

CountOnMe

Calculatrice CountOnMe

Dossier de Conception Fonctionnel

Version 1.0

Auteur :
Mathieu Janneau
Développeur

TABLE DES MATIERES

1 - Versions	3
2 - Introduction	4
2.1 - Objet du document	4
2.2 - Besoin du client	4
2.2.1 - Contexte	4
2.2.2 - Enjeux et Objectifs	4
3 - Description générale de la solution	5
3.1 - Les principe de fonctionnement	5
3.2 - Les acteurs	5
3.3 - Les cas d'utilisation généraux	5
3.4 - Les cas d'utilisation	6
3.4.1 - Package Calculatrice	6
5.1.1.1 - UC1 – Cas d'utilisation Utilisateur appuie sur un bouton chiffre	7
5.1.1.2 - UC2 – Cas d'utilisation Utilisateur appuie sur un bouton chiffre	8
5.1.1.2.1 Scénario alternatif :	9
5.1.1.3 - UC1 – Cas d'utilisation Utilisateur appuie sur un bouton chiffre	11
5.1.1.3.1 Scénario alternatif :	12

1 - Versions

Auteur	Date	Description	Version
Mathieu Janneau	05/02/2018	Création du document	1.0

2 - Introduction

2.1 - Objet du document

Le présent document constitue le dossier de conception fonctionnelle de l'application XXX...

Le document a pour objectif de présenter les fonctionnalités et le fonctionnement de l'application de calculatrice CountOnMe

Les éléments du présent dossier découlent :

- De l'analyse du besoin client
- Des échanges avec les commerciaux de la société

2.2 - Besoin du client

2.2.1 - Contexte

La société Client a demandé à notre société de développer pour son compte une application de calculatrice disponible sur tous les smartphones iphone et évoluant sur le système iOS 11 et postérieurs.

2.2.2 - Enjeux et Objectifs

L'enjeu est de créer une application qui permet d'effectuer des opérations arithmétiques binaires telles que :

- Addition
- Soustraction
- Multiplication
- Division
- La possibilité d'effectuer les calculs avec des nombres entiers et décimaux
- La possibilité d'utiliser des calculs à partir du résultat d'un précédent calcul
- La possibilité de réinitialiser la calculette à tous moments.

3 - Description générale de la solution

La solution proposée est une application iOS 11 qui permet d'effectuer des calculs binaires.

3.1 - Les principe de fonctionnement

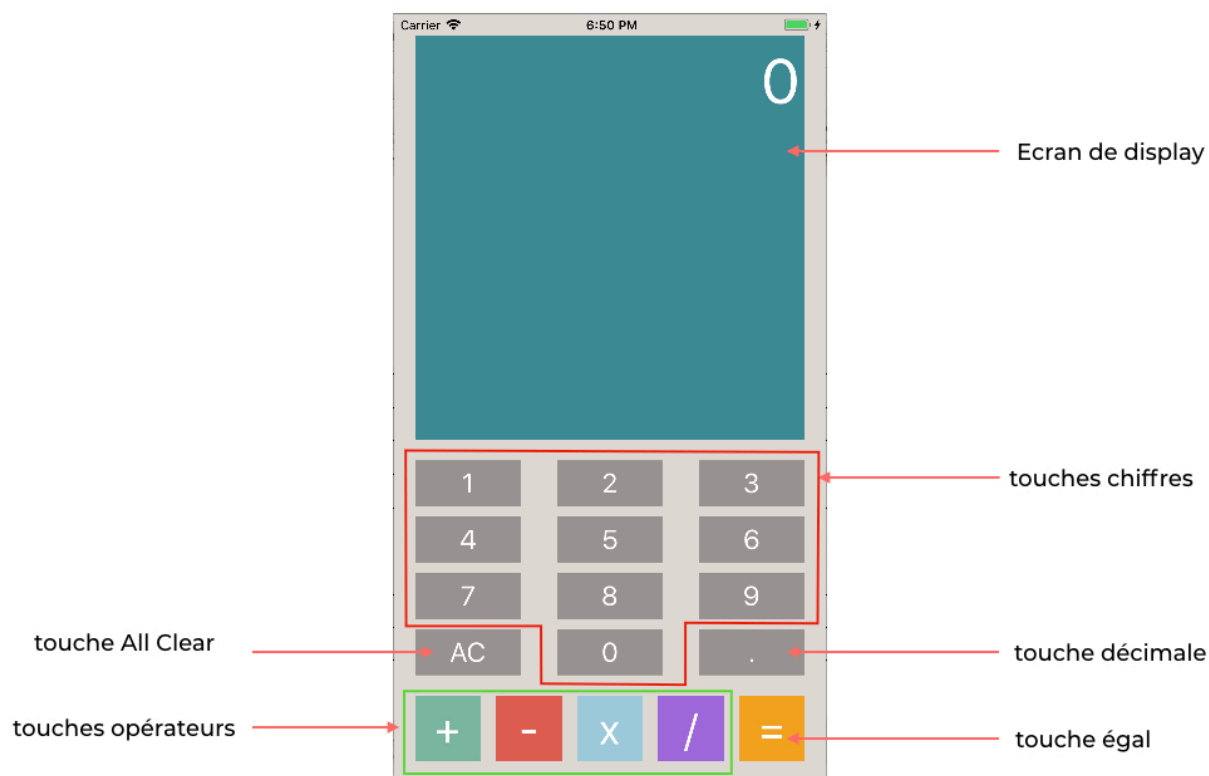
L'utilisateur entre un nombre, entier ou décimal, à l'aide du clavier puis un opérateur (+, -, /, x), ensuite il entre le deuxième nombre. Pour finir, il appuie sur le signe égal (« = ») et le résultat est présenté.

