

h e g

Haute école de gestion
Genève

Impacts des smartphones et tablettes sur les interfaces applicatives, théorie et mise en œuvre actuelle

Travail de Bachelor réalisé en vue de l'obtention du Bachelor HES

par :

Fabio AVELINO

Conseiller au travail de Bachelor :

Rolf HAURI, Chargé d'enseignement

Genève, le 18 septembre 2017

Haute École de Gestion de Genève (HEG-GE)

Filière Informatique de Gestion

Déclaration

Ce travail de Bachelor est réalisé dans le cadre de l'examen final de la Haute école de gestion de Genève, en vue de l'obtention du titre Bachelor of Science en Informatique de Gestion.

L'étudiant atteste que son travail a été vérifié par un logiciel de détection de plagiat.

L'étudiant accepte, le cas échéant, la clause de confidentialité. L'utilisation des conclusions et recommandations formulées dans le travail de Bachelor, sans préjuger de leur valeur, n'engage ni la responsabilité de l'auteur, ni celle du conseiller au travail de Bachelor, du juré et de la HEG.

« J'atteste avoir réalisé seul le présent travail, sans avoir utilisé des sources autres que celles citées dans la bibliographie. »

Fait à Genève, le 18 septembre 2017

Fabio AVELINO

Remerciements

Je remercie le conseiller de mon travail, Rolf HAURI, qui a accepté mon idée et m'a orienté vers plusieurs points. Sa disponibilité et ses conseils m'ont permis d'avancer rapidement dans ce mémoire.

Je remercie aussi ma famille et mes amis de m'avoir soutenu et aidé pour la rédaction et la correction de ce travail.

Résumé

Les smartphones et les tablettes ont changé notre manière de concevoir les applications. Arrivés depuis 2007, ces supports n'ont cessé d'augmenter leur part de marché par rapport au seul compétiteur qu'était le PC. Ils ne peuvent plus être ignorés lors de la conception d'un produit applicatif informatique. Cependant, pour développer des applications pour ces nouveaux supports, plusieurs points doivent être pris en compte.

Plusieurs sujets sont abordés dans ce travail. La première partie expliquera la montée en puissance de ces nouveaux appareils et leurs manières d'être utilisés.

Ensuite, des pratiques actuelles concernant les éléments d'interfaces utilisateurs seront présentées. En effet, des recommandations pour la conception d'interfaces utilisateurs existent afin d'offrir la meilleure expérience à l'utilisateur final.

De plus, quelques outils et méthodologies de développement seront détaillés pour comprendre comment concevoir des applications mobiles et web de nos jours.

Une analyse d'une application, portant sur le respect des best practices des interfaces utilisateurs, sera présentée en se basant sur le point de vue d'une méthodologie. Le but de cette analyse est de montrer le respect et la prise de liberté concernant les pratiques actuelles expliquées dans ce travail.

Finalement, on retrouvera une proposition de solution visant à faciliter et accélérer le développement d'applications pour le web et pour les nouveaux supports de manière native.

Table des matières

Déclaration.....	i
Remerciements	ii
Résumé	iii
Liste des tableaux	vii
Liste des figures.....	viii
1. Introduction.....	1
2. Définitions	2
2.1 Glossaire, abréviations et termes techniques.....	2
2.2 L'expérience utilisateur et l'interface utilisateur.....	2
2.3 Définition d'une application.....	3
3. Changements provoqués par les appareils mobiles	4
3.1 Les ventes	4
3.2 Le trafic web	5
3.3 Google	6
3.4 Les marchés d'applications	7
3.5 Utilisation de ces nouveaux supports	7
4. Courant de design digital.....	8
4.1 Skeuomorphisme	8
4.2 Flat design.....	9
5. Pratiques actuelles	12
5.1 Tactile	12
5.2 Taille d'écran des smartphones.....	12
5.3 Boutons	14
5.3.1 Floating Action Button	16
5.4 Les gestes	17
5.4.1 Pression.....	17
5.4.2 Double pression	17
5.4.3 Pression longue	17
5.4.4 Glisser le doigt vers une direction	17
5.4.5 Écarter / Pincer	17
5.5 La typographie	18
5.6 Les icônes	19
5.7 Les animations	21
5.8 Menus de navigation.....	21
5.8.1 Barre principale.....	22
5.8.2 Tiroir de navigation.....	22

5.8.3	Barre d'onglets.....	23
5.9	Disposition	24
5.9.1	Espacement.....	24
5.9.2	Orientation	25
6.	Développement des interfaces utilisateurs depuis l'arrivée des appareils mobiles.....	26
6.1	Développement des applications natives avec les outils officiels	26
6.1.1	Affichage adaptatif	28
6.2	Web Design	30
6.2.1	Responsive Web Design.....	30
6.2.1.1	CSS : Media Queries.....	30
6.2.1.2	Grille de disposition	32
6.2.1.3	Exemple.....	33
7.	Nouvelles méthodologies de développement.....	35
7.1	Mobile First.....	35
7.2	Atomic Design.....	36
7.2.1	Atomes.....	36
7.2.2	Molécules.....	37
7.2.3	Organismes.....	37
7.2.4	Modèle	37
7.2.5	Page	37
8.	Analyse de l'interface utilisateur d'une application	38
8.1	Analyse atomique	38
8.1.1	Atomes.....	38
8.1.1.1	Icônes.....	38
8.1.1.2	Typographie.....	40
8.1.1.3	Autres éléments.....	40
8.1.2	Molécules.....	41
8.1.2.1	Barre principale	41
8.1.2.2	Barre d'onglets	41
8.1.2.3	Barre d'interaction avec une photo.....	42
8.1.2.4	Profil de la personne ayant publié le contenu	42
8.1.2.5	Nom du profil et commentaire	43
8.1.3	Organismes.....	43
8.1.3.1	Barre principale sur la version web PC	43
8.1.3.2	Contenu principal : Photo ou vidéo	43
8.1.4	Modèle de page	45
8.2	Conclusion de l'analyse	46
9.	Proposition de solution.....	47
9.1	React et React Native.....	48
9.2	Structure du projet.....	49
9.3	Création d'un composant natif.....	50

9.4 Explication de ces choix.....	51
10. Conclusion	53
Bibliographie	54

Liste des tableaux

Tableau 1 : Répartition du trafic mondial sur le web, juillet 2017	5
Tableau 2 : Répartition du trafic sur le web en Suisse, juillet 2017	6
Tableau 3 : Répartition des OS mobiles concernant le trafic web, juillet 2017	6

Liste des figures

Figure 1 : Répartition du trafic mondial sur le web, janvier 2009 à juillet 2017	5
Figure 2 : Exemples de skeuomorphisme sur les premières versions d'iOS	8
Figure 3 : Exemple de poster de la période de création du style suisse.....	9
Figure 4 : Interface de Windows 8	10
Figure 5 : Exemples de flat design à partir d'iOS 7	10
Figure 6 : Exemple d'interface adoptant le Material Design de Google.....	11
Figure 7 : Comparaison de la zone de pression d'un doigt avec une pièce équivalente à 0.50 CHF et une touche de clavier	12
Figure 8 : Zone de confort d'utilisation des iPhone	13
Figure 9 : Apple Maps avant et après iOS 10	14
Figure 10 : Les 4 états d'un bouton	15
Figure 11 : Floating Action Button sur Twitter Android.....	16
Figure 12 : Comparaisons du nombre de caractères par ligne sur appareils mobiles ..	18
Figure 13 : Signification des icônes iOS par Apple	20
Figure 14 : Exemples d'icônes : Android à gauche et iOS à droite	20
Figure 15 : Exemple de barre principale	22
Figure 16 : Exemple de tiroir de navigation	23
Figure 17 : Recommandations de placement de la barre d'onglets	24
Figure 18 : marges et alignement vertical se basant sur des multiples de 8	25
Figure 19 : Développement d'une interface pour Android	27
Figure 20 : Développement d'une interface pour iOS	27
Figure 21 : Disposition d'éléments pour une interface adaptative	28
Figure 22 : Exemple de disposition avec l'application Zalando	29
Figure 23 : CSS – Media Queries.....	31
Figure 24 : Exemple visuel de grille de disposition	32
Figure 25 : Exemple de Responsive Web Design.....	33
Figure 26 : Règles d'affichage de l'exemple	34
Figure 27 : Atomic Design : les 5 niveaux.....	36
Figure 28 : Icônes utilisées par Instagram	38
Figure 29 : Exemple des différences d'icônes	39
Figure 30 : Différentes zones de texte utilisées sur Instagram.....	40
Figure 31 : Différentes barres principales sur Instagram	41
Figure 32 : Différentes barres d'onglets sur Instagram	41
Figure 33 : Différentes barres d'interaction avec une photo sur Instagram	42
Figure 34 : Élément du profil de la personne de la publication sur Instagram	42
Figure 35 : Barre principale sur la version web PC d'Instagram.....	43
Figure 36 : Contenu principal d'Instagram	43
Figure 37 : Modèles de pages génériques sur Instagram	45
Figure 38 : Application native mobile réalisée par moi-même.....	47
Figure 39 : Exemple de composant réalisé avec React	48
Figure 40 : Exemple de composant avec React Native	48
Figure 41 : Structure des composants avec le concept Atomic Design.....	49
Figure 42 : Barre principale de l'application de démonstration.....	51
Figure 43 : Alignement spécifique du titre de la page selon l'OS	51
Figure 44 : Affichage d'un composant selon l'OS	51

1. Introduction

Selon une infographie de Experience Dynamics¹, on estime que 70 % des projets échouent dû au manque d'acceptation de l'utilisateur final par rapport au produit. L'un des motifs d'adoption d'une application est son efficacité et sa facilité d'utilisation. Les utilisateurs pensent que 72 % du temps, c'est ce point qui décide de l'acceptation d'une application. Les fonctionnalités proposées ne représentent que 16 % du critère de choix d'adoption.

Autre statistique intéressante, dépenser 10 % du budget de développement pour l'ergonomie d'un programme informatique augmenterait l'acceptation par rapport à l'utilisateur de 83 %. En moyenne, les entreprises investissent 11.5 % du budget de développement d'un projet dans l'expérience utilisateur.

Nul doute que le design des interfaces graphiques est donc un des points essentiels du succès d'une application. L'aspect esthétique et l'ergonomie doivent être bien pensés afin de livrer une solution agréable et efficace en termes d'utilisation.

Mais, depuis 2007, 2 nouveaux supports ont vu le jour : les smartphones et les tablettes. Pourquoi et comment ces appareils ont-ils changé notre manière de concevoir les interfaces utilisateurs des applications d'aujourd'hui ?

¹ <https://www.experiencedynamics.com/blog/2014/07/making-strong-business-case-roi-ux-infographic>

2. Définitions

2.1 Glossaire, abréviations et termes techniques

UX (User eXperience) est une abréviation anglaise désignant l'expérience utilisateur.

UI (User Interface) est une abréviation anglaise désignant l'interface utilisateur.

CSS (Cascading Style Sheets) est un langage informatique utilisé pour mettre en forme le contenu d'une page web.

W3C (World Wide Web Consortium) est l'organisme de standardisation des technologies web.

OS (Operating System) est une abréviation anglaise désignant le système d'exploitation d'un ordinateur.

Un **IDE** (Integrated Development Environment) est un environnement de développement facilitant la programmation d'applications.

Du code **natif** est du code qui est exécuté directement par la machine, ce qui offre des performances optimales. En comparaison, du code web est exécuté par un intermédiaire (le navigateur Internet) et voit ses performances réduites.

2.2 L'expérience utilisateur et l'interface utilisateur

L'expérience utilisateur désigne les interactions d'un utilisateur final avec un produit ou un service. On y retrouve les attentes exactes du client, le contenu à proposer, la structure de l'information, la conception visuelle et interactive ainsi que l'ergonomie. Le but de cette science est qu'un public cible ait envie d'utiliser le produit ou le service en question et qu'il s'y fidélise.

Pour arriver à une excellente expérience utilisateur, plusieurs outils et méthodologies existent comme la recherche des besoins de l'utilisateur, la définition de profils types d'utilisateurs finaux, la définition des cas d'utilisation, du prototypage et des tests d'utilisation par un utilisateur final sur une portion ou la totalité du produit.

Le résultat de tout ceci est alors retranscrit dans l'interface utilisateur en y apportant des touches d'esthétique comme la marque du produit, des couleurs et la typographie par exemple.

2.3 Définition d'une application

Tout au long de ce travail, le terme « application » sera employé. Bon nombre d'applications sont disponibles via des marchés virtuels ou des pages web.

Une application a pour but d'offrir plusieurs fonctionnalités et plusieurs interactions à un utilisateur, pouvant aller d'une simple tâche à une action complexe.

Prenons l'exemple d'un blog de recettes de cuisine pour mieux comprendre ce qu'est une application. On distingue deux utilisateurs : le propriétaire du blog et le lecteur. Le premier utilisateur a la possibilité de publier un article, le styliser, l'éditer ou le supprimer. Ce sont des fonctionnalités et donc, il a affaire à une application. Par contre, le deuxième utilisateur consulte le blog et ne peut que lire le contenu publié : il n'est par conséquent pas sur une application, car aucune fonctionnalité lui est proposée.

3. Changements provoqués par les appareils mobiles

3.1 Les ventes

Le premier smartphone à avoir révolutionné le marché n'est autre que l'iPhone. Sorti en 2007 par Apple, cet appareil a posé les fondements de ce qui caractérise un téléphone intelligent : un écran tactile, un appareil photo et un système d'exploitation mobile capable de naviguer sur Internet, lancer des applications ainsi que d'en télécharger d'autres à l'aide d'un magasin en ligne. Ce système d'exploitation mobile s'appelle iOS. La sortie de l'iPhone a été rapidement suivie par le HTC G1, premier smartphone Android qui est le système d'exploitation mobile développé par Google. Ce téléphone propose des caractéristiques similaires à l'appareil d'Apple.

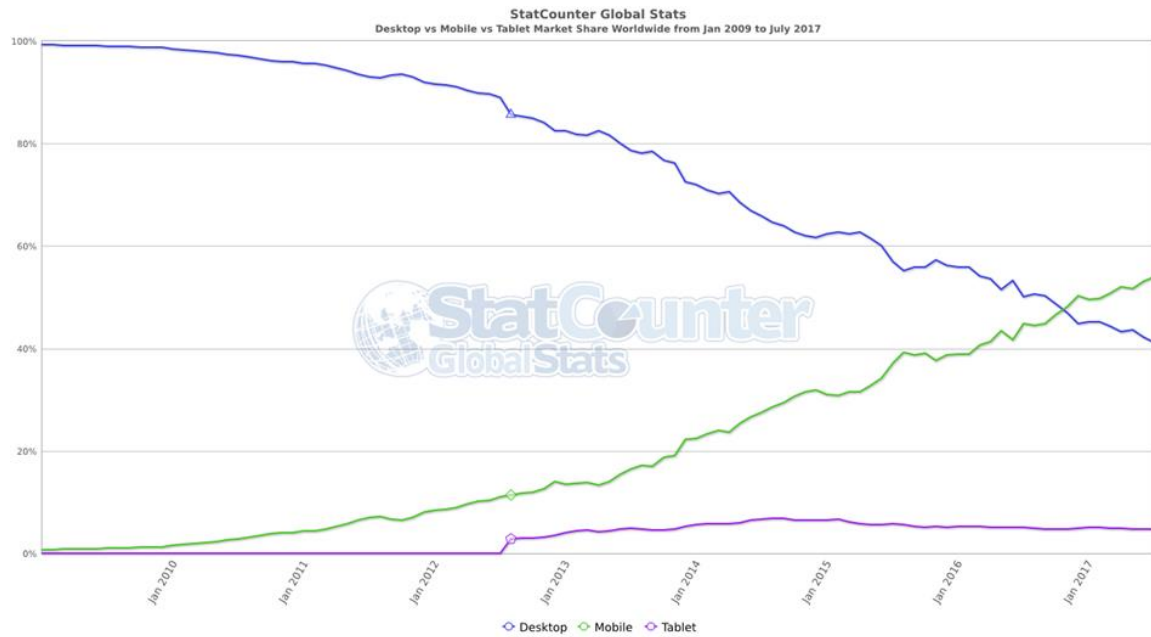
3 ans après l'arrivée du premier iPhone, Apple décide de bouleverser à nouveau le marché en proposant la tablette iPad. À mi-chemin entre un ordinateur portable et un smartphone, on reconnaît ce produit par son écran tactile d'une taille très proche à un ordinateur portable, mais en étant démunie d'un clavier et d'une souris. L'iPad a le même système d'exploitation et les mêmes capacités que l'iPhone. Bien évidemment, les premières tablettes Android sont rapidement arrivées par la suite.

