

Polytech Annecy	MODÈLE <b>Plan de test</b>	Projet: VPUB
Raguenes Masson		WP 5.3
IDU4		

## Plan de test

### RESUME

Ce document contient le plan de test de la bibliothèque Java VPUB.  
Le lien se trouve dans le tableau des livrables (page 5)

### HISTORIQUE DU DOCUMENT

Version	Modification	Date	Auteur
0.10	Version initiale	03-mar-21	Pierre
0.11	Version initiale avec première phase de tests	07-mar-21	Yoann Enzo

### DIFFUSION

Organisme	Nom	Mode de diffusion
	Équipe de développement	push github

Polytech Annecy	MODÈLE <b>Plan de test</b>	Projet: VPUB
Raguenes Masson		WP 5.3
IDU4		

## Introduction

### I.1 BUT

Ce plan de test pour le projet *VPUB* a les objectifs suivants :

Le projet est un Pub virtuel constitué d'une cave où sont stockées les réserves et d'un bar qui ne peut être approvisionné que par la cave. Ainsi toute livraison du Pub se fait directement dans la cave. Le seul point d'entrée de cette bibliothèque est le Pub, il peut être livré ou servir des boissons aux clients ( Remarque : aucune gestion financière ne vous est demandée dans cette version de « Virtual Pub » ). « Virtual Pub » doit être capable de gérer deux catégories de boissons : les boissons de base et les cocktails. Les boissons de base se subdivisent en trois catégories : les boissons sans alcool, celles avec alcool et pour finir les boissons chaudes. Les Cocktails quant à eux se décomposent en deux catégories : ceux sans et ceux avec alcool. Un Cocktail est un assemblage proportionné de différentes boissons de base et est considéré alcoolisé dès que l'un de ses éléments de base l'est.

En temps qu'équipe de test, notre but est de mettre en place un ensemble de tests unitaires et faire remonter de manière claire les bugs à l'équipe de développement.

### I.2 PORTÉE

Nous allons ici tester les classes suivantes :

- Boisson
- Cave
- Bar
- Cocktail

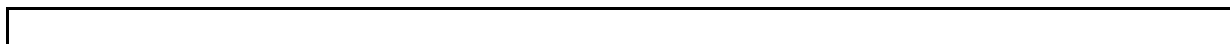
Dans ces classes nous allons tester les différents constructeurs et méthodes. Nous allons effectuer des tests boîtes noires et des tests à la limite.

### I.3 DOCUMENTS

La table ci-dessous indique les documents disponibles pour l'activité de test.

Document	Version	Disponible	Notes
Java-Doc		Oui	Incomplète, peu de descriptions

*Table 1 : Tableau des documents disponibles*



Polytech Annecy	MODÈLE <b>Plan de test</b>	Projet: VPUB
Raguènes Masson		WP 5.3
IDU4		

## La stratégie de test

### I.4 LES TYPES DE TEST

#### I.4.1 Les tests boîtes noires

<b>Objectif du test</b>	S'assurer du bon fonctionnement des classes décrites dans la Java-Doc
<b>Technique</b>	Utilisation de J-unit
<b>Critère d'achèvement</b>	
<b>Considération spéciale</b>	
<b>Responsable</b>	Yoann-Enzo

#### I.4.2 Les tests limites

<b>Objectif du test</b>	S'assurer que les méthodes ne peuvent pas dépasser certaines limites (ex : prix inférieur à 0)
<b>Technique</b>	Utilisation de J-Unit
<b>Critère d'achèvement</b>	
<b>Considération spéciale</b>	
<b>Responsable</b>	Yoann-Enzo

### I.5 LES OUTILS

Outil	Version	Commentaire
Java	8	Language
Eclipse IDE	2020-03	Logiciel IDE
EclEmma	3.1.3	plug-in
SonnarLint	5.0	plug-in
JUnit	5	Bibliothèque

Table 5 : Liste des outils utilisés par les tests

Polytech Annecy	<div> <div>MODÈLE</div> <div>Plan de test</div> </div>	Projet: VPUB
Raguenes Masson		WP 5.3
IDU4		

II Les ressources

II.1 LES RESSOURCES HUMAINES

Ressources humaines		
Personne	Rôle(s)	Responsabilité(s)
Yoann Raguenes	testeur	faire le plan de test et implémenter les tests
Enzo Masson	testeur 2	faire le plan de test et implémenter les tests

Table 6 : Tableau des ressources humaines

--

Polytech Annecy	MODÈLE <b>Plan de test</b>	Projet: VPUB
Raguenes Masson		WP 5.3
IDU4		

## Les livrables de l'activité de test

Délivrable	Destination	Origine	Date
Récapitulatif des tests			05/03/2021
Les tests implantés	<a href="#">GitHub</a>		05/03/2021
Rapport de couverture			05/03/2021
Rapport métriques de qualités			05/03/2021

### Levée de bug :

#### Classe Boisson:

Le constructeur de la classe permet de créer une boisson avec un nom et un degré d'alcool. Or il ne permet pas de prendre en compte les autres attributs (qui sont d'ailleurs définis en "public" au lieu de "private"). Par exemple, il sera impossible de définir le prix d'une boisson lors de sa création.

Dans le constructeur, le degré d'alcool est un float, ce qui n'est pas pratique car on doit écrire par exemple `boisson= new Boisson("bière", (float) 5)`

#### Classe Cocktail:

Il est impossible de paramétrer le boolean alcoolisé pour un cocktail

#### Classe Bar:

On ne peut pas ajouter une boisson chaude au bar (peut-être un choix)

#### Classe Cave:

Lorsqu'on demande au programme de prendre une boisson qui n'est pas dans la cave, ce dernier enlève une boisson aléatoire au lieu de nous prévenir que la boisson n'est pas dans la cave.

**La documentation n'est pas complète !**

**Tous les paramètres sont "public" ⇒ problème de sécurité.**

**Attention erreurs de français (ex : ingredients, alcoole ...)**