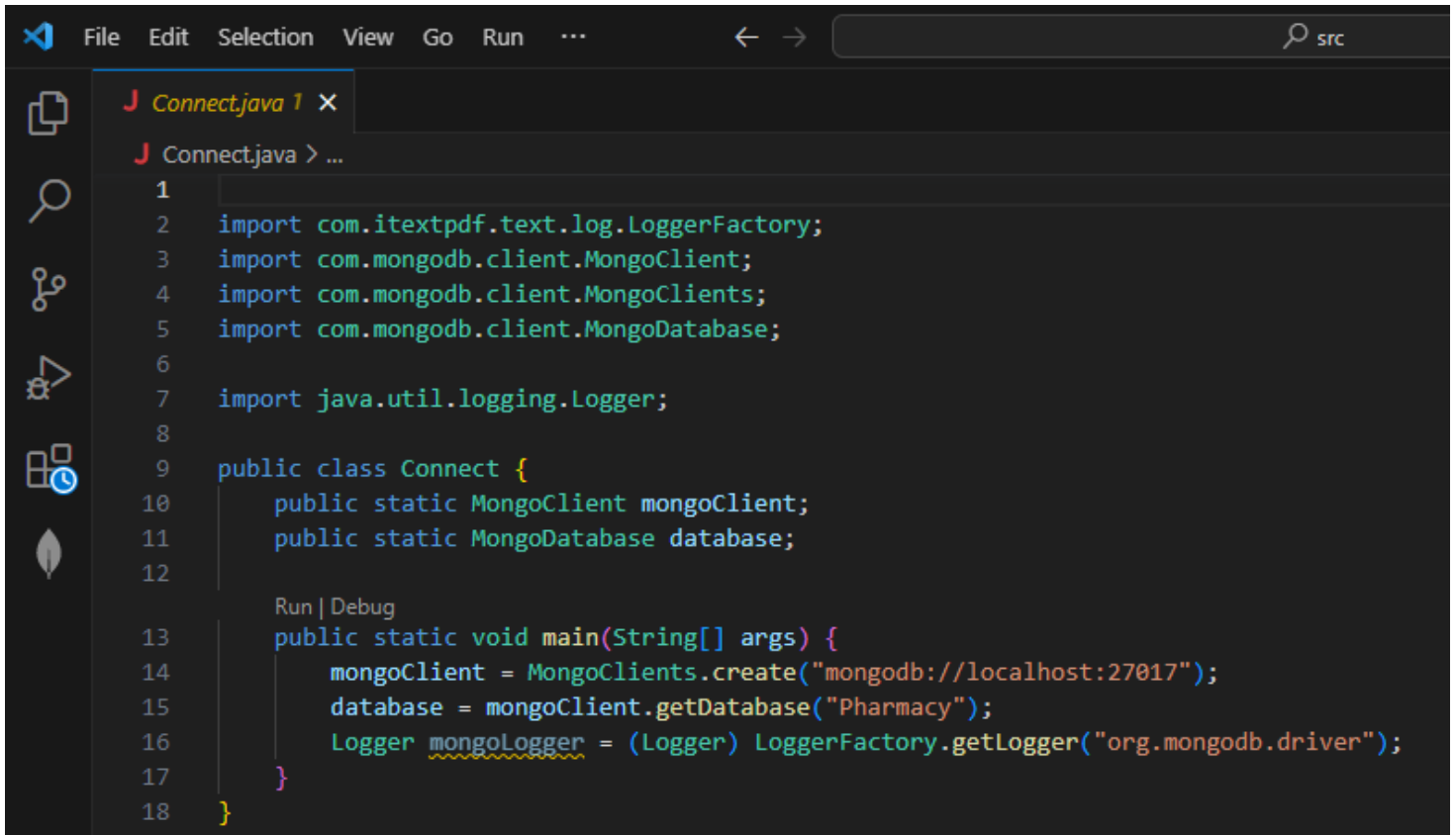
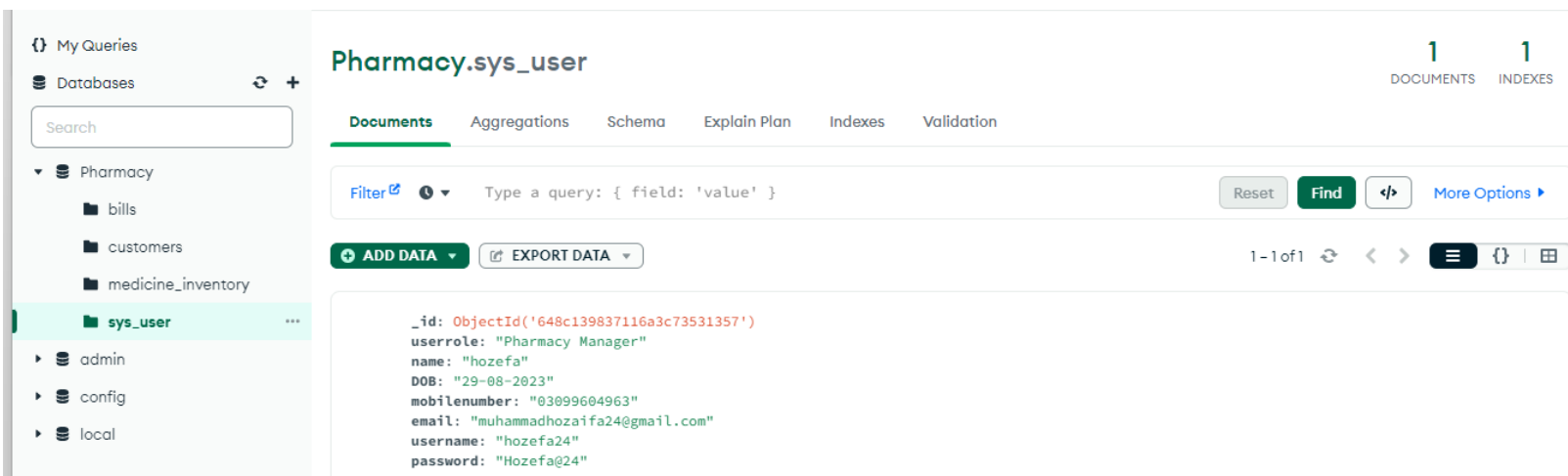


