

## ETAPE 5 : ajout (création) d'une excursion

Utilisateur	Application THALI_MINI
Clique sur le bouton "ajouter une excursion"	
	Affiche la fenêtre de visualisation d'une excursion, les attributs sont initialisés avec les attributs de l'excursion sélectionnée. Aucune donnée n'est modifiable
Remplit les attributs Clique sur le bouton "OK" pour valider la saisie	
	Réalise les contrôles de validité : tous les champs sont obligatoires et compatibles avec leur format en base de données
	Cas 1 Si le contrôle de validité est conforme : - Enregistre les données en base - Ferme la fenêtre - La fenêtre principale est réinitialisée avec la nouvelle liste des excursions
	Cas 2 Si le contrôle de validité n'est pas conforme : - Affiche les erreurs rencontrées Conserve la fenêtre ouverte avec les données saisies

Utilisateur	Application THALI_MINI
Clique sur le bouton "ajouter une excursion"	
	Affiche la fenêtre de création d'une excursion, les attributs sont initialisés à vide
Remplit les attributs Clique sur le bouton "Annuler"	
	Ferme la fenêtre et retourne sur la fenêtre principale Aucun contrôle n'est réalisé, aucune donnée n'est modifié en base

### Utilisation bouton création d'expédition :

Le bouton de création d'expédition ouvre une fenêtre JDialog avec le code suivant :

```
private void jButtonAddExcursionActionPerformed(java.awt.event.ActionEvent evt) {
    // ouvrir la fenêtre de création d'expédition
    MiniExcursion excursionCourante = (MiniExcursion) jComboBoxExcursions.getSelectedItem();
    JDialogAjouterExcursion dialog = new JDialogAjouterExcursion(parent: this, modal: true);
    dialog.setVisible(b: true);
}
```

Dans la fenêtre **JDialogAjouterExcursion.java**, On va remplir les champs nécessaires pour ensuite les insérer dans la table **Excursion**

```

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)
    //</editor-fold>

    /* Create and display the dialog */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            JDialogAjouterExcursion dialog = new JDialogAjouterExcursion(new javax.swing.JFrame(), modal: true);
            dialog.addWindowListener(new java.awt.event.WindowAdapter() {
                @Override
                public void windowClosing(java.awt.event.WindowEvent e) {
                    System.exit(status: 0);
                }
            });
            dialog.setVisible(b: true);
        }
    });
}

private void ajouterExcursion(String code, String nom, int nbPlaces, double tarif) {

    String sql = "INSERT INTO Excursion (Code, Libelle, NbPlaces, Tarif) VALUES (?, ?, ?, ?)";

    try (Connection conn = ConnexionBDD.getConnexion();
        PreparedStatement stmt = conn.prepareStatement(sql)) {

        stmt.setString(1, code);
        stmt.setString(2, nom);
        stmt.setInt(3, nbPlaces);
        stmt.setDouble(4, tarif);
        stmt.executeUpdate();
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(parentComponent: this, "Erreur lors de l'ajout : " + e.getMessage(), title: "Erreur", messageType: JOptionPane.ERROR_MESSAGE);
    }
}

```

Le bouton **Annuler** ferme simplement la page, en revanche le bouton **OK** va vérifier les informations entrées et les envoyer à la table :

```

private void jButtonOkActionPerformed(java.awt.event.ActionEvent evt) {
    // valider l'ajout
    String code = jTextFieldCode.getText().trim();
    String nom = jTextFieldNom.getText().trim();
    String nbPlacesStr = jTextFieldNbPlaces.getText().trim();
    String tarifStr = jTextFieldTarif.getText().trim();

    if (code.isEmpty() || nom.isEmpty() || nbPlacesStr.isEmpty() || tarifStr.isEmpty()) {
        JOptionPane.showMessageDialog(parentComponent: this, message: "Veuillez remplir tous les champs", title: "Erreur", messageType: JOptionPane.ERROR_MESSAGE);
        return;
    }

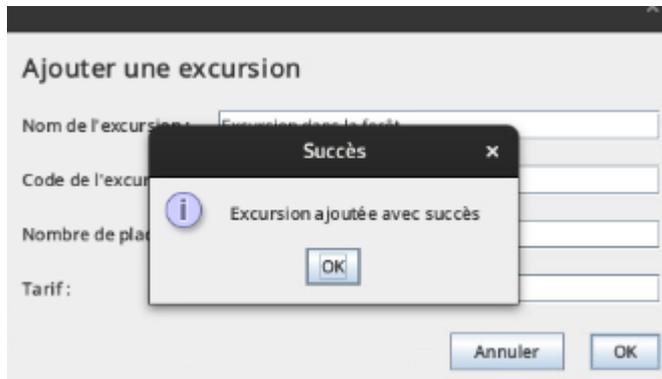
    try {
        int nbPlaces = Integer.parseInt(nbPlacesStr);
        double tarif = Double.parseDouble(tarifStr);
        ajouterExcursion(code, nom, nbPlaces, tarif);
        JOptionPane.showMessageDialog(parentComponent: this, message: "Excursion ajoutée avec succès", title: "Succès", messageType: JOptionPane.INFORMATION_MESSAGE);
    } catch (NumberFormatException e) {
        JOptionPane.showMessageDialog(parentComponent: this, message: "Nombre de places ou tarif invalide", title: "Erreur", messageType: JOptionPane.ERROR_MESSAGE);
    }

    // fermeture de la fenêtre après que les données soient enregistrées
    this.dispose();
}

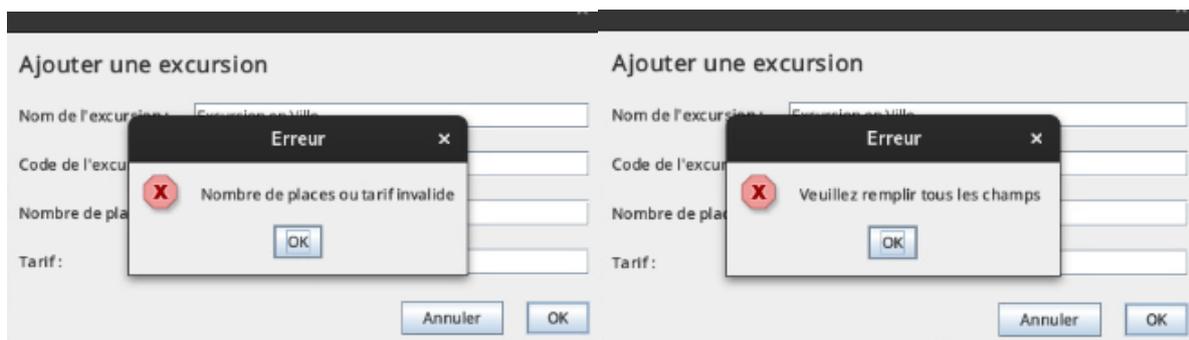
private void jButtonAnnulActionPerformed(java.awt.event.ActionEvent evt) {
    // Fermer la fenêtre
    this.dispose();
}

```

Lorsqu'on entre des informations valides dans les champs, un message précise que les données sont bien ajoutés et ferme directement la fenêtre



Plusieurs messages d'erreur sont prévus si les champs ne sont pas rempli correctement :



### [Informations supplémentaires :](#)

La fonctionnalité de rafraichissement de la fenêtre JFrame ne fonctionne pas malgré plusieurs tentatives de correction même avec d'autres camarades plus en avance sur le travail.