```
 1 package application;
 2
 3 import javafx.application.Application;
25
26 public class Main extends Application {
27     private Stage primaryStage;
28     private Scene loginScene;
29     private Scene createUserScene;
30     private HealthDataEntry healthDataEntry;
31     private User<HealthData<?>> user; // Declare user as an instance variable
32
33     private static final String url = "jdbc:sqlite:healthtracker.db";
34 //    private void printUserRecords() {
35 //        try (Connection conn = DriverManager.getConnection(url);
36 //             Statement stmt = conn.createStatement()) {
37 //            String selectUsersSql = "SELECT * FROM BmiData";
38 //            ResultSet resultSet = stmt.executeQuery(selectUsersSql);
39 //
40 //            while (resultSet.next()) {
41 //                String weight = resultSet.getString("weight");
42 //                String height = resultSet.getString("height");
43 ////                String email = resultSet.getString("email");
44 //                // Retrieve other columns as needed
45 //
46 //                System.out.println("weight: " + weight + " " + "height: " + height);
47 //            }
48 //        } catch (SQLException e) {
49 //            System.out.println("Error retrieving user data: " + e.getMessage());
50 //        }
51 //    }
52
53     @Override
54     public void start(Stage primaryStage) {
55         this.primaryStage = primaryStage;
56
57         createLoginScene();
58         createCreateUserScene();
59
60         // Set the initial scene to the login scene
61         primaryStage.setScene(loginScene);
62         primaryStage.setTitle("Login/Create User");
63         primaryStage.show();
64     }
65     private void createLoginScene() {
66         // Create UI components for login scene
67         Label usernameLabel = new Label("Username:");
68         TextField usernameTextField = new TextField();
69         Label passwordLabel = new Label("Password:");
70         PasswordField passwordTextField = new PasswordField();
71         Button loginButton = new Button("Login");
72         Button createUserButton = new Button("Create User");
73
74         // Create layout container for login scene
75         GridPane root = new GridPane();
76         root.setAlignment(Pos.CENTER);
77         root.setHgap(10);
78         root.setVgap(5);
79         root.setPadding(new Insets(10));
80
```

```
 81         // Add UI components to the grid pane
 82         root.add(usernameLabel, 0, 0);
 83         root.add(usernameTextField, 1, 0);
 84         root.add(passwordLabel, 0, 1);
 85         root.add(passwordTextField, 1, 1);
 86         root.add(loginButton, 0, 2);
 87         root.add(createUserButton, 1, 2);
 88 
 89         // Create login scene
 90         loginScene = new Scene(root, 400, 200);
 91 
 92         // Handle create user button click event
 93         createUserButton.setOnAction(event -> {
 94             primaryStage.setScene(createUserScene);
 95         });
 96 
 97         // Handle login button click event
 98         loginButton.setOnAction(event -> {
 99             String email = usernameTextField.getText();
100             String password = passwordTextField.getText();
101 
102             // Perform login validation
103             boolean loginSuccessful = validateLogin(email, password);
104 
105             if (loginSuccessful) {
106                 // Login successful
107                 System.out.println("Login successful. Email: " + email);
108 
109                 // Proceed with further logic or switch to another scene
110                 try (Connection conn = DriverManager.getConnection(url);
111                      Statement stmt = conn.createStatement()) {
112                     // Retrieve user data from the User table
113                     String getUserSql = "SELECT * FROM User WHERE email = '" + email +
    "'";
114                     ResultSet resultSet = stmt.executeQuery(getUserSql);
115 
116                     if (resultSet.next()) {
117                         // Retrieve user data from the result set
118                         String firstName = resultSet.getString("firstName");
119                         String lastName = resultSet.getString("lastName");
120                         String storedEmail = resultSet.getString("email");
121                         String storedPassword = resultSet.getString("password");
122                         LocalDate dateOfBirth = resultSet.getObject("dateOfBirth",
    LocalDate.class);
123                         String gender = resultSet.getString("gender");
124                         String phoneNumber = resultSet.getString("phoneNumber");
125 
126                         // Create User object
127                         user = new User<>(firstName, lastName, storedEmail,
    storedPassword, dateOfBirth, gender, phoneNumber);
128 
129                         // Switch to the health data entry scene
130                         healthDataEntry = new HealthDataEntry(primaryStage, user);
131 //                        printUserRecords();
132                         healthDataEntry.showHealthDataEntryScene();
133                     }
134                 } catch (SQLException e) {
135                     System.out.println("Error retrieving user data: " + e.getMessage
    ());
```

```
136                 }
137             } else {
138                 // Login failed
139                 System.out.println("Invalid email or password. Please try again.");
140 
141                 // Display an error message to the user
142                 Alert alert = new Alert(Alert.AlertType.ERROR);
143                 alert.setTitle("Login Error");
144                 alert.setHeaderText(null);
145                 alert.setContentText("Invalid email or password. Please try again.");
146                 alert.showAndWait();
147             }
148         });
149     }
150 
151 
152 
