

Ce document présente l'analyse  
conception de l'application Vinci Thermo  
Green V3.2.0

# Analyse conception

Vinci Thermo Green

GUILET Julien

---

## Page de service

**Référence :** Vinci Thermo Green

**Plan de classement :** stadium-technic-analyse-conception-thermo-green

**Niveau de confidentialité :** confidential

### Mises à jour

Version	Date	Auteur	Description du changement
1.0.0	23/11/2020	GUILET Julien	Création du document
2.0.0	13/12/2020	GUILET Julien	Finalisation du document

## Table des matières

Page de service.....	1
Objet du document .....	2
Description complète du modèle .....	2
Cas d'utilisation « Modifier les températures supportées par le stade » .....	2
Acteur principal : .....	2
Objectifs : .....	2
Préconditions : .....	2
Postconditions : .....	2
Scénario nominal : .....	2
Extensions : .....	3
Diagramme scénario boîte noire .....	3
Diagramme classes métier .....	3
Diagramme expérience utilisateur .....	4
Diagramme scénario nominal détaillé .....	5

## Objet du document

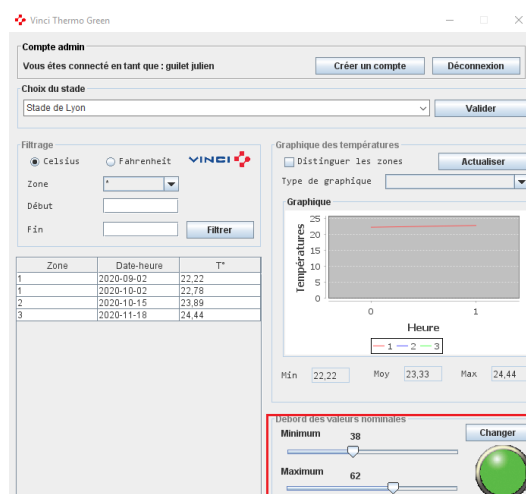
Ce document spécifie l'analyse conception de la version 3.2.0 de l'application Java qui permet d'alerter le gérant du stade par SMS si les températures étaient amenées à être anormales.

## Description complète du modèle

Dans cette version, un administrateur pourra modifier les températures minimums et maximums supportées d'un stade. De plus, le gérant du stade recevra une notification par SMS si les températures relevées n'étaient comprises entre les températures minimums et maximums.

### Cas d'utilisation « Modifier les températures supportées par le stade »

Dans cette version, aucune maquette n'est nécessaire étant donné que la modification à apporter sera sur une vue déjà existante. Voici cette vue :



L'encadré rouge correspond à l'interface où l'administrateur sera amené à modifier les températures.

Acteur principal :

Un administrateur.

Objectifs :

Modifier les températures maximums et minimums d'un stade.

Préconditions :

L'administrateur s'est authentifié et il a déjà sélectionné un stade.

Postconditions :

Les températures supportées par le stade ont été modifiées dans la base de données.

Scénario nominal :

1. Le système affiche un slider minimum et maximum avec au-dessus, la température sélectionnée.

2. L'administrateur bouge les curseurs du slider afin de mettre les températures souhaitées et clique sur le bouton « Changer ».

**3.** Le système enregistre dans la base de données les changements et lui affiche un message lui indiquant que l'opération s'est bien déroulée.

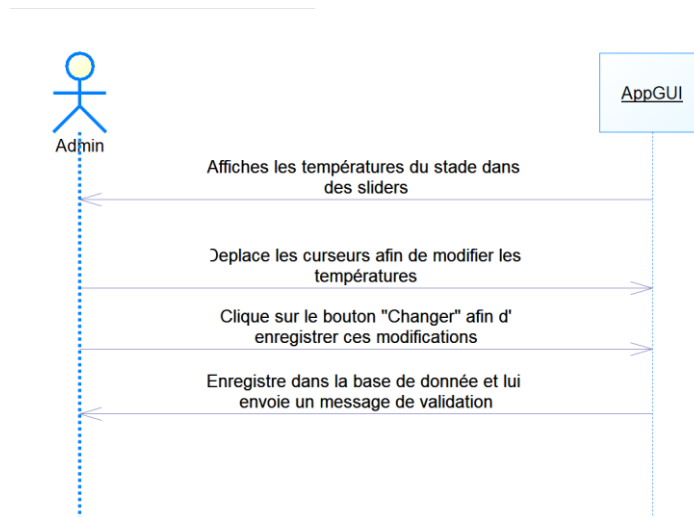
Extensions :

**3.1.** La modification dans la base de données ne s'est passée correctement.

a. Le système lui envoie un message d'erreur.

