport à l'autre pour indiquer à l'utilisateur la direction à suivre. En effet, elle est différente selon si l'utilisateur se trouve au point A ou au point B. La distance entre les points de passage peut être utilisée pour donner une indication à l'utilisateur sur la distance à parcourir ou pour choisir le chemin le plus court (voir le point précédent (3.3.3.)).

Comme expliqué dans le point 3.3.2.c, pour la gestion des changements d'étages, l'orientation est utilisée afin de savoir si l'utilisateur doit monter ou descendre. S'il doit passer d'un point de passage de type escalier ou ascenseur à un autre de même type, l'application doit regarder l'orientation correspondante entre ces deux points. Si l'orientation est "+1", l'utilisateur doit monter. Si c'est "-1", il doit descendre.

3.3.4.3. QR codes

Si la technologie des QR codes est utilisée pour matérialiser les points de passage, il est nécessaire de stocker l'association entre les QR codes et les points de passage. Pour cela, il faut stocker l'identifiant du point de passage dans le QR code. Lorsque l'utilisateur scanne le QR code, l'application peut ainsi identifier le point de passage correspondant en récupérant l'identifiant.

3.3.5. Stockage des points de passage

Pour stocker les informations des points de passage et des liaisons, il est nécessaire d'utiliser une base de données. Une base de données est un système qui permet de stocker et de gérer des données de manière structurée. Il existe plusieurs types de bases de données, notamment les bases de données relationnelles et les bases de données NoSQL. Les bases de données relationnelles sont composées de tables qui sont reliées entre elles par des clés étrangères. Elles sont adaptées pour stocker des données structurées comme des objets avec des attributs. Les bases de données NoSQL sont plus flexibles et permettent de stocker des données structurées différemment, par exemple sous forme de documents ou de graphes. Le choix de la base de données dépend des besoins du système de navigation.

Dans le cas du système de navigation basé sur des points de passage, il est nécessaire, comme expliqué dans le point précédent, de non seulement retenir des informations sur des objets (les points de passage) mais aussi sur des relations entre ces objets (les liaisons). Les deux types de bases de données peuvent répondre à ce besoin.

3.3.5.1. Base de données relationnelle

Dans une base de données relationnelle, les points de passage et les liaisons peuvent être stockés dans des tables séparées. Voici un schéma de base de données relationnelle pour stocker les points de passage et les liaisons :

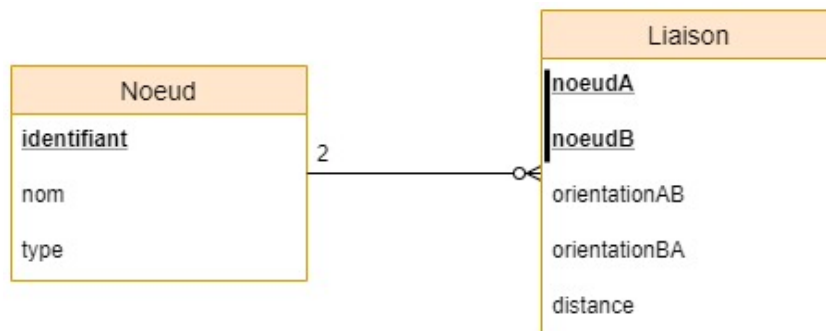


Figure 7 - Schéma de base de données relationnelle

3.3.5.2. Base de données NoSQL orientée graphe

Dans une base de données NoSQL orientée graphe, les points de passage et les liaisons peuvent être stockés respectivement sous forme de nœuds et de relations. Voici un exemple de schéma de base de données NoSQL de type graphe pour stocker les points de passage et les liaisons :

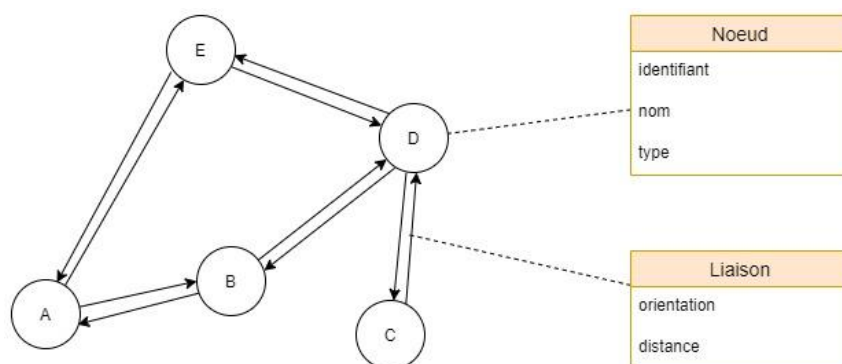


Figure 8 - Schéma de base de données NoSQL orientée graphe

Dans la figure 8, il est à remarquer qu'il y a deux relations entre chaque nœud. Cela est dû au fait qu'une base de données NoSQL orientée graphe est orientée. Cela veut dire que les relations ont un sens. Dans le cas du système de navigation, les liaisons sont bidirectionnelles. Une liaison doit donc être traduite par deux relations de sens opposés dans la base de données. De plus, chaque relation a l'orientation correspondant à son sens.

Une base de données NoSQL orientée graphe paraît plus adaptée pour le système de navigation. En effet, les points de passage et les liaisons peuvent être stockés de manière plus naturelle sous forme de nœuds et de relations. Cela permet de représenter plus facilement le réseau de points de passage reliés entre eux.

3.4. Utilisation

Cette partie va décrire comment les utilisateurs peuvent utiliser l'application de navigation en intérieur. Elle va détailler les différentes étapes de l'utilisation de l'application, de l'installation des points de passage à la navigation de l'utilisateur.

3.4.1. Mise en place

La première étape avant d'utiliser l'application de navigation est l'installation des points de passage dans le bâtiment. Comme déterminé précédemment, les points de passage sont matérialisés par des QR codes ou des balises BLE. La suite de cette explication se basera sur l'utilisation des QR codes pour les points de passage.

Lors de la mise en place du système, l'administrateur du bâtiment doit ajouter les points de passage un à un dans l'application de navigation. Pour cela, il peut générer QR codes à l'avance. Pour cela, l'application génère les identifiants et construit les QR codes avec ceux-ci. Ensuite, il doit placer ces QR codes dans le bâtiment aux endroits correspondants aux points de passage. Une fois le QR code placé, l'administrateur peut l'associer à un point de passage dans l'application en scannant le QR code avec son smartphone. Il peut alors renseigner le nom et le type du point de passage. Il doit également préciser les liaisons entre ce point de passage et les autres points de passage déjà enregistrés. Pour cela, il doit donner la direction vers le point voisin en orientant son appareil mobile vers ce point de passage et enregistrer l'orientation. Il doit ensuite se déplacer vers le point de passage voisin et scanner le QR code pour enregistrer la liaison. Pendant le déplacement, l'application compte le nombre de pas effectués pour estimer la distance entre les points de passage. Après avoir scanner le QR code voisin, l'administrateur enregistre l'orientation vers le nouveau point de passage afin d'avoir les deux orientations pour la liaison. Il répète cette opération pour chaque point de passage voisin en revenant au nouveau point de passage à chaque fois. Une fois cela terminé, il peut passer au point de passage suivant.

