UNIVERSITY of GREENWICH

Mobile Application Design and Development

Logbook report

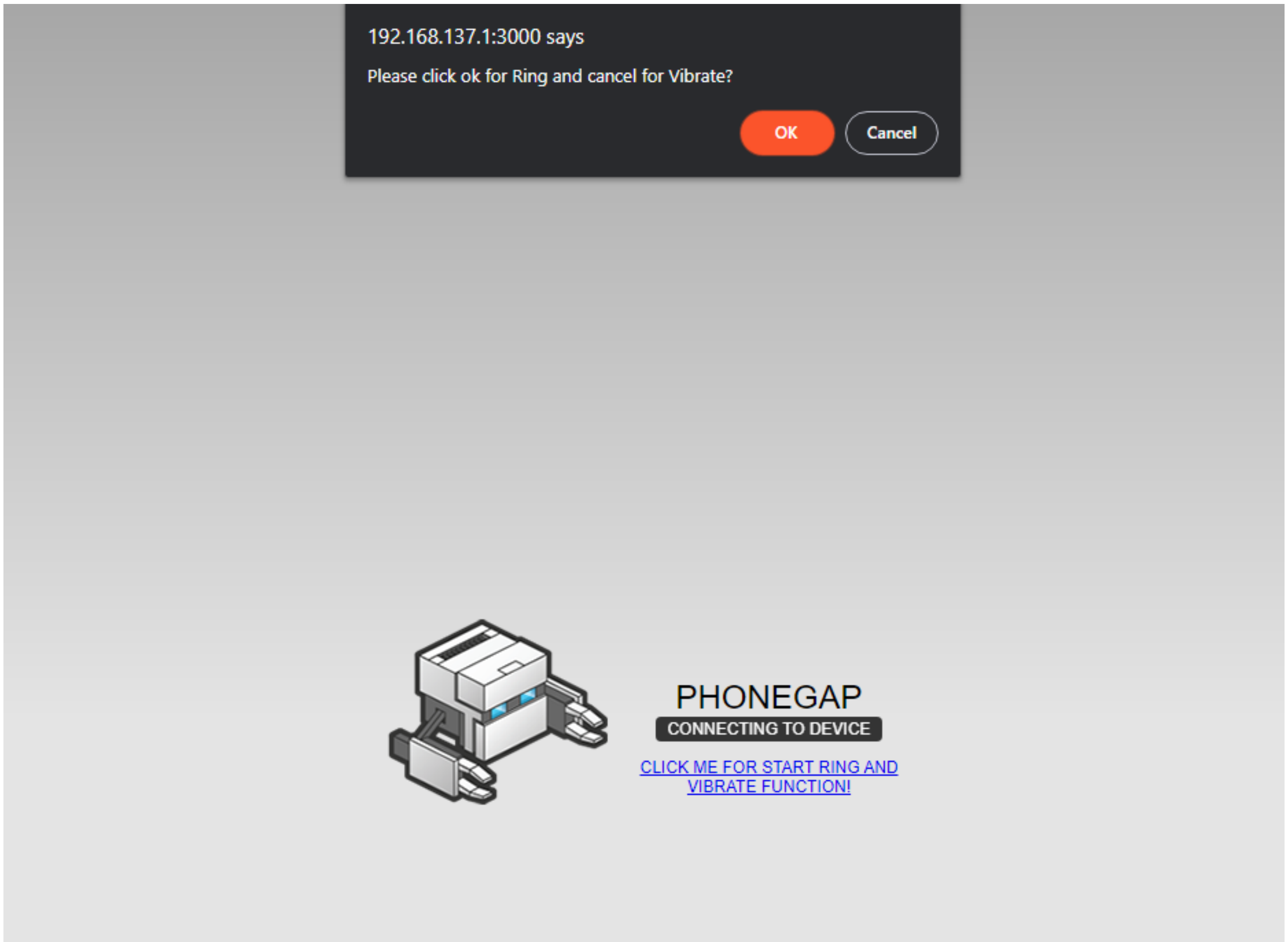
Course: **COMP1661**

Exercise 1: Notification buzzer

Basic Information

1.1 Student name	**Nguyen Tran Tien Phat** **Login id: np6380i**
1.2 Who did you work with? Note that for logbook exercises you are allowed to work with one other person as long as you give their name and login id and both contribute to the work. Both students need to provide description of their contribution.	**Name:** **Login id:**
1.3 Which Exercise is this? Tick as appropriate.	Create a PhoneGap application that uses the Notification API
1.4 How well did you complete the exercise? Tick as appropriate.	• I tried but couldn't complete it □ • I did it but I feel I should have done better □ • I did everything that was asked ☒ • I did more than was asked for □
1.5 Briefly explain your answer to question 1.4	A notification buzzer has been created. It will ring when the user clicks the ring button and vibrates when the vibrate button is pressed.

Exercise explanation



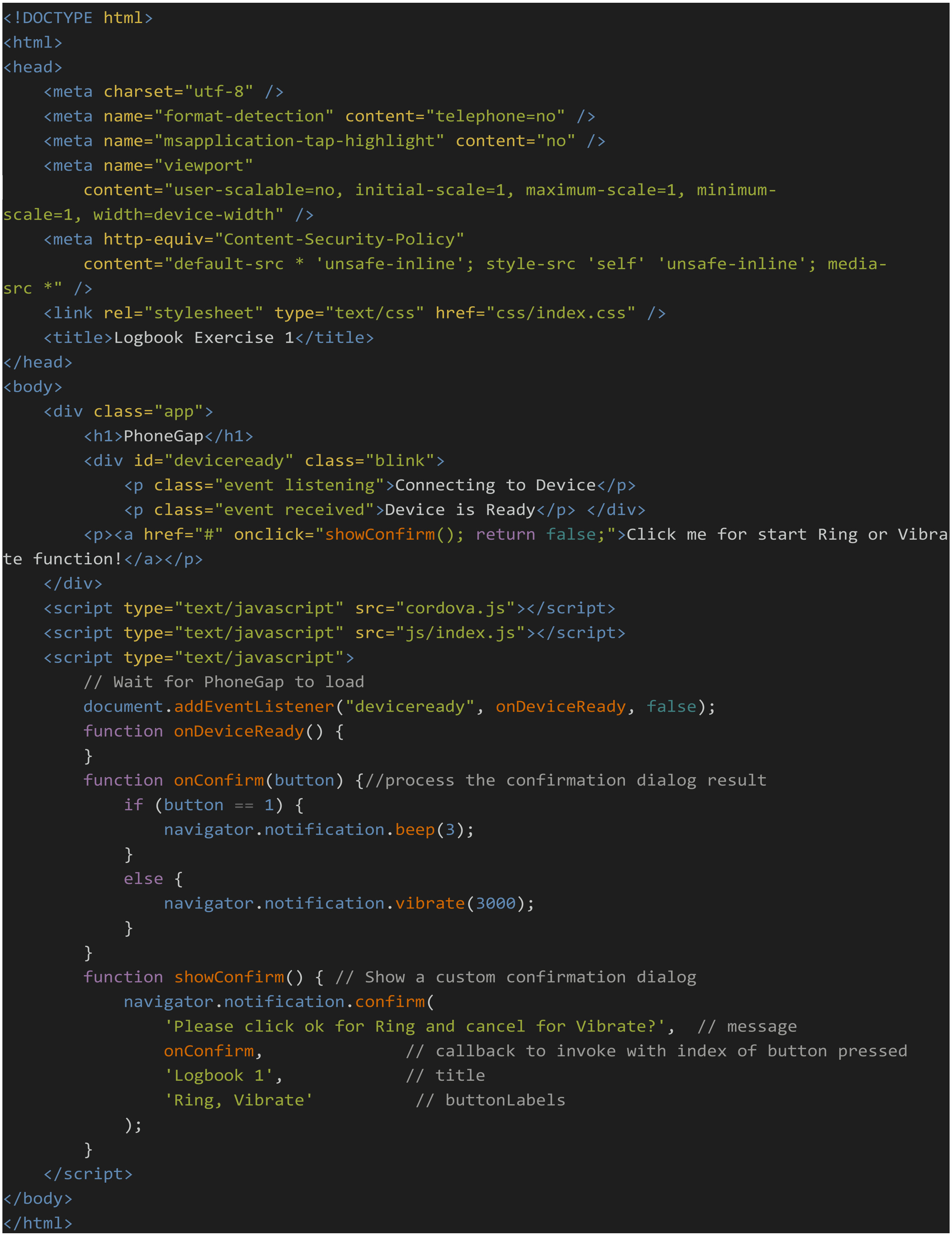
192.168.137.1:3000 says

Please click ok for Ring and cancel for Vibrate?

Figure 1 Result demo of Ring and Vibrate function

At this screen, users can click on 'Click me for start ring and vibrate function!' to enable the function. Then press Ok to Ring and Cancel to Vibrate.

Code Snippet



```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="format-detection" content="telephone=no" />
    <meta name="msapplication-tap-highlight" content="no" />
    <meta name="viewport"
        content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width" />
    <meta http-equiv="Content-Security-Policy"
        content="default-src * 'unsafe-inline'; style-src 'self' 'unsafe-inline'; media-src *" />
    <link rel="stylesheet" type="text/css" href="css/index.css" />
    <title>Logbook Exercise 1</title>
</head>
<body>
    <div class="app">
        <h1>PhoneGap</h1>
        <div id="deviceready" class="blink">
            <p class="event listening">Connecting to Device</p>
            <p class="event received">Device is Ready</p> </div>
        <p><a href="#" onclick="showConfirm(); return false;">Click me for start Ring or Vibrate function!</a></p>
    </div>
    <script type="text/javascript" src="cordova.js"></script>
    <script type="text/javascript" src="js/index.js"></script>
    <script type="text/javascript">
        // Wait for PhoneGap to load
        document.addEventListener("deviceready", onDeviceReady, false);
        function onDeviceReady() {
        }
        function onConfirm(button) {//process the confirmation dialog result
            if (button == 1) {
                navigator.notification.beep(3);
            }
            else {
                navigator.notification.vibrate(3000);
            }
        }
        function showConfirm() { // Show a custom confirmation dialog
            navigator.notification.confirm(
                'Please click ok for Ring and cancel for Vibrate?',  // message
                onConfirm,              // callback to invoke with index of button pressed
                'Logbook 1',            // title
                'Ring, Vibrate'          // buttonLabels
            );
        }
    </script>
</body>
</html>
```

Figure 2 Code snippet for exercise1

In this code, after the user clicks on 'Click me for start ring and vibrate function!' the system will call the *showConfirm*() function to show confirm notification for the user to choose. The *showConfirm* () function will then call the function *onConfirm*(). If the user clicks Ok, the system will call *navigator.notification.beep*(3); to activate the ringtone, or if the user presses Cancel, the system will call *navigator.notification.vibrate*(3000); to activate vibrate.