Depuis l'iPhone premier du nom, les ventes de smartphones n'ont cessé d'augmenter. On ne dénombre rien que 1.47 milliard de livraisons de smartphone en 2016. Pour les tablettes, on compte 1,067 milliard d'appareils vendus depuis 2010, date de sortie du premier iPad.

3.2 Le trafic web

L'analyse du trafic web est un bon moyen de connaître la popularité de ces nouveaux supports. Depuis l'arrivée des smartphones, ces derniers ont toujours augmenté leur part de marché au détriment du PC. Pour la première fois en novembre 2016, les mobiles ont surpassé le PC. Concernant les tablettes, la part est restée basse, mais non négligeable.

Figure 1 : Répartition du trafic mondial sur le web, janvier 2009 à juillet 2017



(gs.statcounter.com)

Combinés ensemble, le PC et la tablette, qui ont des tailles d'écrans et des résolutions similaires, font quasiment part égale avec le mobile concernant le trafic mondial.

Tableau 1 : Répartition du trafic mondial sur le web, juillet 2017

Support	Pourcentage total
Mobile	54.08 %
PC	41.12 %
Tablette	4.8 %

(gs.statcounter.com)

Quant à la Suisse, le constat est différent. Le mobile est en dessous du PC. Ce dernier possède plus de la moitié du trafic web.

Tableau 2 : Répartition du trafic sur le web en Suisse, juillet 2017

Support	Pourcentage total
Mobile	37.79 %
PC	54.35 %
Tablette	7.85 %

(gs.statcounter.com)

Il est intéressant de noter que, à ce jour, Android et iOS ont le quasi-monopole des systèmes d'exploitation mobiles.

Tableau 3 : Répartition des OS mobiles concernant le trafic web, juillet 2017

OS	Pourcentage total
Android	70.27 %
iOS	22.49 %
Autres	7.24 %

(gs.statcounter.com)

3.3 Google

Google, plus grand moteur de recherche web et concepteur du système d'exploitation Android, a très bien compris l'importance de ces supports portables et a ainsi œuvré pour la satisfaction des clients utilisant ces appareils. Le 21 avril 2015, Google a décidé d'inclure l'optimisation mobile faite par un site dans son algorithme de recherche. Par conséquent, si un site web est adapté pour mobile, par exemple avec une version mobile ou un page s'adaptant au support de consultation, il sera alors propulsé vers les meilleurs résultats par rapport à un autre site qui n'est pas optimisé pour une navigation mobile. Le but de cette initiative est de motiver la conception orientée mobile des sites web existants et à venir.

Actuellement, Google souhaite aller encore plus loin en revoyant l'indexation des pages web pour prendre en compte seulement les pages web mobiles. Leur mesure du 21 avril 2015 prend en compte la version PC pour la recherche et vérifie si elle est adaptée pour une navigation mobile. Selon l'explication de Google, l'utilisateur peut du coup se retrouver sur le site mobile alors que l'information qu'il cherche n'y est pas disponible dessus, car elle n'est visible que sur la version PC. C'est pourquoi le moteur de recherche ne veut se baser que sur la version mobile d'un site, même pour les recherches effectuées depuis un PC.

3.4 Les marchés d'applications

Les systèmes d'exploitation des smartphones et des tablettes incluent des marchés d'applications. On retrouve l'App Store pour les appareils iOS d'Apple et le Play Store pour Android de Google.

Avec 149 milliards de téléchargements de produits applicatifs pour 61 milliards de dollars de chiffres d'affaires annuels en 2016², les marchés d'applications ont une importance capitale dans ces appareils mobiles. On compte 2.8 millions de programmes informatiques sur le Play Store contre 2.2 millions sur l'App Store.³

Des prévisions pour 2021 annonceraient le nombre de téléchargements d'applications à 352 milliards et des revenus à 139 milliards de dollars.⁴

3.5 Utilisation de ces nouveaux supports

Ces appareils ont des caractéristiques différentes par rapport à un ordinateur de bureau et offrent une nouvelle manière d'interagir avec un système informatique.

L'écran est une des principales nouveautés. Pour les smartphones, la taille des écrans est plus petite par rapport à un ordinateur personnel. Celle des tablettes est plutôt proche d'un PC. Mais ce qui change, c'est la navigation faite par le toucher de l'écran avec les doigts et non pas par un mouvement de souris. L'affichage prend aussi en compte l'orientation : si, par défaut d'utilisation, l'appareil est tenu à la verticale, il est possible de le tourner pour avoir un affichage à l'horizontale.

² http://files.appannie.com.s3.amazonaws.com/reports/1704_Report_Market_Forecast_FR.pdf

³ cf. note de bas de page n°2

⁴ cf. note de bas de page n°2

4. Courant de design digital

4.1 Skeuomorphisme

Avec la sortie de l'iPhone, Apple a décidé d'adopter le principe de skeuomorphisme pour ses interfaces graphiques. Cela se fera encore plus ressentir sur l'iPad.

Le skeuomorphisme, provenant du grec « skeuos » signifiant l'ornement, est un choix de design où, lorsque la fonctionnalité première d'une application rappelle un objet réel, l'interface et ses éléments vont être modélisés de sorte à reproduire l'aspect visuel de cet objet réel.

L'argument premier d'Apple d'adopter ce principe était de familiariser les consommateurs à la nouveauté d'interagir avec un système en touchant un écran. En adoptant un aspect visuel familier, l'utilisateur trouve directement ses repères dans une application, il comprend alors à quoi elle sert et comment elle fonctionne.

Figure 2 : Exemples de skeuomorphisme sur les premières versions d'iOS



<https://99designs.co.uk/blog/tips-en-gb/long-road-googles-material-design>

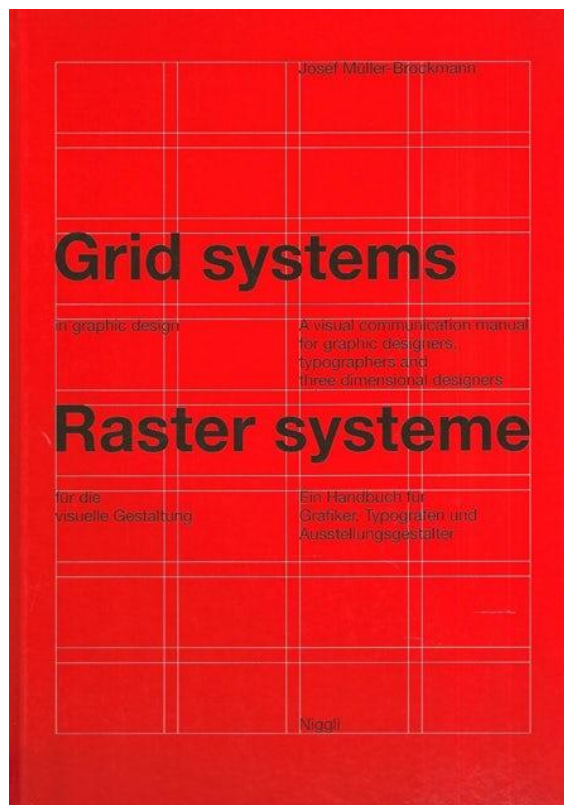
Sur les exemples ci-dessus, on constate que l'application « Notes » reproduit l'aspect visuel d'une feuille de note ainsi qu'une écriture manuscrite. Pour l'application « Enregistrement vocal », on retrouve un grand microphone, des boutons métalliques et un modulomètre analogique (indication du niveau de signal audio), rappelant une ancienne station d'enregistrement.

Un changement a commencé à s'amorcer. En effet, les consommateurs se sont habitués à comment utiliser une application tactile avec le temps. De plus, le skeuomorphisme a été de plus en plus critiqué : l'interface devenait trop chargée en détail et était trop compliquée à être réalisée.

4.2 Flat design

L'inspiration première de ce courant est le « style suisse ». Né dans les années 1940-1950, le « Swiss style design » porte l'attention sur la lisibilité, la netteté et la simplicité. Le principe est toujours le même : choisir une police d'écriture facile à lire (dites « Sans Serif ») et se baser sur une grille pour disposer des éléments.

Figure 3 : Exemple de poster de la période de création du style suisse



(<https://en.99designs.ch/blog/trends/skeuomorphism-flat-design-material-design>)

En 2012, Windows 8, système d'exploitation pour PC, est livré avec une nouvelle interface utilisateur adoptant le principe du flat design. Le changement est flagrant : les éléments d'interface sont disposés sur une grille et se retrouvent simplifiés en perdant leurs ombres, en adoptant une couleur unique et une forme basique.

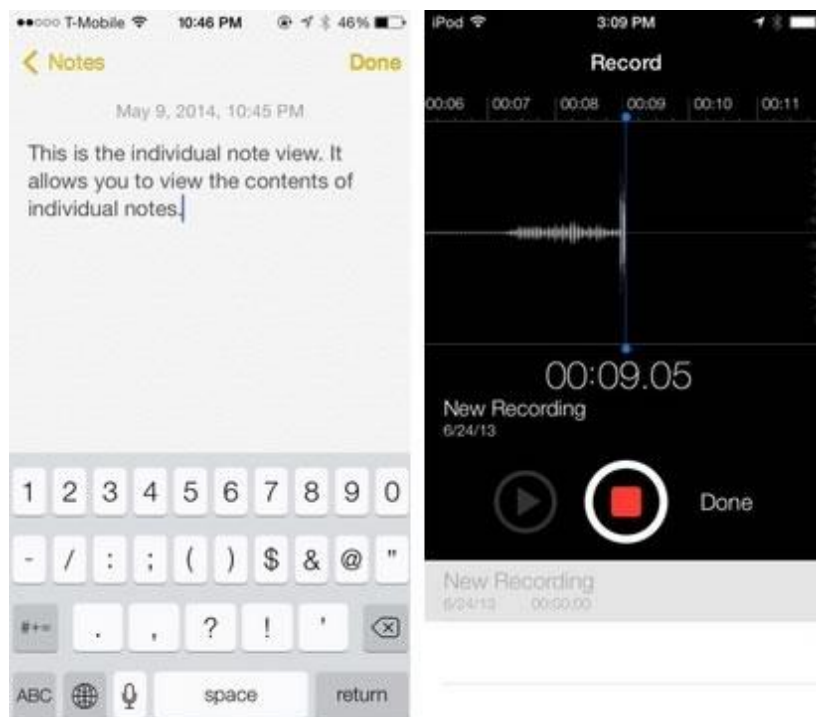
Figure 4 : Interface de Windows 8



<https://en.99designs.ch/blog/trends/skeuomorphism-flat-design-material-design>

En 2013, Apple décide d'abandonner le skeuomorphisme au sein de ses systèmes d'exploitation et d'adopter une approche « flat design » pour ses interfaces utilisateurs. La version 7 d'iOS apporte une géométrie simple et une palette de couleur limitée, qui sont des caractéristiques de l'approche « flat design ».

Figure 5 : Exemples de flat design à partir d'iOS 7

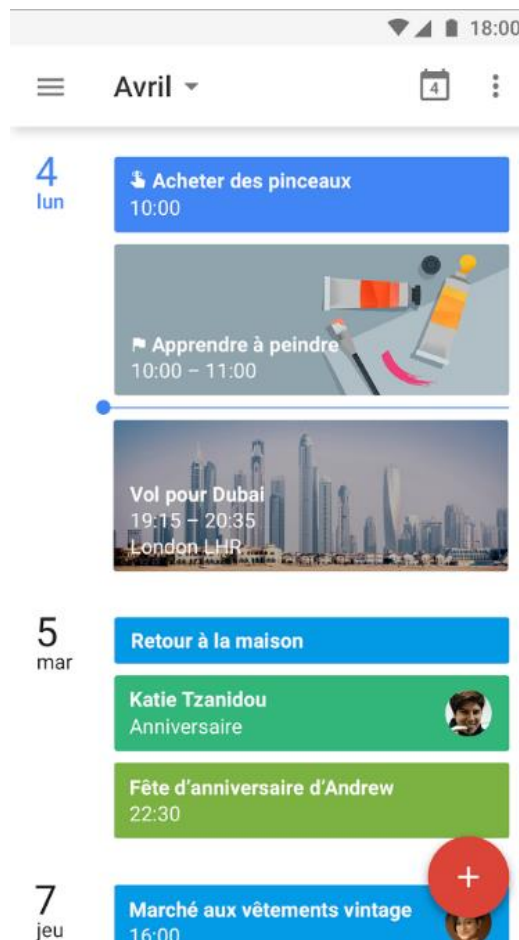


<https://99designs.co.uk/blog/tips-en-gb/long-road-googles-material-design>

En 2014, Google présente le Material Design. Android ayant connu plusieurs changements visuels depuis sa création, Google décide de concevoir une identité visuelle commune à ses produits.

Les aspects visuels sont très proches du flat design. Mais une attention est particulièrement portée sur la création d'un champ de profondeur. En effet, un élément va produire de l'ombre afin de donner une impression qu'il est devant d'autres éléments. En outre, un accent est mis sur l'animation des éléments, en prenant en compte qu'une animation doit avoir une signification. Tout ceci dans le but d'avoir une logique naturelle d'utilisation et de comportement dans une interface utilisateur.

Figure 6 : Exemple d'interface adoptant le Material Design de Google



<https://play.google.com/store/apps/details?id=com.google.android.calendar>

5. Pratiques actuelles

Plusieurs bonnes pratiques, servant à améliorer l'expérience utilisateur et à perfectionner les éléments des interfaces graphiques, se sont développées depuis l'arrivée des appareils mobiles.

Des documentations comme le Material Design de Google et l'Human Interface Guidelines d'Apple fournissent une multitude d'informations sur le design d'interface utilisateur pour les applications. Elles ont été créées dans le but d'expliquer comment offrir une expérience utilisateur cohérente et consistante.

Rapidement adoptées par les développeurs, ces bonnes pratiques sont maintenant utilisées sur les applications mobiles et web. D'autres conseils ont vu le jour à travers d'articles et de marches à suivre, jusqu'à devenir aussi des références pour la conception d'interface.

5.1 Tactile

Les supports mobiles ont, comme caractéristiques principales, des écrans tactiles qui permettent à l'utilisateur d'interagir directement avec le système en appuyant avec les doigts dessus.

De ce fait, la précision de pression sur l'écran est à revoir. En effet, le toucher de doigt a une zone de pression de 10mm sur 10mm, ce qui nécessite d'agrandir la taille de nos éléments interactifs afin d'être atteignable sans difficulté. Apple recommande une zone de 44 pixels sur 44 pour un confort optimal.