Pour mieux comprendre le processus d'installation des points de passage, voici un schéma illustrant les différentes étapes :

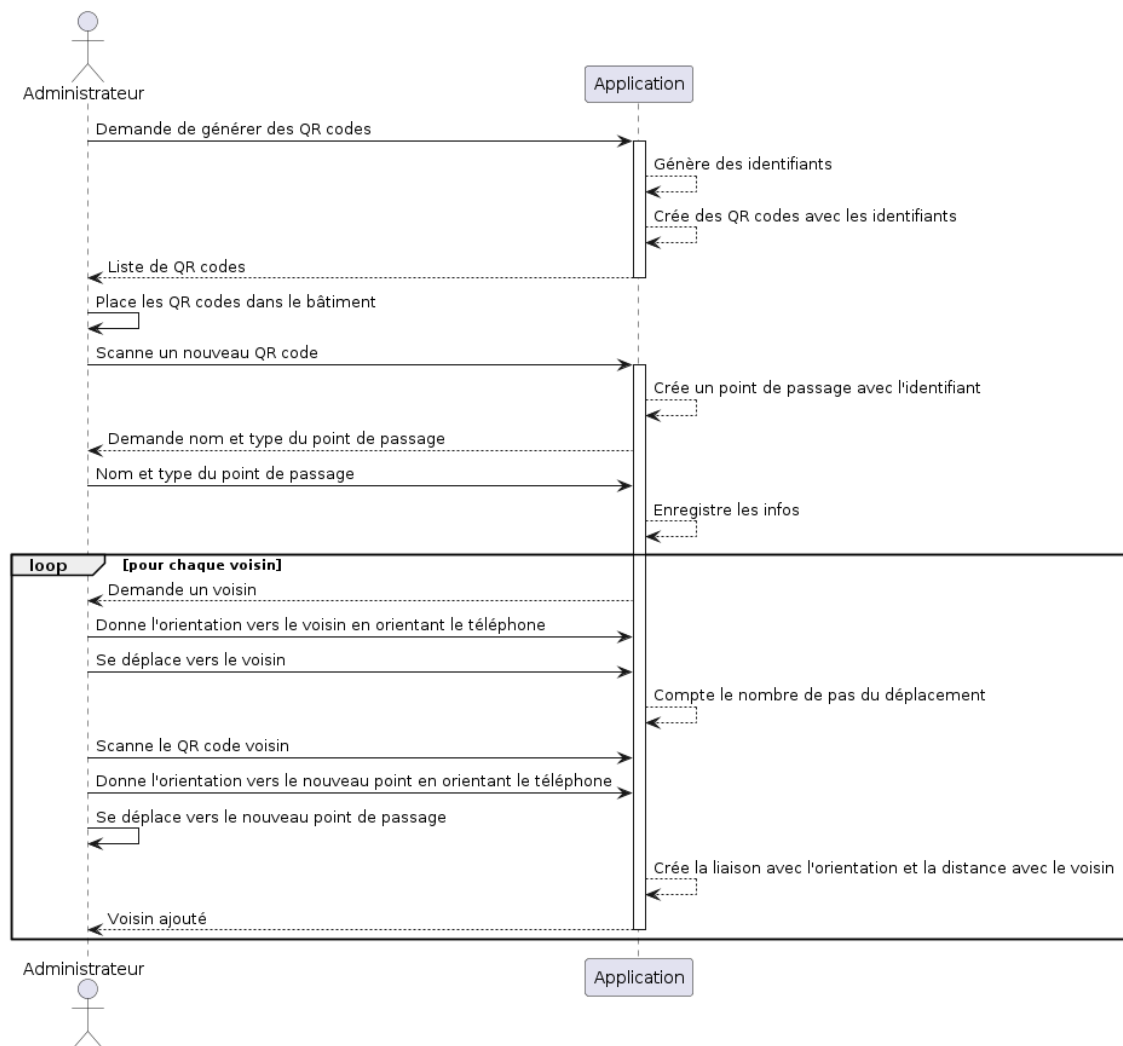


Figure 9 - Diagramme de séquences - Installation des points de passage

Dans le point 3.2.2., des règles pour établir les liaisons entre les points de passage ont été définies. L'administrateur doit respecter ces règles pour garantir une navigation efficace. Il doit également veiller à ce que les QR codes soient bien placés et visibles pour les utilisateurs. Il est également spécifié dans ce point qu'il ne peut y avoir qu'une seule liaison entre deux points de passage. Cela ne pose pas de problème pour l'administrateur car l'application est capable de vérifier si une liaison existe déjà avant d'en ajouter une nouvelle.

Pour que l'installation soit réalisable, le premier point de passage installé n'aura, bien sûr, pas de voisin. En revanche, les suivants devront obligatoirement être reliés.

3.4.2. Navigation

Une fois les points de passage installés dans le bâtiment, les utilisateurs peuvent commencer à utiliser l'application de navigation. Le processus de navigation a déjà été évoqué dans le point 3.1. mais est détaillé ci-après.

L'utilisateur ouvre l'application de navigation sur son smartphone et scanne le QR code où il se trouve dans le bâtiment. L'application identifie le point de passage correspondant et lui demande sa destination. L'utilisateur entre sa destination dans l'application. L'application détermine alors le chemin à parcourir pour se rendre à la destination en utilisant l'algorithme déterminé dans le point 3.3.3. L'application lui indique la direction à suivre pour se rendre au prochain point de passage. L'utilisateur suit les indications de l'application et scanne le QR code du prochain point de passage pour confirmer qu'il est bien arrivé. L'application lui indique alors le prochain point de passage à atteindre. Il répète cette opération jusqu'à atteindre sa destination.

