```java
package application;

import javafx.application.Platform;


public class HealthDataEntry {
    private Stage primaryStage;
    private Scene healthDataEntryScene;
    private TableView<HealthData<?>> tableView;
    private User<HealthData<?>> user;
    private static final String url = "jdbc:sqlite:healthtracker.db";

    public Stage getPrimaryStage() {
        return primaryStage;
    }
    public User<HealthData<?>> getUser() {
        return user;
    }
    private HealthData<CommonHealthData> healthData;
    /**
     * Creates a new instance of HealthDataEntry.
     * @param primaryStage the primary stage for the application
     * @param user the user associated with the health data
     * PRECONDITION: primaryStage is not null
     * PRECONDITION: user is not null
     */
    public HealthDataEntry(Stage primaryStage, User<HealthData<?>> user) {
        this.primaryStage = primaryStage;
        this.user = user;
        createHealthDataEntryScene();

    }
    /**
     * Displays the health data entry scene.
     * PreCondiotn: User created successfully
     * POSTCONDITION: The health data entry scene is shown on the primary stage.
     */
    public void showHealthDataEntryScene() {
        primaryStage.setScene(healthDataEntryScene);
        primaryStage.setTitle("Health Data Entry");
        primaryStage.show();
    }
    /**
     * Creates the health data entry scene and sets up event handlers for the buttons.
     * Precondition: ShowHealthDataEntryScene works
     * POSTCONDITION: The health data entry scene is created with all the necessary UI components and event handlers.
     */
    private void createHealthDataEntryScene() {
        // Create UI components for health data entry scene
        Label titleLabel = new Label("Health Data Entry");

        Button bloodPressureButton = new Button("Blood Pressure");
        Button cholesterolButton = new Button("Cholesterol");
        Button bmiButton = new Button("BMI");
        Button bloodSugarButton = new Button("Blood Sugar");
        Button customHealthNoteButton = new Button("Custom Health Note");
        Button historyButton = new Button("History");

```

```
77          // Create layout container for health data entry scene
78          VBox root = new VBox(10);
79          root.setAlignment(Pos.CENTER);
80          root.setPadding(new Insets(10));
81          root.getChildren().addAll(titleLabel, bloodPressureButton, cholesterolButton,
    bmiButton,
82                  bloodSugarButton, customHealthNoteButton, historyButton);
83
84          // Create health data entry scene
85          healthDataEntryScene = new Scene(root, 400, 300);
86
87          // Handle blood pressure button click event
88          bloodPressureButton.setOnAction(event -> {
89            showBloodPressureScene(null, false, tableView);
90          });
91
92          // Handle cholesterol button click event
93          cholesterolButton.setOnAction(event -> {
94              showCholesterolScene(null, false, tableView);
95          });
96
97          // Handle BMI button click event
98          bmiButton.setOnAction(event -> {
99              showBMIScene(null, false, tableView);
100         });
101
102         // Handle blood sugar button click event
103         bloodSugarButton.setOnAction(event -> {
104             showBloodSugarScene(null, false, tableView);
105         });
106
107         // Handle custom health note button click event
108         customHealthNoteButton.setOnAction(event -> {
109             showCustomHealthNoteScene(null, false, tableView);
110         });
111
112         // Handle history button click event
113         historyButton.setOnAction(event -> {
114             showHistoryScreen();
115         });
116     }
117     /**
118      * Shows the blood pressure scene for data entry.
119      * Precondition: Button exist and click-able
120      * POSTCONDITION: A new Scene object for blood pressure entry is created and
    returned.