Exercise 2: Data entry screen (created by PhoneGap)

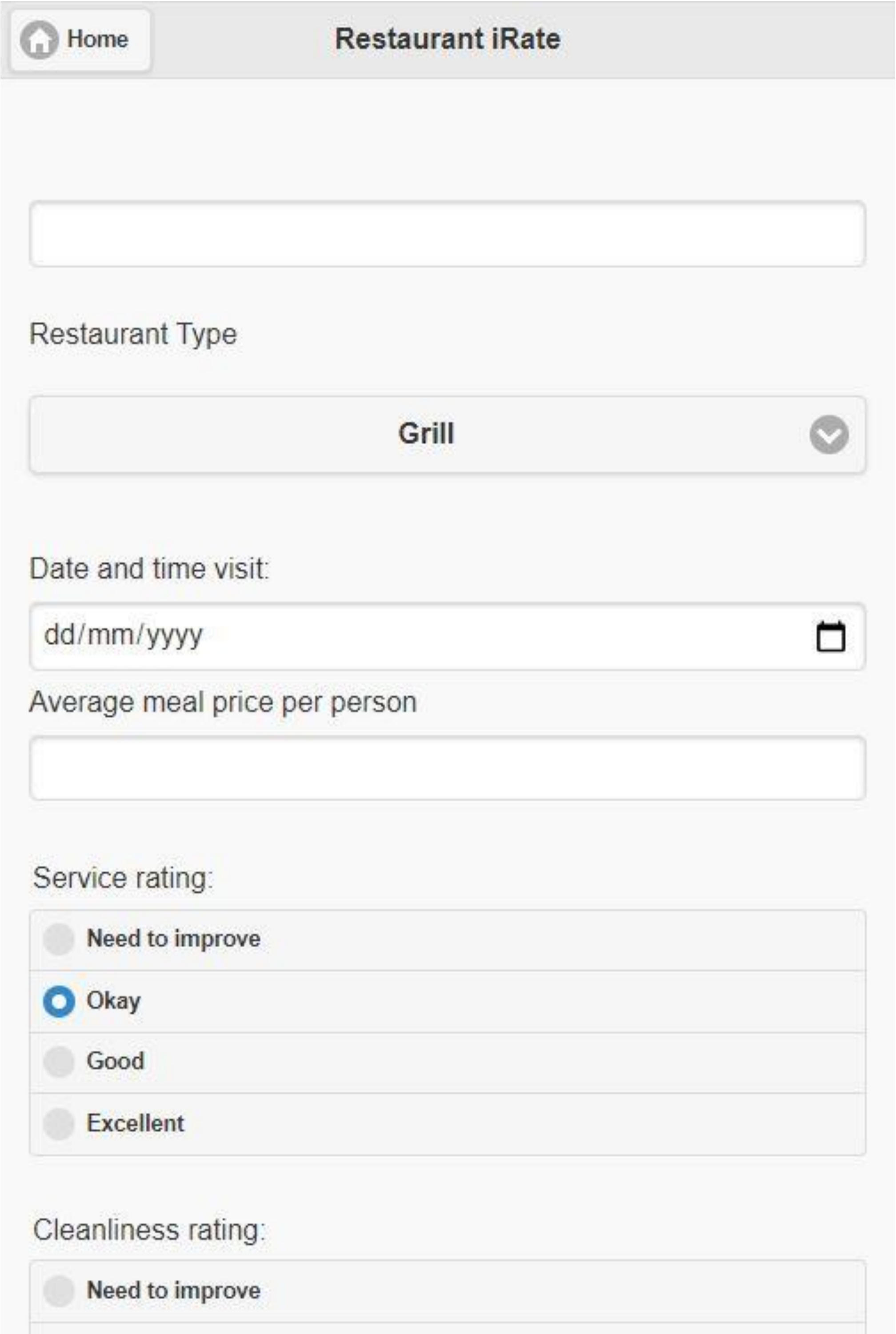
Basic Information

1.1 Student name	**Nguyen Tran Tien Phat** **Login id: np6380i**
1.2 Who did you work with? Note that for logbook exercises you are allowed to work with one other person as long as you give their name and login id and both contribute to the work. Both students need to provide description of their contribution.	**Name:** **Login id:**
1.3 Which Exercise is this? Tick as appropriate.	Creating a form display with Phonegap allows users to enter data and some validations and error messages about the data that the user entered.
1.4 How well did you complete the exercise? Tick as appropriate.	• I tried but couldn't complete it ☐ • I did it but I feel I should have done better ☐ • I did everything that was asked ☒ • I did more than was asked for ☐

1.5 Briefly explain your answer to question 1.4	An application includes a data entry screen created via Phonegap, which allows the user to enter data and select the information they want. Some validation for the entered fields is also implemented. These functions will be more detailed in the following sections.

Exercise explanation

Demo results



Home

Restaurant iRate

Restaurant Type

Grill

Date and time visit:

dd/mm/yyyy

Average meal price per person

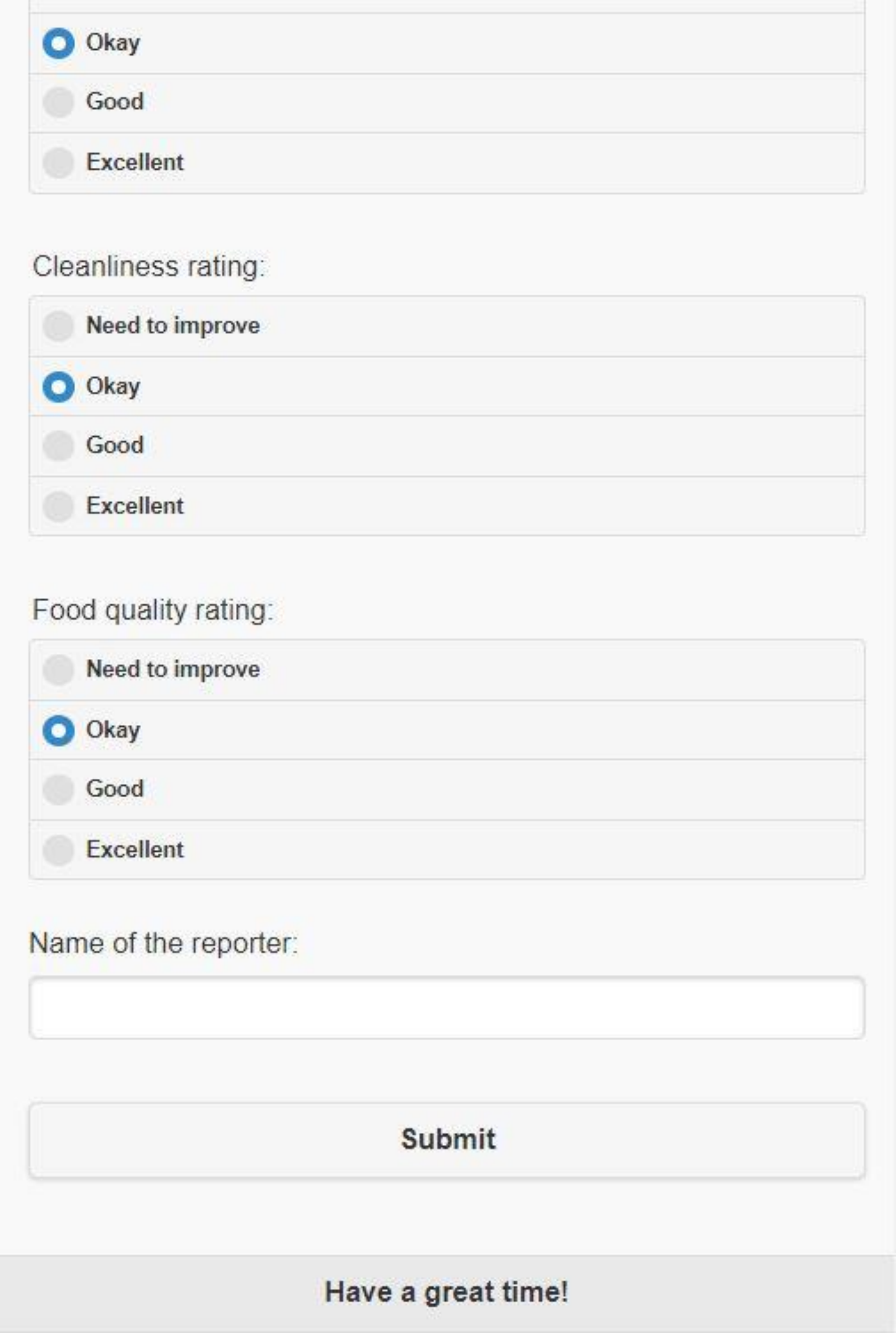
Service rating:

○ Need to improve
● Okay
○ Good
○ Excellent

Cleanliness rating:

○ Need to improve

Figure 3 Data entry screen by PhoneGap



● Okay

○ Good

○ Excellent

Cleanliness rating:

○ Need to improve

● Okay

○ Good

○ Excellent

Food quality rating:

○ Need to improve

● Okay

○ Good

○ Excellent

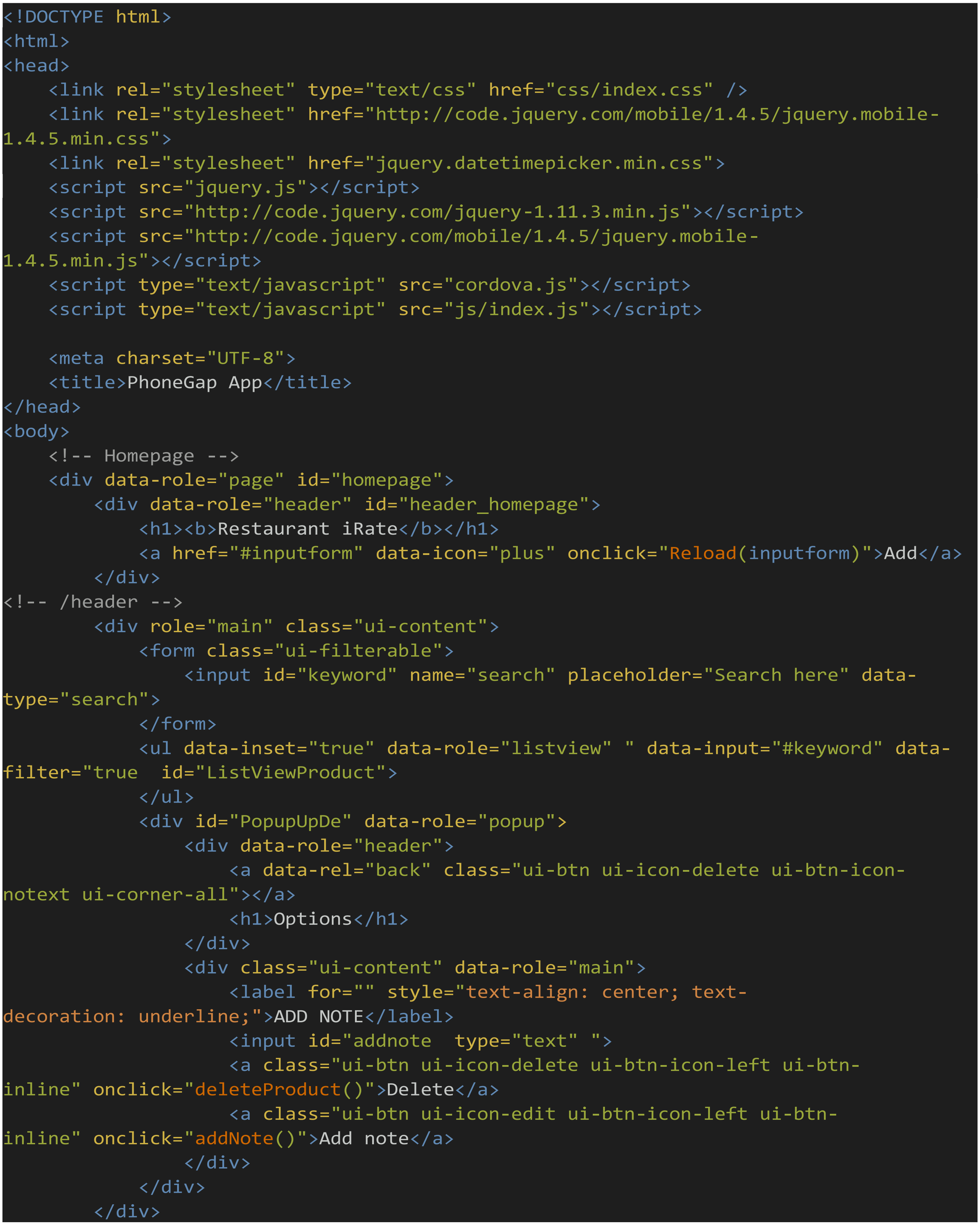
Name of the reporter:

Have a great time!