Pour mieux comprendre le processus de navigation, voici un schéma illustrant les différentes étapes :

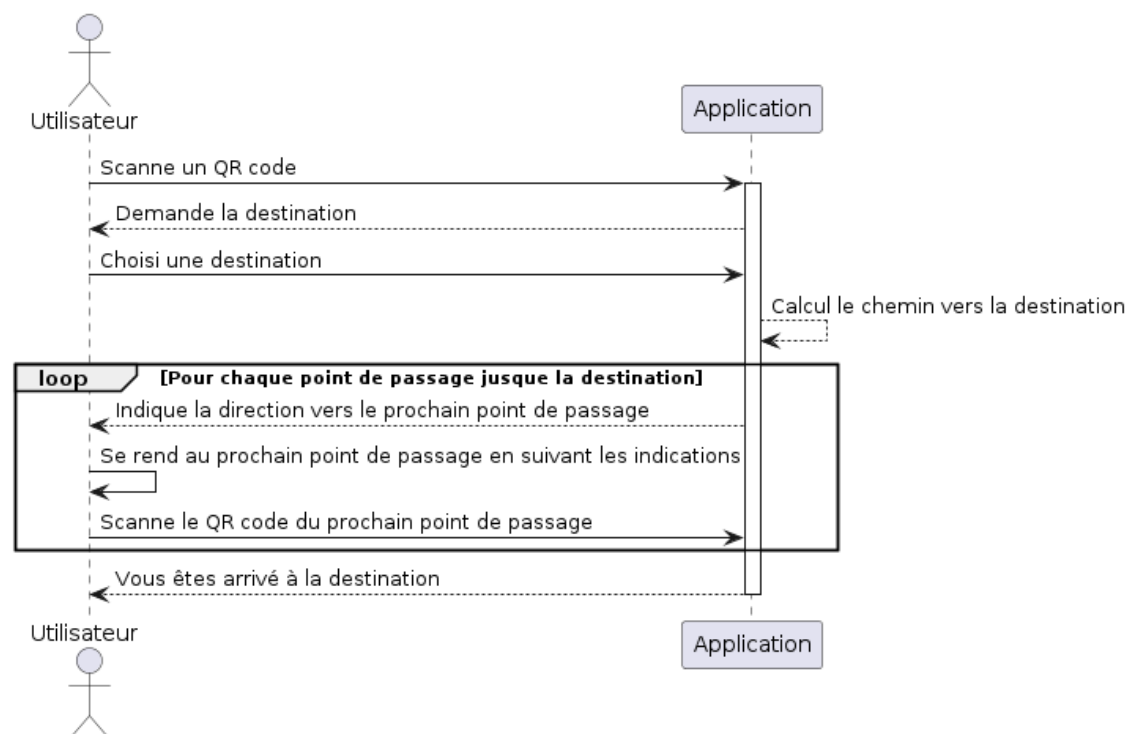


Figure 10 - Diagramme de séquences - Navigation

Dans le cas où l'utilisateur se trompe et se dirige vers un mauvais point de passage, l'application recalculera le chemin vers la destination à partir du point où l'utilisateur se trouve.

3.5. Résilience aux erreurs

Le système de navigation n'est pas basé sur une localisation précise de l'utilisateur et des points de passage. Il se base sur des valeurs entrées par l'administrateur qui a installé les points de passage. Ces valeurs peuvent ne pas être précises. Par exemple, l'administrateur a peut-être mal orienté son téléphone ou le comptage des pas a été quelque peu imprécis. Cela peut entraîner des erreurs dans la navigation de l'utilisateur, notamment dans les indications

de direction vers le prochain point de passage. Pour pallier à cela, des solutions peuvent être mises en place pour améliorer la précision du système.

3.5.1. Correction des distances

L'application peut corriger les distances entre les points de passage en utilisant les données des utilisateurs. Lorsque l'utilisateur se déplace entre deux points de passage, l'application peut compter le nombre de pas effectués. Elle peut ensuite réaliser une moyenne des distances relevées par les différents utilisateurs pour estimer la distance réelle entre les points de passage. Cette moyenne peut être utilisée pour corriger les distances enregistrées dans la base de données.

3.5.2. Correction des orientations

L'application peut également corriger les orientations des points de passage en utilisant les données des utilisateurs. Lorsque l'utilisateur se déplace d'un point de passage à un autre, l'application peut enregistrer l'orientation moyenne de son smartphone. Elle peut ensuite réaliser une moyenne des orientations relevées par les différents utilisateurs pour estimer l'orientation réelle du point de passage vers le voisin. Cette moyenne peut être utilisée pour corriger les orientations enregistrées dans la base de données.

3.6. Conclusion et perspectives

Dans cette partie d'analyse de l'hypothèse retenue, le fonctionnement du système de navigation basé sur des points de passage a été détaillé. Ce système permet de guider l'utilisateur dans un bâtiment en utilisant une infrastructure physique minimale. Cela permet ainsi de réduire les coûts d'installation. Il se base sur des QR codes pour matérialiser des points de passage. Ces points permettent de traduire le bâtiment en un réseau de points dans lequel il est possible de calculer un chemin entre un point de départ et une destination. Ainsi, l'utilisateur scanne le QR code où il se trouve, il entre sa destination et l'application lui indique le chemin à suivre.

Ce système peut grandir et s'améliorer en intégrant de nouvelles fonctionnalités. Par exemple, il serait possible d'ajouter de la réalité augmentée pour guider l'utilisateur en affichant un chemin virtuel devant lui. L'application pourrait également afficher à l'utilisateur sa position dans le bâtiment au moyen d'un plan interactif. Il serait aussi intéressant d'ajouter des paramètres de navigation comme la prise en compte des personnes à mobilité réduite (PMR) pour éviter les escaliers et privilégier les ascenseurs. Comme expliqué dans le point 3.5., il est possible que les données de navigation ne soient pas précises. Il serait donc pertinent d'analyser la navigation des utilisateurs pour les corriger. Enfin, il pourrait être avantageux pour l'administrateur de consulter des statistiques sur la navigation des utilisateurs dans le bâtiment.

En conclusion, le système de navigation basé sur des points de passage est une solution viable pour guider l'utilisateur dans un bâtiment. Il offre une alternative aux systèmes de navigation existants qui nécessitent un investissement financier important. Il permet de réduire les coûts d'installation tout en offrant une bonne expérience utilisateur. Il peut être amélioré en intégrant de nouvelles fonctionnalités pour répondre aux besoins des utilisateurs et des administrateurs.

4. Conception d'un prototype

Cette partie va détailler la conception d'un prototype pour le système de navigation basé sur des points de passage. Ce prototype de mettre en application le système de navigation par points de passage décrit dans les parties précédentes. Il permettra de valider le fonctionnement du système et d'identifier les éventuels problèmes à résoudre. Pour cela, il est nécessaire de définir un cahier des charges, une analyse fonctionnelle et une analyse technique.

4.1. Cahier des charges

Le prototype permettra de mettre en place un système de navigation en intérieur basé sur des points de passage. Il devra permettre à l'utilisateur d'installer des points de passage dans le bâtiment et de se guider vers une destination en suivant ces points.

4.1.1. Objectifs

Le prototype devra remplir les objectifs suivants :

- L'administrateur est capable d'installer des points de passage dans le bâtiment.
 - Il peut associer un QR code avec un point de passage.
 - Il peut installer plusieurs points de passage reliés entre eux.
- L'utilisateur est capable de se diriger dans le bâtiment en suivant les points de passage.
 - Il peut identifier un point de passage en scannant un QR code.
 - Les indications de l'application lui permettent de se rendre à sa destination.