121      */
122     public Scene showBloodPressureScene(CommonHealthData existingHealthData, boolean
    isEdit, TableView<HealthData<?>> tableView) {
123         Stage bloodPressureStage = new Stage(); // Create a new stage
124
125         // Create UI components for blood pressure scene
126         Label titleLabel = new Label("Blood Pressure");
127
128         Label systolicBPLabel = new Label("Systolic BP:");
129         TextField systolicBPTextField = new TextField();
130
131         Label diastolicBPLabel = new Label("Diastolic BP:");
132         TextField diastolicBPTextField = new TextField();
```

```
133
134         Button submitButton = new Button("Submit");
135
136         // Create layout container for blood pressure scene
137         VBox root = new VBox(10);
138         root.setAlignment(Pos.CENTER);
139         root.setPadding(new Insets(10));
140         root.getChildren().addAll(titleLabel, systolicBPLabel, systolicBPTextField,
    diastolicBPLabel,
141                 diastolicBPTextField, submitButton);
142
143         // Create blood pressure scene
144         Scene bloodPressureScene = new Scene(root, 400, 300);
145
146         // Handle submit button click event
147         submitButton.setOnAction(event -> {
148             int systolicBP = Integer.parseInt(systolicBPTextField.getText());
149             int diastolicBP = Integer.parseInt(diastolicBPTextField.getText());
150
151             try (Connection conn = DriverManager.getConnection(url);
152                  Statement stmt = conn.createStatement()) {
153
154                 // Fetch the user ID from the User table based on the user's email
155                 String selectUserSql = "SELECT id FROM User WHERE email = '" +
    user.getEmail() + "'";
156                 ResultSet resultSet = stmt.executeQuery(selectUserSql);
157                 int userId = resultSet.getInt("id");
158
159                 if (isEdit && existingHealthData instanceof CommonHealthData) {
160                     CommonHealthData commonHealthData = (CommonHealthData)
    existingHealthData;
161                     commonHealthData.setSystolicBP(systolicBP);
162                     commonHealthData.setDiastolicBP(diastolicBP);
163
164                     try {
165                         commonHealthData.validate();
166                     } catch (HealthDataException e) {
167                         e.printStackTrace();
168                     }
169
170                     // Create the SQL UPDATE statement to update the blood pressure
    data
171                     String updateDataSql = "UPDATE BloodPressureData SET systolicBP =
    " + systolicBP
172                             + ", diastolicBP = " + diastolicBP + " WHERE userId = " +
    userId;
173
174                     // Execute the UPDATE statement
175                     stmt.executeUpdate(updateDataSql);
176
177                     tableView.refresh();
178                     Platform.runLater(() -> {
179                         Stage editbloodpreStage = (Stage) bloodPressureScene.getWindow
    ();
180                         editbloodpreStage.close();
181                         bloodPressureStage.close();
182                     });
183                 } else {
184                     // Create the SQL INSERT statement to insert the blood pressure
```

```
    data
185                     String insertDataSql = "INSERT INTO BloodPressureData (userId,
    systolicBP, diastolicBP, dateRecorded) VALUES ("
186                             + userId + ", " + systolicBP + ", " + diastolicBP + ", '"
    + new Date() + "')";
187
188                     // Execute the INSERT statement
189                     stmt.executeUpdate(insertDataSql);
190
191                     // Create a new health data entry and add it to the user's health
    data
192                     String name = "Blood Pressure";
193                     Date date = new Date();
194                     Date originalDate = date;
195                     String metric = "Blood Pressure";
196                     CommonHealthData healthDataEntry = new CommonHealthData(name,
    originalDate, metric, systolicBP, diastolicBP);
197                     try {
198                         healthDataEntry.validate();
199                     } catch (HealthDataException e) {
200                         e.printStackTrace();
201                     }
202                     user.addHealthData(healthDataEntry);
203                     HealthDataChecker.checkBloodPressure(healthDataEntry);
204
205                     Platform.runLater(() -> {
206                         bloodPressureStage.close();
207                         showHealthDataEntryScene();
208                     });
209                 }
210             } catch (SQLException e) {
211                 e.printStackTrace();
212             }
213         });
214
215         bloodPressureStage.setScene(bloodPressureScene); // Set the blood pressure
    scene to the new stage
216         bloodPressureStage.show(); // Show the new stage
217
218         return bloodPressureScene;
219     }
220
221
222     /**
223      * Shows the Cholesterol scene for data entry.
224      * Precondition: Button exists and is clickable.
225      * POSTCONDITION: A new Scene object for cholesterol entry is created and
    returned.