Figure 4 Data entry screen by PhoneGap (cont)

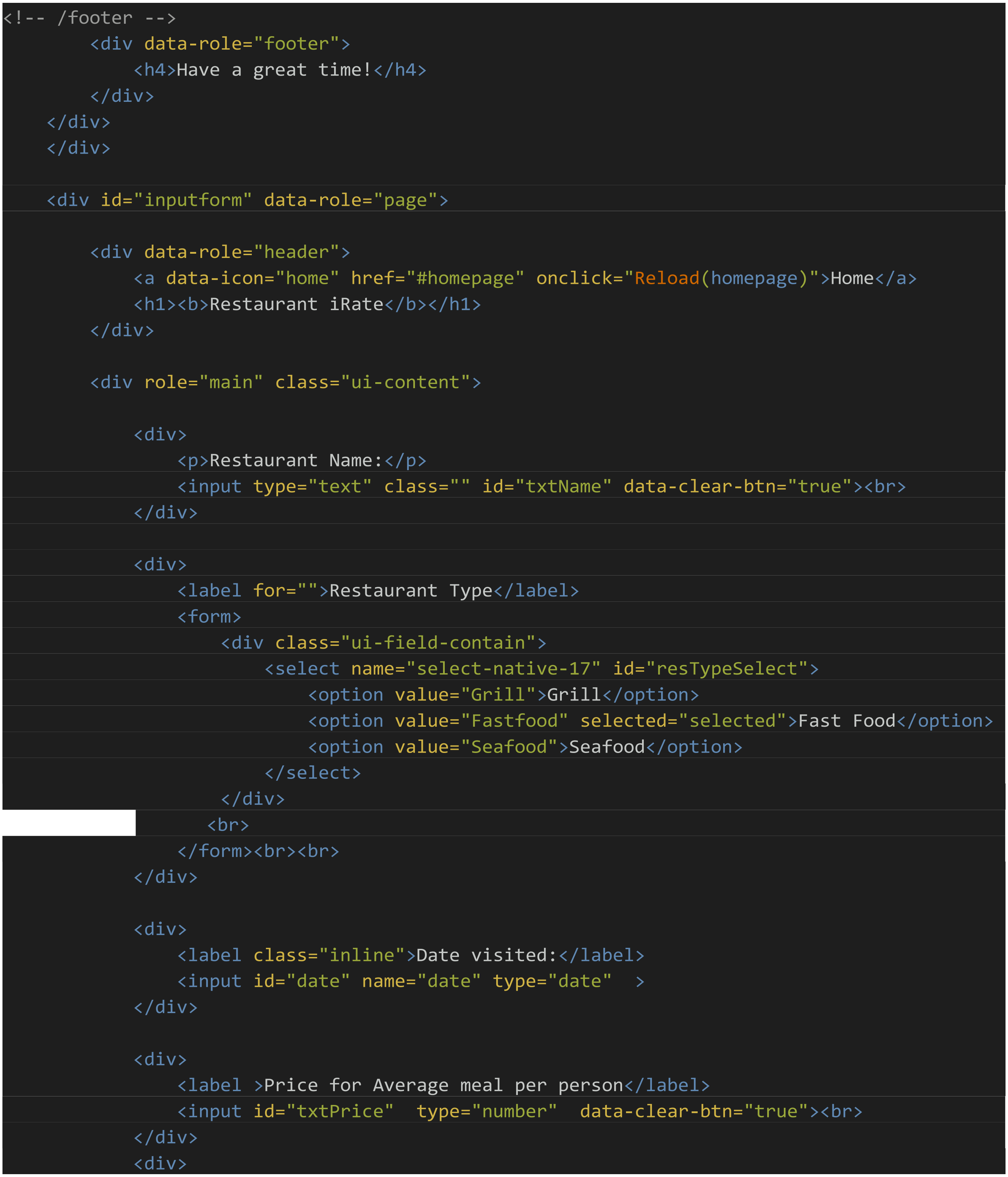
This screen is designed with clearly components and labels and in easy-to-understand language. As a result, users can easily understand what is everything they need to do on the app. In addition, the text is also enlarged to ensure visibility.

Code Snippet



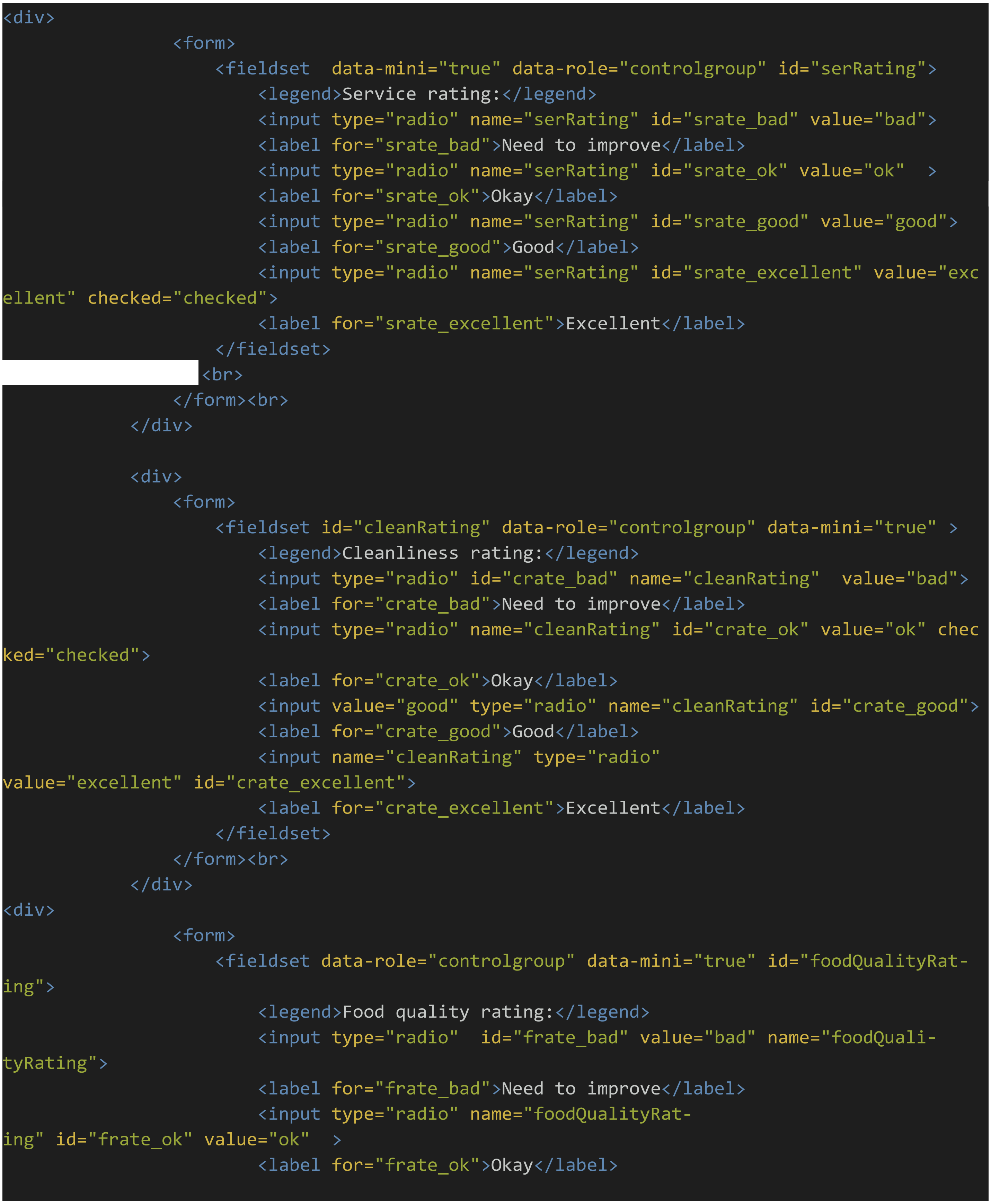
```html
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="css/index.css" />
    <link rel="stylesheet" href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
    <link rel="stylesheet" href="jquery.datetimepicker.min.css">
    <script src="jquery.js"></script>
    <script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
    <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>
    <script type="text/javascript" src="cordova.js"></script>
    <script type="text/javascript" src="js/index.js"></script>

    <meta charset="UTF-8">
    <title>PhoneGap App</title>
</head>
<body>
    <!-- Homepage -->
    <div data-role="page" id="homepage">
        <div data-role="header" id="header_homepage">
            <h1><b>Restaurant iRate</b></h1>
            <a href="#inputform" data-icon="plus" onclick="Reload(inputform)">Add</a>
        </div>
<!-- /header -->
        <div role="main" class="ui-content">
            <form class="ui-filterable">
                <input id="keyword" name="search" placeholder="Search here" data-type="search">
            </form>
            <ul data-inset="true" data-role="listview" " data-input="#keyword" data-filter="true  id="ListViewProduct">
            </ul>
            <div id="PopupUpDe" data-role="popup">
                <div data-role="header">
                    <a data-rel="back" class="ui-btn ui-icon-delete ui-btn-icon-notext ui-corner-all"></a>
                    <h1>Options</h1>
                </div>
                <div class="ui-content" data-role="main">
                    <label for="" style="text-align: center; text-decoration: underline;">ADD NOTE</label>
                    <input id="addnote  type="text" ">
                    <a class="ui-btn ui-icon-delete ui-btn-icon-left ui-btn-inline" onclick="deleteProduct()">Delete</a>
                    <a class="ui-btn ui-icon-edit ui-btn-icon-left ui-btn-inline" onclick="addNote()">Add note</a>
                </div>
            </div>
        </div>
```