4.1.2. Limitations

Pour remplir les objectifs précités, le prototype devra prendre en charge les fonctionnalités générales suivantes :

- L'installation des points de passage ;
- La navigation de l'utilisateur.

Toutefois, le prototype ne sera qu'une version simplifiée du système de navigation basé sur des points de passage. En effet, il a pour objectif principal de vérifier que le concept fonctionne. Il ne prendra donc pas en charge les fonctionnalités suivantes :

- La génération des QR codes ;
- La correction des distances et des orientations ;
- Les perspectives d'amélioration énumérées dans le point 3.6. ;
- La modification des points de passages installés.

4.1.3. Fonctionnalités

Voici les fonctionnalités, de façon un peu plus détaillée, que le prototype devra prendre en charge.

Pour l'installation des points de passage, l'application devra permettre à l'administrateur de :

- Scanner un QR code pour associer un point de passage ;
- Entrer un nom et un type pour le point de passage ;
- Orienter le point de passage par rapport aux points de passage voisins.

Pour la navigation de l'utilisateur, l'application devra permettre à l'utilisateur de :

- Scanner un QR code pour identifier un point de passage ;
- Entrer une destination pour calculer un chemin ;
- Suivre les indications de l'application pour se rendre à sa destination.

4.1.4. Contraintes

Le prototype devra respecter les contraintes suivantes :

- il devra être développé pour une plateforme mobile (iOS et/ou Android) ;
- il devra être capable de scanner des QR codes ;
- il devra être capable de stocker des données localement ou sur un serveur distant.

4.2. Analyse fonctionnelle

4.2.1. Cas d'utilisation

Le système de navigation basé sur des points de passage peut être décomposé en plusieurs cas d'utilisation. Voici un diagramme de cas d'utilisation pour le prototype :

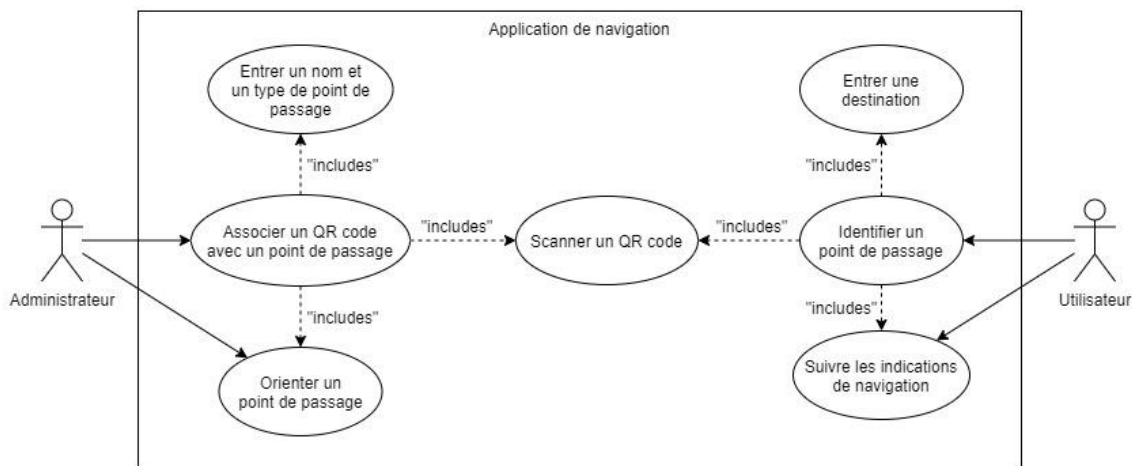


Figure 11 - Diagramme de cas d'utilisation

Dans le diagramme ci-dessus, il y a deux acteurs principaux : l'administrateur et l'utilisateur. L'administrateur est responsable de l'installation des points de passage dans le bâtiment. Pour réaliser cela, il utilise l'application pour associer un QR code avec un point de passage. Cela se fait en scannant le QR code avec l'application et en entrant le nom et le type du point de passage. Après cela, l'application lui permet d'orienter le point de passage par rapport aux autres. L'utilisateur peut également utiliser l'application pour se diriger dans le bâtiment. Pour cela, il peut identifier un point de passage en scannant un QR code. Pour la navigation, il doit entrer une destination et suivre ensuite les indications que lui donne l'application.

4.2.2. Fonctionnement

L'application de navigation doit suivre le fonctionnement décrit dans la partie d'analyse du système de navigation basé sur des points de passage (voir le point 3.4.).

4.2.2.1. Installation

Lors de l'installation du système, l'administrateur doit mettre en place les points de passage. Pour cela, il doit créer les QR codes à l'avance. Le prototype ne prendra pas en charge la

génération des QR codes. L'administrateur doit ensuite placer les QR codes dans le bâtiment aux endroits correspondants aux points de passage. Il doit ensuite associer les QR codes aux points de passage en scannant le QR code avec l'application. L'application lui demande alors de renseigner le nom et le type du point de passage. Il doit également orienter le point de passage par rapport aux autres points de passage. Pour cela, il oriente son smartphone dans la direction du point de passage voisin et enregistre l'orientation. Il se déplace ensuite vers le point de passage voisin et scanne le QR code pour enregistrer la liaison. Il oriente à nouveau son smartphone dans la direction du nouveau point de passage. Il peut revenir au nouveau point de passage et répéter l'opération pour chaque point de passage voisin.

Ce procédé est représenté dans le diagramme de séquence ci-après.