Figure 7 : Comparaison de la zone de pression d'un doigt avec une pièce équivalente à 0.50 CHF et une touche de clavier



<https://blogs.adobe.com/creativecloud/basic-patterns-of-mobile-navigation/>

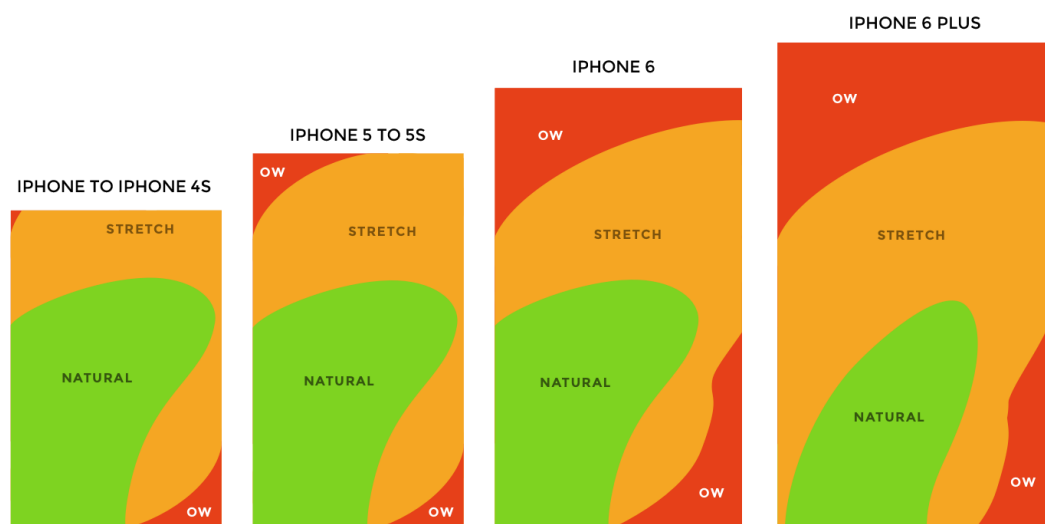
5.2 Taille d'écran des smartphones

Les fabricants de smartphones n'ont cessé d'agrandir la taille des écrans de leurs appareils. Les premiers iPhone avaient un écran de 3.5 pouces. Les 2 dernières versions

7 des iPhone, qui sont la version normale et la version Plus, mesurent respectivement 4.7 et 5.5 pouces. Pour les smartphones Android les plus populaires, la taille moyenne est entre 5.1 et 5.7 pouces.

Avoir des écrans aussi grands nécessite de repenser la structure même de l'interface utilisateur, car la taille des écrans s'agrandit, mais pas nos mains. En effet, le toucher sur l'écran se fait habituellement avec le pouce en tenant le téléphone d'une main. Dans ce cas de figure, on va retrouver plusieurs zones de confort d'utilisation sur l'écran. Si avant, cela n'était pas un problème du fait de la taille relativement petite d'un écran, c'est tout le contraire lorsqu'on est confronté à des écrans plus grands comme celui de l'iPhone 6, 6s ou 7 (4.7 pouces) et leurs versions Plus (5.5 pouces).

Figure 8 : Zone de confort d'utilisation des iPhone



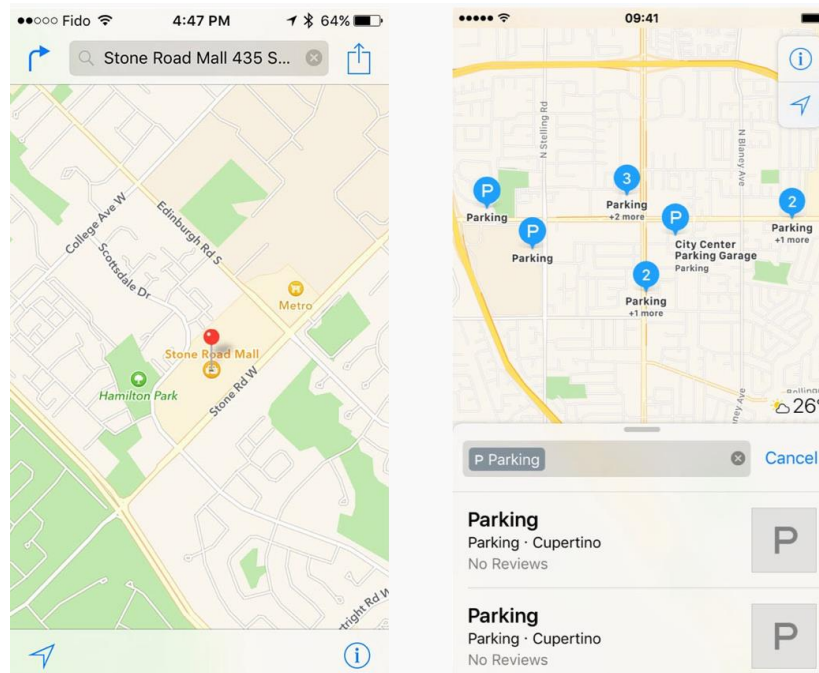
(<http://scotthurff.com/posts/how-to-design-for-thumbs-in-the-era-of-huge-screens>)

3 zones sont identifiables :

- La zone verte, dites la zone naturelle. C'est la plus confortable pour interagir.
- La zone orange, dites la zone d'étirement. Il faut déjà faire un petit effort avec le pouce pour atteindre cette zone.
- La zone rouge, qui est la zone où l'effort est plus difficile, car il faut bouger la main qui tient le téléphone pour pouvoir atteindre la zone avec le pouce.

Plusieurs applications ont déjà eu un changement dans la disposition des éléments, pour mettre les composants les plus utilisés dans la zone de confort optimale. Par exemple, la barre de navigation, placée de base en haut, commence à être déplacée vers le bas.

Figure 9 : Apple Maps avant et après iOS 10



(<https://uxdesign.cc/mobile-uis-are-turning-upside-down-90ba1cb049e0>)

En outre, Apple propose une autre solution, appelé « Reachability » ou « Accessibilité » en français. Cette fonctionnalité permet de descendre à la moitié inférieure de l'écran la totalité de l'affichage, afin d'accéder aux éléments présents initialement dans la zone rouge supérieure. Elle s'active à l'aide d'un double toucher sur le bouton d'accueil.

5.3 Boutons

Cet élément principal de l'interface utilisateur réagit différemment si l'on est sur PC ou sur les supports mobiles. En effet, sur PC, on parle de clic, car on interagit avec la souris. Du côté des appareils avec des écrans tactiles, on parle plutôt de pression avec le doigt. 4 états principales sont identifiables :

- L'état statique, c'est l'état de base. Cet état est visible sur PC et supports mobiles.
- L'état de survol, signifiant que le pointeur est au-dessus du bouton. Cet état n'est essentiel que sur PC, car il faut répondre à l'événement de survol par le pointeur de la souris, pour montrer que le bouton est cliquable.

- L'état de pression, indiquant que le bouton est actuellement cliqué / pressé, visible pour tous.
- L'état désactivé, où le bouton est momentanément impossible à cliquer / presser.

Figure 10 : Les 4 états d'un bouton



5.3.1 Floating Action Button

Composant issu du Material Design de Google et ayant connu un franc succès, le Floating Action Button est un bouton qui représente l'action principale du programme ou de la page actuelle de l'application. Il peut aussi afficher d'autres éléments comme des sous-actions.

Cet élément est facilement distinguable, car il a une forme ronde, une couleur particulière, une icône symbolisant l'action et, comme son nom l'indique, il « flotte » au-dessus de l'interface utilisateur grâce à sa position fixe et l'ombre qu'il produit.

Le bouton ne doit apparaître qu'en nombre très limité sur l'écran pour ne pas causer de confusion. Il est censé représenter une action positive comme la création ou le partage d'une ressource et non pas une action négative comme une suppression.

Figure 11 : Floating Action Button sur Twitter Android



5.4 Les gestes

L'interaction avec un écran tactile peut se faire de plusieurs manières. En effet, il est possible d'interagir à l'aide de divers gestes. Le but de ces derniers est d'alléger l'affichage en enlevant des éléments de navigation et d'offrir une interaction naturelle.

Mais, la courbe d'apprentissage d'utilisation de l'application augmente si on intègre plusieurs gestes. En effet, c'est à l'utilisateur de les découvrir pour interagir avec l'application. Pour qu'il ne soit pas perdu, il est bon de proposer des indications concernant les gestes réalisables.

C'est pourquoi Google et Apple recommandent d'utiliser des gestes standards. Les utilisateurs s'attendent à ce que ces gestes aient le même résultat à travers diverses applications.

5.4.1 Pression

Provoque la sélection d'un élément ou l'activation d'un élément.

5.4.2 Double pression

Le fait de tapoter rapidement 2 fois sur l'écran permet de zoomer ou de dézoomer sur une image.

5.4.3 Pression longue

Permet de rentrer dans un mode de sélection multiple d'éléments dans une liste. Dans une zone de texte, la pression longue provoque une sélection du mot pressée ou l'apparition d'une loupe sur iOS. Il permet aussi de déplacer un élément ou d'en afficher le menu contextuel.

5.4.4 Glisser le doigt vers une direction

Permet de défiler des éléments, se déplacer de page en page, de revenir en arrière ou d'afficher plus d'actions sur un élément.

5.4.5 Écarter / Pincer

Le fait d'écartier deux doigts permet de zoomer sur l'image alors que pincer permet de dézoomer.

5.5 La typographie

La typographie est l'élément le plus important lors de la conception d'une interface utilisateur. Il est nécessaire de rendre la lecture le plus agréable que possible. Pour cela, plusieurs bonnes pratiques existent.

Premièrement, un trop grand nombre de polices d'écriture différente dégage un aspect général non structuré. Il est bon de n'en garder que 3 au maximum. Google et Apple recommandent même de rester sur la police d'écriture de base de l'appareil sur lequel on se trouve : Roboto pour Android et San Francisco pour iOS.

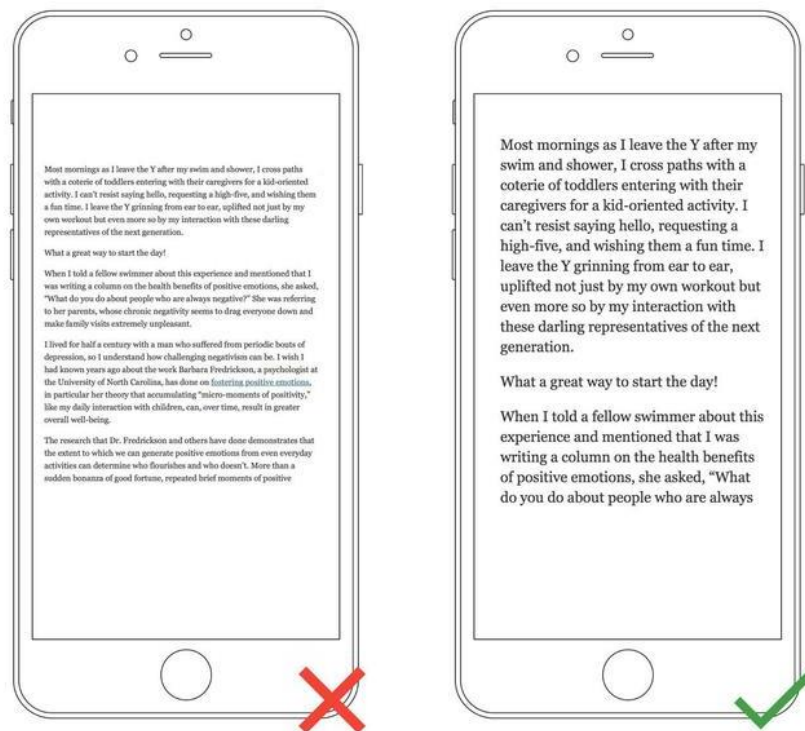
La taille minimale d'un élément textuel dépend du support de consultation :

- Pour les PC et tablettes, 16 pixels sont recommandés.
- Pour smartphones, il faut compter 11 pixels.

Il faut ensuite jouer avec la taille, l'intensité et la couleur pour différencier les zones de texte telles que les titres ou les boutons par exemple. C'est pourquoi il faut choisir une police d'écriture qui s'accorde bien à ces changements.

Limiter le nombre de caractères par ligne à 60 sur PC et à 30-40 sur appareils mobiles permet une lecture fluide, un rythme agréable et donc une fatigue amoindrie des yeux.

Figure 12 : Comparaisons du nombre de caractères par ligne sur appareils mobiles



(<https://medium.springboard.com/the-guide-to-getting-typography-right-in-digital-design-bb61214ff3ad>)

L'espacement entre les lignes est aussi important : il est recommandé d'utiliser 1.5x la taille du texte comme écart pour un confort de lecture optimal.

Le dernier élément à prendre en compte est le contraste. En effet, les appareils mobiles peuvent être utilisés à plusieurs endroits. Mais, la lumière en plein jour réduit la visibilité d'un écran et rend alors la lecture plus difficile. C'est pourquoi, pour faciliter le plus possible la lecture, il faut choisir une bonne combinaison de couleurs entre celle des caractères d'écriture et celle de fond. W3C recommandent un contraste d'un ratio de 4.5:1 au minimum entre une petite zone de texte et son fond. Pour les textes larges, comme ceux en gras ou ceux ayant une taille plus grande que la normale, un ratio de 3:1 est recommandé. Des outils⁵ existent pour mesurer le contraste.

5.6 Les icônes

Une icône est un symbole graphique représentant un objet, une idée ou une action. Elles sont présentes dans l'écran d'accueil d'un système d'exploitation mobile, sur le bureau des ordinateurs personnels et dans chaque coin des applications.

L'utilisation d'icônes dans les interfaces applicatives mobiles est devenue très courante :




- Une icône peut former un bouton assez grand pour être facilement pressée par un doigt sur un écran tactile tout en interagissant aussi bien avec une souris.
- Une icône est un symbole, elle communique plus rapidement une idée que du texte.
- De l'espace est gagné en affichant une icône, car elle est compacte, comparée à du texte.
- Une icône, si elle est bien choisie, est reconnaissable à travers toutes les cultures et toutes les langues, au contraire du texte où il faut le traduire pour toucher des utilisateurs internationaux.
- Une icône est un élément esthétique pouvant être plaisant à voir.

Mal employées, les icônes causent souvent des problèmes de compréhension à l'utilisateur. En effet, il faut tenir compte de plusieurs points lors du choix des icônes.

⁵ <http://webaim.org/resources/contrastchecker/>

Les icônes dites « universelles » sont rares. Ce sont les icônes que l'on peut utiliser sans apporter une précision textuelle complémentaire, car elles sont connues et reconnaissables par tout le monde. Par exemple, les symboles comme la loupe et l'imprimante font directement penser à la fonction de recherche et d'impression. Le fait est qu'une icône, pour qu'elle devienne universelle, doit toujours être utilisée pour la même fonction.

Figure 13 : Signification des icônes iOS par Apple

	Favorites	Shows the person's favorite items.
	Featured	Shows content featured by the app.
	Top Rated	Shows the highest-rated items.

(<https://developer.apple.com/ios/human-interface-guidelines/graphics/system-icons/>)

Dans cet exemple de la ligne directrice des interfaces rédigée par Apple, l'icône de l'étoile a 3 significations différentes. Cela prouve qu'il est difficile d'avoir une définition universelle pour une icône.

Pour régler ce problème, il suffit d'utiliser un label textuel comme complément à l'icône. Mais, en contrepartie, on perd la compacité qu'on avait en employant seulement l'icône. Un test simple, appelé « la règle des 5 secondes », permet de prendre la décision d'accompagner ou non une icône par une précision écrite. Si trouver une icône pour la signification de quelque chose prend plus de 5 secondes, alors il est recommandé d'apporter un complément textuel.