Figure 5 Code snippet for data entry screen



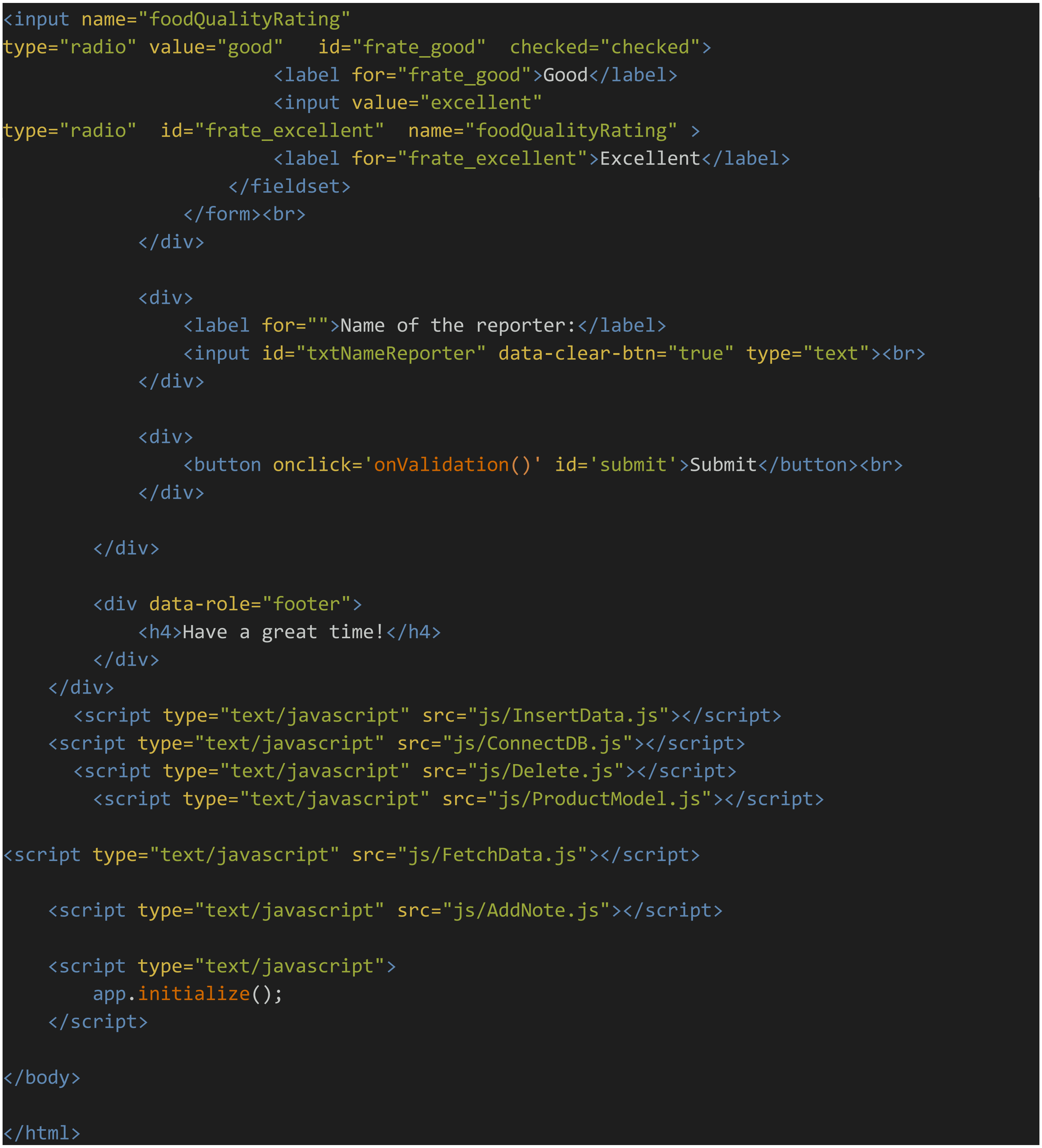
```
<!-- /footer -->
        <div data-role="footer">
            <h4>Have a great time!</h4>
        </div>
    </div>
    </div>

    <div id="inputform" data-role="page">

        <div data-role="header">
            <a data-icon="home" href="#homepage" onclick="Reload(homepage)">Home</a>
            <h1><b>Restaurant iRate</b></h1>
        </div>

        <div role="main" class="ui-content">

            <div>
                <p>Restaurant Name:</p>
                <input type="text" class="" id="txtName" data-clear-btn="true"><br>
            </div>

            <div>
                <label for="">Restaurant Type</label>
                <form>
                    <div class="ui-field-contain">
                        <select name="select-native-17" id="resTypeSelect">
                            <option value="Grill">Grill</option>
                            <option value="Fastfood" selected="selected">Fast Food</option>
                            <option value="Seafood">Seafood</option>
                        </select>
                    </div>
                   <br>
                </form><br><br>
            </div>

            <div>
                <label class="inline">Date visited:</label>
                <input id="date" name="date" type="date"   >
            </div>

            <div>
                <label >Price for Average meal per person</label>
                <input id="txtPrice"  type="number"  data-clear-btn="true"><br>
            </div>
            <div>
```

Figure 6 Code snippet for data entry screen (cont)



```
<div>
                <form>
                    <fieldset  data-mini="true" data-role="controlgroup" id="serRating">
                        <legend>Service rating:</legend>
                        <input type="radio" name="serRating" id="srate_bad" value="bad">
                        <label for="srate_bad">Need to improve</label>
                        <input type="radio" name="serRating" id="srate_ok" value="ok"  >
                        <label for="srate_ok">Okay</label>
                        <input type="radio" name="serRating" id="srate_good" value="good">
                        <label for="srate_good">Good</label>
                        <input type="radio" name="serRating" id="srate_excellent" value="exc
ellent" checked="checked">
                        <label for="srate_excellent">Excellent</label>
                    </fieldset>
                   <br>
                </form><br>
            </div>

            <div>
                <form>
                    <fieldset id="cleanRating" data-role="controlgroup" data-mini="true" >
                        <legend>Cleanliness rating:</legend>
                        <input type="radio" id="crate_bad" name="cleanRating"  value="bad">
                        <label for="crate_bad">Need to improve</label>
                        <input type="radio" name="cleanRating" id="crate_ok" value="ok" chec
ked="checked">
                        <label for="crate_ok">Okay</label>
                        <input value="good" type="radio" name="cleanRating" id="crate_good">
                        <label for="crate_good">Good</label>
                        <input name="cleanRating" type="radio"
value="excellent" id="crate_excellent">
                        <label for="crate_excellent">Excellent</label>
                    </fieldset>
                </form><br>
            </div>
<div>
                <form>
                    <fieldset data-role="controlgroup" data-mini="true" id="foodQualityRat-
ing">
                        <legend>Food quality rating:</legend>
                        <input type="radio"  id="frate_bad" value="bad" name="foodQuali-
tyRating">
                        <label for="frate_bad">Need to improve</label>
                        <input type="radio" name="foodQualityRat-
ing" id="frate_ok" value="ok"  >
                        <label for="frate_ok">Okay</label>
```

Figure 7 Code snippet for data entry screen (cont)



```
<input name="foodQualityRating"
type="radio" value="good"   id="frate_good"  checked="checked">
                        <label for="frate_good">Good</label>
                        <input value="excellent"
type="radio"  id="frate_excellent"  name="foodQualityRating" >
                        <label for="frate_excellent">Excellent</label>
                    </fieldset>
                </form><br>
            </div>

            <div>
                <label for="">Name of the reporter:</label>
                <input id="txtNameReporter" data-clear-btn="true" type="text"><br>
            </div>

            <div>
                <button onclick='onValidation()' id='submit'>Submit</button><br>
            </div>

        </div>

        <div data-role="footer">
            <h4>Have a great time!</h4>
        </div>
    </div>
      <script type="text/javascript" src="js/InsertData.js"></script>
    <script type="text/javascript" src="js/ConnectDB.js"></script>
      <script type="text/javascript" src="js/Delete.js"></script>
        <script type="text/javascript" src="js/ProductModel.js"></script>

<script type="text/javascript" src="js/FetchData.js"></script>

    <script type="text/javascript" src="js/AddNote.js"></script>

    <script type="text/javascript">
        app.initialize();
    </script>

</body>

</html>
```

Figure 8 Code snippet for data entry screen (cont)

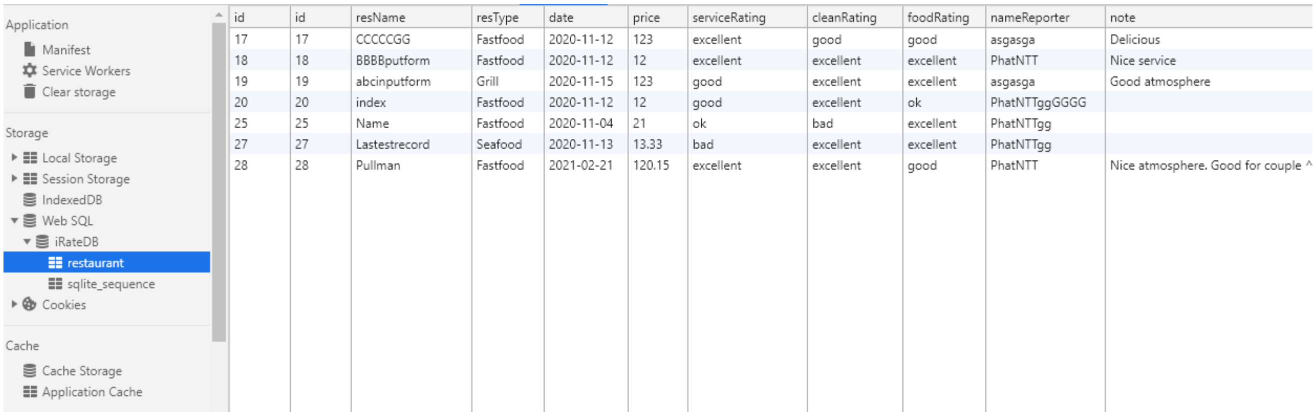
Exercise 3: Connect database for restaurant detail storing

Basic Information

1.1 Student name	**Nguyen Tran Tien Phat** **Login id: np6380i**
1.2 Who did you work with? Note that for logbook exercises you are allowed to work with one other person as long as you give their name and login id and both contribute to the work. Both students need to provide description of their contribution.	**Name:** **Login id:**
1.3 Which Exercise is this? Tick as appropriate.	Create an SQLite database to store the event details entered into the i-Rate App
1.4 How well did you complete the exercise? Tick as appropriate.	• I tried but couldn't complete it □ • I did it but I feel I should have done better □ • I did everything that was asked ☒ • I did more than was asked for □
1.5 Briefly explain your answer to question 1.4	A database is created on the SQLite database administration system, which stores details of the restaurants that a user has rated in the i-Rate application.

Exercise explanation

Demo results



id	id	resName	resType	date	price	serviceRating	cleanRating	foodRating	nameReporter	note
17	17	CCCCCGG	Fastfood	2020-11-12	123	excellent	good	good	asgasga	Delicious
18	18	BBBBputform	Fastfood	2020-11-12	12	excellent	excellent	excellent	PhatNTT	Nice service
19	19	abcinputform	Grill	2020-11-15	123	good	excellent	excellent	asgasga	Good atmosphere
20	20	index	Fastfood	2020-11-12	12	good	excellent	ok	PhatNTTggGGGG	
25	25	Name	Fastfood	2020-11-04	21	ok	bad	excellent	PhatNTTgg	
27	27	Lastestrecord	Seafood	2020-11-13	13.33	bad	excellent	excellent	PhatNTTgg	
28	28	Pullman	Fastfood	2021-02-21	120.15	excellent	excellent	good	PhatNTT	Nice atmosphere. Good for couple ^

Figure 9 Database connection result

The picture above shows the application is connected to the database and the records have been saved to the database.