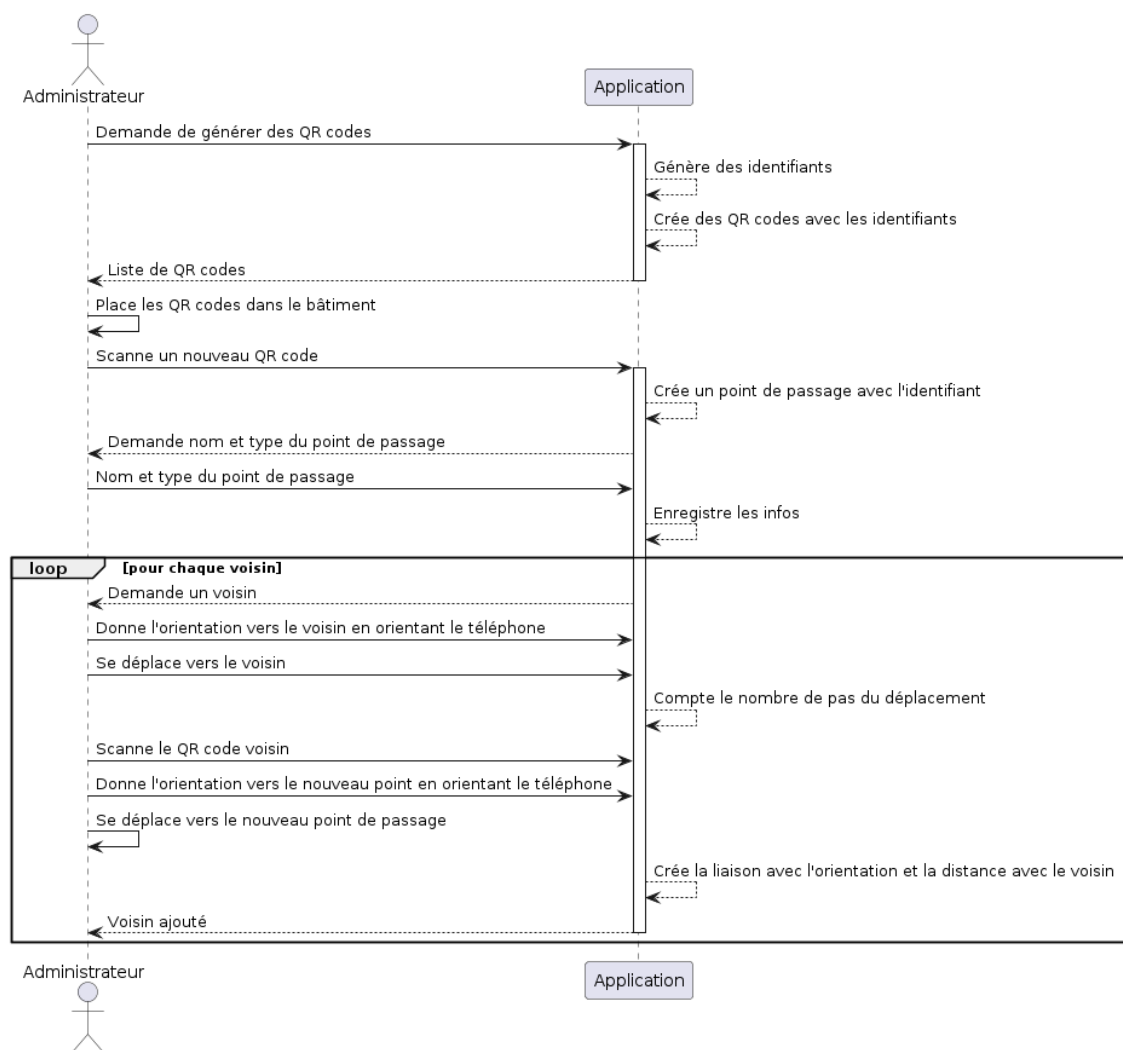


Figure 9 - Diagramme de séquences - Installation des points de passage

Une précision doit être apportée à propos de la boucle pour les voisins. Le premier point de passage installé n'aura forcément pas encore de voisin. L'application doit donc tolérer que ce point soit installé sans voisin. Toutefois, tous les points de passage suivants devront obligatoirement être connectés à au moins un point existant.

4.2.2.2. Navigation

Pour se diriger dans un bâtiment à l'aide de l'application, l'utilisateur doit scanner le QR code où il se trouve. L'application identifie le point de passage correspondant et lui demande sa destination. L'utilisateur entre sa destination dans l'application. L'application détermine ensuite le chemin à parcourir pour se rendre à la destination. Elle indique enfin à l'utilisateur la direction vers laquelle se rendre pour atteindre le prochain point de passage. L'utilisateur suit ces indications et scanne le QR code du prochain point de passage. L'application indique la direction de la suite du parcours jusqu'à atteindre la destination.

Ce procédé pour la navigation est illustré dans le diagramme de séquence ci-après.

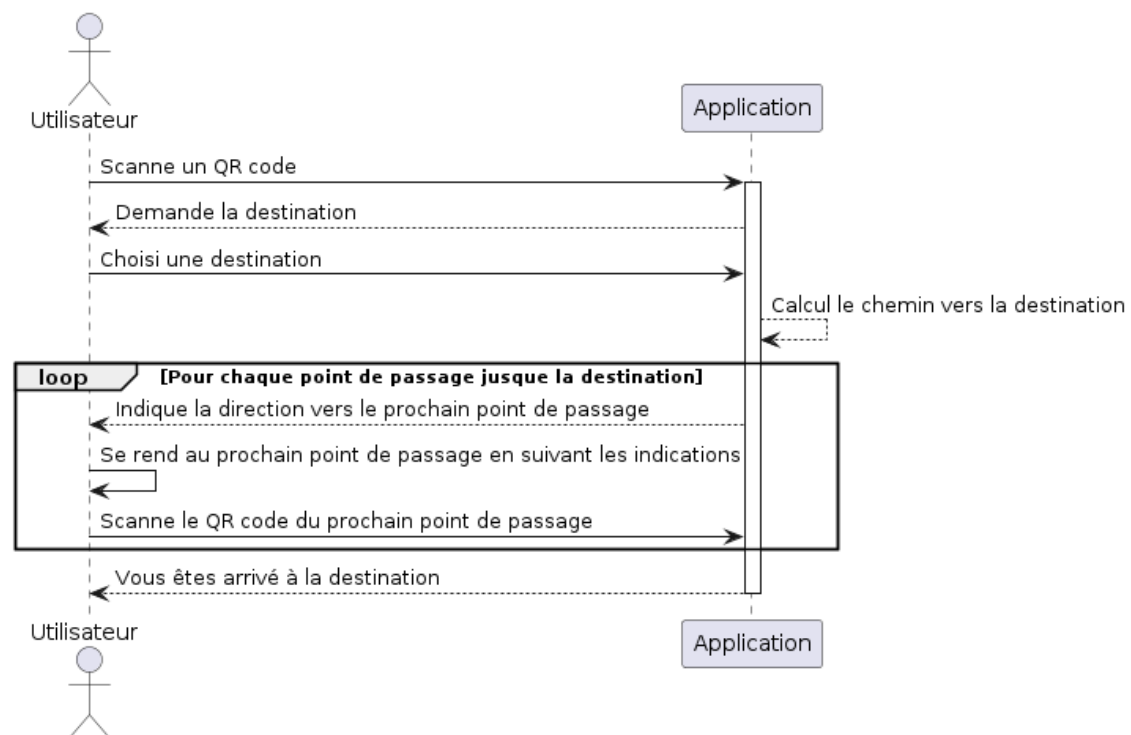


Figure 10 - Diagramme de séquences - Navigation

4.2.3. Maquettes d'écrans

Pour mieux visualiser le fonctionnement de l'application, voici des maquettes d'écrans. Elles représentent les différentes étapes de l'installation des points de passage et de la navigation de l'utilisateur. Elles permettent de fixer les visuels et les interactions de l'application.

4.2.3.1. Page d'accueil

La page d'accueil de l'application est la première page que l'utilisateur voit lorsqu'il ouvre l'application. Elle lui permet de choisir entre deux options : l'installation des points de passage ou la navigation.