Mongo DB Connection



```
1
2 import com.itextpdf.text.log.LoggerFactory;
3 import com.mongodb.client.MongoClient;
4 import com.mongodb.client.MongoClients;
5 import com.mongodb.client.MongoDatabase;
6
7 import java.util.logging.Logger;
8
9 public class Connect {
10     public static MongoClient mongoClient;
11     public static MongoDatabase database;
12
13     Run | Debug
14     public static void main(String[] args) {
15         mongoClient = MongoClients.create("mongodb://localhost:27017");
16         database = mongoClient.getDatabase("Pharmacy");
17         Logger mongoLogger = (Logger) LoggerFactory.getLogger("org.mongodb.driver");
18     }
19 }
```

Mongo DB Compass



My Queries

Databases

Search

Pharmacy

- bills
- customers
- medicine_inventory
- sys_user**
- admin
- config
- local

Pharmacy.sys_user

1 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Filter Type a query: { field: 'value' } Reset Find More Options

ADD DATA EXPORT DATA

1 - 1 of 1

```
{
  "_id": ObjectId("648c139837116a3c73531357"),
  "userrole": "Pharmacy Manager",
  "name": "hozefa",
  "DOB": "29-08-2023",
  "mobilenumber": "03099604963",
  "email": "muhammadhozaifa24@gmail.com",
  "username": "hozefa24",
  "password": "Hozefa@24"
}
```

Add User Code

```
J Add_User.java 1 x
J Add_User.java > ...
395     try{
396         MongoClient mongoClient = MongoClient.create("mongodb://localhost:27017");
397
398         // Accessing the database
399         MongoDB database = mongoClient.getDatabase("Pharmacy");
400
401         // Accessing the collection
402         MongoCollection<Document> collection = database.getCollection("sys_user");
403
404         // Creating a query filter
405         Document query = new Document("username", username);
406
407         // Executing the query
408         MongoCursor<Document> cursor = collection.find(query).iterator();
409
410         // Checking if the username exists
411         boolean usernameExists = cursor.hasNext();
412
413         // Updating the icon and label based on the result
414         if (usernameExists) {
415             checkUsername = 1;
416             iconLabel.setIcon(new ImageIcon(filename:"C:\\Users\\Abdullah Ahmad\\Desktop\\Pharmacy-Management-System-master\\Pharmacy_Management_System\\IMG\\no.png"));
417             iconLabel.setText(text:"");
418         }
419
420         // Closing the cursor
421         cursor.close();
422
423         // Closing the MongoDB client connection
424         mongoClient.close();
425
426         }catch(Exception e){
427             JOptionPane.showMessageDialog(parentComponent:null,e);
428         }
429     }else {
430         iconLabel.setVisible(aFlag:false);
431     }
432 }
```