L'aspect visuel d'une icône change d'un OS à un autre. Sur Android, les lignes d'une icône sont plus épaisses ou l'icône est même pleine, alors que sur iOS, on utilise que des icônes faites de lignes fines.

Figure 14 : Exemples d'icônes : Android à gauche et iOS à droite



(<https://webdesign.tutsplus.com/articles/a-tale-of-two-platforms-designing-for-both-android-and-ios--cms-23616>)

De plus, 2 icônes importantes changent complètement leur aspect sur chaque OS. Sur Android, l'icône « retour » est une flèche pointant vers la gauche. Sur iOS, l'icône de la flèche perd la tige pour ne garder que la direction. L'icône « plus d'options », représentée par 3 points, est orientée à la verticale sur Android et à l'horizontale sur iOS.

5.7 Les animations

Les animations permettent d'apporter des indications supplémentaires de l'état de l'application à un utilisateur.

Elles informent sur le statut actuel de l'application (si elle est en chargement par exemple) et donnent un retour par rapport à une action effectuée par un utilisateur, comme le fait de montrer une transition de « glissement » lors d'un changement de page.

Elles ont pris une place importante dans les interfaces utilisateurs, car un élément peut alors prendre vie et avoir un comportement naturel. Elles facilitent la compréhension d'une application et peuvent avoir un aspect visuel très plaisant. Bien évidemment, une animation ne doit pas distraire un utilisateur des informations affichées à l'écran.

Google va plus loin en définissant certains critères :

- Une animation se doit d'être rapide pour ne pas impatienter l'utilisateur. Une animation normale sur smartphone dure, en moyenne 300 millisecondes. Une animation sur tablette dure 390 millisecondes, ce qui représente 30 % de plus que sur smartphone. Quant au PC, il faut se baser sur la distance de déplacement d'un élément et animer de sorte que ce soit rapide et naturel.
- Plusieurs rythmes d'animations existent, mais le timing recommandé est quand une animation prend du temps à atteindre sa vitesse maximum et ralentie progressivement à la fin. On parle alors de « ease in out » ou d'atténuation au début et à la fin.
- Le terme « chorégraphie » désigne plusieurs éléments animés en même temps. Cette animation doit être claire, simple et cohérente avec l'affichage actuel pour ne pas créer de confusion.

5.8 Menus de navigation

Les menus de navigation sont communs à toutes applications : ils permettent d'accéder à d'autres pages ou fonctionnalités qu'un programme propose. Sur les supports mobiles,

ils ont été adaptés de plusieurs manières pour faire face à la contrainte de l'espace limité sur les petits écrans.

5.8.1 Barre principale

La barre principale se place en général en haut de l'affichage. Elle comprend habituellement le logo de l'application ou le titre de la page actuelle. On y trouve souvent la fonctionnalité de recherche et d'autres actions principales. C'est aussi sur cette barre que l'on va retrouver l'accès au tiroir de navigation. Selon où on se situe dans l'application, le contenu de la barre principale change.

Figure 15 : Exemple de barre principale



<https://material.io/guidelines/layout/structure.html>

L'action de pouvoir revenir à la page précédente est positionnée à gauche. Sur iOS, il est nécessaire d'avoir cette action présente dans la barre principale, contrairement à un appareil Android, qui est doté d'une touche « retour » faisant partie à part entière du système d'exploitation.

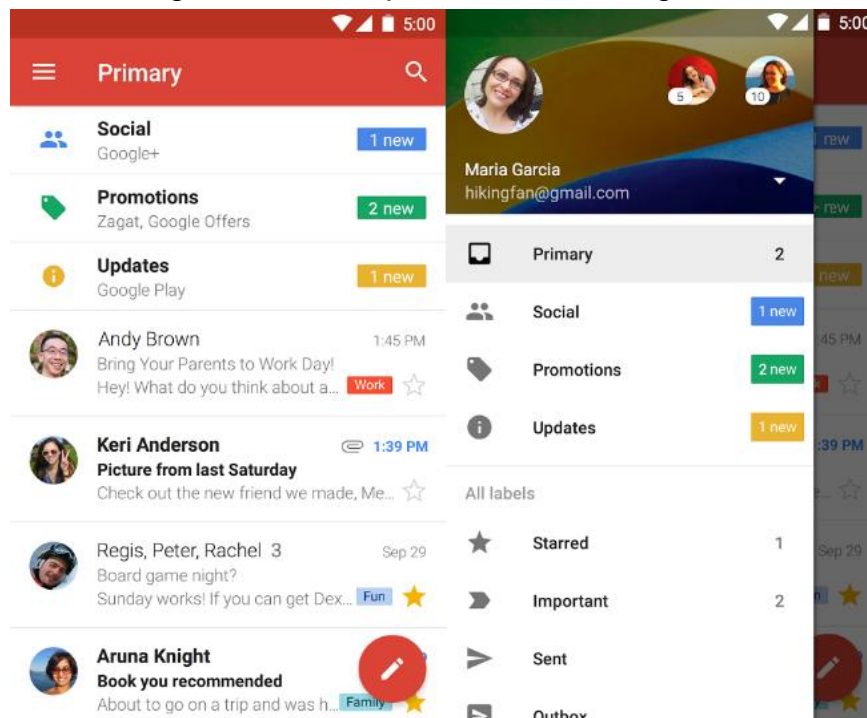
5.8.2 Tiroir de navigation

Le tiroir de navigation a été mis en avant par Google dans le Material Design. Comme l'espace est très limité sur les écrans mobiles, le tiroir de navigation, appelé aussi le tiroir latéral, est devenu un élément de navigation populaire. En effet, les fonctionnalités ne sont pas affichées directement sur l'écran. Elles sont présentes sous forme de liste dans un composant qui s'affiche au-dessus de la page actuelle lors de l'appui sur l'icône représentant ce menu de navigation.

Appelé péjorativement « hamburger menu » à cause de la ressemblance de l'icône avec un hamburger, il est aussi très critiqué, car il cache les fonctionnalités de l'application à l'utilisateur. En outre, il faut effectuer une action de plus, c'est-à-dire cliquer sur le bouton du menu, pour accéder à une fonctionnalité. Il est à noter que cette icône est devenue la représentation standard de ce composant pour les utilisateurs au sein des applications.

Il est aussi utilisé sur les applications web de bureau. Même si un PC offre plus d'espace d'affichage qu'un mobile, le tiroir de navigation permet de cacher des fonctionnalités et ainsi de réduire le nombre d'informations affichées à l'écran.

Figure 16 : Exemple de tiroir de navigation



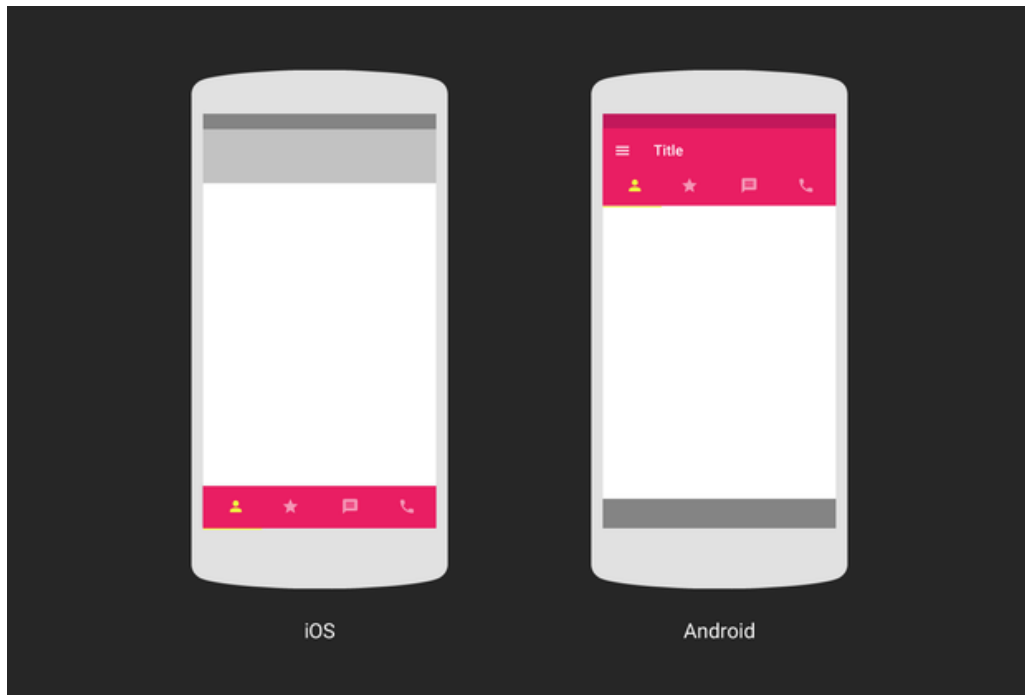
<https://play.google.com/store/apps/details?id=com.google.android.gm&hl=fr>

5.8.3 Barre d'onglets

Dans la barre d'onglets, on retrouve les destinations les plus importantes de l'application. Présente dans n'importe quelle page, elle permet de facilement changer de fonctionnalité dans une application et de savoir dans quelle page on se situe. Du fait de l'espace limité sur l'écran mobile, il est conseillé de ne pas y proposer plus de 5 éléments.

Les bonnes pratiques de design recommandent un emplacement différent pour la barre d'onglets selon le système d'exploitation mobile. Sur iOS, ce composant est placé en bas de l'écran, alors que sur Android, elle se place en haut.

Figure 17 : Recommandations de placement de la barre d'onglets



(<https://design.google/library/design-ios-android-and-back-again/>)

5.9 Disposition

La disposition des éléments d'une interface graphique permet d'indiquer une hiérarchie et une signification pour l'utilisateur. Ce dernier peut alors mieux comprendre ce qui est affiché à l'écran et profiter pleinement du contenu offert.

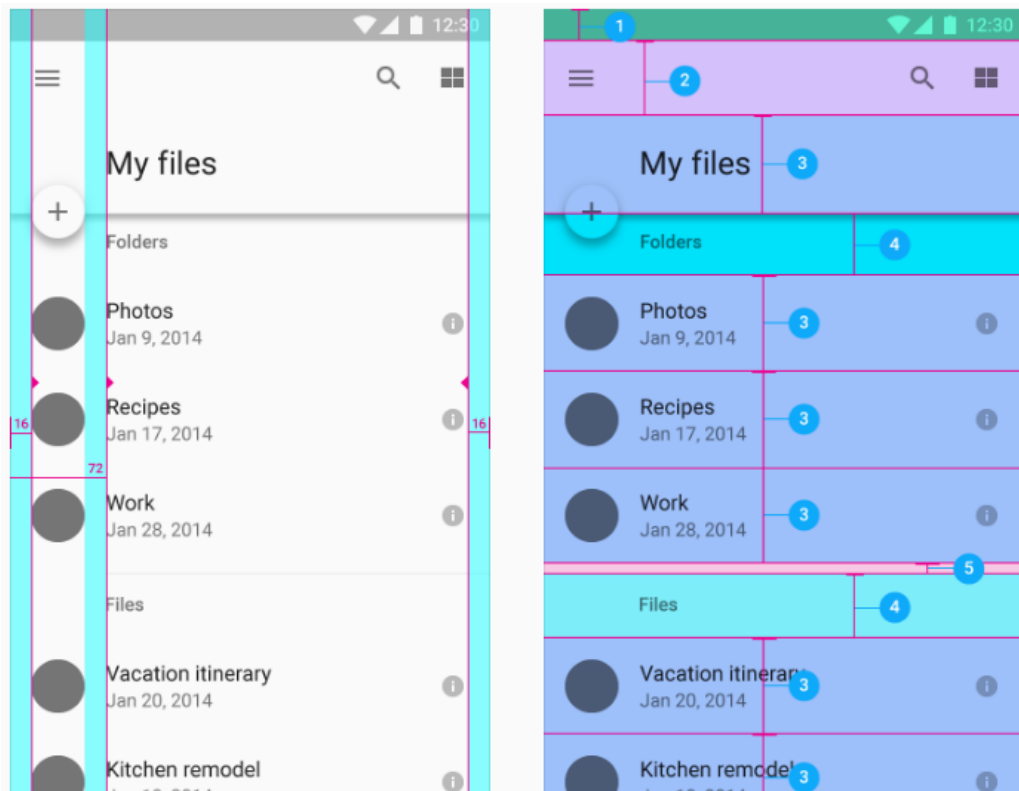
Plusieurs éléments cités auparavant ont des contraintes liées à leur disposition. C'est grâce à leur emplacement que ces composants prennent leur sens.

5.9.1 Espacement

L'espacement entre éléments crée de la consistance et de la cohérence dans une interface utilisateur et offre une meilleure lisibilité du contenu. En effet, les écarts permettent de former une hiérarchie, de constituer des groupes d'éléments, de mettre en valeur un composant et bien plus encore.

Pour espacer les éléments de manière constante, le système proposé par Google est efficace : il suffit de se baser sur une grille de placement de 8 pixels. Ce qui veut dire que chaque espacement, marge et hauteur ou longueur d'un élément a une taille qui est un multiple de 8.

Figure 18 : marges et alignement vertical se basant sur des multiples de 8



<https://material.io/guidelines/layout/metrics-keylines.html#metrics-keylines-keylines-spacing>

En utilisant ce système, il suffit ensuite de placer les éléments de sorte qu'ils soient en harmonie entre eux. Dans l'exemple ci-dessus, on reconnaît les types d'éléments (ceux qui ont les mêmes numéros dans l'image de droite) comme les listes ou les sous-titres selon leur rapprochement et leur espacement régulier. Ainsi, notre œil analyse naturellement les informations affichées à l'écran.

5.9.2 Orientation

Les appareils mobiles peuvent être tenus de 2 manières différentes : à la verticale, mais aussi à l'horizontale. C'est pourquoi il est recommandé d'offrir la possibilité d'utiliser une application dans les 2 sens. De plus, il faut que le changement d'orientation garde une disposition similaire dans les 2 sens pour que l'utilisateur s'y retrouve.

6. Développement des interfaces utilisateurs depuis l'arrivée des appareils mobiles

6.1 Développement des applications natives avec les outils officiels

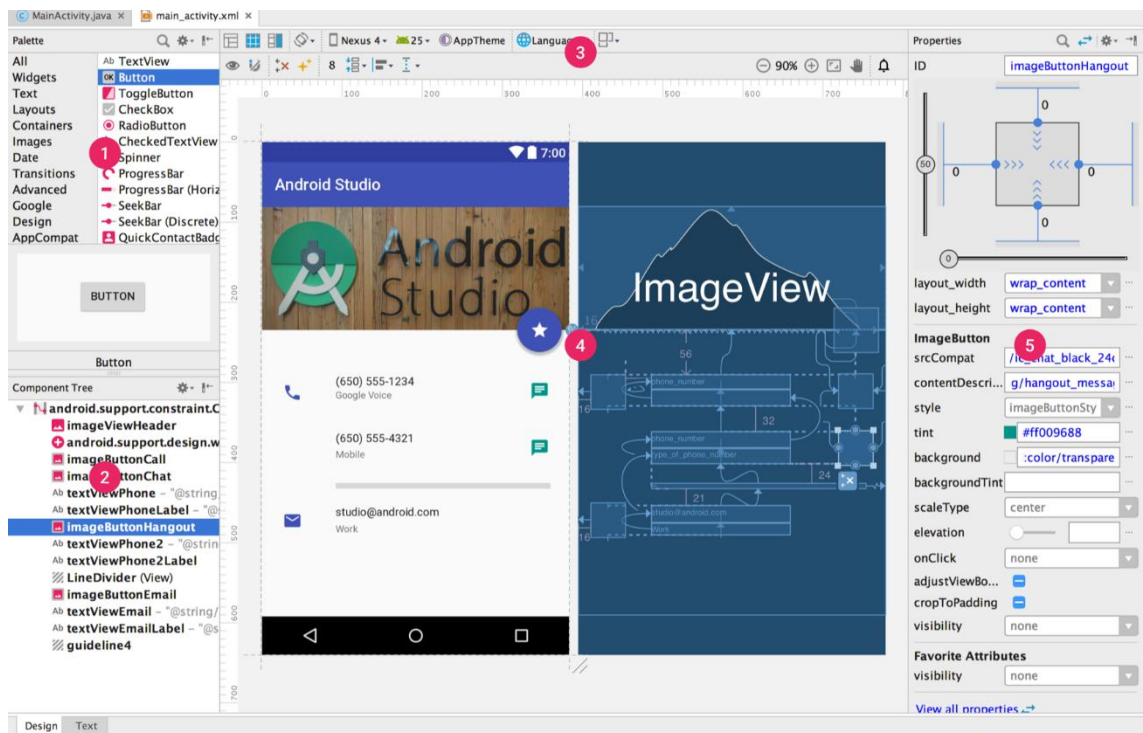
Le développement d'une application native procure un avantage certain face à une application web : on peut utiliser toute les fonctionnalités offertes par l'appareil en question comme la pleine puissance du processeur par exemple. De plus, elle se retrouve sur le marché des applications, qui est un endroit avec une exposition importante.