3.2 - Les acteurs

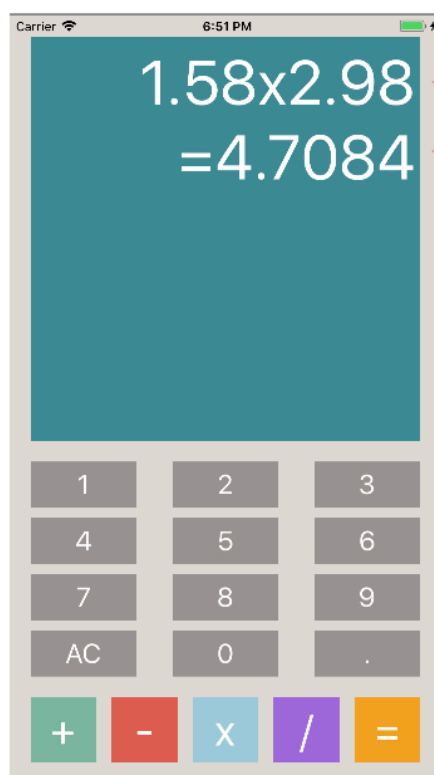
La solution est en interaction avec un seul acteur principal appelé UTILISATEUR.

3.3 - Les cas d'utilisation généraux

Interface graphique



Présentation opération



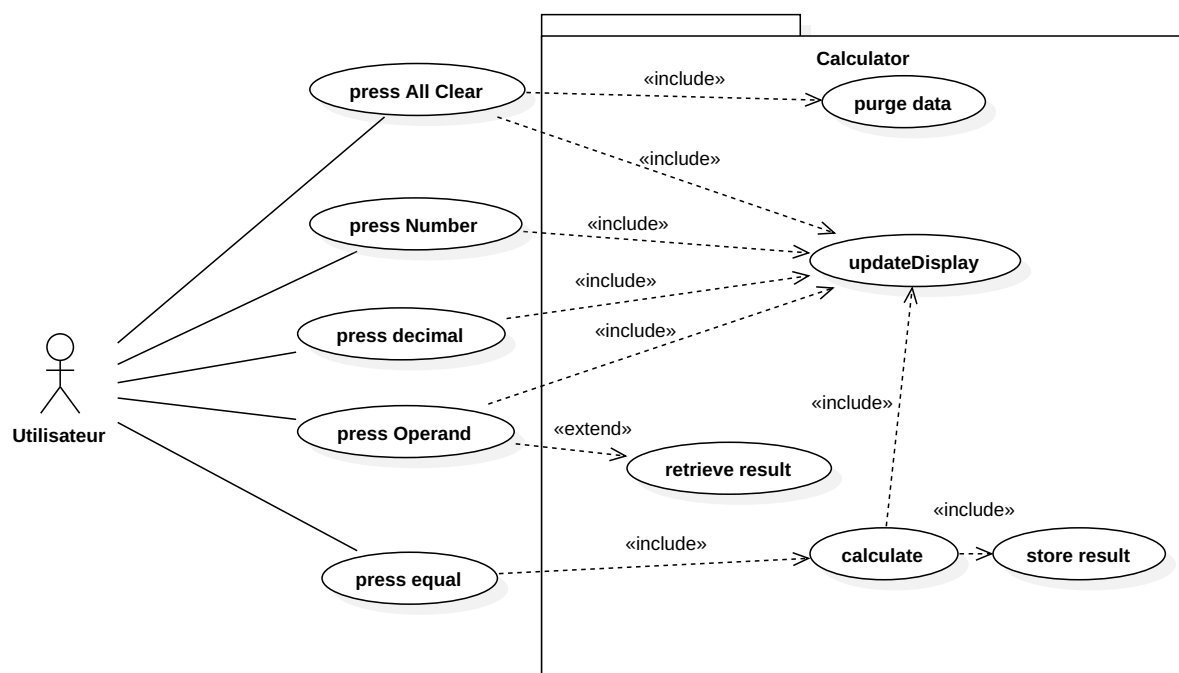
affichage opération

affichage résultat

3.4 - Les cas d'utilisation

3.4.1 - Package Calculatrice