Code Snippet

Create and access database

```
var db = window.openDatabase("iRateDB", "1.0", "iRate Database", 1000000);
document.addEventListener("deviceready", onDeviceReady, false);

function onDeviceReady() {

    // Create Table
    db.transaction(populateDB, errorCB, successCB);
}

function populateDB(tx) {
    tx.executeSql('CREATE TABLE IF NOT EXISTS restaurant (id INTEGER PRIMARY KEY
AUTOINCREMENT, ' +
        'resName TEXT NOT NULL, resType TEXT, date INTEGER, price DECIMAL(10,2),
serviceRating INT, ' +
        'cleanRating INT, foodRating INT, nameReporter TEXT, note TEXT)');
}

function errorCB(err) {
    alert("Error processing SQL: " + err.message);
}

function successCB() {
    alert("Success!");
}
```

Figure 10 Create/ access database and create table for database

window.openDatabase () function is supported to access or create database. It takes four parameters: (db_name) as iRateDB, (db_version) as 1.0, (db_display name) as iRate Database and (db_size) as 1000000.

Then, db.transaction () will call populateDB () to run the query to create the table in the database, then it will return successCB () function to announce success if the create the table query successfully and returns errorCB () function if an error is encountered during the execution of the query.

Exercise 4: Data entry screen (created by Native Android)

Basic Information

1.1 Student name	**Nguyen Tran Tien Phat** **Login id: np6380i**

1.2 Who did you work with? Note that for logbook exercises you are allowed to work with one other person as long as you give their name and login id and both contribute to the work. Both students need to provide description of their contribution.	**Name:** **Login id:**
1.3 Which Exercise is this? Tick as appropriate.	Creating a form display with Native Android allows users to enter data and some validations and error messages about the data that the user entered.
1.4 How well did you complete the exercise? Tick as appropriate.	• I tried but couldn't complete it ☐ • I did it but I feel I should have done better ☐ • I did everything that was asked ☒ • I did more than was asked for ☐
1.5 Briefly explain your answer to question 1.4	An application includes a data entry screen created via Native Android, which allows the user to enter data and select the information they want. Some validation for the entered fields is also implemented. These functions will be more detailed in the following sections.

Exercise explanation

Demo results interface of input entry screen

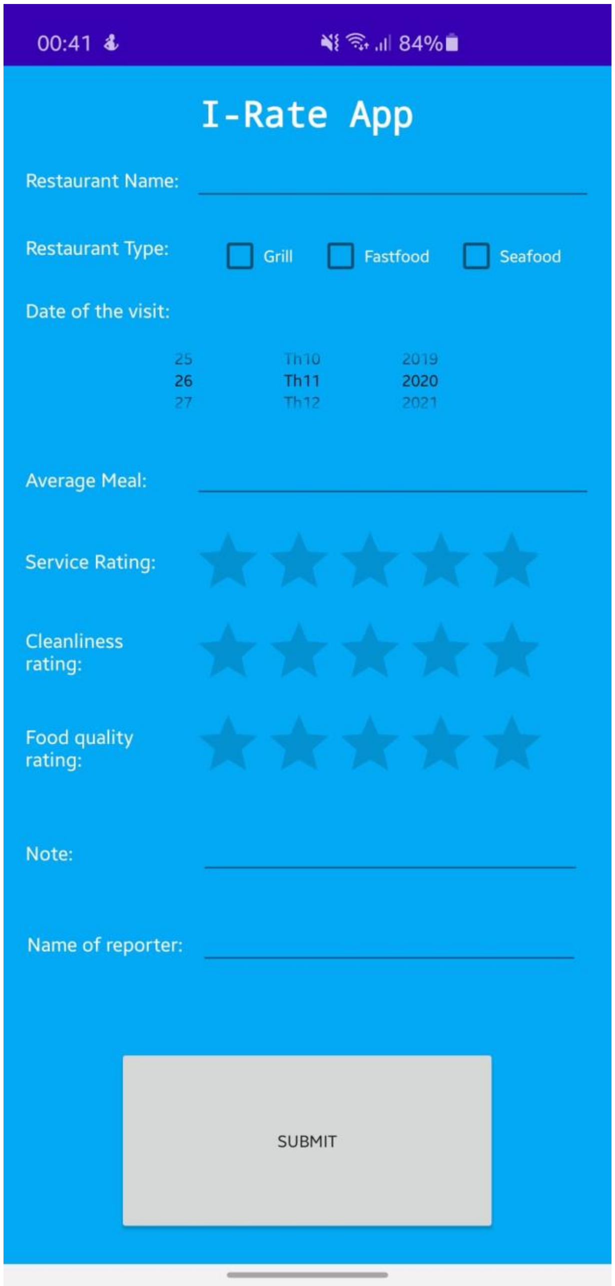
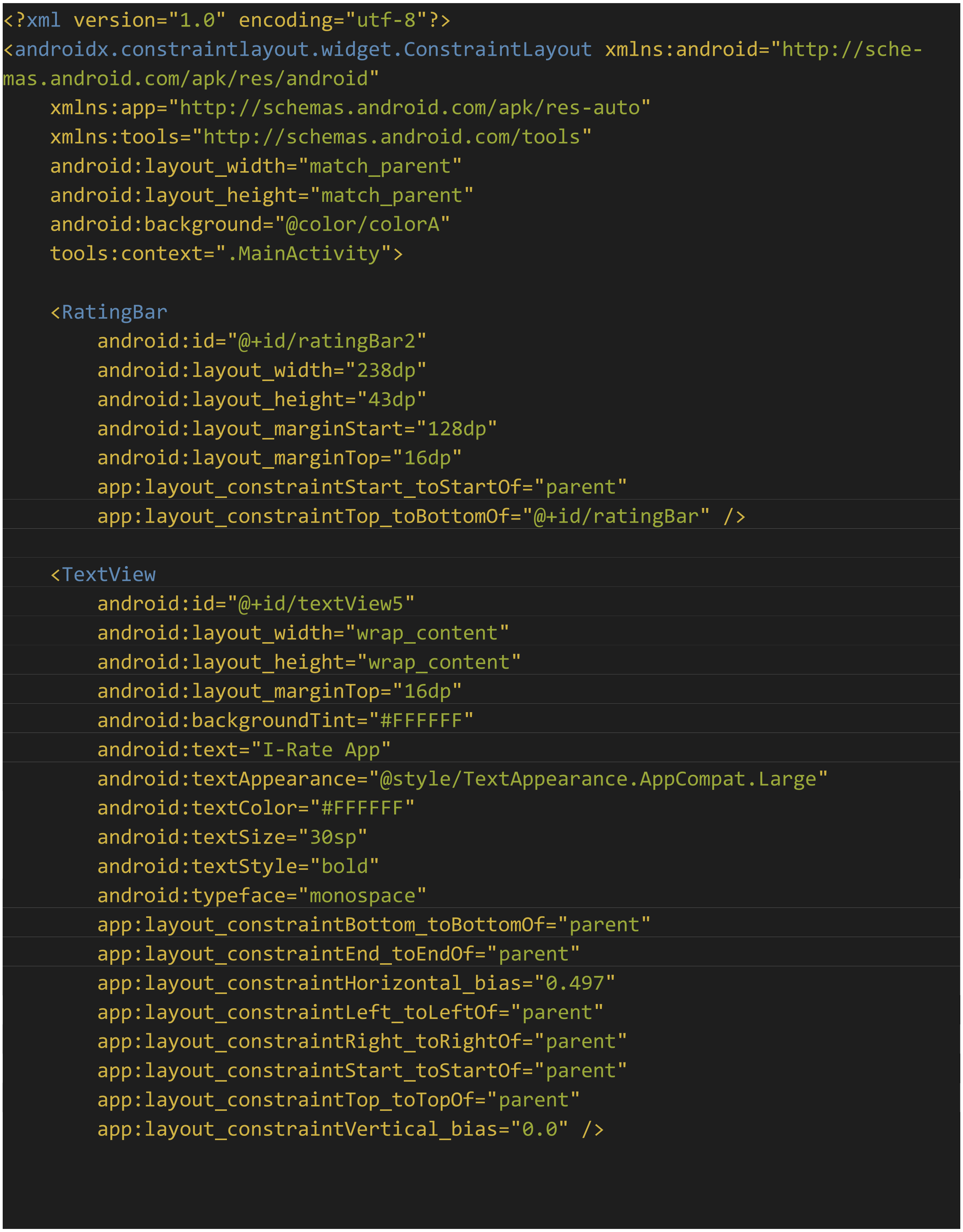


Figure 11 Data entry screen

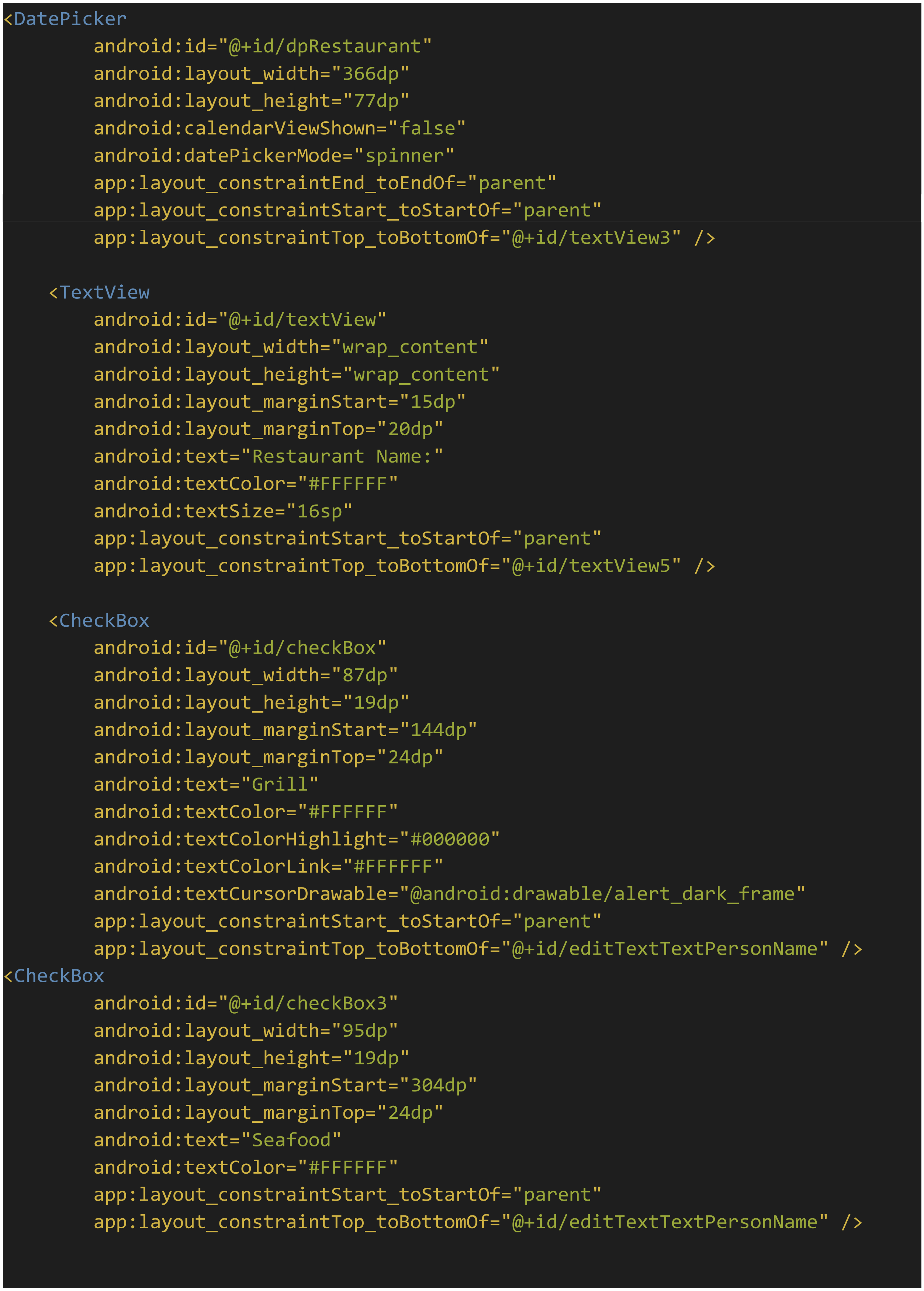
Unlike the application made from Hybrid app, the data entry screen made from Native Android is somewhat more complete in terms of user interface. Namely the components are more modern and that use some metaphor to describe properties. For example, a sequence of stars to describe the rating property. Besides, everything is standard design on one screen, so the user doesn't take the effort to scroll down. However, still ensure the visibility of the components on the mobile application.