Figure 12 - Maquette d'écran - Page d'accueil

4.2.3.2. Installation

a) Scanner du nouveau QR code

Si l'utilisateur choisit l'installation, il est dirigé vers la partie d'installation des points de passage de l'application. Il se trouve sur une première page qui lui permet de scanner le QR code à associer avec un point de passage.



Figure 13 - Maquette d'écran - Scanner le nouveau QR code

Un bouton « Quitter » est situé dans le coin supérieur gauche. Il permet à l'utilisateur de revenir à la page d'accueil. S'il désire quitter ou annuler le processus d'installation d'un nouveau point de passage, toutes les informations entrées seront perdues.

b) Entrer le nom et le type du point de passage

Après avoir scanné le nouveau QR code, l'utilisateur est redirigé vers une page qui lui permet d'entrer le nom et le type du point de passage.

La maquette d'écran a un fond brun foncé. En haut à gauche, il y a un bouton bleu avec le texte "< Annuler". Au centre, le titre "Ajout d'un nouveau point de passage" est écrit en blanc. En dessous, il y a deux champs de saisie. Le premier est un champ de texte gris avec l'étiquette "Un nom" au-dessus. Le second est un menu déroulant gris avec l'étiquette "Le type" au-dessus et une flèche noire vers le bas à droite. En bas de l'écran, il y a un bouton bleu avec le texte "Suivant".

Figure 14 - Maquette d'écran - Entrer le nom et le type du point de passage

Pour entrer les informations, l'utilisateur doit remplir le champ de texte pour le nom et sélectionner le type dans une liste déroulante. Pour accéder à la suite de l'installation, il doit absolument remplir les deux champs.

c) Ajouter un nouveau voisin

Après avoir entré le nom et le type du point de passage, l'utilisateur est redirigé vers une page qui lui permet d'ajouter un nouveau voisin au point de passage. Sur cette page, il a le choix d'ajouter un nouveau voisin ou de terminer l'installation.



Figure 15 - Maquette d'écran - Ajouter un point de passage voisin

L'utilisateur doit ajouter au moins un voisin pour le point de passage. Une seule exception est faite pour le premier point de passage qui n'a pas de voisin. Si l'utilisateur choisit de terminer l'installation, il est redirigé vers la page d'accueil.

Pour les étapes suivantes, le bouton "Annuler", situé dans le coin supérieur gauche, permet à l'utilisateur d'annuler l'ajout d'un nouveau voisin et de revenir à la page cette page.

d) Orienter vers le voisin

Si l'utilisateur choisit d'ajouter un nouveau voisin, il est redirigé vers une page qui lui permet d'orienter le voisin depuis le point de passage à associer. Sur cette page, il doit orienter son smartphone dans la direction du voisin et déterminer l'orientation en appuyant sur le bouton au centre de l'écran. Une flèche indique alors la direction. S'il est satisfait de l'orientation, il peut appuyer sur le bouton "Suivant" au bas de l'écran pour enregistrer l'orientation et passer à l'étape suivante.



Figure 16 - Maquette d'écran - Orienter vers le point de passage voisin

e) Scanner le voisin

Après avoir orienté le voisin, l'utilisateur doit se déplacer vers le voisin et scanner son QR code. L'application détecte le QR code et associe le voisin avec le point de passage.



Figure 17 - Maquette d'écran - Scanner le QR code voisin

f) Orienter vers le nouveau point de passage

Une fois le voisin associé, l'utilisateur doit orienter le nouveau point de passage depuis le voisin. Le processus est le même que pour l'orientation du voisin (étape d). Cette étape termine l'installation du point de passage. Ce dernier est alors enregistré dans la base de données de l'application. L'utilisateur est ensuite redirigé vers la page d'ajout d'un nouveau voisin (étape c).



Figure 18 - Maquette d'écran - Orienter vers le nouveau point de passage

4.2.3.3. Navigation

a) Scanner le QR code de départ

Si l'utilisateur choisit la navigation, il est redirigé vers la partie de navigation de l'application. Il se trouve sur une première page qui lui permet de scanner le QR code où il se trouve.



Figure 19 - Maquette d'écran - Scanner le QR code de départ

Un bouton « Quitter » est situé dans le coin supérieur gauche. Il permet à l'utilisateur de revenir à la page d'accueil. S'il désire quitter durant le processus de navigation, le chemin en cours sera perdu. Cependant, il lui est possible de reprendre la navigation en scannant le QR code où il se trouve et en entrant à nouveau sa destination.

b) Entrer la destination

Après avoir scanné le QR code de départ, l'utilisateur est redirigé vers une page qui lui permet d'entrer sa destination. Il doit choisir la destination dans une liste déroulante.

La maquette d'écran est présentée sur un fond brun foncé. En haut à gauche, il y a un bouton bleu avec le texte "< Annuler". Au centre, le mot "Navigation" est écrit en blanc. En dessous, le texte "La destination" est suivi d'un menu déroulant gris avec une flèche noire vers le bas. En bas de l'écran, il y a un bouton bleu avec le texte "Suivant".

Figure 20 - Maquette d'écran - Entrer la destination

c) Suivre les indications et scanner le QR code suivant

Après avoir entré sa destination, l'utilisateur est redirigé vers une page qui lui permet de scanner le QR code du prochain point de passage. L'application lui indique la direction à suivre, au moyen d'une flèche, pour se rendre au prochain point de passage. L'utilisateur suit les indications et scanne le QR code du prochain point de passage. L'application lui indique alors le prochain point de passage à atteindre. Il répète cette opération jusqu'à atteindre sa destination.



Figure 21 - Maquette d'écran - Suivre les indications et scanner le prochain QR code

d) Arriver à la destination

Lorsque l'utilisateur atteint sa destination, il est redirigé vers une page qui lui indique qu'il est arrivé à destination. Il peut alors terminer la navigation, ce qui le ramène à la page d'accueil.



Figure 22 - Maquette d'écran - L'utilisateur est arrivé à la destination

4.2.3.4. Schéma de navigation dans l'application

Pour mieux visualiser le fonctionnement de l'application, voici un schéma de navigation qui reprend les différentes étapes de l'installation des points de passage et de la navigation de l'utilisateur. Il permet de voir comment les différentes pages de l'application sont reliées entre elles.

