Ce document présente la réalisation de l'application Vinci Thermo Green V3.2.0

Réalisation

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

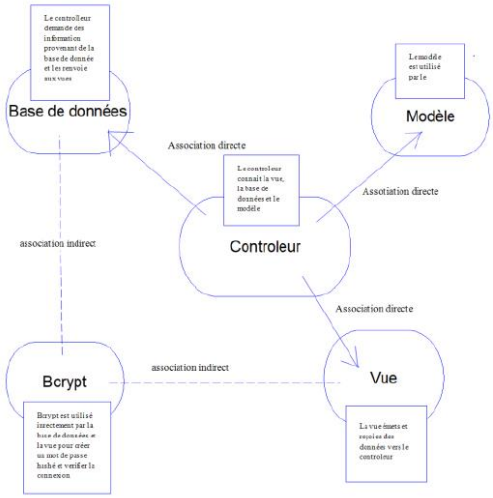
Page de service 1
Objet du document 2
Architecture 2
Changement des classes métiers 2
Donnee 2
getMinMaxTemp(String nomstade) 2
enregistreTemp(String nomstade, int min, int max) 3
verifAlert() 3

Objet du document

Ce document spécifie la réalisation 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.

Architecture

Voici l'architecture globale de la version 3.2.0 de l'application Vinci Thermo Green. Celle-ci n'a pas été modifier lors de ce changement de version.



Changement des classes métiers

Pour mener à bien cette nouvelle version de l'application, des ajouts ont été nécessaires.

Donnee

Dans cette version, trois fonctions ont été rajoutées.

getMinMaxTemp(String nomstade)

Cette fonction permet de récupérer les valeurs min et max de température en fonction du stade mis en paramètre. Elle renvoie une ArrayList<Integer> possédant deux Integer, le premier correspond à la valeur minimale du stade et second correspond à la valeur maximale.

Cette fonction sera utilisée lorsqu'un utilisateur sélectionne un stade, elle permettra de récupérer les valeurs min et max des températures et de les affecter aux deux sliders.

Voici la méthode :

```
public ArrayList<Float> getMinMaxTemp(String nomstade) throws SQLException{
		ArrayList<Float> minMax = new		<>();

		ResultSet resultset = null;
		resultset = statement.executeQuery("select temp_min, temp_max from stade
where nomStade = '" + nomstade + "';");
		resultset.next();
		minMax.	(resultset.getFloat("temp_min"));
		minMax.	(resultset.getFloat("temp_max"));


		return minMax;
	}
```

enregistreTemp(String nomstade, int min, int max)

Cette fonction permet d'enregistrer les nouvelles valeurs de min et max en fonction du stade, cela actualise également la date du changement. Elle retourne un Boolean, true si les requêtes d'update se sont déroulées sans erreurs, et False dans le cas contraire.

Cette fonction sera utilisée lorsqu'un administrateur change les températures supportées par le stade.

Voici la méthode :

```
public boolean engistreTemp(String nomstade,int min, int max) {
		try {
			statement.execute("  update stade set temp_min = '"+min+"' where
nomStade ='"+nomstade+"';");
			statement.execute("  update stade set temp_max = '"+max+"' where
nomStade ='"+nomstade+"';");
			statement.execute("  update stade set dateTemp =
'"+java.time.LocalDate.now()+" "+java.time.LocalTime.now()+"' where nomStade
='"+nomstade+"';");
			return true;
		} catch(SQLException e) {
			System.out.	(e);
			return false;
		}
	}
```

verifAlert()

Cette fonction permet de vérifier lors de la connexion d'un utilisateur, si dans la base donnée nous possédons une mesure qui dépasse les minimums ou maximums de température qui peut supporter un stade. De plus, elle envoie un SMS à tous les gérants possédant une mesure anormale, et enregistre une alerte dans la BDD. Elle retourne un Interger, il correspond au nombre d'alertes qui auront été détectées.

Cette fonction utilisée lorsque quelqu'un se connecte à l'application.

Voici la méthode :

```
public int verifAlerte() throws SQLException {
                ResultSet resultset = null;
                int nbAlert =0;
                ArrayList<Integer> lesMesuresAlerte = new          <>();
                ArrayList<String> lesStadesAlerte = new          <>();
                ArrayList<Integer> lesTemperaturesAlerte = new          <>();
                ArrayList<Integer> lesTempMin = new          <>();
                ArrayList<Integer> lesTempMax = new          <>();
                ArrayList<String> lesGerants = new          <>();


                resultset = statement.executeQuery("select idMesure, mesure.nomStade,
mesure.temperature, temp_min, temp_max , stade.gerant from mesure"
                                +" inner join stade on mesure.nomStade = stade.nomStade"
                                +" where mesure.horoDate >= stade.dateTemp and (temperature <
stade.temp_min OR temperature >  stade.temp_max) AND mesure.idMesure not in (select IdAlert from
alerte);");
                resultset.next();
                try {
                        lesMesuresAlerte.    (resultset.getInt("idMesure"));
                        lesStadesAlerte.    (resultset.getString("nomStade"));
                        lesTemperaturesAlerte.    (resultset.getInt("temperature"));
                        lesTempMin.    (resultset.getInt("temp_min"));
                        lesTempMax.    (resultset.getInt("temp_max"));
                        lesGerants.    (resultset.getString("gerant"));
                        for (int i = 0; i < lesMesuresAlerte.    (); i++) {

                                String message = "La mesure : "+lesMesuresAlerte.    (i)+" a été envoyé
à : "+lesGerants.    (i)+" concernant le stade : "+lesStadesAlerte.    (i)+" car il possedait une
temperature de : "+lesTemperaturesAlerte.    (i)+" alors que la température minimum est de :
"+lesTempMin.    (i)+" et maximum de : "+lesTempMax.    (i)+".";
                                System.out.        (message);
                                statement.execute("  insert into alerte VALUES "
                                                + "("+lesMesuresAlerte.    (i)+",'"+message+"');");
                                nbAlert++;

                                resultset = statement.executeQuery("select tel from user where login =
'"+lesGerants.    (i)+"';");
                                resultset.next();
                                String numTel = resultset.getString("tel");


                                //Envoie de message avec Twilio
                                Twilio.init("ACf8bd48b1d88b4cd64938e365bd165ea7",
"cabda4940afa91ef9dcae658bb0c31dd");
                                Message messagee = Message.creator(
                                new com.twilio.type.          (numTel),
                                new com.twilio.type.          ("+17073407596"),
                                message)
                                .      ();
                        System.out.        (messagee.      ());
                        }

                        return nbAlert;
                } catch(SQLException e) {
                        return nbAlert;
                }
        }
```