Code Snippet



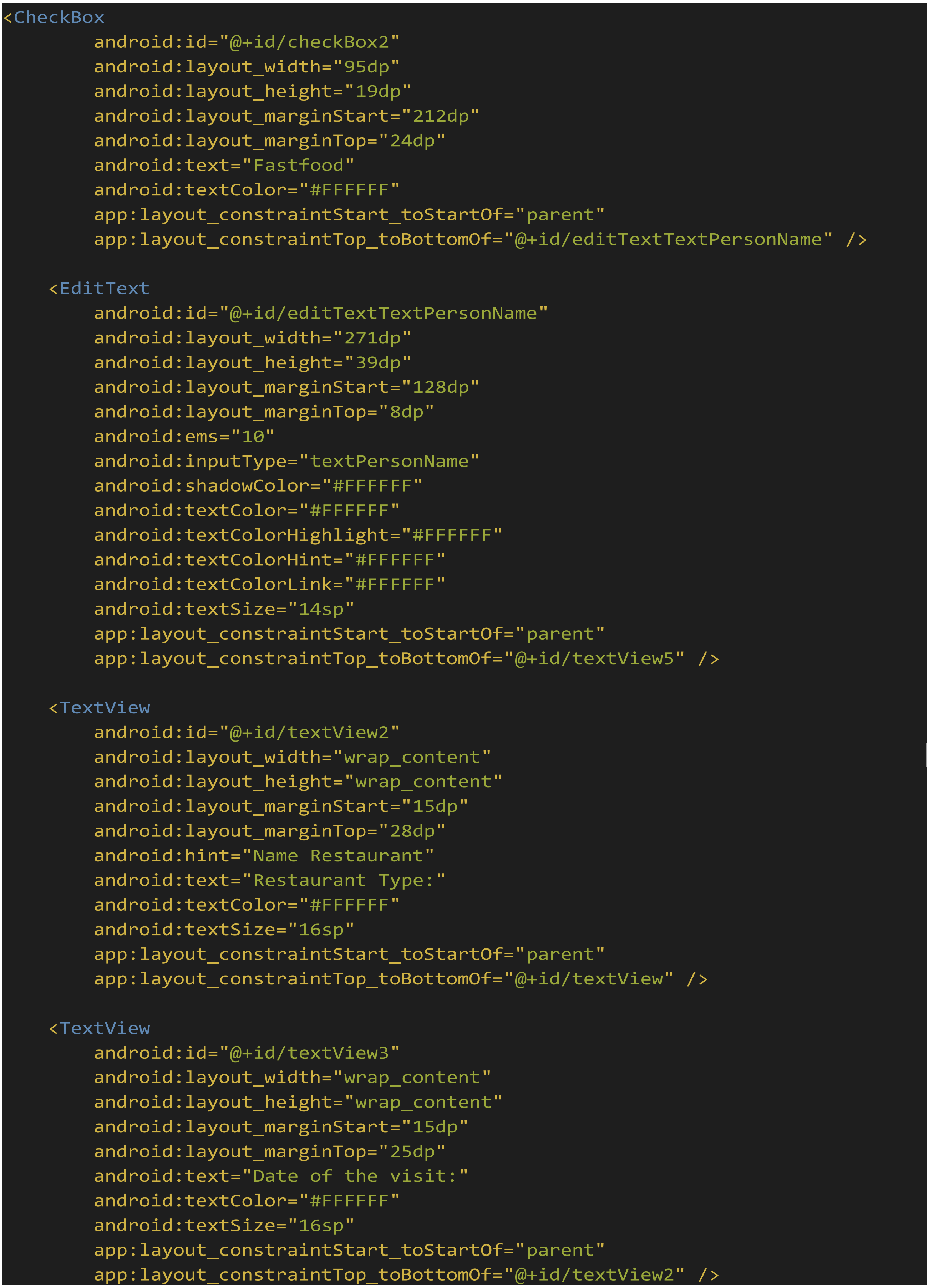
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorA"
    tools:context=".MainActivity">

    <RatingBar
        android:id="@+id/ratingBar2"
        android:layout_width="238dp"
        android:layout_height="43dp"
        android:layout_marginStart="128dp"
        android:layout_marginTop="16dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/ratingBar" />

    <TextView
        android:id="@+id/textView5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:backgroundTint="#FFFFFF"
        android:text="I-Rate App"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:textColor="#FFFFFF"
        android:textSize="30sp"
        android:textStyle="bold"
        android:typeface="monospace"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.497"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0" />
```

Figure 12 Code snippet for data entry screen created via Native Android



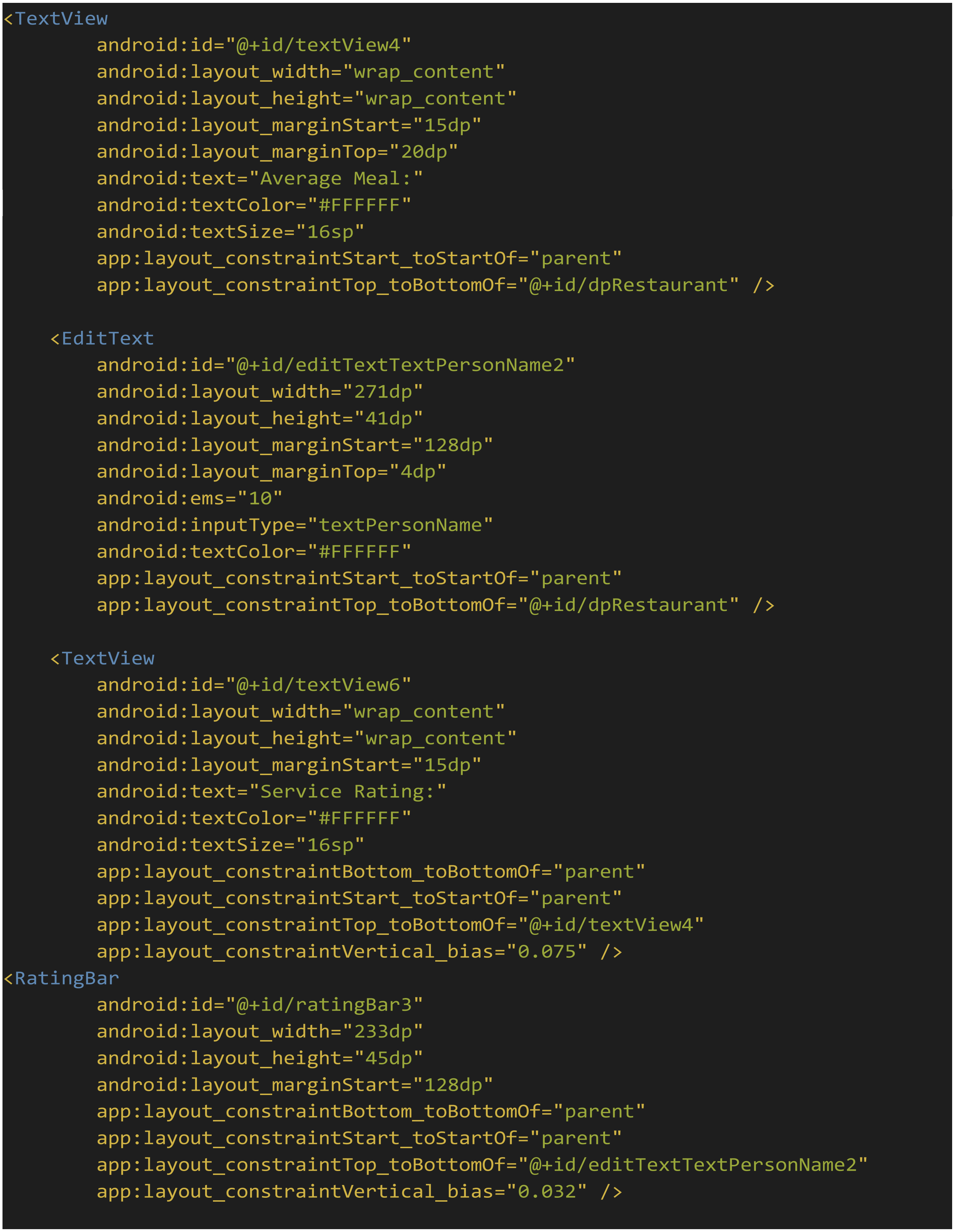
```
<DatePicker
        android:id="@+id/dpRestaurant"
        android:layout_width="366dp"
        android:layout_height="77dp"
        android:calendarViewShown="false"
        android:datePickerMode="spinner"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView3" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="15dp"
        android:layout_marginTop="20dp"
        android:text="Restaurant Name:"
        android:textColor="#FFFFFF"
        android:textSize="16sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView5" />