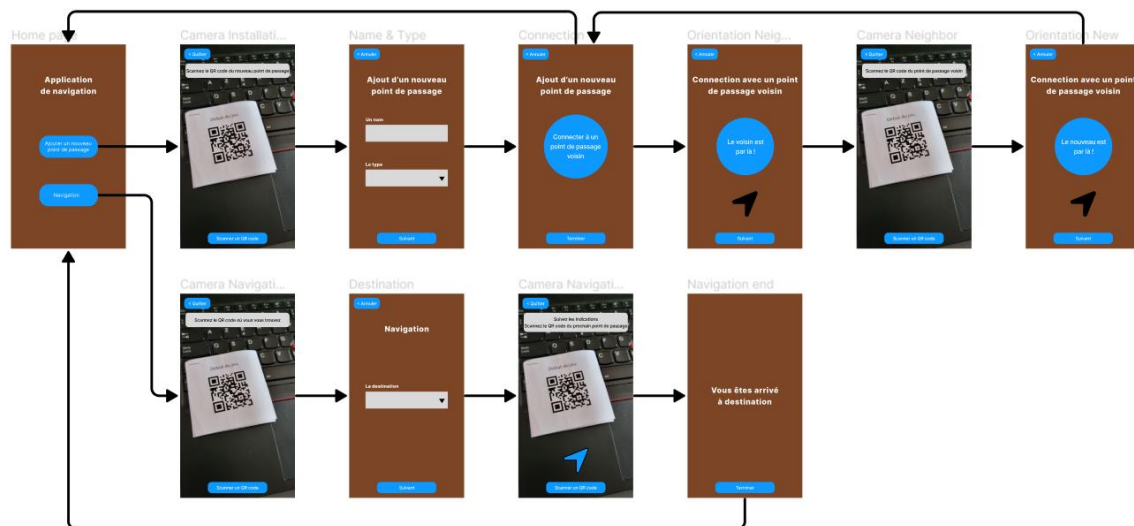


Figure 23 - Schéma de navigation dans l'application

4.3. Analyse technique

L'analyse technique du prototype va déterminer les choix technologiques à faire pour sa réalisation. Il va déterminer les technologies à utiliser pour le développement de l'application, la base de données à utiliser et la structure de l'application.

4.3.1. Choix technologiques

4.3.1.1. Technologie mobile

Comme spécifié dans le cahier des charges, le prototype doit être développé pour une plateforme mobile (iOS et/ou Android). Pour cela, plusieurs types d'application existent : les applications natives, les applications hybrides/cross-plateforme, les applications PWA et les applications web. Les applications natives sont développées spécifiquement pour une plateforme (iOS ou Android) en utilisant les langages de programmation et les outils propres à la plateforme. Les applications hybrides/cross-plateforme sont développées en utilisant des technologies web (HTML, CSS, JavaScript) et sont ensuite compilées pour les différentes plateformes. Les applications PWA sont des applications web qui peuvent être installées sur le smartphone de l'utilisateur. Les applications web sont des applications qui s'exécutent dans un navigateur web.

Le prototype n'a pas de vocation commerciale. Il est destiné à vérifier que le concept de navigation basé sur des points de passage fonctionne. Il n'est donc pas nécessaire de choisir un type d'application pour ce motif. Toutefois, il doit respecter les contraintes du cahier des charges : il doit être capable de scanner des QR codes et de stocker des données localement

ou sur un serveur distant. Pour diriger le choix, une matrice de décision pondérée peut être réalisée. Les critères à considérer sont les suivants :

- Multi-plateforme : la capacité de l'application à fonctionner sur plusieurs plateformes.
- Accès aux fonctionnalités du smartphone : la capacité de l'application à accéder aux fonctionnalités du smartphone (appareil photo, stockage, etc.).
- Maintenabilité : la facilité de maintenir l'application.
- Connaissances : évaluation des connaissances de l'équipe de développement.

Tableau 3 - Matrice de décision - Technologie mobile

Critères	Pondération	Natif	Hybride	PWA	Web
Multi-plateforme	1	0 (Non)	1 (Oui)	1 (Oui)	1 (Oui)
Fonctionnalités du téléphone	3	5	4	2	2
Maintenabilité	2	2	4	4	4
Connaissances	2	2	4	4	4
	Total	23	29	23	23

Un commentaire sur les critères est nécessaire. Tout d'abord, l'accès aux fonctionnalités du smartphone est évalué à 2 pour les PWA et les applications web car l'accès est limité par le navigateur sur lequel elles s'exécutent. Ensuite, la maintenabilité des applications natives est plus fastidieuse car il faut maintenir une application pour chaque plateforme. Enfin, l'équipe de développement est spécialisée dans les technologies web, ce qui explique la note de 4 pour les applications hybrides/cross-plateforme, les PWA et les applications web.

La matrice de décision ci-avant permet de déterminer que les applications hybrides/cross-plateforme semblent être le meilleur choix pour le prototype. Elles obtiennent un score total de 29, contre 23 pour les autres types d'application. En effet, elles permettent un accès aux fonctionnalités du smartphone, elles sont facilement maintenables et l'équipe de développement a les connaissances nécessaires pour les développer.

Maintenant que le type d'application est déterminé, il est nécessaire de choisir un framework pour le développement de l'application. Pour les applications hybrides/cross-plateforme, plusieurs frameworks existent, notamment React Native, Flutter, Native Script et Ionic. Ces frameworks sont très semblables. La décision dépend principalement des préférences de l'équipe de développement. Toutefois, des critères comme la popularité et la documentation peuvent être pris en compte.

Une première décision peut être prise en fonction du langage de programmation utilisé. L'équipe de développement a des connaissances avancées en JavaScript. Voici un tableau comparatif des frameworks en fonction du langage de programmation utilisé :

Tableau 4 - Tableau comparatif des frameworks hybrides en fonction de leur langage de programmation

Framework	Langage de programmation
React Native	Javascript
Flutter	Dart
Native Script	Javascript
Ionic	Javascript

Flutter est écarté car il utilise le langage Dart. Ce langage est moins connu de l'équipe de développement.

La suite de la décision peut se baser sur la popularité, la documentation et l'appréciation de l'équipe de développement pour les frameworks restants. Voici une matrice de décision pondérée pour choisir le framework :

Tableau 5 - Matrice de décision - Technologie mobile hybride

Critère	Pondération	React Native	Native Script	Ionic
Popularité	1	5	2	3
Documentation	2	4	3	4
Appréciation de l'équipe	3	4	2	3
	Total	25	14	20

D'après la matrice de décision, React Native semble être le meilleur choix pour le développement du prototype. Il obtient un score total de 25, contre 14 pour Native Script et 20 pour Ionic. Il est le plus populaire, il a une bonne documentation et c'est le framework le mieux apprécié par l'équipe de développement. Cette appréciation est due à l'utilisation régulière de ce framework et à la connaissance approfondie de son fonctionnement.

React Native peut être utilisé avec Expo, un ensemble d'outils de développement pour React Native. Expo apporte, par exemple, des accès prêts à l'emploi aux fonctionnalités du smartphone comme l'appareil photo. Il permet également d'exécuter l'application sur un émulateur ou un smartphone physique. Il est donc un choix judicieux pour le développement du prototype. React Native permet également d'utiliser TypeScript, une surcouche de JavaScript. C'est ce langage qui sera utilisé pour le développement du prototype. Il permettra d'éviter les erreurs de typage et d'améliorer la qualité du code.