En contrepartie, si on souhaite viser les 2 OS mobiles principaux, il faut alors développer une application spécifique pour chaque support, car chaque système d'exploitation utilise un langage de programmation différent. Google et Apple offrent aux développeurs un IDE pour leurs OS. Google propose Android Studio et Apple propose Xcode.

Ces IDE permettent de créer des interfaces utilisateurs de manière visuelle. En effet, chaque IDE propose de glisser-déposer des composants dans une page afin de concevoir facilement une interface.

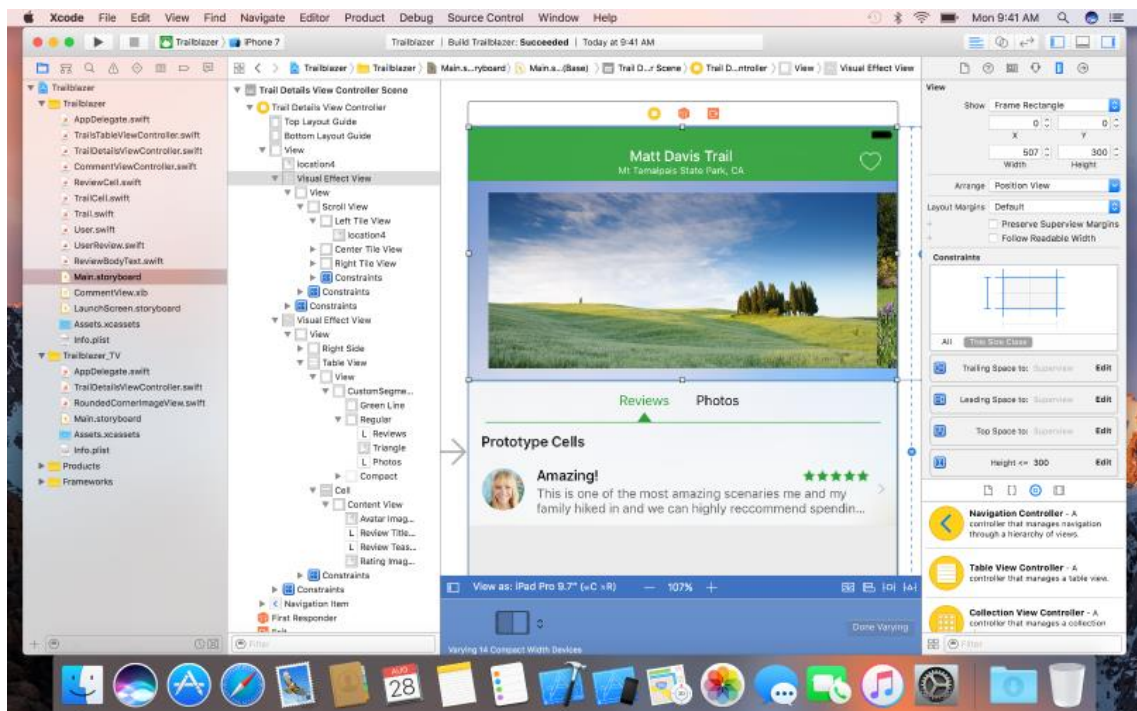
Chaque élément a des propriétés éditables, semblables à ce qui est utilisé en CSS. Il suffit de les changer pour arriver au résultat escompté.

Figure 19 : Développement d'une interface pour Android



<https://developer.android.com/studio/write/layout-editor.html>

Figure 20 : Développement d'une interface pour iOS



<https://developer.apple.com/xcode/interface-builder/>

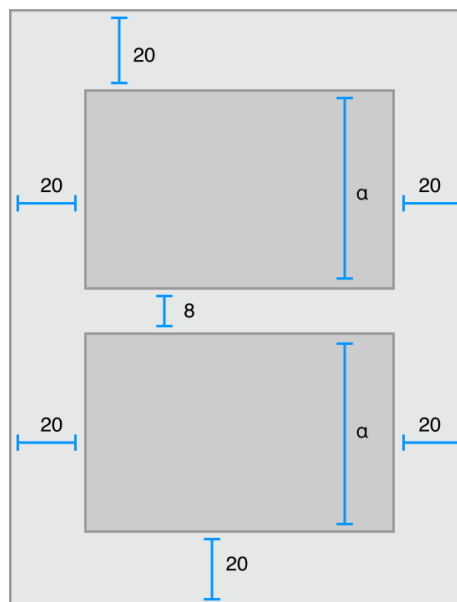
6.1.1 Affichage adaptatif

Une des parties les plus importantes lors de la conception d'une interface est la disposition des éléments. Les applications natives pour les OS mobiles doivent supporter plusieurs tailles d'écrans. En effet, il faut prendre en compte tous les smartphones et toutes les tablettes. De plus, l'orientation d'affichage ne doit pas être ignorée si on souhaite offrir une expérience utilisateur idéale.

Pour résoudre ces problèmes, des grilles de disposition (layout en anglais) sont disponibles sur Android et iOS : ConstraintLayout et AutoLayout respectivement.

Le principe est le même pour les deux : chaque élément peut se voir attribuer des règles appelées « contraintes ». Une contrainte représente un lien ou un alignement entre un composant et un autre. En bref, c'est la position d'un élément par rapport à son environnement. Par exemple, une règle peut définir qu'un élément s'étire en fonction de la résolution d'affichage ou qu'il reste à une distance spécifique du bord gauche de l'écran.

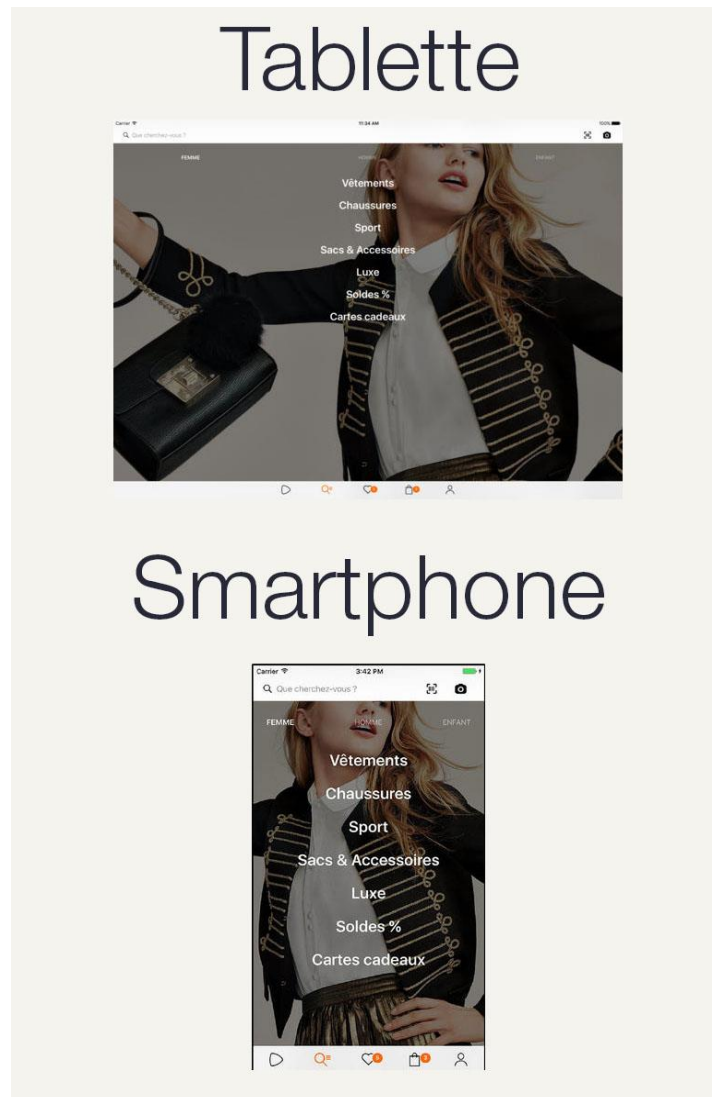
Figure 21 : Disposition d'éléments pour une interface adaptative



(<https://developer.apple.com/library/content/documentation/UserExperience/Conceptual/AutolayoutPG/index.html>)

Dans cet exemple, on peut voir que sur chaque affichage, les 2 éléments resteront ancrés par rapport aux bords de l'écran. Ils seront liés par une distance fixée entre les 2 et ils partageront la même longueur.

Figure 22 : Exemple de disposition avec l'application Zalando



<https://itunes.apple.com/ch/app/zalando-mode-shopping-und-fashion-trends/id585629514?mt=8&v0=WWW-EUCR-ITSTOP100-FREEAPPS&l=fr&ign-mpt=uo%3D4>

Sur cet exemple de l'application Zalando, on repère différentes contraintes :

- La barre principale voit la zone de recherche fixée à gauche et les boutons de fonctionnalités fixés à droite.
- Les catégories humaines sont alignées au centre et séparées par la même longueur entre elles. Elles ont une distance fixe par rapport à la barre principale et les types d'habits.
- Les éléments des types d'habits et de la barre d'onglets ont une distance fixe entre elles.
- Les types d'habits et la barre d'onglets ont un alignement central.

6.2 Web Design

Avec l'arrivée des smartphones et tablettes, la première solution en Web Design a été les sites dédiés pour chacun de ces supports. Bien que cette solution soit encore utilisée de nos jours, cette méthode de conception duplique le code source. Il faut alors passer beaucoup de temps pour concevoir et maintenir à jour chaque version. C'est pourquoi une solution plus robuste et plus efficace a vu le jour.

6.2.1 Responsive Web Design

Le Responsive Web Design est un ensemble de pratiques appliqué sur une page web servant à promouvoir une navigation confortable pour les smartphones, les tablettes et ordinateurs de bureau.

En effet, le but est d'avoir l'interface d'une page web qui s'adapte à l'appareil consultant la page afin d'éviter une mauvaise expérience utilisateur, telle que le défilement horizontal ou le besoin de zoomer pour mieux voir un élément.

6.2.1.1 CSS : Media Queries

Recommandées depuis le 19 juin 2012 par la W3C, les Media Queries sont des règles CSS qui permettent de viser un type de média avec des conditions particulières. En définissant ces règles dans une page web, on peut ainsi s'adapter à un support spécifique qui consulterait la page.

Plus précisément, les Media Queries permettent de sélectionner plusieurs types de médias tels qu'un écran, une impression, une télévision, un appareil auditif, un appareil de synthèse vocale, un appareil de lecture Braille et bien plus. On peut ensuite restreindre un média en définissant des conditions telles que la résolution d'un média, le ratio d'affichage, l'orientation d'écran ou les couleurs d'affichage disponibles.

Finalement, dans les Media Queries, on y écrit les contraintes de style voulues pour le support visé.

Figure 23 : CSS – Media Queries

```
@media only screen and (min-width: 1025px) {  
  /* Règles pour PC */  
}  
  
@media only screen and (min-width: 768px) and (max-width: 1024px) {  
  /* Règles pour tablette */  
}  
  
@media only screen and (max-width: 767px) {  
  /* Règles pour smartphone */  
}
```

Dans cet exemple, on retrouve 3 sélections : chacune cible seulement les écrans et définit des conditions relatives à la largeur de l'écran, pour préciser quel support est visé. Du coup, il est possible d'écrire des règles pour un smartphone, pour une tablette ou pour un PC en utilisant ces 3 Media Queries. On pourrait aller encore plus loin en précisant l'orientation pour améliorer le confort de l'utilisateur.

6.2.1.2 Grille de disposition

Un des éléments pratiques du Responsive Web Design est la grille de disposition. Les « grids » sont des conteneurs d'éléments, servant à la disposition des composants.

Une grille comporte des lignes qui contiennent des colonnes. Chaque ligne occupe toute la largeur disponible. Chaque colonne occupe une largeur différente selon les supports définis. Une ligne peut alors contenir un certain nombre de colonnes. Lorsque ce nombre de colonnes est atteint, les colonnes en surplus sont renvoyées à la ligne d'en dessous.

Tout l'intérêt de ce système est de construire une disposition qui s'adapte pour chaque affichage, sans devoir fixer des hauteurs et des largeurs pour chaque composant. En effet, les éléments prendront la largeur de la colonne qui les contient. Plusieurs types de colonnes sont possibles afin de pouvoir former des dispositions complexes. Ensuite, il suffit de fixer une largeur aux colonnes qui change selon le support de consultation.

Figure 24 : Exemple visuel de grille de disposition



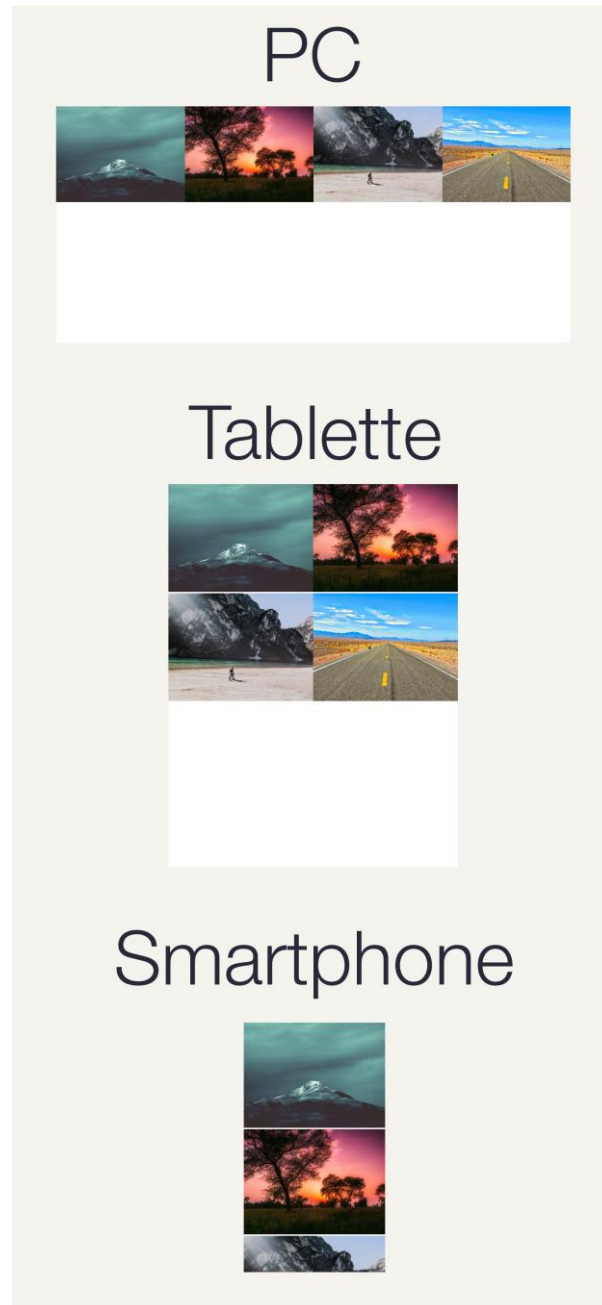
(https://www.slideshare.net/sajed_shaikh/responsive-web-design-bootstrap-and-media-queries)

Dans cette illustration, nous voyons que sur le PC, chaque ligne contient 4 colonnes : la largeur de ces dernières est de 25 % chacune. Pour la tablette, la largeur est de 50 % pour une colonne. Finalement, la largeur d'une colonne sur smartphone est de 100 %.