226      */
227     public Scene showCholesterolScene(CommonHealthData existingHealthData, boolean
    isEdit, TableView<HealthData<?>> tableView) {
228         // Create UI components for cholesterol scene
229         Stage cholesterolStage = new Stage(); // Create a new stage
230
231         Label titleLabel = new Label("Cholesterol");
232
233         Label ldlCholesterolLabel = new Label("LDL Cholesterol:");
234         TextField ldlCholesterolTextField = new TextField();
235
```

```
236         Label hdlCholesterolLabel = new Label("HDL Cholesterol:");
237         TextField hdlCholesterolTextField = new TextField();
238 
239         Label triglycerideCholesterolLabel = new Label("Triglyceride Cholesterol:");
240         TextField triglycerideCholesterolTextField = new TextField();
241 
242         Button submitButton = new Button("Submit");
243 
244         // Create layout container for cholesterol scene
245         VBox root = new VBox(10);
246         root.setAlignment(Pos.CENTER);
247         root.setPadding(new Insets(10));
248         root.getChildren().addAll(titleLabel, ldlCholesterolLabel,
    ldlCholesterolTextField, hdlCholesterolLabel,
249                 hdlCholesterolTextField, triglycerideCholesterolLabel,
    triglycerideCholesterolTextField, submitButton);
250 
251         // Create cholesterol scene
252         Scene cholesterolScene = new Scene(root, 400, 300);
253 
254         // Handle submit button click event
255         submitButton.setOnAction(event -> {
256             int ldlCholesterol = Integer.parseInt(ldlCholesterolTextField.getText());
257             int hdlCholesterol = Integer.parseInt(hdlCholesterolTextField.getText());
258             int triglycerideCholesterol =
    Integer.parseInt(triglycerideCholesterolTextField.getText());
259 
260             if (isEdit) {
261                 if (existingHealthData instanceof CommonHealthData) {
262                     CommonHealthData commonHealthData = (CommonHealthData)
    existingHealthData;
263                     commonHealthData.setLdlCholesterol(ldlCholesterol);
264                     commonHealthData.setHdlCholesterol(hdlCholesterol);
265                     commonHealthData.setTriglycerideCholesterol
    (triglycerideCholesterol);
266 
267                     try (Connection conn = DriverManager.getConnection(url);
268                          Statement stmt = conn.createStatement()) {
269 
270                         // Create the SQL UPDATE statement to update the cholesterol
    data
271                         String selectUserSql = "SELECT id FROM User WHERE email = '" +
    user.getEmail() + "'";
272                         ResultSet resultSet = stmt.executeQuery(selectUserSql);
273                         int userId = resultSet.getInt("id");
274                         String updateDataSql = "UPDATE CholesterolData SET
    ldlCholesterol = " + ldlCholesterol
275                                 + ", hdlCholesterol = " + hdlCholesterol + ",
    triglycerideCholesterol = " + triglycerideCholesterol
276                                 + " WHERE userId = " + userId;
277 
278                         // Execute the UPDATE statement
279                         stmt.executeUpdate(updateDataSql);
280 
281                         commonHealthData.validate();
282                     } catch (SQLException | HealthDataException e) {
283                         e.printStackTrace();
284                     }
285                 }
```

```
286
287                 tableView.refresh(); // Refresh the table view to reflect the changes
288                 Platform.runLater(() -> {
289                     Stage editCholesterolStage = (Stage) cholesterolScene.getWindow();
290                     editCholesterolStage.close();
291                     cholesterolStage.close();
292                 });
293             } else {
294                 String name = "Cholesterol";
295                 Date date = new Date();
296                 Date originalDate = date;
297                 String metric = "Cholesterol";
298
299                 CommonHealthData healthDataEntry = new CommonHealthData(name,
    originalDate, metric, ldlCholesterol,
300                         hdlCholesterol, triglycerideCholesterol);
301
302                 try (Connection conn = DriverManager.getConnection(url);
303                      Statement stmt = conn.createStatement()) {
304
305                     String selectUserSql = "SELECT id FROM User WHERE email = '" +
    user.getEmail() + "'";
306                     ResultSet resultSet = stmt.executeQuery(selectUserSql);
307                     int userId = resultSet.getInt("id");
308
309                     // Create the SQL INSERT statement to insert the cholesterol data
310                     String insertDataSql = "INSERT INTO CholesterolData (userId,
    ldlCholesterol, hdlCholesterol, triglycerideCholesterol, dateRecorded) VALUES ("
311                             + userId + ", " + ldlCholesterol + ", " + hdlCholesterol +
    ", " + triglycerideCholesterol + ", '" + originalDate + "')";
312
313                     // Execute the INSERT statement
314                     stmt.executeUpdate(insertDataSql);
315
316                     healthDataEntry.validate();
317                 } catch (SQLException | HealthDataException e) {
318                     e.printStackTrace();
319                 }
320
321                 user.addHealthData(healthDataEntry);
322                 HealthDataChecker.checkCholesterol(healthDataEntry);
323                 Platform.runLater(() -> {
324                     cholesterolStage.close();
325                     showHealthDataEntryScene();
326                 });
327             }
328         });
329
330         cholesterolStage.setScene(cholesterolScene);
331         cholesterolStage.show();
332         return cholesterolScene;
333     }
334
335     /**
336      * Shows the BMI scene for data entry.
