19/11/2019

Julien GUILLAN - Salayna DOUKOURE - Thibault BALSAMO-Henri GOURGUE - DELAGE Corentin

ESGI

Audit de code

AK OKLM Radio

Table des matières

[Contexte 2](#_Toc25010985)

[Problèmes de design et interface 3](#_Toc25010986)

[IHM 4](#_Toc25010987)

[Code source 5](#_Toc25010988)

[Compte rendu de l’étude 7](#_Toc25010989)

# Contexte

Une entreprise externe AKA OKLM Radio a fait appel à nos services. Ils souhaitent faire auditer leur application (réalisée par un développeur Junior). L'application est disponible sous IOS (IPhone, IPad). L’objectif final de ce rapport est d’étudier le code fourni et d’en tirer des conclusions quant à son bon fonctionnement, ainsi ce rapport jugera de l’efficacité de ce code.

Ce code a été réalisée en swift 5. Il comporte plusieurs problèmes à différents niveaux. Cet audit aura pour but de répertorier et expliquer ces erreurs. Pour traiter ce problème, nous parlerons en premier temps de l'interface de l'application puis nous nous pencherons sur le contenu des fichiers de code.

Pour chaque problème d'exécution, de code relevé dans les paragraphes suivants, un tableau de criticité du problème sera fourni. Ce tableau a pour but de décrire au demandeur l’importance du problème souligné.

Ci-dessous, un exemple de tableau de criticité d’un problème quelconque :

|  |  |  |  |
| --- | --- | --- | --- |
|  | Faible | Moyenne | Forte |
| Criticité du problème |  |  | X |
| Facilité de résolution |  |  | X |

Ainsi, le problème précédent est un problème très critique, qui pose problème à l’expérience de l’utilisateur (il peut nuire à celle-ci). Toutefois, ce problème est classé facile à résoudre.

Dans la suite du document, ce tableau sera inséré à la suite de l’explication de chaque portion de code posant problème.

# Problèmes de design et interface

Concernant les problèmes de design et interface, nous pouvons noter dans un premier temps qu'il manque des informations nécessaires au bon fonctionnement visuel de l'application. En effet, lorsque l’utilisateur souhaite connaître les détails météorologiques d’une journée, il est redirigé vers une activité ayant pour but de délivrer ces détails. Toutefois, les unités sont manquantes et l’utilisateur ne peut donc pas prendre connaissance des réelles informations météorologiques (les informations affichées ne sont pas pertinentes). D’autre part, l’application n’offre pas un aspect responsif fonctionnel, l'exécution du projet sous un Ipad n’est pas optimale, l’utilisateur est surchargé d’informations et ne s’y retrouve pas facilement. Autre problème détecté, l'icône de l’application n’est pas présente. Toutefois, l’application reste quand même fluide, qu’elle soit exécutée sur un Iphone ou un Ipad.

L’aspect non responsif de l’application a été noté à la suite d’une série de test qui s’est déroulé de la manière suivante :

Affichage multiple de l'exécution de l’application sur un Iphone 8 et deux Iphone SE (un réel et un simulé). Nous avons constaté que les images affichées sur les Iphone SE sont rognées tandis que l’affichage sur un Iphone 8 s’effectue correctement.

Criticité du design et de l’interface :

|  |  |  |  |
| --- | --- | --- | --- |
|  | Faible | Moyenne | Forte |
| Criticité du problème |  |  | X |
| Facilité de résolution |  | X |  |

# IHM

A l’installation de l’application, les données de la météo ne sont pas affichées nativement. L’utilisateur est obligé de rafraichir lui-même les données pour qu’il les reçoive alors que l’on pourrait automatiquement le faire dès le lancement de l’application. C’est une légère perte de temps et une fonctionnalité peu pratique pour l’utilisateur. L’application va aller regarder dans la base de données dès le lancement sauf que la base de données n’a pas eu le temps de récupérer les informations de l’API.

Criticité du problème du premier lancement :

|  |  |  |  |
| --- | --- | --- | --- |
|  | Faible | Moyenne | Forte |
| Criticité du problème | X |  |  |
| Facilité de résolution |  | X |  |

Des tests poussés concernant l’utilisation de cette application sur un Ipad ont été réalisés. Ces tests ont révélé qu’il existe beaucoup de problème de design et d’interface sur ce périphérique. En effet, lors du premier lancement de l’application, la météo est affichée deux fois (chaque vignette météo est répétée). Cette mauvaise gestion des informations sur ce périphérique est susceptible de nuire à l’expérience de l’utilisateur.

# Code source

Au niveau du code source de l’application, on constate de multiples avertissements de compilations. Cela ne va pas générer de problèmes majeurs pouvant porter atteinte à l’application mais elles peuvent générer des problèmes mineurs seraient susceptibles de ralentir l’application ou la rendre plus consommatrice en termes de performances. De plus, pour nous développeur qui analysons le code, on remarque que les conventions de codage ne sont pas respectées, par exemple le nom des variables est écrit en Français, les valeurs des variables sont codées en dur (coordonnées GPS). De plus on peut apercevoir une redondance du code qui pourrait être optimisée et de ce fait, réduire la taille des fichiers sources (alléger l’application).

|  |  |  |  |
| --- | --- | --- | --- |
|  | Faible | Moyenne | Forte |
| Criticité du problème |  | X |  |
| Facilité de résolution |  |  | X |

Toujours concernant le code source, on peut apercevoir que les jetons d’identifications sont codés en dur. Cela peut-être dangereux de laisser ça en clair puisque des utilisateurs mal intentionnés pourraient récupérer ces informations comme elles le souhaitent. Il aurait fallu crypter ces jetons. Aussi, on relève un problème au sein de l’app qui affiche la même météo.

Une autre analyse des fichiers sources du projet nous a révélé un problème sur les images (icônes) destinées à afficher la météo. Une seule image a été trouvée, celle affichant un soleil (beau temps). Ceci implique que l’application ne pourra jamais afficher un temps différent que “beau temps”. Autre problème aussi relevé au niveau de la pression de la mer, on a un mauvais traitement de la donnée qui affiche une valeur incohérente.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Faible | Moyenne | Forte |
| Criticité du problème |  |  | X |
| Facilité de résolution | X |  |  |

Le code de cette application comporte différentes erreurs, qui pourraient, chacunes à différents niveaux être la cause mère de problèmes bug et plantage de l’application. En effet, certaines fuites mémoires ont été décelées grâce à l’utilisation de plusieurs outils informatiques, ce qui peut entraîner des fermetures et plantage inopinés de l’application, portant atteinte au bon fonctionnement de celle-ci.

Criticité des problèmes mémoire :

|  |  |  |  |
| --- | --- | --- | --- |
|  | Faible | Moyenne | Forte |
| Criticité du problème |  |  | X |
| Facilité de résolution | X |  |  |

# Compte rendu de l’étude

Le travail effectué par nos équipes concernant l’étude et les tests de code, nous a révélé un certain nombre de points. D’une part, l’application est « fonctionnelle » et peut être utilisée sur plusieurs périphériques et systèmes d’exploitation. Le code s’exécute correctement et l’application se lance sans problèmes apparents. D’autre part, cet aspect fonctionnel cache des erreurs dans le code qui pourraient nuire à l’expérience utilisateur en ralentissant son périphérique. Aussi, de nombreuses erreurs d’affichage et de design sont présentes au sein de l’application. Ces erreurs sont susceptibles d’induire en erreur l’utilisateur.