6.2.1.3 Exemple

Afin de comprendre ces deux concepts ci-dessus, voici un exemple minimaliste d'affichage de plusieurs images.

Figure 25 : Exemple de Responsive Web Design



Sur un écran de PC, on retrouve 4 images par ligne, car chaque image occupe 25 % de largeur d'écran. La taille d'écran se réduisant, on tombe sur l'affichage pour tablette : 2 images par ligne, car chaque image prend 50 % de largeur d'écran. Pour finir, une image occupe toute la largeur de l'écran dans l'affichage pour smartphone.

Figure 26 : Règles d'affichage de l'exemple

```
/* Règles communes aux 3 supports */
.une-ligne {
  width: 100%; /* Une ligne occupe toute la largeur de l'écran */
}

/* Règles pour PC */
@media only screen and (min-width: 1025px) {
  .une-colonne {
    width: 25%; /* Une colonne prend le quart de la largeur de la ligne */
  }
}

/* Règles pour tablette */
@media only screen and (min-width: 768px) and (max-width: 1024px) {
  .une-colonne {
    width: 50%; /* Une colonne prend la moitié de la largeur de la ligne */
  }
}

/* Règles pour smartphone */
@media only screen and (max-width: 767px) {
  .une-colonne {
    width: 100%; /* Une colonne prend toute la largeur de la ligne */
  }
}
```

7. Nouvelles méthodologies de développement

7.1 Mobile First

L'approche « Mobile First » part du principe que le design est plus dur à réaliser sur une petite taille d'écran que sur une grande. C'est pourquoi il faudrait commencer par concevoir les interfaces applicatives pour les smartphones et réaliser pour les autres supports dans l'ordre croissant de la taille des écrans, c'est-à-dire les tablettes et les PC.

Comme on est restreint par la petite taille de l'écran, on ne va y inclure que les éléments essentiels. Du coup, on se retrouve avec un design réfléchi pour un téléphone, doté d'une expérience utilisateur optimisée, qu'il suffit d'adapter aux tablettes et ensuite aux PC. Il est facile alors de rajouter des éléments au fur et à mesure que l'espace d'affichage s'agrandit.

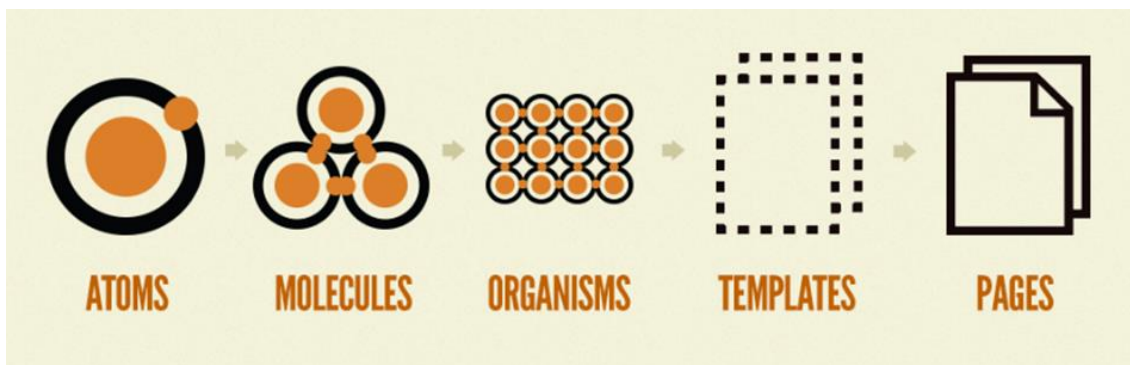
C'est à l'opposé de l'approche appelée « Graceful Degradation » ou dégradation progressive, où on se retrouve en tout premier avec une version pensée pour le PC, qui est très complète avec tous les composants essentiels et optionnels. Ensuite, cette version est réduite en enlevant certains éléments pour tenir sur un écran d'affichage plus petit. Le problème est qu'il est difficile de distinguer entre tous ces éléments ce qui est essentiel à une expérience utilisateur. Au final, on se retrouve avec une version qui n'est pas forcément adaptée à un smartphone, car elle a subi plusieurs réajustements depuis la version initiale.

L'approche Mobile First place le contenu en priorité dans le design de l'interface applicative. En effet, l'utilisateur utilise une application pour le contenu offert. On adopte alors une vision centrée sur le besoin de l'utilisateur. Après avoir défini comment le contenu se présente sur un affichage réduit, il est plus facile de l'adapter à un écran plus grand qu'à l'inverse, l'adapter à un écran plus petit.

7.2 Atomic Design

Avec tous les supports à prendre en compte lors de la conception d'un produit, garder une consistance et une continuité en général établit une identité à l'application et permet à l'utilisateur final de se repérer d'un support à un autre. Le développement est aussi accéléré grâce à la reprise des composants sur les divers appareils visés. Une méthodologie intéressante pour la conception du design d'une application est l'approche atomique ou « Atomic Design » en anglais. Cette méthodologie consiste à décomposer les éléments visuels de l'interface. Selon cette approche, on peut distinguer 5 niveaux en termes de design.

Figure 27 : Atomic Design : les 5 niveaux



<http://bradfrost.com/blog/post/atomic-web-design/>

L'intérêt d'une telle approche est de définir des éléments plus abstraits afin de former des éléments plus concrets. Cela prend toute son importance lorsque l'on développe sur plusieurs supports. De cette manière, on garde une consistance et une cohésion forte et on compose plus rapidement des interfaces spécifiques pour chaque support grâce aux composants réutilisables.

7.2.1 Atomes

Un atome est un élément basique et indécomposable. Il représente un simple élément visuel : cela peut être un champ de texte, un label, ou un bouton par exemple. La police d'écriture, les couleurs ou même les animations sont aussi comprises dans cette catégorie. En soit, un atome seul est inutile, car il est trop abstrait pour avoir un sens. Le fait de les définir permet d'établir une cohérence par la suite dans tout le produit.

7.2.2 Molécules

En rassemblant plusieurs atomes, on obtient une molécule. Cette unité gagne déjà en utilité et forme un bloc fondamental et réutilisable pour notre design. En reprenant nos exemples d'atomes, on peut composer un formulaire de recherche avec un label, un champ de texte et un bouton.

7.2.3 Organismes

Si on met ensemble plusieurs molécules, cela forme un organisme. On reconnaît un organisme par sa complexité. Il représente une section distincte de l'interface. Par exemple, un en-tête est un organisme où on retrouve souvent un formulaire de recherche. Du coup, l'organisme « en-tête » pourrait avoir un logo, un menu de navigation et notre exemple de molécule de formulaire de recherche.

7.2.4 Modèle

Une fois les organismes mis bout à bout, on se trouve avec un modèle. Ce dernier est une représentation d'une page, mais vierge de toutes informations. Il permet d'avoir une idée de ce que ressemble une page avec tous ces éléments regroupés ensemble. Par exemple, si notre application est un site de commande d'habits, on aura un modèle de présentation de produit. Ce modèle de page pourra donc être réutilisé pour tous les habits possibles.

7.2.5 Page

Une page est un modèle avec du contenu réel : c'est le rendu final avec lequel l'utilisateur interagira. On peut tester si les éléments s'affichent correctement et si le résultat est satisfaisant.

8. Analyse de l'interface utilisateur d'une application

Pour cette analyse, l'application choisie est Instagram, qui est un réseau social de partage de photos et de vidéos qui a vu le jour en 2010. En 2012, Facebook rachète Instagram pour 1 milliard de dollars américains, mais décide de laisser l'application en tant que telle et ne pas la rattacher à ses services. Instagram compte 700 millions d'utilisateurs actifs en avril 2017.

L'analyse se focalisera sur la version Android, iOS et la version web. Il est à noter que l'affichage de l'application web et native sur smartphone et sur tablette est identique sur la plupart des points. C'est pourquoi ces affichages sont regroupés, selon si on parle en termes d'OS ou en termes de supports.

L'analyse se fera en utilisant l'approche « Atomic Design », expliquée précédemment dans ce mémoire. Nous verrons 4 catégories possibles d'éléments : atomes, molécules, organismes et modèles de page. En effet, la catégorie « page » n'apporte rien de plus dans l'analyse.

8.1 Analyse atomique

8.1.1 Atomes

8.1.1.1 Icônes

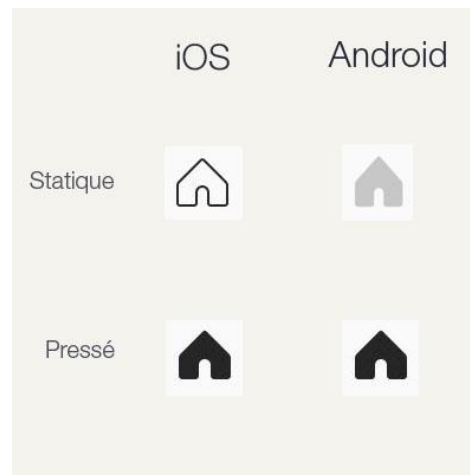
Figure 28 : Icônes utilisées par Instagram



Les icônes sont, à quelques détails près, toutes semblables d'une plateforme à une autre.

Pour la version Android, Instagram a pris en compte les bonnes pratiques concernant les icônes et a décidé d'utiliser une version des icônes pleine ou avec des lignes plus grosses. Mais, cette variante d'icônes est aussi affichée sur iPhone : lors qu'un bouton d'icône est pressé / actif, ce sont ces icônes plus marquées qui représentent l'état pressé. Pour l'état statique de ces boutons sur Android, Instagram a utilisé la même icône, mais avec une couleur plus grisâtre, pour montrer que l'état n'est pas pressé / actif.

Figure 29 : Exemple des différences d'icônes



De plus, les icônes « retour en arrière » et « plus d'options » respectent bien les spécificités liées à chaque plateforme.

Les icônes propres à une seule version sont celles de la boussole (n'apparait que sur PC) et de l'ajout d'ami (n'apparait que sur la version mobile web). L'icône de la boussole représente la fonctionnalité « Découvrir ». Cette dernière est accessible sur les versions mobiles à l'aide de l'icône de la loupe. Sur mobiles, cette fonctionnalité permet d'accéder à une barre de recherche ainsi que d'avoir des propositions de contenus. C'est pourquoi ce changement sur la version web PC s'explique par la présence en permanence de la barre de recherche, et de ne donc pas utiliser une icône de la loupe pour la fonctionnalité « Découvrir ».

Le logo de la marque est le même sur les versions mobiles, mais, sur la version PC, le logo change.

Pour le reste des icônes sur la version web, elles proviennent de la version iOS.

8.1.1.2 Typographie

Figure 30 : Différentes zones de texte utilisées sur Instagram



Les différents types de zones de textes trouvés sont tous égaux sur chaque support en termes de taille, d'intensité et de couleur.

Les bonnes pratiques concernant le choix de la police d'écriture sont respectées : selon le système d'exploitation, la police d'écriture recommandée est utilisée. Il est impossible de savoir la taille exacte de la typographie sur l'application native. Mais sur web, la taille de la police est de 14 pixels et elle ne change pas en fonction du support de consultation. Du coup, cette taille est dans la norme sur mobile, mais ne l'est pas sur PC. Cela n'est relativement pas grave, car le contenu principal est les photos et les vidéos et non pas du texte.

8.1.1.3 Autres éléments

On retrouve un élément qui correspond à une photo ou une vidéo.

Sur version web PC, une zone de saisie existe, servant pour effectuer une recherche.

Quant aux couleurs, Instagram utilise principalement du blanc et du noir avec quelques tons en gris, un dégradé violet-rouge-orange sur quelques éléments et une touche de bleu.

8.1.2 Molécules

8.1.2.1 Barre principale

Figure 31 : Différentes barres principales sur Instagram



En mettant ensemble l'icône de l'appareil photo, le logo Instagram et l'icône de l'avion en papier, cela forme la barre principale de la page d'accueil. Selon l'OS mobile, les icônes changent. Mais, il n'y a pas de labels qui accompagnent ces logos pour expliquer leur sens. Le logo Instagram est aligné au centre que ce soit sur Android ou iOS, alors que sur Android, la bonne pratique veut qu'il soit aligné à gauche.

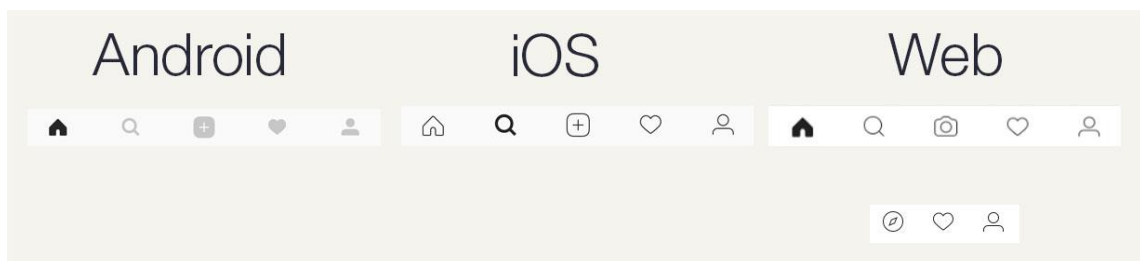
Lorsque l'on est sur la page d'un profil ou d'une photo, la barre principale change. On se retrouve avec un composant qui respecte totalement les best practices des 2 OS : les icônes spécifiques et le texte aligné selon les recommandations.

Sur la version web mobile, la barre principale est dénuée des icônes présentes sur iOS, car ces fonctionnalités ne sont pas implémentées sur la version web. Par contre, l'icône d'ajout d'ami fait son apparition quand on est sur la page d'accueil sur la version web mobile.

Pour la version web PC, la barre principale devient plus complexe et passe dans la catégorie « Organisme ».

8.1.2.2 Barre d'onglets

Figure 32 : Différentes barres d'onglets sur Instagram

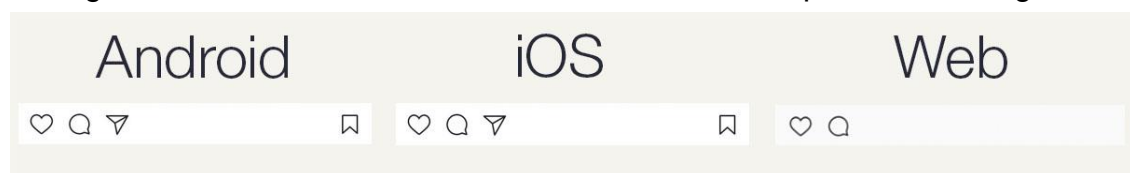


La barre d'onglets reprend les icônes en fonction de l'OS de la version comme expliqué avant. Encore une fois, il n'y a pas de labels qui accompagnent les icônes pour expliquer leur signification. Le fonctionnement de la barre sur les 2 OS est exactement pareil : elle offre les mêmes pages à consulter.