337      * Precondition: Button exists and is clickable.
338      * POSTCONDITION: A new Scene object for BMI entry is created and returned.
339      */
340     public Scene showBMIScene(CommonHealthData existingHealthData, boolean isEdit,
```

```
    TableView<HealthData<?>> tableView) {
341         // Create UI components for BMI scene
342         Stage bmiStage = new Stage(); // Create a new stage
343 
344         Label titleLabel = new Label("BMI");
345 
346         Label weightLabel = new Label("Weight:");
347         TextField weightTextField = new TextField();
348 
349         Label heightLabel = new Label("Height:");
350         TextField heightTextField = new TextField();
351 
352         Button submitButton = new Button("Submit");
353 
354         // Create layout container for BMI scene
355         VBox root = new VBox(10);
356         root.setAlignment(Pos.CENTER);
357         root.setPadding(new Insets(10));
358         root.getChildren().addAll(titleLabel, weightLabel, weightTextField,
    heightLabel,
359                 heightTextField, submitButton);
360 
361         // Create BMI scene
362         Scene bmiScene = new Scene(root, 400, 300);
363 
364         // Handle submit button click event
365         submitButton.setOnAction(event -> {
366             double weight = Double.parseDouble(weightTextField.getText());
367             double height = Double.parseDouble(heightTextField.getText());
368 
369             try (Connection conn = DriverManager.getConnection(url);
370                  Statement stmt = conn.createStatement()) {
371 
372                 // Fetch the user ID from the User table based on the user's email
373                 String selectUserSql = "SELECT id FROM User WHERE email = '" +
    user.getEmail() + "'";
374                 ResultSet resultSet = stmt.executeQuery(selectUserSql);
375                 int userId = resultSet.getInt("id");
376 
377                 if (isEdit && existingHealthData instanceof CommonHealthData) {
378                     CommonHealthData commonHealthData = (CommonHealthData)
    existingHealthData;
379                     commonHealthData.setWeight(weight);
380                     commonHealthData.setHeight(height);
381 
382                     try {
383                         commonHealthData.validate();
384                     } catch (HealthDataException e) {
385                         e.printStackTrace();
386                     }
387 
388                     // Create the SQL UPDATE statement to update the BMI data
389                     String updateDataSql = "UPDATE BMIData SET weight = " + weight +
    ", height = " + height
390                             + " WHERE userId = " + userId;
391 
392                     // Execute the UPDATE statement
393                     stmt.executeUpdate(updateDataSql);
394 
```

```
395                     tableView.refresh();
396                     Platform.runLater(() -> {
397                         Stage editBmiStage = (Stage) bmiScene.getWindow();
398                         editBmiStage.close();
399                         bmiStage.close();
400                     });
401                 } else {
402                     // Create the SQL INSERT statement to insert the BMI data
403                     String insertDataSql = "INSERT INTO BMIData (userId, weight,
    height, dateRecorded) VALUES ("
404                             + userId + ", " + weight + ", " + height + ", '" + new
    Date() + "')";
405 
406                     // Execute the INSERT statement
407                     stmt.executeUpdate(insertDataSql);
408 
409                     // Create a new health data entry and add it to the user's health
    data
410                     String name = "BMI";
411                     Date date = new Date();
412                     Date originalDate = date;
413                     String metric = "BMI";
414 
415                     CommonHealthData healthDataEntry = new CommonHealthData(name,
    originalDate, metric, weight, height);
416                     try {
417                         healthDataEntry.validate();
418                     } catch (HealthDataException e) {
419                         e.printStackTrace();
420                     }
421                     user.addHealthData(healthDataEntry);
422                     HealthDataChecker.checkBMI(healthDataEntry);
423 
424                     Platform.runLater(() -> {
425                         bmiStage.close();
426                         showHealthDataEntryScene();
427                     });
428                 }
429             } catch (SQLException e) {
430                 e.printStackTrace();
431             }
432         });
433 
434         bmiStage.setScene(bmiScene);
435         bmiStage.show();
436         return bmiScene;
437     }
438 
439 
440     /**
441      * Shows the blood sugar scene for data entry.