    <CheckBox
        android:id="@+id/checkBox"
        android:layout_width="87dp"
        android:layout_height="19dp"
        android:layout_marginStart="144dp"
        android:layout_marginTop="24dp"
        android:text="Grill"
        android:textColor="#FFFFFF"
        android:textColorHighlight="#000000"
        android:textColorLink="#FFFFFF"
        android:textCursorDrawable="@android:drawable/alert_dark_frame"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextTextPersonName" />
<CheckBox
        android:id="@+id/checkBox3"
        android:layout_width="95dp"
        android:layout_height="19dp"
        android:layout_marginStart="304dp"
        android:layout_marginTop="24dp"
        android:text="Seafood"
        android:textColor="#FFFFFF"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextTextPersonName" />
```

Figure 13 Code snippet for data entry screen created via Native Android (cont)



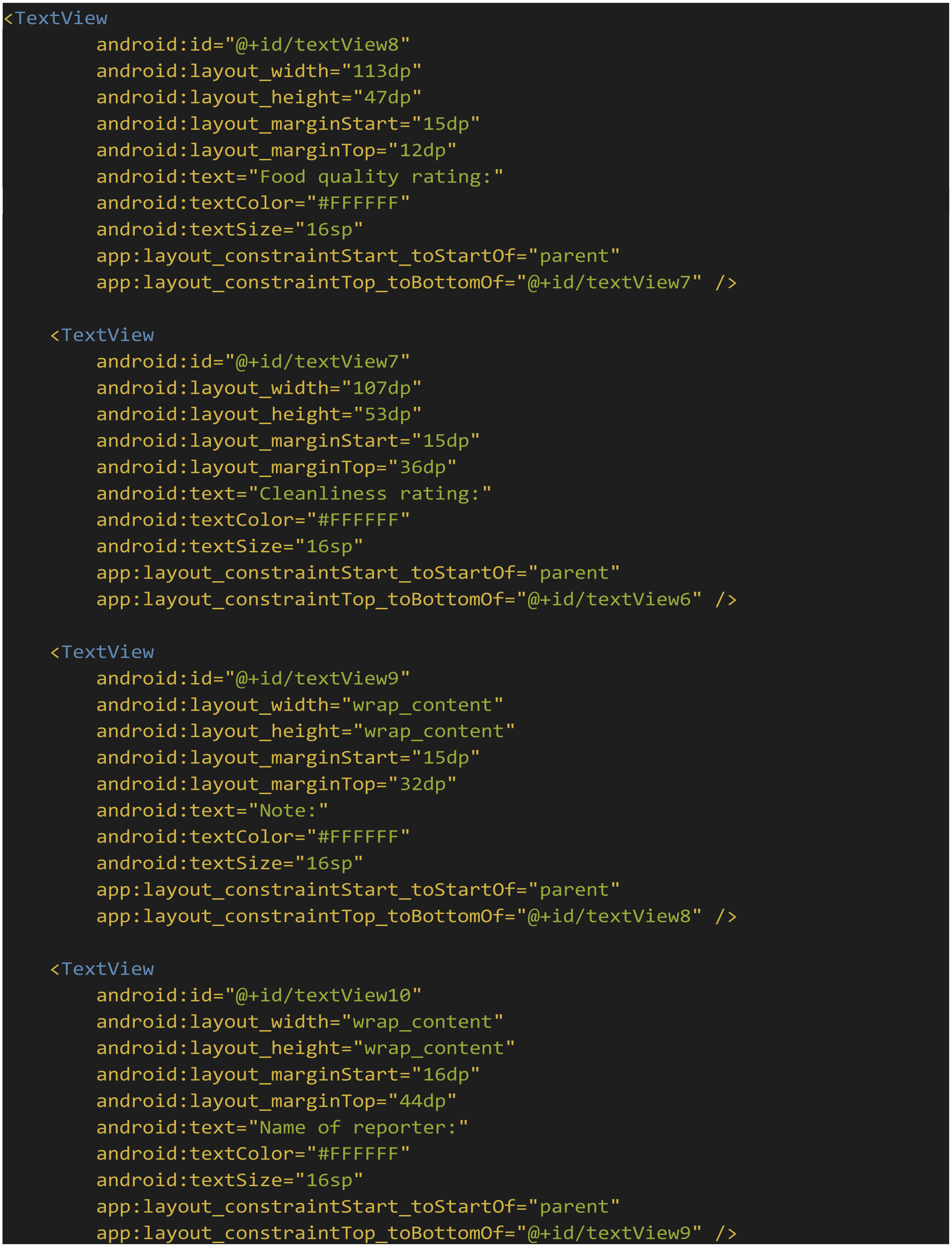
```
<CheckBox
        android:id="@+id/checkBox2"
        android:layout_width="95dp"
        android:layout_height="19dp"
        android:layout_marginStart="212dp"
        android:layout_marginTop="24dp"
        android:text="Fastfood"
        android:textColor="#FFFFFF"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextTextPersonName" />

    <EditText
        android:id="@+id/editTextTextPersonName"
        android:layout_width="271dp"
        android:layout_height="39dp"
        android:layout_marginStart="128dp"
        android:layout_marginTop="8dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:shadowColor="#FFFFFF"
        android:textColor="#FFFFFF"
        android:textColorHighlight="#FFFFFF"
        android:textColorHint="#FFFFFF"
        android:textColorLink="#FFFFFF"
        android:textSize="14sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView5" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="15dp"
        android:layout_marginTop="28dp"
        android:hint="Name Restaurant"
        android:text="Restaurant Type:"
        android:textColor="#FFFFFF"
        android:textSize="16sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="15dp"
        android:layout_marginTop="25dp"
        android:text="Date of the visit:"
        android:textColor="#FFFFFF"
        android:textSize="16sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView2" />
```

Figure 14 Code snippet for data entry screen created via Native Android (cont)



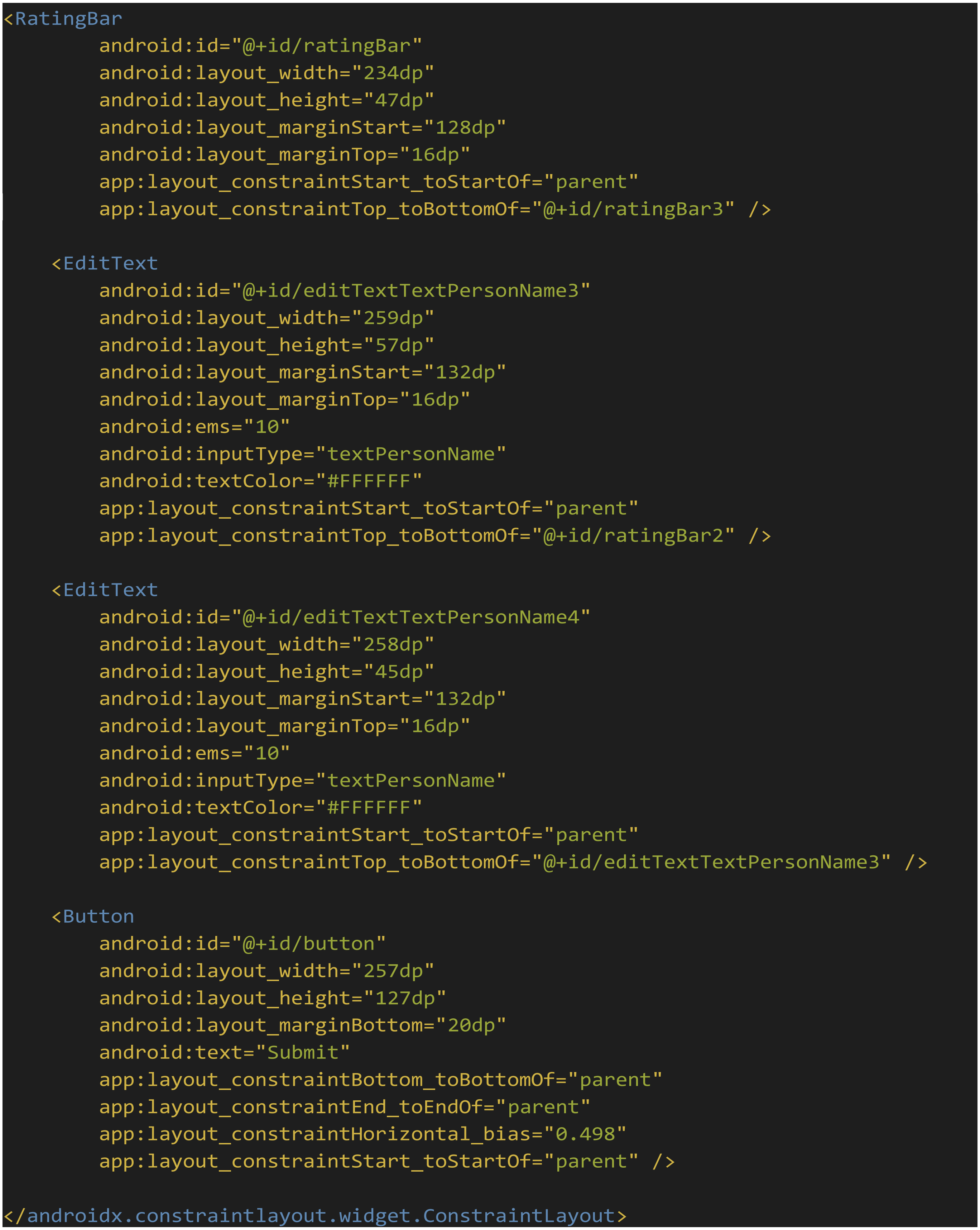
```
<TextView
        android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="15dp"
        android:layout_marginTop="20dp"
        android:text="Average Meal:"
        android:textColor="#FFFFFF"
        android:textSize="16sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/dpRestaurant" />

    <EditText
        android:id="@+id/editTextTextPersonName2"
        android:layout_width="271dp"
        android:layout_height="41dp"
        android:layout_marginStart="128dp"
        android:layout_marginTop="4dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:textColor="#FFFFFF"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/dpRestaurant" />

    <TextView
        android:id="@+id/textView6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="15dp"
        android:text="Service Rating:"
        android:textColor="#FFFFFF"
        android:textSize="16sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView4"
        app:layout_constraintVertical_bias="0.075" />
<RatingBar
        android:id="@+id/ratingBar3"
        android:layout_width="233dp"
        android:layout_height="45dp"
        android:layout_marginStart="128dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextTextPersonName2"
        app:layout_constraintVertical_bias="0.032" />
```

Figure 15 Code snippet for data entry screen created via Native Android (cont)



```
<TextView
        android:id="@+id/textView8"
        android:layout_width="113dp"
        android:layout_height="47dp"
        android:layout_marginStart="15dp"
        android:layout_marginTop="12dp"
        android:text="Food quality rating:"
        android:textColor="#FFFFFF"
        android:textSize="16sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView7" />

    <TextView
        android:id="@+id/textView7"
        android:layout_width="107dp"
        android:layout_height="53dp"
        android:layout_marginStart="15dp"
        android:layout_marginTop="36dp"
        android:text="Cleanliness rating:"
        android:textColor="#FFFFFF"
        android:textSize="16sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView6" />

    <TextView
        android:id="@+id/textView9"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="15dp"
        android:layout_marginTop="32dp"
        android:text="Note:"
        android:textColor="#FFFFFF"
        android:textSize="16sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView8" />

