



AP-4 Proposition des solutions aux problématiques

Compétence validée :

- Analyser les objectifs et les modalités d'organisation d'un projet
- Planifier les activités
- Évaluer les indicateurs de suivi d'un projet et analyser les écarts

On a dû préparer le projet d'un point de vue conceptuel avant sa réalisation. Avec un outil de partage comme GitHub, Notion. Ensuite on devait faire la projection du projet, imaginez les risques et comment les contourner. On a utilisé beaucoup d'outils pour planifier nos activités comme un diagramme de Gantt, un diagramme de classe, un triangle de projet, un diagramme de cas d'utilisations et un diagramme de d'activité pour la connexion et l'achat.

Mais dans ce cas-là, on doit faire une application sous Flutter pour la gestion des produits par les administrateurs.

Triangle de projet:

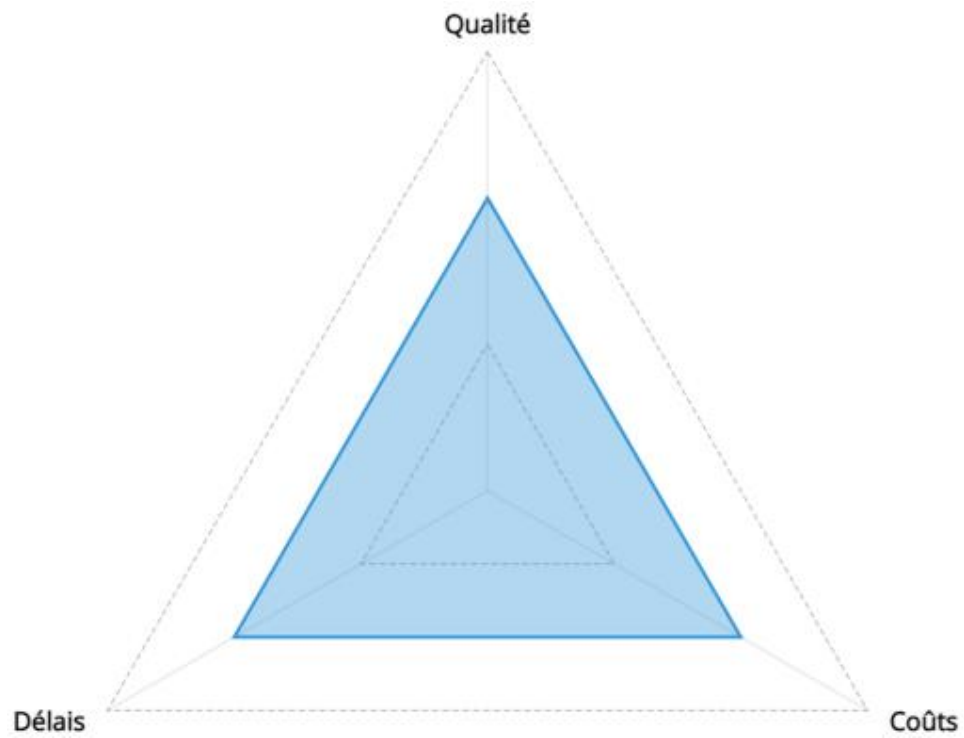


Diagramme de classe :

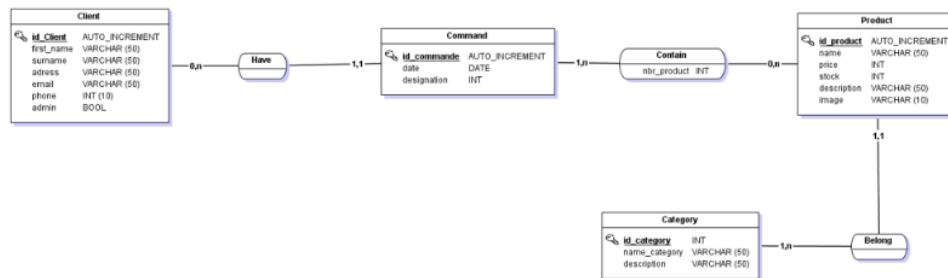


Diagramme de cas d'utilisation:

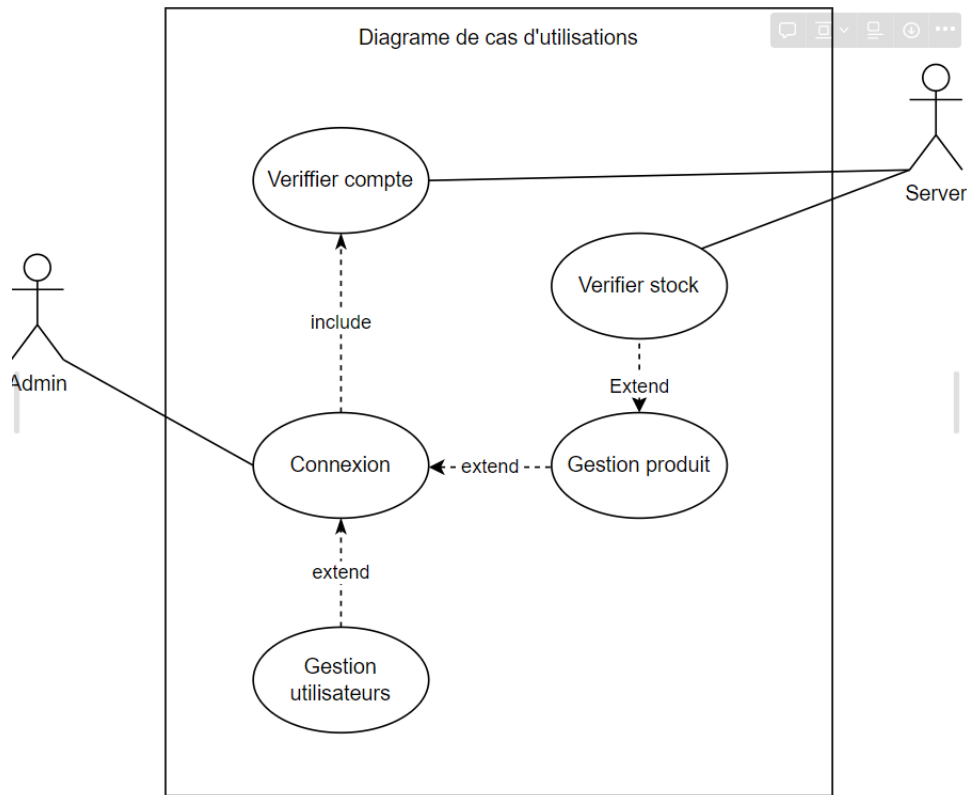


Diagramme d'activité pour la connexion d'un admin :