442      * Precondition: Button exists and is clickable.
443      * POSTCONDITION: A new Scene object for blood sugar entry is created and
    returned.
444      */
445     public Scene showBloodSugarScene(CommonHealthData existingHealthData, boolean
    isEdit, TableView<HealthData<?>> tableView) {
446         // Create UI components for blood sugar scene
447         Stage bloodSugarStage = new Stage(); // Create a new stage
```

```
448
449         Label titleLabel = new Label("Blood Glucose");
450
451         Label glucoseLevelLabel = new Label("Glucose Level:");
452         TextField glucoseLevelTextField = new TextField();
453
454         Button submitButton = new Button("Submit");
455
456         // Create layout container for blood sugar scene
457         VBox root = new VBox(10);
458         root.setAlignment(Pos.CENTER);
459         root.setPadding(new Insets(10));
460         root.getChildren().addAll(titleLabel, glucoseLevelLabel,
    glucoseLevelTextField, submitButton);
461
462         // Create blood sugar scene
463         Scene bloodSugarScene = new Scene(root, 400, 300);
464
465         // Handle submit button click event
466         submitButton.setOnAction(event -> {
467             double bloodSugarLevel = Double.parseDouble(glucoseLevelTextField.getText
    ());
468
469             try (Connection conn = DriverManager.getConnection(url);
470                  Statement stmt = conn.createStatement()) {
471
472                 // Fetch the user ID from the User table based on the user's email
473                 String selectUserSql = "SELECT id FROM User WHERE email = '" +
    user.getEmail() + "'";
474                 ResultSet resultSet = stmt.executeQuery(selectUserSql);
475                 int userId = resultSet.getInt("id");
476
477                 if (isEdit && existingHealthData instanceof CommonHealthData) {
478                     CommonHealthData commonHealthData = (CommonHealthData)
    existingHealthData;
479                     commonHealthData.setGlucoseLevel(bloodSugarLevel);
480
481                     try {
482                         commonHealthData.validate();
483                     } catch (HealthDataException e) {
484                         e.printStackTrace();
485                     }
486
487                     // Create the SQL UPDATE statement to update the blood sugar data
488                     String updateDataSql = "UPDATE DiabetesData SET bloodSugarLevel =
    " + bloodSugarLevel
489                             + " WHERE userId = " + userId;
490
491                     // Execute the UPDATE statement
492                     stmt.executeUpdate(updateDataSql);
493
494                     tableView.refresh();
495                     Platform.runLater(() -> {
496                         Stage editBloodSugarStage = (Stage) bloodSugarScene.getWindow
    ();
497                         editBloodSugarStage.close();
498                         bloodSugarStage.close();
499                     });
500                 } else {
```

```
501                     // Create the SQL INSERT statement to insert the blood sugar data
502                     String insertDataSql = "INSERT INTO DiabetesData (userId,
    bloodSugarLevel, dateRecorded) VALUES ("
503                             + userId + ", " + bloodSugarLevel + ", '" + new Date() +
    "')";
504
505                     // Execute the INSERT statement
506                     stmt.executeUpdate(insertDataSql);
507
508                     // Create a new health data entry and add it to the user's health
    data
509                     String name = "Blood Glucose";
510                     Date date = new Date();
511                     Date originalDate = date;
512                     String metric = "Blood Glucose";
513
514                     CommonHealthData healthDataEntry = new CommonHealthData(name,
    originalDate, metric, bloodSugarLevel);
515                     try {
516                         healthDataEntry.validate();
517                     } catch (HealthDataException e) {
518                         e.printStackTrace();
519                     }
520                     user.addHealthData(healthDataEntry);
521                     HealthDataChecker.checkBloodGlucose(healthDataEntry);
522
523                     Platform.runLater(() -> {
524                         bloodSugarStage.close();
525                         showHealthDataEntryScene();
526                     });
527                 }
528             } catch (SQLException e) {
529                 e.printStackTrace();
530             }
531         });
532
533         bloodSugarStage.setScene(bloodSugarScene);
534         bloodSugarStage.show();
535         return bloodSugarScene;
536     }
537
538
539     /**
540      * Shows the custom health note scene for data entry.