153     private void createCreateUserScene() {
154         // Create UI components for create user scene
155         Label firstNameLabel = new Label("First Name:");
156         TextField firstNameTextField = new TextField();
157         Label lastNameLabel = new Label("Last Name:");
158         TextField lastNameTextField = new TextField();
159         Label emailLabel = new Label("Email:");
160         TextField emailTextField = new TextField();
161         Label passwordLabel = new Label("Password:");
162         PasswordField passwordTextField = new PasswordField();
163         Label dateOfBirthLabel = new Label("Date of Birth:");
164         DatePicker dateOfBirthPicker = new DatePicker();
165         Label genderLabel = new Label("Gender:");
166         TextField genderTextField = new TextField();
167         Label phoneNumberLabel = new Label("Phone Number:");
168         TextField phoneNumberTextField = new TextField();
169         Button createUserButton = new Button("Create User");
170 
171         // Create layout container for create user scene
172         VBox root = new VBox(5);
173         root.setAlignment(Pos.CENTER);
174         root.setPadding(new Insets(10));
175         root.getChildren().addAll(
176                 firstNameLabel, firstNameTextField,
177                 lastNameLabel, lastNameTextField,
178                 emailLabel, emailTextField,
179                 passwordLabel, passwordTextField,
180                 dateOfBirthLabel, dateOfBirthPicker,
181                 genderLabel, genderTextField,
182                 phoneNumberLabel, phoneNumberTextField,
183                 createUserButton
184         );
185 
186         // Create create user scene
187         createUserScene = new Scene(root, 500, 500);
188 
189         // Handle create user button click event
190         createUserButton.setOnAction(event -> {
191             // Retrieve user input
192             String firstName = firstNameTextField.getText();
193             String lastName = lastNameTextField.getText();
194             String email = emailTextField.getText();
```

```
195             String password = passwordTextField.getText();
196             LocalDate dateOfBirth = dateOfBirthPicker.getValue();
197             String gender = genderTextField.getText();
198             String phoneNumber = phoneNumberTextField.getText();
199 
200             try (Connection conn = DriverManager.getConnection(url);
201                  Statement stmt = conn.createStatement()) {
202                 // Check if the email already exists in the User table
203                 String checkEmailSql = "SELECT * FROM User WHERE email = '" + email +
    "'";
204                 ResultSet resultSet = stmt.executeQuery(checkEmailSql);
205 
206                 if (resultSet.next()) {
207                     // Email already exists, show an error message
208                     Alert alert = new Alert(Alert.AlertType.ERROR);
209                     alert.setTitle("User Creation Error");
210                     alert.setHeaderText(null);
211                     alert.setContentText("Email already exists. Please use a different
    email.");
212                     alert.showAndWait();
213                 } else {
214                     // Email is unique, proceed with user creation
215                     // Create User object
216                     User<HealthData<?>> user = new User<>(firstName, lastName, email,
    password, dateOfBirth, gender, phoneNumber);
217 
218                     // Insert the user into the User table
219                     String insertUserSql = "INSERT INTO User (firstName, lastName,
    email, password, dateOfBirth, gender, phoneNumber) VALUES ('"
220                             + user.getFirstName() + "', '"
221                             + user.getLastName() + "', '"
222                             + user.getEmail() + "', '"
223                             + user.getPassword() + "', '"
224                             + user.getDateOfBirth() + "', '"
225                             + user.getGender() + "', '"
226                             + user.getPhoneNumber() + "')";
227 
228                     stmt.executeUpdate(insertUserSql);
229 
230                     // Show a success message or provide feedback to the user
231                     Alert alert = new Alert(Alert.AlertType.INFORMATION);
232                     alert.setTitle("User Creation");
233                     alert.setHeaderText(null);
234                     alert.setContentText("User created successfully!");
235                     alert.showAndWait();
236 
237                     // Switch to the health data entry scene
238                     healthDataEntry = new HealthDataEntry(primaryStage, user);
239                     healthDataEntry.showHealthDataEntryScene();
240                 }
241             } catch (SQLException e) {
242                 System.out.println("Error creating user: " + e.getMessage());
243             }
244         });
245     }
246 
247 
248     private java.sql.Date convertToDate(LocalDate localDate) {
249         return java.sql.Date.valueOf(localDate);
```

```
250     }
251
252     private boolean validateLogin(String email, String password) {
253         try (Connection conn = DriverManager.getConnection(url);
254              Statement stmt = conn.createStatement()) {
255             // Query the User table to validate login credentials
256             String selectUserSql = "SELECT * FROM User WHERE email = '" + email + "'
    AND password = '" + password + "'";
257             ResultSet resultSet = stmt.executeQuery(selectUserSql);
258
259             return resultSet.next();
260         } catch (SQLException e) {
261             System.out.println("Error querying user: " + e.getMessage());
262             return false;
263         }
264     }
265
266
267     public static void main(String[] args) {
268         launch(args);
269     }
270 }
```