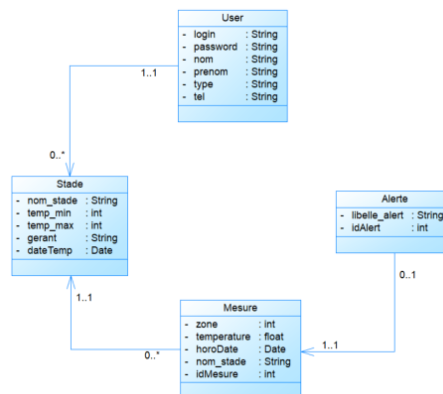
## Diagramme scénario boîte noire

Ce diagramme représente les interactions dans le scénario préciser précédemment.



## Diagramme classes métier

Le diagramme de classes ci-dessous décrit la persistance des données de l'application.



Dans cette version 3.2., une classe supplémentaire ainsi que quelque attribut ont été intégrés.

Pour commencer, dans la classe User, nous avons ajouté une propriété « tel », celui correspond au numéro de téléphone de l'utilisateur grâce à ce numéro de téléphone, nous pourrions lui envoyer une alerte si cet utilisateur est gérant d'un stade. De plus, comparé à la version précédente un user peut-être inscrit dans un stade. Cela permet de déterminer un gérant par stade.

Ensuite, dans la classe Stade quatre propriétés les premières sont « temp\_min » et « temp\_max » celle-ci correspond aux températures limite que soit supporté un stade.

Comme dit précédemment il y aura par stade un gérant, c'est pour cela que nous ajoutons cette propriété. Dans cette propriété, nous allons enregistrer le login de l'utilisateur gérant le stade.

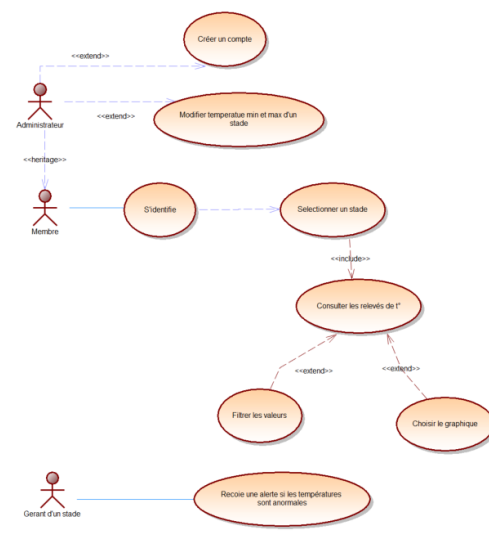
La dernière propriété de stade est « dateTemp », celle-ci permet de retenir la dernière fois que nous avons modifié les températures limites du stade. Cette propriété est là pour des raisons de fiabilité. En effet, si un administrateur baisse la température maximale d'un stade, mais que de ce fait dans la base de données une mesure possède maintenant une température supérieure à celle maximale, une notification sera donc envoyée, mais cette mesure date peut-être depuis plusieurs semaines ou mois. Il serait problématique de recevoir des notifications de problème sur des mesures datées. Donc pour contrer ce problème nous vérifierons, lors de la détection d'une anomalie, que la date et l'heure de la mesure sont supérieures à celle de la dernière modification des mesures du stade.

Ensuite, dans la classe Mesure, nous avons ajouté une propriété « idMesure » comme clé primaire afin de faciliter son enregistrement dans la nouvelle classe Alerte.

Cette nouvelle classe Alerte, contiendra toutes les mesures qui ont été détectées comme anormale et une notification sera donc envoyée aux gérants. La propriété « idAlert » contiendra l'id des mesures anormales et « libelle\_alert » contiendra le message qui sera envoyé par SMS au gérant.

## Diagramme expérience utilisateur

Le diagramme ci-dessous donne une vision synthétique des fonctionnalités attendues par l'application.



Dans cette nouvelle version de ce diagramme d'expérience utilisateur, un administrateur peut modifier les températures minimums et maximums d'un stade. De plus, un gérant recevra sur son téléphone un message lui indiquant si mesure enregistrée est anormale aux vues des températures minimums et maximums enregistrés.

## Diagramme scénario nominal détaillé

Voici le diagramme de séquence de l'application Vinci Thermo Green v3.2.0.