4.3.1.2. Base de données

4.3.2. Fonctionnement technique

4.3.2.1. QR codes

Les QR codes sont des codes-barres en deux dimensions qui peuvent stocker des informations. Ils sont utilisés pour matérialiser les points de passage dans le bâtiment. Chaque QR code contient un identifiant unique qui permet d'identifier le point de passage associé. Lorsque

l'utilisateur scanne un QR code, l'application récupère l'identifiant et identifie le point de passage correspondant dans la base de données.

Pour scanner les QR codes, l'application utilisera l'appareil photo du smartphone. Pour cela, nous utiliserons le module [Camera](#) de Expo/React Native. Ce module permet d'accéder à l'appareil photo du smartphone pour prendre des photos et scanner des QR codes. Il permet également de récupérer les données des QR codes scannés.

4.3.2.2. Orientation

L'orientation des points de passage est une information importante pour la navigation de l'utilisateur. Elle permet de déterminer la direction à suivre pour se rendre au point de passage voisin. L'orientation est déterminée par l'administrateur lors de l'installation des points de passage. Pour récupérer l'orientation, nous utiliserons la boussole du smartphone. Ainsi, l'application pourra déterminer l'orientation du smartphone par rapport au nord. La boussole est accessible via le module [Magnetometer](#) de Expo/React Native.

4.3.2.3. Estimation de la distance

La distance entre les points de passage sera estimée en fonction du nombre de pas effectués par l'administrateur lors de l'installation des points de passage. Pour cela, l'application utilisera le module [Pedometer](#) de Expo/React Native.

4.3.3. Structure de l'application

5. Validation

(analyse proto si ça fonctionne, s'il remplit les casus farfelus)

Table des figures

Figure 1 - Photos du fléchage dans le bâtiment IESN de l'Hénallux.....	6
Figure 2 - Panneaux de direction [1]	6
Figure 3 - Capture d'écran de l'application "plan" de Walibi Belgium.....	7
Figure 4 - Schéma des points de passage.....	11
Figure 5 - Schéma des liaisons entre les points de passage.....	12
Figure 6 - Schéma du tracé entre un point A et un point B	12
Figure 7 - Schéma de base de données relationnelle	18
Figure 8 - Schéma de base de données NoSQL orientée graphe	18
Figure 9 - Diagramme de séquences - Installation des points de passage.....	20
Figure 10 - Diagramme de séquences - Navigation	21
Figure 11 - Diagramme de cas d'utilisation.....	24
Figure 12 - Maquette d'écran - Page d'accueil.....	27
Figure 13 - Maquette d'écran - Scanner le nouveau QR code	28
Figure 14 - Maquette d'écran - Entrer le nom et le type du point de passage.....	29
Figure 15 - Maquette d'écran - Ajouter un point de passage voisin.....	30
Figure 16 - Maquette d'écran - Orienter vers le point de passage voisin	31
Figure 17 - Maquette d'écran - Scanner le QR code voisin	32
Figure 18 - Maquette d'écran - Orienter vers le nouveau point de passage	33
Figure 19 - Maquette d'écran - Scanner le QR code de départ.....	34
Figure 20 - Maquette d'écran - Entrer la destination	35
Figure 21 - Maquette d'écran - Suivre les indications et scanner le prochain QR code	36
Figure 22 - Maquette d'écran - L'utilisateur est arrivé à la destination.....	37
Figure 23 - Schéma de navigation dans l'application	38

Table des tableaux

Tableau 1 - Matrice de décision - Matérialisation des points de passage	13
Tableau 2 - Matrice de décision - Algorithme de recherche de chemin dans un graphe	16
Tableau 3 - Matrice de décision - Technologie mobile	39
Tableau 4 - Tableau comparatif des frameworks hybrides en fonction de leur langage de programmation	40
Tableau 5 - Matrice de décision - Technologie mobile hybride	40

Références

La rédaction des textes de ce document a été réalisée avec l'aide d'une intelligence artificielle.

- [1] « Fléchage interne par routes », Cliniques universitaires Saint-Luc. Consulté le: 4 avril 2024. [En ligne]. Disponible sur: <https://www.saintluc.be/fr/routes>
- [2] VAISHAK, « Top 10 Indoor Navigation Services Companies », Medium. Consulté le: 29 avril 2024. [En ligne]. Disponible sur: https://medium.com/@VAISHAK_CP/top-10-indoor-navigation-service-companies-d15ae7ebf074
- [3] J. Kunthoth, A. Karkar, S. Al-Maadeed, et A. Al-Attiyah, « Comparative analysis of computer-vision and BLE technology based indoor navigation systems for people with visual impairments », *Int. J. Health Geogr.*, vol. 18, n° 1, p. 29, déc. 2019, doi: 10.1186/s12942-019-0193-9.
- [4] A. Omisakin, « RF based Indoor Positioning using RSSI fingerprints », sept. 2016.
- [5] M. Shchekotov, « Indoor localization method based on Wi-Fi trilateration technique. », *Proceeding 16th Conf. Fruct Assoc.*, p. 177-9, 2014.
- [6] T. H. T. Pham, « Apports et difficultés d'une collecte des données à l'aide du récepteur GPS pour réaliser une enquête sur la mobilité », Theses, Université Paris-Est, 2016. [En ligne]. Disponible sur: <https://hal.science/tel-01548851>
- [7] E. Mastrostefano et M. Bernaschi, « Efficient breadth first search on multi-GPU systems », *J. Parallel Distrib. Comput.*, vol. 73, n° 9, p. 1292-1305, 2013, doi: <https://doi.org/10.1016/j.jpdc.2013.05.007>.
- [8] J. Hopcroft et R. Tarjan, « Efficient Planarity Testing », *J ACM*, vol. 21, n° 4, p. 549-568, oct. 1974, doi: 10.1145/321850.321852.
- [9] E. W. Dijkstra, « Ewd 316: a short introduction to the art of programming », 1971. [En ligne]. Disponible sur: <https://www.cs.utexas.edu/users/EWD/transcriptions/EWD03xx/EWD316.html>
- [10] A. Schrijver, « On the History of the Shortest Path Problem », 2012. [En ligne]. Disponible sur: https://ftp.gwdg.de/pub/misc/EMIS/journals/DMJDMV/vol-ismv/32_schrijver-alexander-sp.pdf
- [11] O. K. Sulaiman, A. M. Siregar, K. Nasution, et T. Haramaini, « Bellman Ford algorithm - in Routing Information Protocol (RIP) », *J. Phys. Conf. Ser.*, vol. 1007, n° 1, p. 012009, avr. 2018, doi: 10.1088/1742-6596/1007/1/012009.

Annexes