Update User Code

```
J Update_User.java 4 x
J Update_User.java > ...
185
186     private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
187         // TODO add your handling code here:
188         int checkUE = 0;
189         String username = username_tf.getText();
190         if(username.equals(anObject:"")){
191             JOptionPane.showMessageDialog(parentComponent:null,message:"Username Field is Required");
192         }else {
193             SimpleDateFormat dateFormat = new SimpleDateFormat(pattern:"dd-MM-yyyy");
194             try{
195                 MongoClient mongoClient = MongoClient.create("mongodb://localhost:27017");
196
197                 // Accessing the database
198                 MongoDB database = mongoClient.getDatabase("Pharmacy");
199
200                 // Accessing the collection
201                 MongoCollection<Document> collection = database.getCollection("sys_user");
202
203                 // Creating a query filter
204                 Document query = new Document("username", username);
205
206                 // Executing the query
207                 MongoCursor<Document> cursor = collection.find(query).iterator();
208                 if (cursor.hasNext()) {
209                     Document document = cursor.next();
210                     // Process the document as needed
211                     // For example:
212                     username_tf.setEditable(b:false);
213                     checkUE = 1;
214                     name_tf.setText(document.getString("name"));
215                     number_tf.setText(document.getString("mobilenumber"));
216                     email_tf.setText(document.getString("email"));
217                     String dob = document.getString("DOB");
218                     if (!dob.isEmpty()) {
219                         try {
220                             Date date = new SimpleDateFormat(pattern:"yyyy-MM-dd").parse(dob);
221                             date_tf.setDate(date);
222                         } catch (Exception e) {
223                             JOptionPane.showMessageDialog(parentComponent:null, e);
224                         }
225                     }
226                 }
227             }
228         }
229     }
```

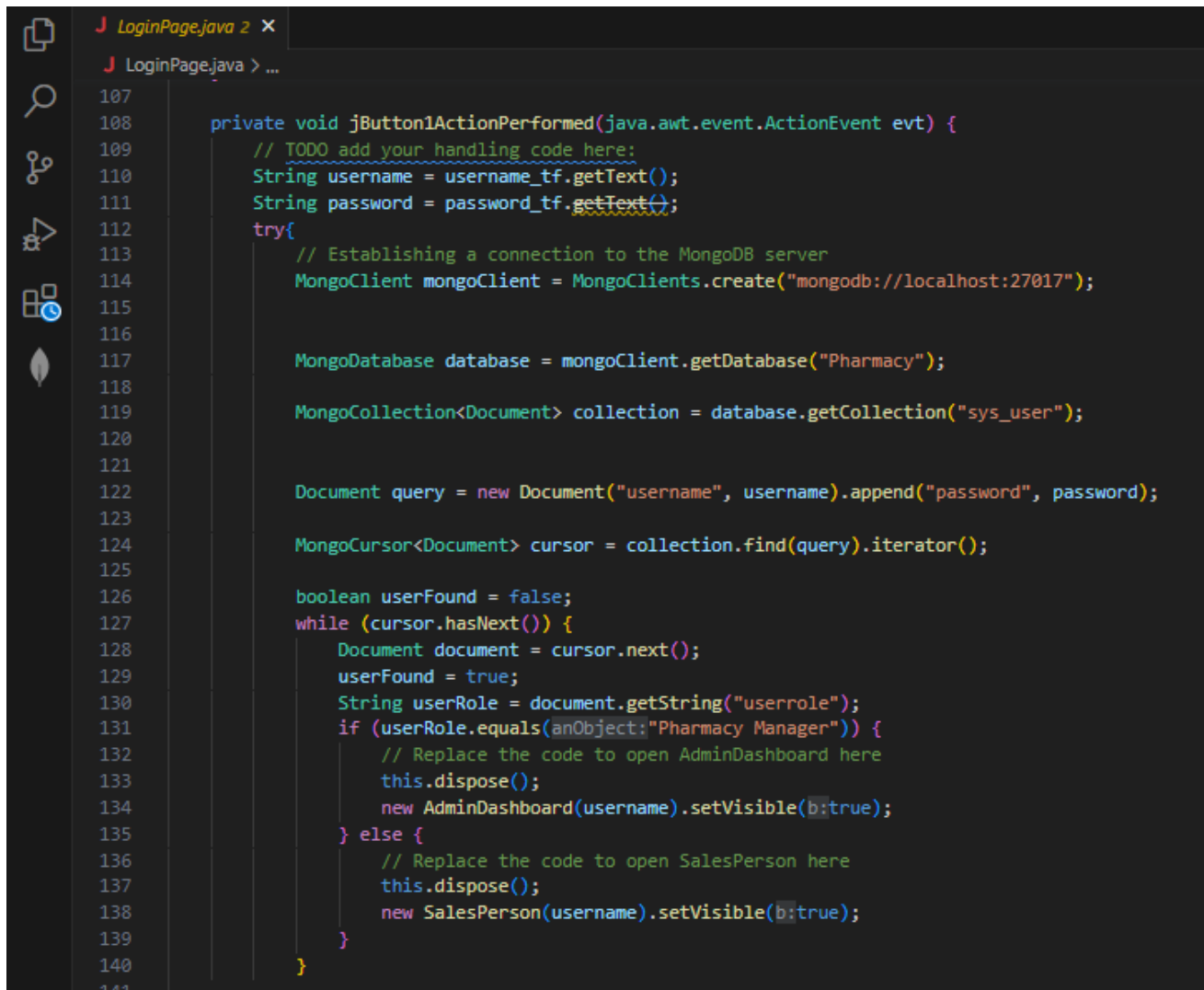
Add Medicine Code

```
J Add_Medicine.java 2 x
J Add_Medicine.java > ...
224         try{
225             // Establishing a connection to the MongoDB server
226             MongoClient mongoClient = MongoClient.create("mongodb://localhost:27017");
227
228             // Accessing the database
229             MongoDB database = mongoClient.getDatabase("Pharmacy");
230
231             // Accessing the collection
232             MongoCollection<Document> collection = database.getCollection("medicine_inventory");
233
234             // Creating a new document
235             Document document = new Document("uniqueID", uniqueID)
236                 .append("name", name)
237                 .append("companyName", companyName)
238                 .append("quantity", Integer.parseInt(quantity))
239                 .append("price", Integer.parseInt(price))
240                 .append("expiry", expiry);
241
242             // Inserting the document into the collection
243             collection.insertOne(document);
244
245             // Closing the MongoDB client connection
246             mongoClient.close();
247
248             JOptionPane.showMessageDialog(parentComponent:null, message:"Medicine Added Successfully");
249             setVisible(b:false);
250             new Add_Medicine().setVisible(b:true);
251         }
252         catch(Exception e){
253             JOptionPane.showMessageDialog(parentComponent:null,e);
254         }
255     }
```

