**{EPITECH.}**

09

**Cahier des charges**

**Graphity**

**{EPITECH.}**

Table des matières

[1. Présentation du projet 3](#_Toc243907577)

[1.1. L’objet du projet 3](#_Toc243907578)

[1.2. Le contexte du projet 4](#_Toc243907579)

[1.3. Public-cible du projet 4](#_Toc243907580)

[1.4. Les acteurs du projet 4](#_Toc243907581)

[1.5. Nos concurrents 4](#_Toc243907582)

# Présentation du projet

## L’objet du projet

Le but de ce projet est la réalisation d’une bibliothèque logicielle multiplateforme orientée objet et offrant à l’utilisateur des composants tels que les interfaces graphiques, la gestion des connexions réseaux, des fils d’exécution, des bases de données... Cette librairie sera développée dans les langages les plus en vogue actuellement, à savoir le C++, le java ainsi que le c#.

Elle permettra la portabilité des applications que ce soit sur ordinateur ou mobile qui n’utilisent que ses composants simplement en recompilant le code source sur le système d’exploitation souhaité afin que les utilisateurs de cette bibliothèque puissent diffuser leurs programmes sur les différentes plateformes mentionnées ci-dessous sans difficultés.

Le développement d’applications sur les mobiles sera ainsi simplifié, si l’application développée dans un langage n’est pas portable sur une plateforme (par exemple Symbian si application initialement développée en c++), il suffira de porter le code dans le langage approprié (dans ce cas le java).

Les environnements supportés seront Windows, Mac OS, Iphone, les distributions Linux (Gentoo, Fedora, Red Hat , Ubuntu…), les distributions BSD (OpenBSD, FreeBSD, NetBSD…) ainsi que les systèmes dérivés basés sur le noyau UNIX (Solaris, System V).

Notre librairie utilisera les API natives de chaque système. Par exemple en c++, elle sera codée en win32 sous Windows, en XLib sous UNIX et en Cocoa sous macOS, afin d’exploiter toutes les possibilités de chaque système. Nous utiliserons le moins possible de librairies externes afin de conserver la rapidité du langage (en réduisant les appels de fonctions). Les seules bibliothèques utilisées seront libres de droits. Nous utiliseront notamment openssl pour le SSL car il nous sera difficile de faire mieux en terme de performance que celle-ci.

La librairie sera composée de deux parties, une partie concernant tout ce qui a attrait au visuel et une autre qui touchera à tout ce qui est système.

Côté système, voici une brève description des classes déjà implémentées : gestions des threads, mutex, processus, chaînes de caractères, librairies dynamiques, conteneurs, traitement d’images, fichiers, répertoires, Dates, Temps, Musique, parseur XML, réseaux (UDP, TCP, SMTP, POP…), Volumes, etc…

Côté graphique, nous trouverons tout ce qui ce fait de basique en visuel, fenêtres, boutons, champs de texte…

Une documentation détaillée de la bibliothèque sera disponible sur un site internet ainsi que sous forme de « man » sous les systèmes UNIX.

Notre réel objectif est d’uniformiser le développement entre les différentes plateformes et les différents langages et de créer un SDK complet et fonctionnel.

## Le contexte du projet

Depuis quelques années, le marché des communications fixes et mobiles ne cesse d’augmenter. En effet, en ce début de siècle, le téléphone mobile et l’informatique sont les outils de communication les plus utilisés.

Secteur incontournable et aujourd'hui indispensable, la communication à travers l'outil informatique  est une réelle valeur ajoutée aux applications qui la proposent. Tout le monde y a recours, quel que soit le secteur, les entreprises ne peuvent plus se passer de l’outil informatique.

## Public-cible du projet

Cette offre s’adressera en premier lieu aux sociétés de développement logiciels, aux développeurs individuels ainsi qu’à toute personne désireuse d’apprendre la programmation sans avoir à apprendre les spécificités des API natives de chaque système d’exploitation.

## Les acteurs du projet

## Nos concurrents

Nos concurrents sont peu nombreux, parmi eux nous trouvons wxWidget, GTK+ et QT. Ce dernier, développé initialement par Trolltech, et racheté récemment par Nokia, a de l’ancienneté et commence à être populaire depuis quelques années.

QT offre un SDK multiplateforme qui s’est fait connaître grâce à son système de signal / slot. Cependant, côté système, beaucoup de fonctionnalités sont manquantes. Nous pouvons notamment citer : les processus (seulement le lancement des processus, impossible de lister les processus en cours sur le système), aucune gestion du registre sous Windows, le Système (pas de Reboot, Shutdown, opération basique sur les lecteurs), pas de gestion du bureau, des icônes, etc…