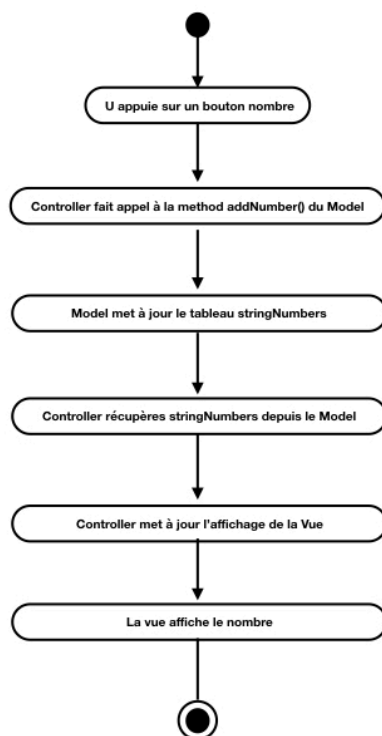
4 - La solution est constituée d'un seul package : le package calculatrice



5 - Diagramme UML des cas d'utilisation généraux.

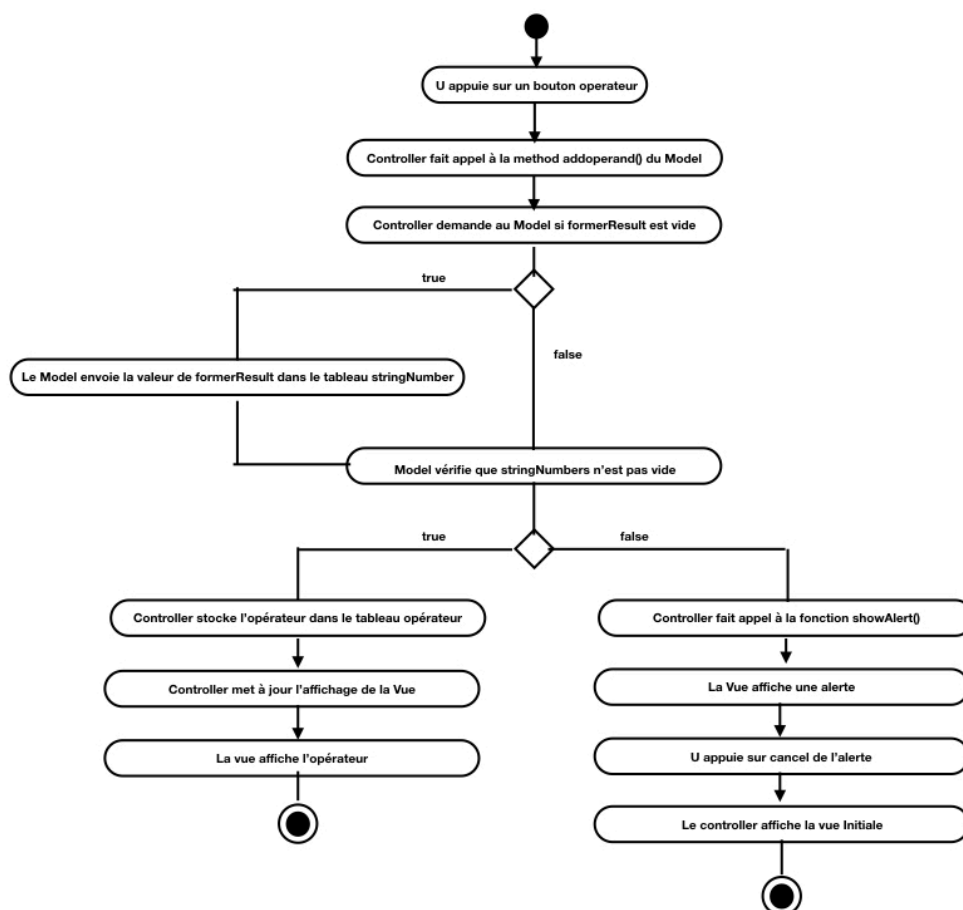
Le package Calculatrice contient toutes les fonctionnalités permettant d'effectuer les calculs, mémoriser des résultats et gérer l'affichage.