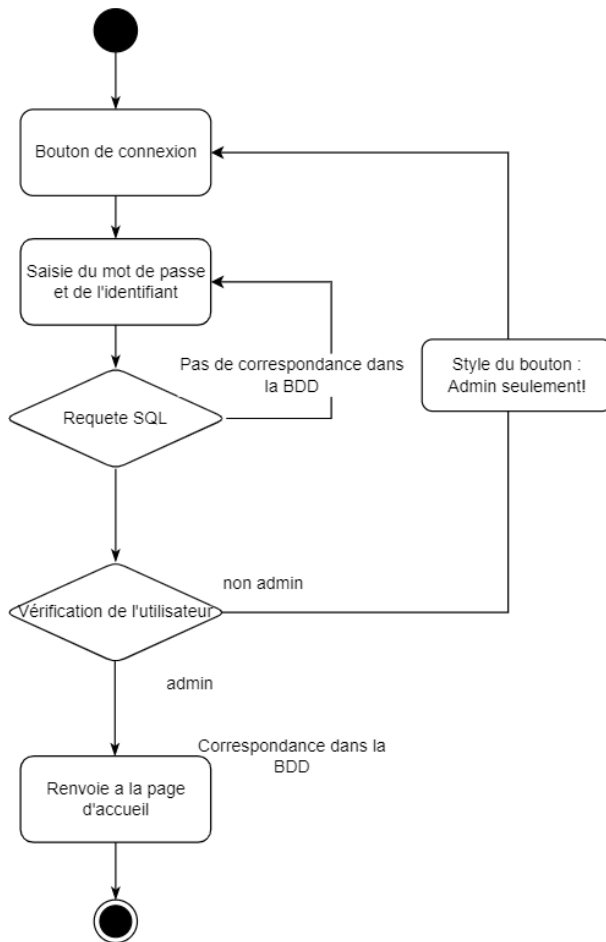
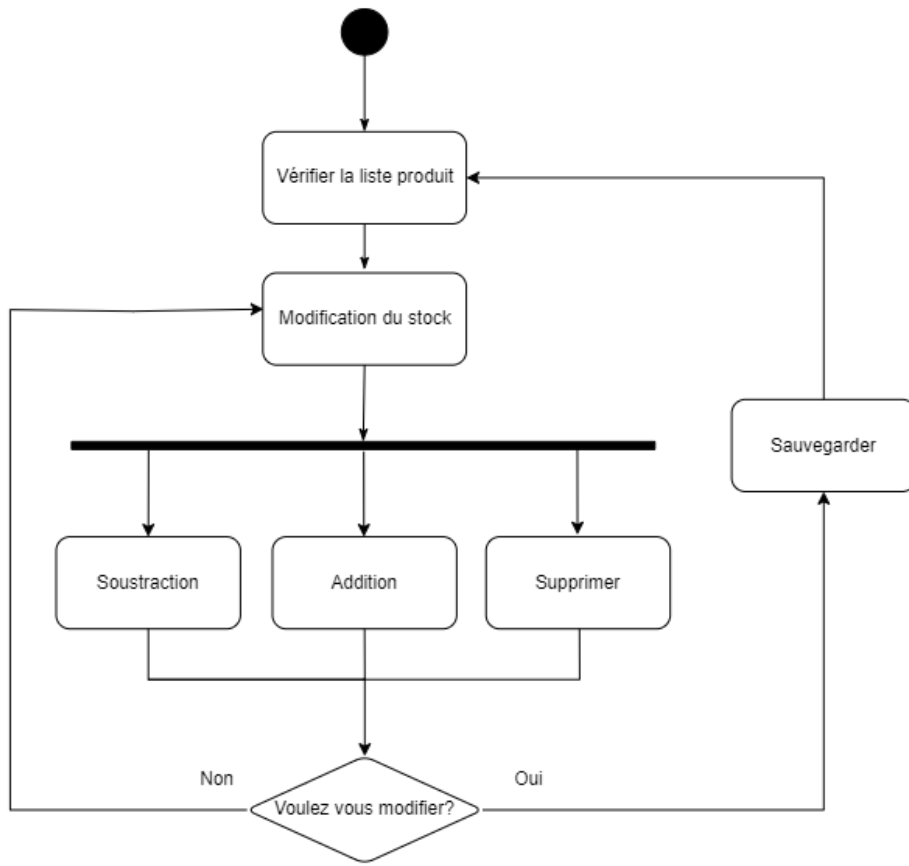


Diagramme d'activité pour la gestion de Stock :

...



Au niveau de la réalisation du code, pour l'instant, on utilise le langage Flutter/Dart (Inclure le CRUD plus tard)

M2L

DEBUT

Connexion :

Email

mot de passe

login

You, 3 minutes ago | 2 authors (You and others)

```
import 'package:flutter/material.dart';
import 'package:flutterapp/livre.dart';
import 'package:snippet_coder_utils/FormHelper.dart';
import 'package:snippet_coder_utils/ProgressHUD.dart';
```

You, 3 minutes ago | 2 authors (souchie88 and others)

```
class LoginPage extends StatefulWidget {
  const LoginPage({Key? key}) : super(key: key);

  @override
  State<LoginPage> createState() => _LoginPageState();
}
```

You, 3 minutes ago | 2 authors (You and others)

```
class _LoginPageState extends State<LoginPage> {
  bool isAPICallProcess = false;
  bool hidePassword = true;
  GlobalKey<FormState> globalFormKey = GlobalKey<FormState>();
  String? email;
  String? password;
  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        backgroundColor: Color(0xffffcfcda),
        body: Stack(
          children: [
            Positioned(
              top: 0
```