**Interface :**

1. Présentation :

Une [interface](https://fr.wikipedia.org/wiki/Interface) définit la frontière de communication entre deux entités, comme des éléments de logiciel, des composants de [matériel informatique](https://fr.wikipedia.org/wiki/Mat%C3%A9riel_informatique), ou un [utilisateur](https://fr.wikipedia.org/wiki/Utilisateur_(informatique)). Elle se réfère généralement à une image abstraite qu'une entité fournit d'elle-même à l'extérieur. Cela permet de distinguer les méthodes de communication avec l'extérieur et les opérations internes, et autorise à modifier les opérations internes sans affecter la façon dont les entités externes interagissent avec elle, en même temps qu'elle en fournit des abstractions multiples. On appelle aussi interfaces des dispositifs fournissant un moyen de traduction entre des entités qui n'utilisent pas le même langage, comme entre un [être humain](https://fr.wikipedia.org/wiki/%C3%8Atre_humain) et un [ordinateur](https://fr.wikipedia.org/wiki/Ordinateur).  
  
-Variétés :

* L'interface entre un [être humain](https://fr.wikipedia.org/wiki/%C3%8Atre_humain) et un [ordinateur](https://fr.wikipedia.org/wiki/Ordinateur) est appelée [interface utilisateur](https://fr.wikipedia.org/wiki/Interface_utilisateur).
* Les interfaces entre des composants de [matériel informatique](https://fr.wikipedia.org/wiki/Mat%C3%A9riel_informatique) sont des interfaces physiques.

1. Les différents langages :

HTML 5, QML, XAML, des plateformes émergent, chacune avec ses avantages et ses défauts, et le choix se pose dès lors que l'on veut réaliser une application Web, une RIA (Rich Internet Application) ayant la même interface et les mêmes possibilités qu'une application locale comme le font certains sites populaires.  
La crainte de prendre la mauvaise voie et de passer des mois en développement avec un environnement sur lequel on ne pourrait aboutir tandis qu'un autre aurait pu s'avérer plus adapté se traduit dans les forums où la question est très présente.  
Le problème se complique encore quand on s'aperçoit que les différentes solutions proposées ne cessent d'évoluer et de proposer des fonctionnalités jusque-là absentes, et l'on se demande, sera-t-il possible qu'à l'avenir, telle plateforme offre les possibilités actuellement offertes par telle autre?

\_HTML :   
La nouvelle spécification HTML 5 offre une interface universelle pour les applications en lignes et certaines applications locales grâce au mode offline.

Cela est complété par Ajax, qui est la combinaison des techniques formant le HTML dynamique (CSS, JavaScript, DOM) et de l'objet [XMLHttpRequest](http://www.scriptol.fr/ajax/xhr.php) qui permet l'interaction asynchrone entre le navigateur et le serveur. Et plus récemment par WebSocket et WebRTC pour une meilleure interaction avec le serveur ou avec directement d'autres clients dans le cas de WebRTC.  
L'application utilise un langage de script coté serveur comme PHP ou ASP ou JavaScript  
  
Avantages :  
-Probabilité totale d’être reconnu par tous les navigateurs  
-Compatible avec les standard du WEB : JavaScript, CSS…  
-Fonctionne sans connexion Internet  
-Facilité de mise en œuvre  
Inconvénients :  
- L’utilisation du CSS pour la mise en forme est souvent laborieuse.  
-Programmer en JavaScript est difficile pour de grands programmes

\_XAML :   
[XAML](http://www.scriptol.fr/programmation/xaml.php) est comme XUL un langage de description d'interface basé sur XML. Mais il est vu par Microsoft comme un moyen de créer des applications Internet riches. Il offre un langage d'interface à Windows avec .NET, Linux avec Mono et et systèmes dérivés de Linux telle la plateforme Android.

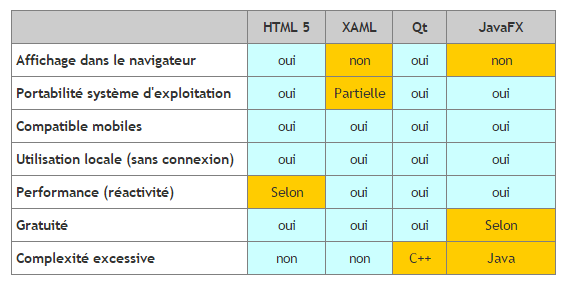
Avantages :  
-Très large bibliothèque de fonctions  
-Choix du langage de programmation sur la plateforme .NET  
Inconvénients :  
-Compatibilité limitée.

\_JavaFX  
La solution Java est complète puisqu'elle propose un framework coté serveur, et coté navigateur, l'emploi d'applets, qui sont de petites applications fonctionnant dans la page Web. Cela peut aussi être complété par un langage de script coté serveur, JSP. Ajoutons à cela les servlets pour les services Web et l'on dispose d'une plateforme portable et complète pour les applications Web d'entreprises.

Avantages :  
-Bibliothèque de fonction très vaste.  
-Compatibilité avec toutes les plateformes  
Inconvénients :  
-Pour les applications WEB le serveur doit supporter Java ce qui implique pratiquement un serveur dédié.  
-La programmation peut être complexe.

\_QML :  
Qt est à l'origine un framework d'interfaces pour Linux qui a été porté sur tous systèmes d'exploitation y compris Android et iOS. Il inclut tous les widgets nécessaires à la réalisation d'une interface et à aussi son langage d'interface, QML. Celui-ci a une syntaxe similaire à celle de JSON ce qui le rapproche de JavaFX.

Avantages :  
-Portabilité sur le bureau et le mobile.  
-QT creator facilite la r éalisation des applications.  
Inconvénients :  
-La programmation se fait en C++, applications rapides mais bugs fréquents.  
-Faire communiquer JavaScript dans l’interface et C++ peut être compliqué.

  
Tableau récapitulatif :  
  
  
III. Notre choix :

Pour créer l’interface nous utiliserons du HTML, du CSS. Nous prenons ce langage là car nous avons des connaissances au niveau du code, cela nous semblera moins complexe.