5.1.1.1 - UC1 – Cas d'utilisation Utilisateur appuie sur un bouton chiffre



Identifiant	UC1 – ajout chiffre – package Calculatrice
Description	Liste des actions réalisées pour afficher un chiffre
Pré-conditions	<ul style="list-style-type: none"> L'utilisateur est dans l'application
Données en entrée	<ul style="list-style-type: none">
Scénario nominal	<ol style="list-style-type: none"> U appuie sur le bouton Contrôleur fait appel à addNumber du Model Model met à jour la propriété stringNumbers Contrôleur récupère stringNumbers Contrôleur met à jour la vue
Résultat	La vue affiche le nombre
Erreurs	<ul style="list-style-type: none">

5.1.1.2 - UC2 – Cas d'utilisation Utilisateur appuie sur un bouton chiffre



Identifiant	UC2 – ajout opérateur – package Calculatrice
Description	Liste des actions pour ajouter un opérateur
Pré-conditions	<ul style="list-style-type: none"> L'utilisateur est dans l'application
Données en entrée	<ul style="list-style-type: none">
Scénario nominal	<ol style="list-style-type: none"> U appuie sur un bouton opérateur Contrôler vérifie former Result du model Contrôler vérifie si stringNumbers est vide Contrôler stocke l'opérateur dans le tableau opérateur Contrôler met à jour la vue
Résultat	La vue est mise à jour, l'opérateur apparaît sur l'écran
Erreurs	<ul style="list-style-type: none"> 2 et 3 sont faux le systeme présentent d'une alerte

5.1.1.2.1 Scénario alternatif :

Etape 2.1 : si formerResult n'est pas vide, le model donne la valeur de formerResult à stringNumbers