    <TextView
        android:id="@+id/textView10"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="44dp"
        android:text="Name of reporter:"
        android:textColor="#FFFFFF"
        android:textSize="16sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView9" />
```

Figure 16 Code snippet for data entry screen created via Native Android (cont)



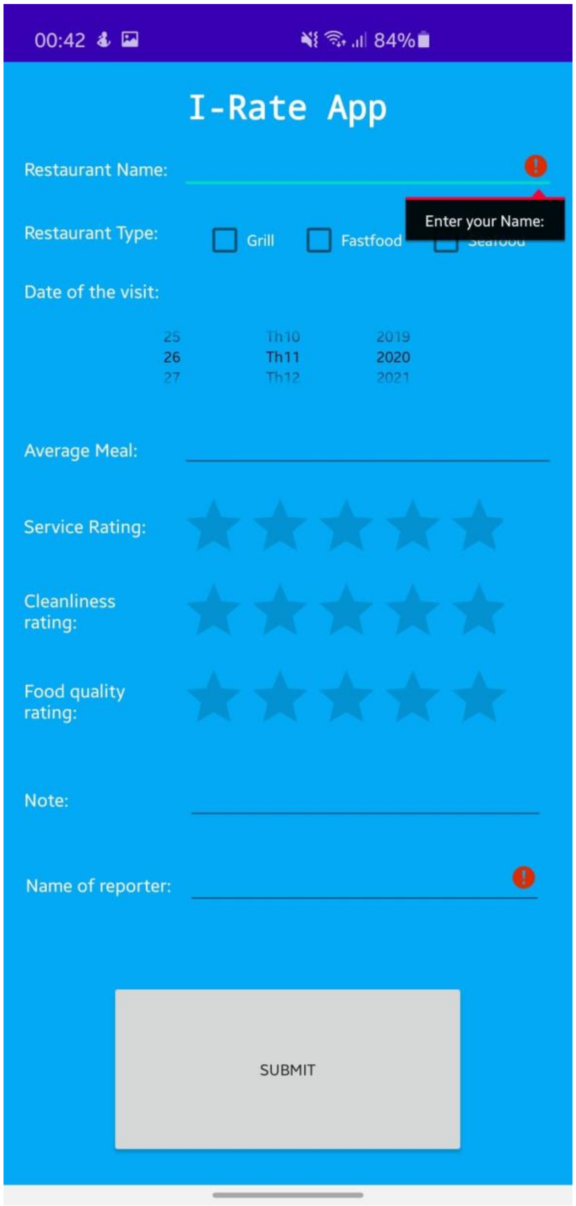
```
<RatingBar
        android:id="@+id/ratingBar"
        android:layout_width="234dp"
        android:layout_height="47dp"
        android:layout_marginStart="128dp"
        android:layout_marginTop="16dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/ratingBar3" />

    <EditText
        android:id="@+id/editTextTextPersonName3"
        android:layout_width="259dp"
        android:layout_height="57dp"
        android:layout_marginStart="132dp"
        android:layout_marginTop="16dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:textColor="#FFFFFF"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/ratingBar2" />

    <EditText
        android:id="@+id/editTextTextPersonName4"
        android:layout_width="258dp"
        android:layout_height="45dp"
        android:layout_marginStart="132dp"
        android:layout_marginTop="16dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:textColor="#FFFFFF"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextTextPersonName3" />

    <Button
        android:id="@+id/button"
        android:layout_width="257dp"
        android:layout_height="127dp"
        android:layout_marginBottom="20dp"
        android:text="Submit"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Figure 17 Code snippet for data entry screen created via Native Android (cont)

The code snippet above is the code for data input screen, which allowed user enter the restaurant details. All information is presented on a specific page and does not need to be scrolled down.

In this part, the interface is designed through *constraintlayout*. Each field is designed and aligned so that everything is streamlined. Finally, the last submit button executes the data submission and at the same time check validation for each data field.

Demo results interface of data validation



00:42 84%

I-Rate App

Restaurant Name:

Enter your Name:

Restaurant Type: □ Grill □ Fastfood □ Seafood

Date of the visit:

25 Th10 2019
26 Th11 2020
27 Th12 2021

Average Meal:

Service Rating:

Cleanliness rating:

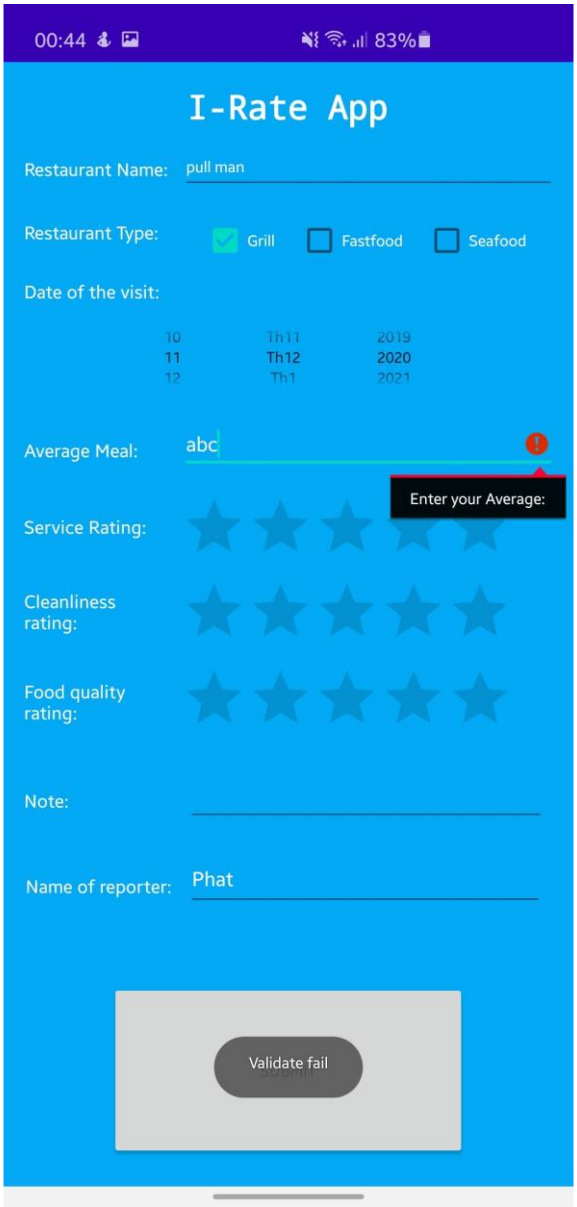
Food quality rating:

Note:

Name of reporter:

SUBMIT

Figure 18 Validate name



00:44 83%

I-Rate App

Restaurant Name: pull man

Restaurant Type: ☑ Grill □ Fastfood □ Seafood

Date of the visit:

10 Th11 2019
11 Th12 2020
12 Th1 2021

Average Meal: abc

Enter your Average:

Service Rating:

Cleanliness rating:

Food quality rating:

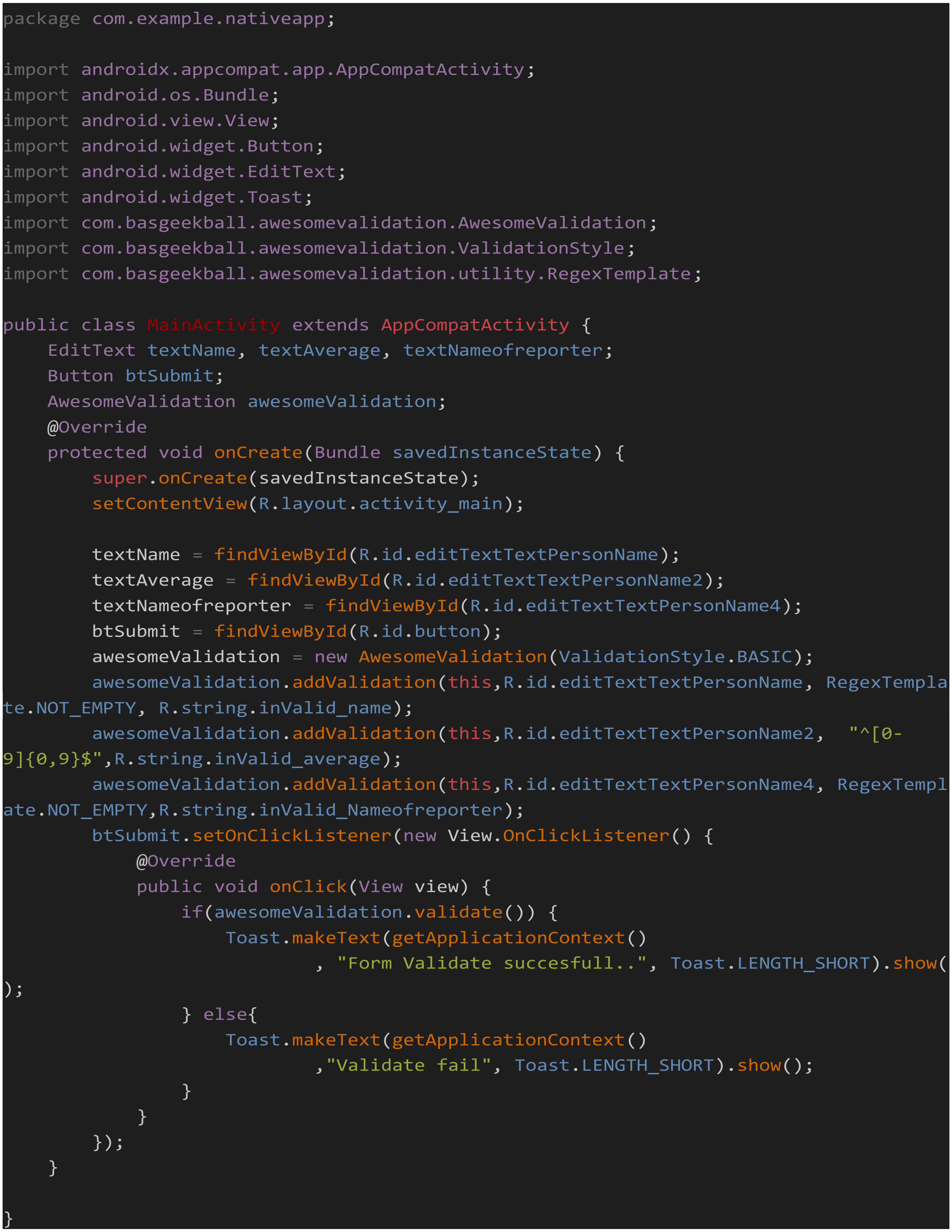
Note:

Name of reporter: Phat

Validate fail

Figure 19 Validate price per person

Code Snipet for validation



```
package com.example.nativeapp;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import com.basgeekball.awesomevalidation.AwesomeValidation;
import com.basgeekball.awesomevalidation.ValidationStyle;
import com.basgeekball.awesomevalidation.utility.RegexTemplate;

public class MainActivity extends AppCompatActivity {
    EditText textName, textAverage, textNameofreporter;
    Button btSubmit;
    AwesomeValidation awesomeValidation;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        textName = findViewById(R.id.editTextTextPersonName);
        textAverage = findViewById(R.id.editTextTextPersonName2);
        textNameofreporter = findViewById(R.id.editTextTextPersonName4);
        btSubmit = findViewById(R.id.button);
        awesomeValidation = new AwesomeValidation(ValidationStyle.BASIC);
        awesomeValidation.addValidation(this,R.id.editTextTextPersonName, RegexTempla
te.NOT_EMPTY, R.string.inValid_name);
        awesomeValidation.addValidation(this,R.id.editTextTextPersonName2,  "^[0-
9]{0,9}$",R.string.inValid_average);
        awesomeValidation.addValidation(this,R.id.editTextTextPersonName4, RegexTempl
ate.NOT_EMPTY,R.string.inValid_Nameofreporter);
        btSubmit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if(awesomeValidation.validate()) {
                    Toast.makeText(getApplicationContext()
                            , "Form Validate succesfull..", Toast.LENGTH_SHORT).show(
);
                } else{
                    Toast.makeText(getApplicationContext()
                            ,"Validate fail", Toast.LENGTH_SHORT).show();
                }
            }
        });
    }

}
```

Figure 20 Code snippet for validate function in data entry screen

The first is the library declaration. Then, we declare some variables to store the values obtained from the fields entered by the user. Finally, check the data has been retrieved and notify the user of the error and success message.