Sur web, la version mobile reprend la même barre que celle sur iOS à une icône près. L'expérience utilisateur est différente sur la version PC, et donc la barre d'onglets aussi : on ne retrouve que 3 fonctionnalités disponibles par rapport aux 5 sur OS mobile. Cet élément fera partie de la barre principale de la version web de bureau.

8.1.2.3 Barre d'interaction avec une photo

Figure 33 : Différentes barres d'interaction avec une photo sur Instagram

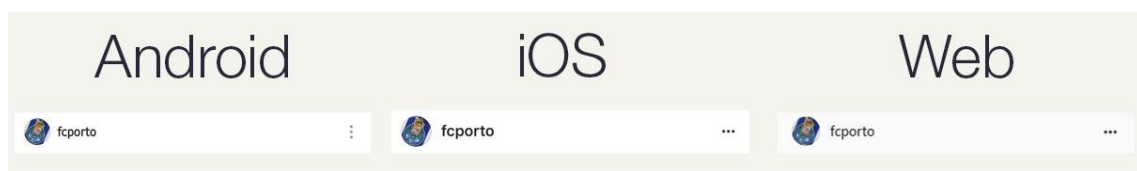


Cet élément ne respecte pas les bonnes pratiques des icônes. C'est les mêmes icônes utilisées sur les 3 supports. Du coup, les contraintes sur Android sont oubliées. Cela crée un manque de cohérence au sein de la version de l'OS de Google. En effet, les icônes du cœur et de l'enveloppe sont différentes de celles vues auparavant.

Sur la version web, des icônes ont été retirées : les fonctionnalités qu'elles représentent n'y sont pas implémentées.

8.1.2.4 Profil de la personne ayant publié le contenu

Figure 34 : Élément du profil de la personne de la publication sur Instagram



L'image de profil est mise ensemble avec le nom du profil et l'icône « plus d'options ». Ce dernier se retrouve sur les 3 plateformes en prenant compte des bonnes pratiques.

8.1.2.5 Nom du profil et commentaire

On retrouve un élément servant à montrer le nom d'une personne et son commentaire, que ce soit sa description sur sa photo ou un commentaire rajouté sur une photo.

8.1.3 Organismes

8.1.3.1 Barre principale sur la version web PC

Figure 35 : Barre principale sur la version web PC d'Instagram



Cette barre principale est plus complexe, car on retrouve des éléments déjà composés à l'intérieur. Elle est constituée du logo de l'application, mais aussi d'une barre de recherche et de la barre d'onglets.

8.1.3.2 Contenu principal : Photo ou vidéo

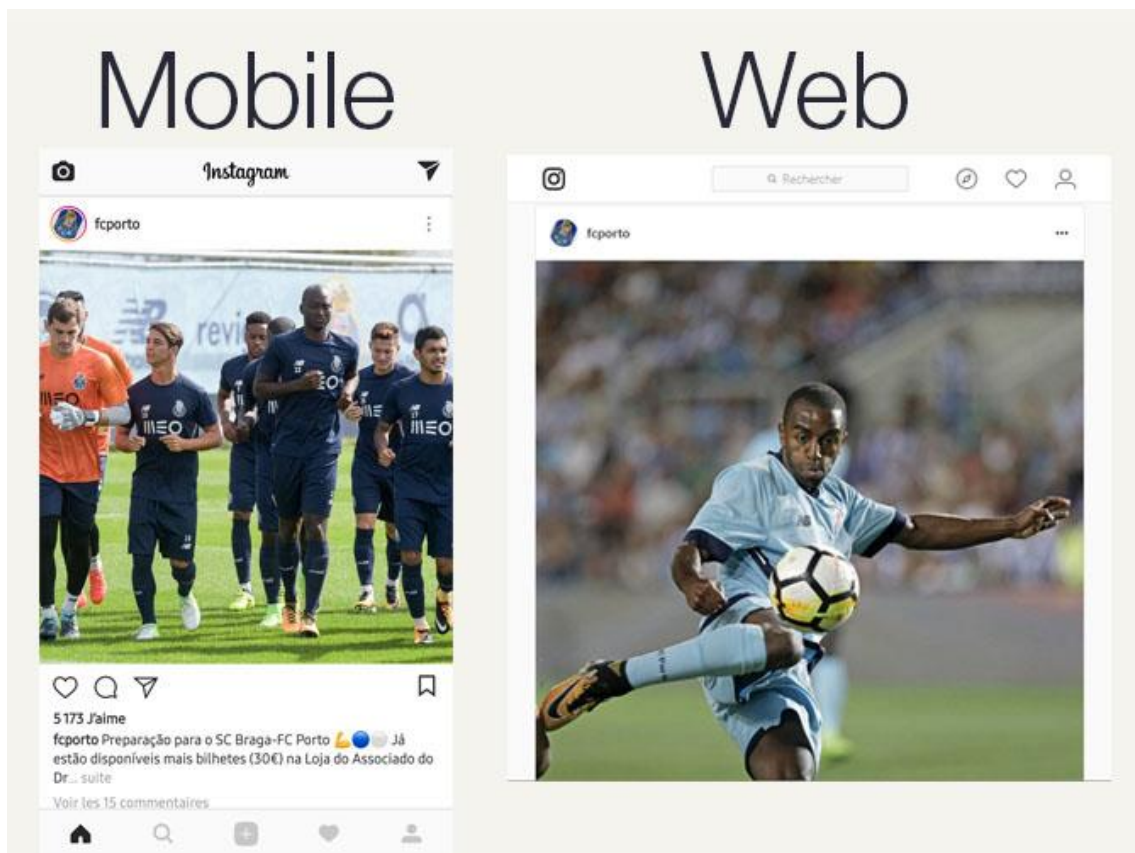
Figure 36 : Contenu principal d'Instagram



Élément principal de l'application, il met ensemble le profil de la personne ayant publié le contenu, l'affichage de la photo ou de la vidéo, la barre d'interaction avec une photo, la description complète, le lien montrant le nombre de commentaires ainsi que la possibilité de les afficher et l'horodatage.

8.1.4 Modèle de page

Figure 37 : Modèles de pages génériques sur Instagram



Pour que l'analyse reste facile à comprendre, seul le modèle de page générique est décortiqué. En effet, l'application garde toujours la même disposition. Le modèle de page générique se compose donc de 2 ou 3 parties, selon si on est sur mobile ou sur la version web de bureau :

- La barre principale, présente en haut de l'écran de chaque version.
- La zone de contenu, là où le contenu principal est affiché.
- La barre d'onglets en bas de l'écran, qui apparaît sur les applications mobiles et sur l'affichage mobile de la version web. Comme cette barre apparaît en bas de l'écran sur Android, elle ne respecte pas les bonnes pratiques de cet OS.

8.2 Conclusion de l'analyse

Instagram applique les principes de l'Atomic Design comme on peut le constater dans cette analyse. On voit que cette méthodologie permet de réutiliser plusieurs composants d'une version à l'autre de manière efficace. La version web finit par ressembler à la version mobile, car elle réutilise la plupart des composants présents sur iOS.

Instagram respecte la plupart du temps les bonnes pratiques de chaque plateforme. Mais, il arrive que des éléments ne suivent pas les conventions, surtout dans la version Android. Plusieurs raisons comme la facilité, le besoin de gagner du temps dans le développement ou une ressemblance voulue entre chaque support sont plausibles. Mais, cela ne nuit pas forcément à l'expérience utilisateur. En effet, les fautes commises ne provoquent pas un apprentissage plus long du fonctionnement de l'application. Par contre, le côté esthétique est touché du fait du manque de cohérence au sein de la version Android.

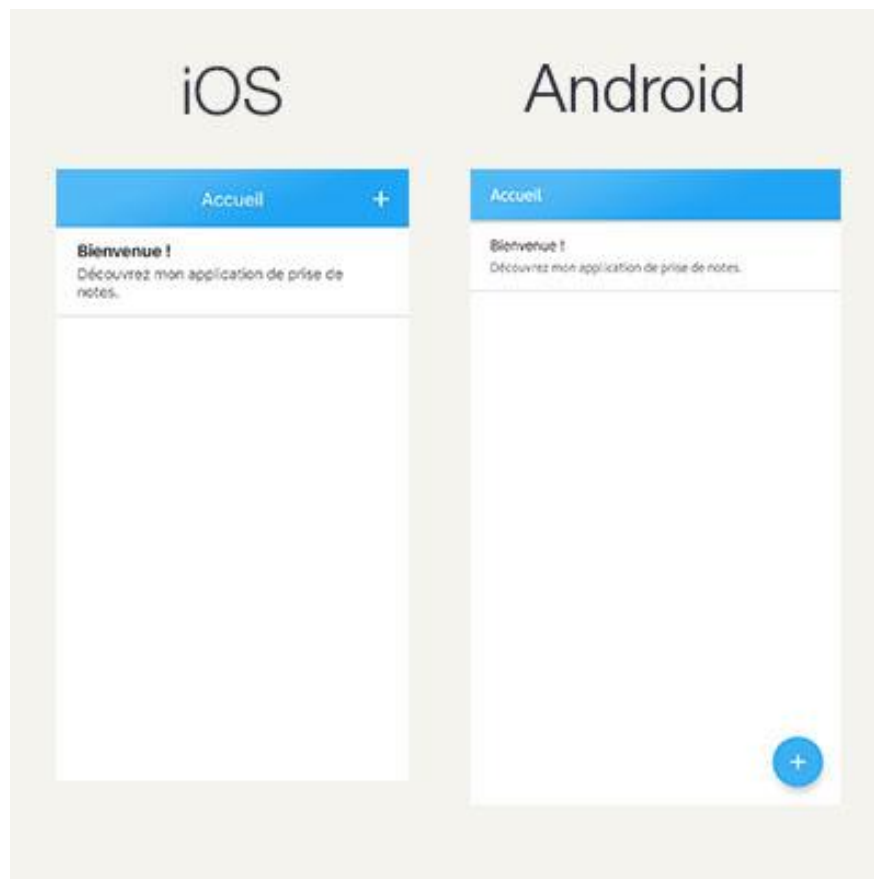
En conclusion, Instagram est relativement une bonne application en termes de design et peut être considéré comme un modèle à suivre sur plusieurs points.

9. Proposition de solution

Comme expliqué précédemment, Android et iOS demandent chacun un développement spécifique à cause du langage de programmation de chaque système d'exploitation, des outils nécessaires et des best practices propres à chaque plateforme. Si on inclut une version web, cela fait 3 développements complètement différents.

Cette solution a pour but de résoudre ces problèmes et ainsi de réduire cette charge supplémentaire qu'apportent les supports mobiles. Afin de mieux la comprendre, une application native mobile a été développée par mes propres soins en mettant en pratique cette proposition. Elle est trouvable à l'adresse <https://github.com/fabioavelino/app-notes-react-native>.

Figure 38 : Application native mobile réalisée par moi-même



9.1 React et React Native

React, une librairie JavaScript conçue par Facebook, permet de développer des interfaces utilisateurs web en créant des composants et en les regroupant afin de former des ensembles plus complexes. Chaque composant peut avoir son propre état et utiliser des attributs reçus en paramètres. Ces composants peuvent être réutilisables par la suite dans d'autres projets.

Figure 39 : Exemple de composant réalisé avec React

```
import React from 'react';

function Welcome (props) {
  return <h1>Hello {props.name}</h1>;
}

const element = <Welcome name="Fabio" />
```

Hello Fabio

Comme on peut le voir dans cet exemple, un composant nommé « Welcome » affiche un titre avec comme contenu « Hello » et le nom reçu en paramètres. On peut ensuite appeler ce composant en lui passant le nom que l'on souhaite comme on peut le voir dans la dernière ligne.

L'affichage avec React se fait sous forme d'un HTML spécial appelé JSX, pouvant être utilisé au milieu du code JavaScript.

Une librairie dépendante de React et toujours conçue par Facebook, appelée React Native, permet de développer des applications natives mobiles, pour Android et iOS, en codant en JavaScript. En effet, on utilise encore React, mais on n'emploie plus des balises HTML comme « div » ou « h1 », mais des éléments fournis par React Native comme « View » et « Text », qui généreront du code natif pour les OS mobiles. Un

Figure 40 : Exemple de composant avec React Native

```
import React from 'react';
import { Text } from 'react-native';

function Welcome (props) {
  return <Text>Hello {props.name}</Text>;
}

const element = <Welcome name="Fabio" />
```

composant est capable de s'afficher sur Android et iOS. La stylisation d'un composant est très similaire au CSS, mais est faite sous forme d'un objet JavaScript.

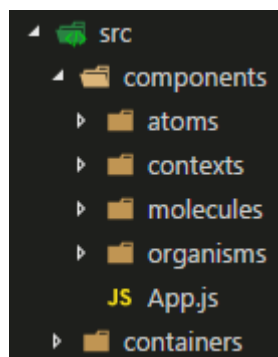
En sachant utiliser React, il n'y a besoin que d'un langage de programmation pour un projet incluant une application native et une application web. Mieux encore, on peut séparer la logique de la partie affichage des composants. Ainsi, on peut réutiliser la partie logique pour le développement de l'application native et pour celui de l'application web. Il suffit de recoder la partie de l'affichage afin qu'elle soit adaptée à React Native avec ses composants natifs ou à React avec ses éléments web.

9.2 Structure du projet

Atomic Design permet de composer de manière modulaire des interfaces utilisateurs en partant du plus petit bloc possible jusqu'à arriver à la page finale. En appliquant ce concept, la création d'une interface utilisateur est facilitée, car des éléments pourront être réutilisés d'une partie de l'UI à une autre.

L'idée ici est d'adopter une approche « Atomic Design » dans la structure du projet.

Figure 41 : Structure des composants avec le concept Atomic Design



Avec cette structure, tous nos composants servant à l'affichage se retrouvent dans un dossier « components ». Ils sont divisés en 4 catégories : atomes, molécules, organismes et contextes. Les 3 premières catégories correspondent à la méthodologie vue précédemment.

Par rapport à la méthodologie originale du design atomique, les catégories « modèles de pages » et « pages » disparaissent. La catégorie « modèles de pages » est utile pour la conception de maquette d'interfaces, mais inadaptée pour le développement de l'application. En outre, une application ne va pas contenir plusieurs pages, mais un affichage qui change selon l'action effectuée par l'utilisateur. C'est pourquoi la catégorie

« contexte » permet d'être plus correcte et plus générique que de parler de « page », même si le principe reste similaire. En effet, c'est dans cette catégorie qu'on va réunir nos composants pour répondre à un but précis de l'application comme l'affichage des notes ou l'affichage d'une seule note par exemple.

Dans chaque catégorie, un élément visuel peut se retrouver dans 2 dossiers :

1. Soit dans un dossier portant le nom de l'élément, dans le cas où il n'y ait pas d'importance s'il est seul.
2. Soit dans un dossier portant le nom du type de composant, par exemple « boutons », dans le cas où il est mis au côté d'autres composants du même type pour des besoins d'organisation. Cela peut aussi être utile si on souhaite partager le style d'affichage entre plusieurs composants du même type par exemple.

Dans le dossier « containers », on trouve des composants complets. Complet, car on appelle un composant visuel en y incorporant des traitements logiques pour avoir un composant utile à l'application. La logique comprend par exemple le remplissage de données d'une liste ou un événement déclenché lors d'un appui. Ainsi, une séparation est faite entre les traitements et l'affichage, ce qui permet de les réutiliser séparément pour les raisons expliquées auparavant. On peut répartir les composants complets dans des sous-dossiers si nécessaire.