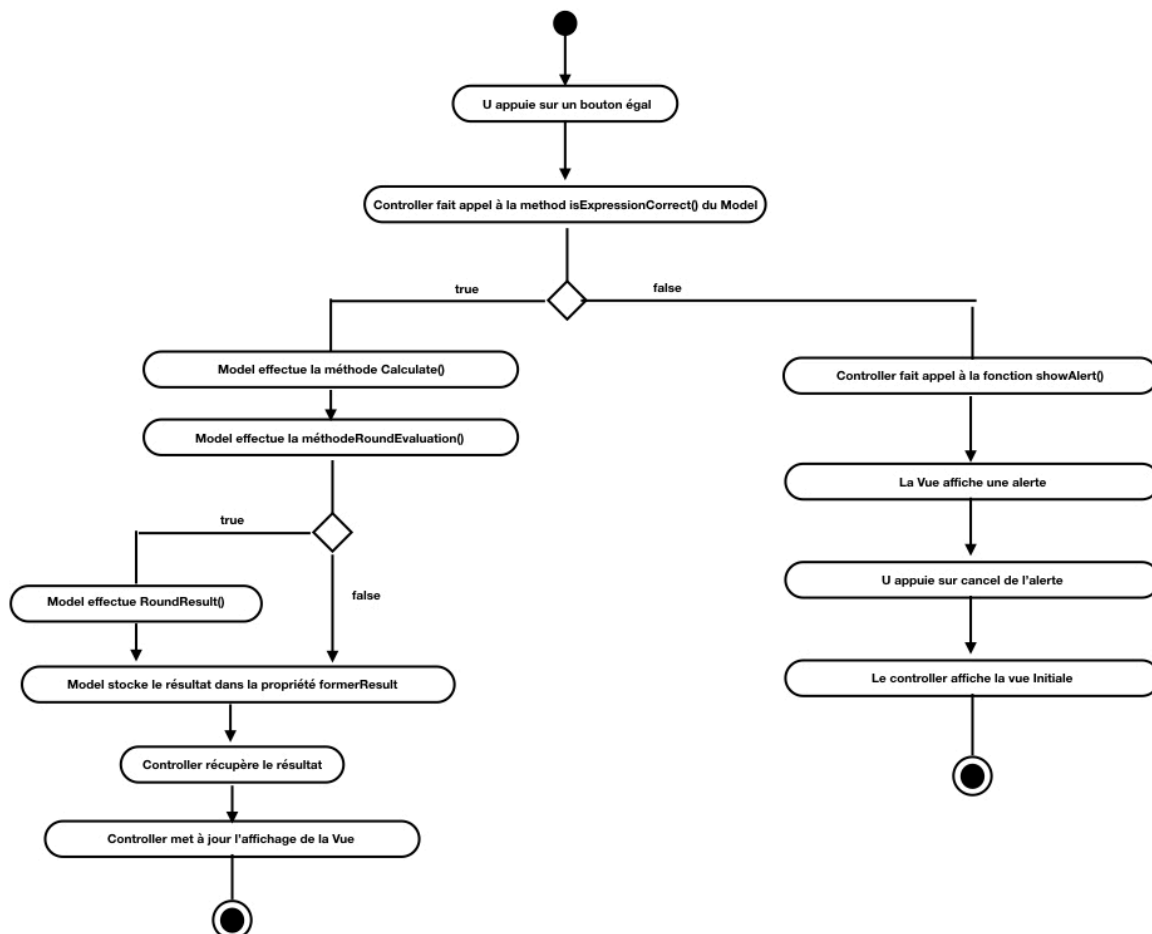
Etape 4.1 : si formerResult et stringNumbers sont vide

Etape 4.2 : Controller fait appel à showAlert()

Etape 4.3 : la Vue présente une alerte en popUp

...

5.1.1.3 - UC1 – Cas d'utilisation Utilisateur appuie sur un bouton chiffre



Identifiant	UC1 – Utilisateur appuie sur bouton chiffre – package Calculatrice
Description	Liste des actions nécessaires pour effectuer un calcul
Pré-conditions	<ul style="list-style-type: none"> L'utilisateur a entré 2 nombres et un opérateur
Données en entrée	<ul style="list-style-type: none"> 2 tableaux de String
Scénario nominal	<ol style="list-style-type: none"> U appuie sur la touche égale Controller évalue si les Pré-conditions sont remplies Controller fait appel à la fonction calculate() du Model Model renvoie un résultat et évalue s'il s'agit d'un entier ou non Model stocke le résultat dans la propriété formerResult Le Controller récupère le résultat La vue affiche le résultat
Résultat	Affichage du résultat
Erreurs	<ul style="list-style-type: none"> Si les préconditions ne sont pas remplies affichage d'une alerte

5.1.1.3.1 Scénario alternatif :

6 - Etape 2.1 : si les préconditions ne sont pas remplies

7 - Etape 3.1 : Controller fait appel à showAlert()

8 - Etape 4.1 : la Vue présente une alerte en popUp

9 - Etape 4.2 : si le résultat est un entier, le model fait appel la méthode roundResult() et renvoie un Integer et non un Double au controller