541      * Precondition: Button exists and is clickable.
542      * POSTCONDITION: A new Scene object for custom health note entry is created and
    returned.
543      */
544     public Scene showCustomHealthNoteScene(CustomHealthData existingHealthData,
    boolean isEdit, TableView<HealthData<?>> tableView) {
545         // Create UI components for custom health note scene
546         Stage customStage = new Stage(); // Create a new stage
547
548         Label titleLabel = new Label("Custom Health Note");
549
550         Label noteLabel = new Label("Note:");
551         TextField noteTextField = new TextField();
552
553         Button submitButton = new Button("Submit");
```

```
554
555         // Create layout container for custom health note scene
556         VBox root = new VBox(10);
557         root.setAlignment(Pos.CENTER);
558         root.setPadding(new Insets(10));
559         root.getChildren().addAll(titleLabel, noteLabel, noteTextField, submitButton);
560
561         // Create custom health note scene
562         Scene customHealthNoteScene = new Scene(root, 400, 300);
563
564         // Handle submit button click event
565         submitButton.setOnAction(event -> {
566             String note = noteTextField.getText();
567
568             try (Connection conn = DriverManager.getConnection(url);
569                  Statement stmt = conn.createStatement()) {
570
571                 // Fetch the user ID from the User table based on the user's email
572                 String selectUserSql = "SELECT id FROM User WHERE email = '" +
    user.getEmail() + "'";
573                 ResultSet resultSet = stmt.executeQuery(selectUserSql);
574                 int userId = resultSet.getInt("id");
575
576                 if (isEdit && existingHealthData instanceof CustomHealthData) {
577                     CustomHealthData customHealthData = (CustomHealthData)
    existingHealthData;
578 //                    customHealthData.setNotes(note);
579
580                     // Create the SQL UPDATE statement to update the custom health
    note data
581                     String updateDataSql = "UPDATE CustomHealthData SET note = '" +
    note
582                             + "' WHERE userId = " + userId;
583
584                     // Execute the UPDATE statement
585                     stmt.executeUpdate(updateDataSql);
586
587                     tableView.refresh();
588                     Stage editCustomStage = (Stage) customHealthNoteScene.getWindow();
589                     Platform.runLater(() -> {
590                         editCustomStage.close();
591                         customStage.close();
592                     });
593                 } else {
594                     // Create the SQL INSERT statement to insert the custom health
    note data
595                     String insertDataSql = "INSERT INTO CustomHealthData (userId,
    note, dateRecorded) VALUES ("
596                             + userId + ", '" + note + "', '" + new Date() + "')";
597
598                     // Execute the INSERT statement
599                     stmt.executeUpdate(insertDataSql);
600
601                     // Create a new health data entry and add it to the user's health
    data
602                     String name = "Custom Health Note";
603                     Date date = new Date();
604                     Date originalDate = date;
605
```

```
606                     CustomHealthData healthDataEntry = new CustomHealthData(name,
    originalDate, note);
607                     user.addHealthData(healthDataEntry);
608                     customStage.close();
609                     Platform.runLater(() -> {
610                         showHealthDataEntryScene();
611                         showSuccessMessage();
612                     });
613                 }
614             } catch (SQLException e) {
615                 e.printStackTrace();
616             }
617         });
618 
619         customStage.setScene(customHealthNoteScene);
620         customStage.show();
621         return customHealthNoteScene;
622     }
623 
624 
625 
626     /**
627      * Shows the history screen with the user's health data entries.
628      * Precondition: Button exist and click-able
629      * POSTCONDITION: The history screen is shown on the primary stage.
630      */
631     private void showHistoryScreen() {
632         HistoryScreen historyScreen = new HistoryScreen(user);
633         historyScreen.display();
634     }
635     private void showSuccessMessage() {
636         Alert alert = new Alert(Alert.AlertType.INFORMATION);
637         alert.setTitle("Success");
638         alert.setHeaderText(null);
639         alert.setContentText("Health data entry saved successfully!");
640         alert.showAndWait();
641     }
642 
643 
644 
645 
646 }
647 
```