Update Medicine Code

```
J UpdateMedicine.java 2 x
J UpdateMedicine.java > ...
259         JOptionPane.showMessageDialog(parentComponent:null,message:"Price Per Unit Field is invalid");
260     }
261     else{
262         try{
263             // Establishing a connection to the MongoDB server
264             MongoClient mongoClient = MongoClient.create("mongodb://localhost:27017");
265
266             // Accessing the database
267             MongoDB database = mongoClient.getDatabase("Pharmacy");
268
269             // Accessing the collection
270             MongoCollection<Document> collection = database.getCollection("medicine_inventory");
271
272             // Creating a query filter
273             Document query = new Document("uniqueID", uniqueId);
274
275             // Creating an update document
276             Document update = new Document("$set", new Document("name", name)
277                 .append("companyName", companyName)
278                 .append("quantity", totalQuantity)
279                 .append("price", price));
280
281             // Updating the document in the collection
282             collection.updateOne(query, update);
283
284             // Closing the MongoDB client connection
285             mongoClient.close();
286
287             JOptionPane.showMessageDialog(parentComponent:null, message:"Medicine Updated Successfully");
288             setVisible(b:false);
289             new UpdateMedicine().setVisible(b:true);
290         }
291         catch(Exception e){
292             JOptionPane.showMessageDialog(parentComponent:null,e);
293         }
294     }
295 }
```

Login Page Code



```
107
108 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
109     // TODO add your handling code here:
110     String username = username_tf.getText();
111     String password = password_tf.getText();
112     try{
113         // Establishing a connection to the MongoDB server
114         MongoClient mongoClient = MongoClient.create("mongodb://localhost:27017");
115
116
117         MongoDB database = mongoClient.getDatabase("Pharmacy");
118
119         MongoCollection<Document> collection = database.getCollection("sys_user");
120
121
122         Document query = new Document("username", username).append("password", password);
123
124         MongoCursor<Document> cursor = collection.find(query).iterator();
125
126         boolean userFound = false;
127         while (cursor.hasNext()) {
128             Document document = cursor.next();
129             userFound = true;
130             String userRole = document.getString("userrole");
131             if (userRole.equals(anObject:"Pharmacy Manager")) {
132                 // Replace the code to open AdminDashboard here
133                 this.dispose();
134                 new AdminDashboard(username).setVisible(b:true);
135             } else {
136                 // Replace the code to open SalesPerson here
137                 this.dispose();
138                 new SalesPerson(username).setVisible(b:true);
139             }
140         }
141     }
```