Ces composants sont ensuite appelés dans un contexte, car ils servent à répondre à un besoin utilisateur.

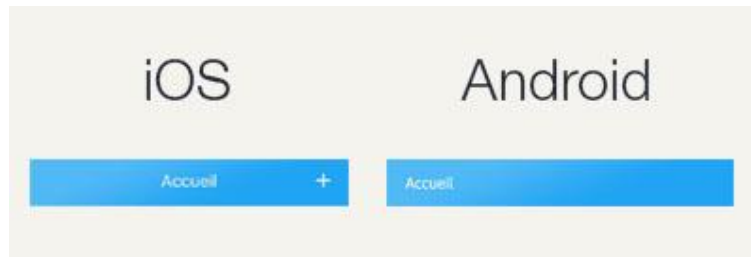
Finalement, le fichier « App.js » s'occupe d'afficher le bon contexte actuel de l'application.

9.3 Création d'un composant natif

Avec React Native, un composant natif s'affiche de manière similaire d'un OS à un autre. Le but de cette librairie étant de développer plus rapidement une application native multiplateforme. Mais, il est aussi possible de spécifier des parties d'un composant, voire le composant en entier, pour que cela soit propre à un OS. Pour cela, il y a 2 manières de le faire.

La première méthode consiste à importer le module « Platform » qui contient une variable appelée « OS », pouvant être égale à « android » ou « ios ». Elle indique sur quel OS s'exécute l'application. De ce fait, on peut se baser sur cette propriété pour afficher un élément d'une certaine manière selon le système d'exploitation.

Figure 42 : Barre principale de l'application de démonstration



Si on prend l'exemple de la barre principale, on constate 2 différences :

- L'alignement du titre de la page. Pour se faire, il suffit de préciser l'alignement en testant l'OS actuel où est exécutée l'application.

Figure 43 : Alignement spécifique du titre de la page selon l'OS

```
textAlign: Platform.OS === 'android' ? 'left' : 'center',
```

- L'affichage du bouton d'ajout, seulement sur la version iOS.

Figure 44 : Affichage d'un composant selon l'OS

```
{Platform.OS === 'ios' && <AddButton onPress={handleOnPressAdd} />}
```

L'autre manière concerne les composants plus complexes. Lorsqu'il y a une très grande différence entre 2 OS, il est préférable de séparer un composant en 2 affichages différents. On peut alors créer 2 fois le même composant avec le même nom dans un même dossier, mais en précisant une extension, « .android.js » ou « .ios.js », pour spécifier quel est l'affichage à adopter selon l'OS. En important le composant en ne précisant que le nom, React Native sait quel composant appeler selon la plateforme de lancement de l'application.

9.4 Explication de ces choix

Le but de cette solution est de développer rapidement un produit où une application web et une native pour les 2 plateformes les plus populaires sont exigées. React et React

Native nous permettent de coder de manière similaire pour le web et pour la version native. L'une des règles fondamentales en développement est le fait d'éviter de répéter du code. L'approche Atomic Design couplée à la séparation du visuel de la logique répond à merveille à cette demande :

- Les éléments visuels sont réutilisables à travers l'application en ayant de la logique différente si nécessaire.
- Un composant visuel complexe peut être écrit en 2 fichiers séparés visant chacun un OS et utiliser la même logique par la suite pour cet élément séparé en 2.
- La réutilisation des composants web vers les supports natifs ou inversement est facilitée, car on peut réutiliser toute la logique et ne devoir réadapter que les composants visuels présents dans le dossier « components ».
- Le projet peut grandir facilement, car la structure est établie pour répondre à un ajout d'un nouveau contexte à l'application tout en gardant un aspect visuel cohérent.

Afin d'offrir un produit adapté pour chaque support de consultation, il faut aussi adopter l'approche « Mobile First » en commençant par concevoir l'interface graphique des versions smartphones, pour ensuite l'adapter pour les tablettes et finir par l'ajuster pour une navigation parfaite sur PC. Pour une version web optimale, il suffit de reprendre les principes expliqués dans le point 6.2.1 « Responsive Web Design ».

10. Conclusion

Du fait de leur grande adoption par les utilisateurs finaux, les tablettes et principalement les smartphones ont pris une importance capitale lors de la conception d'un produit applicatif. Il n'est plus possible de les ignorer, au risque de décevoir le marché cible, voire même de ne plus être visible pour celui-ci. On constate qu'iOS et Android sont les grands gagnants des systèmes d'exploitation mobiles.

Les spécificités de ces nouveaux supports bouleversent la manière d'interagir avec un système informatique. C'est pourquoi Google et Apple ont apporté des lignes directrices de design d'applications pour leurs appareils. D'autres institutions et personnalités ont aussi apporté leur pierre à l'édifice. Le but de toutes ces recommandations est de concevoir une expérience utilisateur agréable et une prise en main facile pour l'utilisateur.

En analysant Instagram, on observe à la fois le respect des bonnes pratiques et la prise de liberté sur certains points. Étant donné que ce sont des conseils, il n'y a pas d'obligations de les employer à la lettre. Mais, le fait de trop s'éloigner de ces recommandations peut rendre une application difficile à utiliser ou peut causer des problèmes d'esthétiques. Tout cela peut provoquer une baisse d'adoption de l'application. Il est nécessaire de bien réfléchir aux choix faits pour les interfaces, afin de toujours offrir une expérience utilisateur optimale.

Même si le web n'a pas des best practices propres à lui-même, beaucoup d'éléments venant du monde mobile y ont été adoptés, comme le tiroir de navigation. Étant à la base une recommandation de Google pour les applications Android, il a fini par être utilisé sur les interfaces utilisateurs web. Dans l'analyse d'Instagram, on constate que l'aspect de l'application web reprend l'aspect des applications natives mobiles. Aujourd'hui, on peut en conclure que les applications web finissent par ressembler aux applications mobiles. Plus étonnant encore, les versions web de bureau s'inspirent fortement des versions mobiles.

Tous ces nouveaux appareils à prendre en compte lors de la conception d'une application provoquent une montée des coûts conséquente. C'est pourquoi une solution est proposée dans ce travail pour réduire ces nouvelles charges que représentent les appareils mobiles. Elle permet de développer une application pour plusieurs supports avec un langage de programmation unique, une grande réutilisation du code créée et une conception efficace d'interface qui facilite le respect des bonnes pratiques.

Bibliographie

WIKIPEDIA, [sans date]. Smartphone [en ligne]. [sans date]. [Consulté le 13 juillet 2017].

Disponible à l'adresse :

<https://fr.wikipedia.org/wiki/Smartphone>

AUFRAY, Christophe, 2017. Chiffres clés : les OS pour smartphones [en ligne]. 12 juin 2017. [Consulté le 13 juillet 2017].

Disponible à l'adresse :

<http://www.zdnet.fr/actualites/chiffres-cles-les-os-pour-smartphones-39790245.htm>

WIKIPEDIA, [sans date]. Tablette tactile [en ligne]. [sans date]. [Consulté le 13 juillet 2017].

Disponible à l'adresse :

https://fr.wikipedia.org/wiki/Tablette_tactile

AUFRAY, Christophe, 2017. Chiffres clés : le marché des tablettes [en ligne]. 12 mai 2017. [Consulté le 13 juillet 2017].

Disponible à l'adresse :

<http://www.zdnet.fr/actualites/chiffres-cles-le-marche-des-tablettes-39789571.htm>

NIELSEN NORMAN GROUP, [sans date]. The Definition of User Experience (UX) [en ligne]. [sans date]. [Consulté le 13 juillet 2017].

Disponible à l'adresse :

<https://www.nngroup.com/articles/definition-user-experience/>

FROST, Brad, 2013. Atomic Web Design [en ligne]. 10 juin 2013. [Consulté le 13 juillet 2017].

Disponible à l'adresse :

<http://bradfrost.com/blog/post/atomic-web-design/>

STATCOUNTER, 2017. Desktop vs Mobile vs Tablet Market Share Worldwide [en ligne]. Juillet 2017. [Consulté le 7 août 2017].

Disponible à l'adresse :

<http://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/>

GOOGLE, 2015. Make sure your website's ready for mobile-friendly Google search results [en ligne]. 2015. [Consulté le 8 août 2017].

Disponible à l'adresse :

<https://support.google.com/adsense/answer/6196932?hl=en-GB>

DOANTAM, Phan, 2015. Mobile-first Indexing [en ligne]. 4 novembre 2016. [Consulté le 8 août 2017].

Disponible à l'adresse :

<https://webmasters.googleblog.com/2016/11/mobile-first-indexing.html>

SPILLERS, Frank, 2014. Making a Strong Business case for the ROI of UX [en ligne]. 24 juillet 2014. [Consulté le 8 août 2017].

Disponible à l'adresse :

<https://www.experiencedynamics.com/blog/2014/07/making-strong-business-case-roi-ux-infographic>

GRANGER, Stevie, 2016. UI vs UX Design [en ligne]. 27 mai 2016. [Consulté le 8 août 2017].

Disponible à l'adresse :

<https://www.uzu-media.com/blog/2016/05/27/ui-vs-ux-design>

W3C, 2012. CSS Media Queries [en ligne]. 19 juin 2012. [Consulté le 9 août 2017].

Disponible à l'adresse :

<https://www.w3.org/TR/css3-mediaqueries/>

HART, Sandee, 2014. Mobile vs. Desktop: 10 Key Differences [en ligne]. 30 avril 2014. [Consulté le 9 août 2017].

Disponible à l'adresse :

<https://www.paradoxlabs.com/blog/mobile-vs-desktop-10-key-differences/>

APP ANNIE, 2016. Prévisions marché 2016-2021 [en ligne]. 2016. [Consulté le 9 août 2017].

Disponible à l'adresse :

http://files.appannie.com.s3.amazonaws.com/reports/1704_Report_Market_Forecast_FR.pdf

STATISTA, 2017. Number of apps available in leading app stores as of March 2017 [en ligne]. Mars 2017. [Consulté le 9 août 2017].

Disponible à l'adresse :

<https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>

RAPHAEL, 2014. C'est quoi le Responsive Web Design ? [en ligne]. Février 2014. [Consulté le 11 août 2017].

Disponible à l'adresse :

<https://www.alsacreations.com/article/lire/1615-cest-quoi-le-responsive-web-design.html>

GOOGLE, [sans date]. Build a Responsive UI with ConstraintLayout [en ligne]. [sans date]. [Consulté le 11 août 2017].

Disponible à l'adresse :

<https://developer.android.com/training/constraint-layout/index.html>

APPLE, [sans date]. Understanding Auto Layout [en ligne]. [sans date]. [Consulté le 11 août 2017].

Disponible à l'adresse :

https://developer.apple.com/library/content/documentation/UserExperience/Conceptual/AutolayoutPG/index.html#//apple_ref/doc/uid/TP40010853-CH7-SW1

GREMILLION, Ben, [sans date]. A Hands-On Guide to Mobile-First Responsive Design [en ligne]. [sans date]. [Consulté le 11 août 2017].

Disponible à l'adresse :

<https://www.uxpin.com/studio/blog/a-hands-on-guide-to-mobile-first-design/>

99DESIGNS, 2016. Skeuomorphism vs. flat design vs material design [en ligne]. Novembre 2016.[Consulté le 15 août 2017].

Disponible à l'adresse :

<https://en.99designs.ch/blog/trends/skeuomorphism-flat-design-material-design/>

99DESIGNS, 2014. Skeuomorphism vs. flat design vs material design [en ligne]. 2014. [Consulté le 15 août 2017].

Disponible à l'adresse :

<https://99designs.co.uk/blog/tips-en-gb/long-road-googles-material-design/>

VISONNEAU, Antoine, 2015. Nouvelle approche du design d'interface sur Android, Material Design [en ligne]. Février 2015. [Consulté le 15 août 2017].

Disponible à l'adresse :

<http://www.servicesmobiles.fr/nouvelle-approche-du-design-dinterface-sur-android-material-design-26596/>

WIKIPEDIA, [sans date]. Skeuomorphisme [en ligne]. [sans date]. [Consulté le 15 août 2017].

Disponible à l'adresse :

<https://fr.wikipedia.org/wiki/Skeuomorphisme>

PANNAFINO, James, 2013. Helping Users Navigate Your Website Design [en ligne]. Avril 2013. [Consulté le 16 août 2017].

Disponible à l'adresse :

<http://www.howdesign.com/web-design-resources-technology/helping-users-navigate-websites/>

HURFF, Scott, [sans date]. How to design for thumbs in the Era of Huge Screens [en ligne]. [sans date]. [Consulté le 16 août 2017].

Disponible à l'adresse :

<http://scotthurff.com/posts/how-to-design-for-thumbs-in-the-era-of-huge-screens>

ELLIS, Brad, 2017. All Thumbs, Why Reach Navigation Should Replace the Navbar in iOS Design [en ligne]. Mai 2017. [Consulté le 16 août 2017].

Disponible à l'adresse :

<https://medium.com/tall-west/lets-ditch-the-nav-bar-3692cb17cc67>

BABICH, Nick, 2017. Basic Patterns of Mobile Navigation [en ligne]. Mai 2017. [Consulté le 16 août 2017].

Disponible à l'adresse :

<https://blogs.adobe.com/creativecloud/basic-patterns-of-mobile-navigation/>

GOOGLE, [sans date]. Material Design [en ligne]. [sans date]. [Consulté le 16 août 2017].

Disponible à l'adresse :

<https://material.io/guidelines/>

APPLE, [sans date]. Human Interface Guidelines [en ligne]. [sans date]. [Consulté le 16 août 2017].

Disponible à l'adresse :

<https://developer.apple.com/ios/human-interface-guidelines/>

BABICH, Nick, 2017. Getting Typography Right in Digital Design [en ligne]. Août 2017. [Consulté le 17 août 2017].

Disponible à l'adresse :

<https://medium.springboard.com/the-guide-to-getting-typography-right-in-digital-design-bb61214ff3ad>

NIELSEN NORMAN GROUP, 2014. Icon Usability [en ligne]. Juillet 2014. [Consulté le 18 août 2017].

Disponible à l'adresse :

<https://www.nngroup.com/articles/icon-usability/>

MALACHI, Stefano, 2017. UI Motion: A Beautiful Problem Solver to Improve UX [en ligne]. Mai 2017. [Consulté le 21 août 2017].

Disponible à l'adresse :

<https://medium.muz.li/malachidigest-2e7f3a371531>

O'SULLIVAN, Chris, 2015. A Tale of Two Platforms: Designing for Both Android and iOS [en ligne]. Avril 2015. [Consulté le 22 août 2017].

Disponible à l'adresse :

<https://webdesign.tutsplus.com/articles/a-tale-of-two-platforms-designing-for-both-android-and-ios--cms-23616>

STATISTA, 2017. Instagram: number of monthly active users 2013-2017 [en ligne]. Avril 2017. [Consulté le 23 août 2017].

Disponible à l'adresse :

<https://www.statista.com/statistics/253577/number-of-monthly-active-instagram-users/>

WIKIPEDIA, [sans date]. Instagram [en ligne]. [sans date]. [Consulté le 23 août 2017].

Disponible à l'adresse :

<https://fr.wikipedia.org/wiki/Instagram>

FACEBOOK, [sans date]. React - A JavaScript library for building user interfaces. [en ligne]. [sans date]. [Consulté le 6 septembre 2017].

Disponible à l'adresse :

<https://facebook.github.io/react/>

FACEBOOK, [sans date]. React Native - A framework for native apps using React. [en ligne]. [sans date]. [Consulté le 6 septembre 2017].

Disponible à l'adresse :

<https://facebook.github.io/react-native/>