

# Pasindu Geevinda

20212016

```
Product.java
...
tblPrdk1.setClmDes(DOC.getString(key: "description"));
tblPrdk1.setClmCtgry(DOC.getString(key: "CategoryName"));
tblPrdk1.setQty(DOC.getString(key: "Quantity"));
List.add(tblPrdk1);

return List;
}

ProductDelete productDelete = new ProductDelete();

@Override
public void initialize(URL location, ResourceBundle resources) {
    productDelete.I0set(product);
    iterate();
    table.setItems(iterate());
}

clmid.setCellValueFactory(new PropertyValueFactory<>("clmid"));
Qty.setCellValueFactory(new PropertyValueFactory<>("Qty"));
clmname.setCellValueFactory(new PropertyValueFactory<>("clmname"));
clmCtgry.setCellValueFactory(new PropertyValueFactory<>("clmCtgry"));
clmDes.setCellValueFactory(new PropertyValueFactory<>("clmDes"));

public void logout(ActionEvent event) throws IOException {
    window.closeWindow(event);

    FXMLLoader fxmlLoader = new FXMLLoader(LoginController.class.getResource(name: "LoginPage.fxml"));
    window.newWindow(fxmlLoader);
}
}
```

```
Product.java
...
FXMLLoader fxmlLoader = new FXMLLoader(LoginController.class.getResource(name: "LoginPage.fxml"));
window.newWindow(fxmlLoader);

public void home(ActionEvent event) throws IOException {
    window.closeWindow(event);

    FXMLLoader fxmlLoader = new FXMLLoader(Home.class.getResource(name: "home.fxml"));
    window.newWindow(fxmlLoader);
}

public void NewProduct(ActionEvent event) throws IOException {
    window.closeWindow(event);
    FXMLLoader fxmlLoader = new FXMLLoader(User.class.getResource(name: "ProductNew.fxml"));
    window.newWindow(fxmlLoader);
}

public void EditProduct(ActionEvent event) throws IOException {
    Error.warning("Enter the id of the product you want to edit first and press the set button");

    FXMLLoader fxmlLoader = new FXMLLoader(User.class.getResource(name: "ProductEdit.fxml"));
    window.newWindow(fxmlLoader);
    window.closeWindow(event);
}

public void DeleteProduct(ActionEvent event) throws IOException {
    FXMLLoader fxmlLoader = new FXMLLoader(User.class.getResource(name: "ProductDelete.fxml"));
    window.newWindow(fxmlLoader);
    window.closeWindow(event);
}
}
```

The screenshot shows the IntelliJ IDEA interface with the code editor open to the `ProductDelete.java` file. The code implements the `Initializable` interface and contains methods for deleting categories and products from a MongoDB database. It uses JavaFX annotations (`@FXML`) and the MongoDB Java Driver (`DBConnection`, `MongoCollection`). The code includes logic to find products by name and ID, delete them, and update the database.

```
public class ProductDelete implements Initializable {
    @FXML
    public TextField DCategory;

    @FXML
    public TextField CategoryID;

    DBConnection dbConnection = DBConnection.getConnection();
    MongoCollection<Document> product = dbConnection.getMongoCollection("Product");

    public void DeleteCategory() {
        String DltProduct = DCategory.getText();
        String ID = CategoryID.getText();

        ArrayList<String> productName = Find.findAll( "ProductName", product );
        ArrayList<String> IDList = Find.find(product, DltProduct, findBy: "ProductName", findField: "ID");

        if (!DltProduct.equals("") || !ID.equals ""){
            if (productName.contains(DltProduct.toUpperCase())){
                if (IDList.contains(ID)){
                    Document document = new Document();
                    document.put("ProductName", DltProduct.toUpperCase());
                    document.put("ID", ID);
                    product.deleteOne(document);
                    Error.statics("Product Deleted");
                    DCategory.setText("");
                    CategoryID.setText("");
                    IDset(product);
                } else {
                    Error.error("Product not Exists");
                }
            } else {
                Error.error("Some fields are empty");
            }
        }
    }

    public void Back(ActionEvent event) throws IOException {
        FXMLLoader fxmlLoader = new FXMLLoader(Home.class.getResource( name: "Product.fxml" ));
        window.newWindow(fxmlLoader);

        window.closeWindow(event);
    }

    public void IDset(MongoCollection<Document> collection) {
        ArrayList<String> count =Find.findAll( "ID",collection );
        for ( int i=0;i<count.size();i++ ) {
            BasicDBObject basicDBObject = new BasicDBObject("ID",count.get(i));
            BasicDBObject basicDBObject2 = new BasicDBObject();
            basicDBObject2.put("ID",String.valueOf(i+1));
            BasicDBObject updateDBObject = new BasicDBObject();
            updateDBObject.put("$set", basicDBObject2); // ($)
            collection.updateOne(basicDBObject, updateDBObject);
        }
    }
}
```

The screenshot shows the continuation of the `ProductDelete.java` code in IntelliJ IDEA. The code handles the back button logic, which loads a FXML file named `Product.fxml`. It also contains the `IDset` method, which updates the database by changing the primary key of documents. The code uses the MongoDB Java Driver to interact with the database.

```
    }
}

public void Back(ActionEvent event) throws IOException {
    FXMLLoader fxmlLoader = new FXMLLoader(Home.class.getResource( name: "Product.fxml" ));
    window.newWindow(fxmlLoader);

    window.closeWindow(event);
}

public void IDset(MongoCollection<Document> collection) {
    ArrayList<String> count =Find.findAll( "ID",collection );
    for ( int i=0;i<count.size();i++ ) {
        BasicDBObject basicDBObject = new BasicDBObject("ID",count.get(i));
        BasicDBObject basicDBObject2 = new BasicDBObject();
        basicDBObject2.put("ID",String.valueOf(i+1));
        BasicDBObject updateDBObject = new BasicDBObject();
        updateDBObject.put("$set", basicDBObject2); // ($)
        collection.updateOne(basicDBObject, updateDBObject);
    }
}
```

```
public class ProductEdit implements Initializable {
    public ChoiceBox<String> Dropbox;
    public TextField Quantity;
    public TextArea ProDescription;
    public TextField NPProduct;
    public TextField ProID;
    public Button editbutton;

    DBCollection dbConnection = DBConnection.getConnection();
    MongoCollection<Document> ProductColl = dbConnection.getMongoCollection("Product");
    MongoCollection<Document> Categories = dbConnection.getMongoCollection("Categories");

    public void EditProduct() {
        if (!Objects.equals(ProID.getText(), "")) {
            String chicebox = Dropbox.getValue();

            ArrayList<String> Name = Find.findall("CategoryName", ProductColl);
            if (!(Name.contains(NPProduct.getText().toUpperCase()))){

                BasicDBObject doc = new BasicDBObject("ID", ProID.getText());
                BasicDBObject basicDBObject2 = new BasicDBObject();
                if (!Objects.equals(NPProduct.getText(), "")) {
                    basicDBObject2.put("ProductName", NPProduct.getText().toUpperCase());
                }
                if (!Objects.equals(ProDescription.getText(), "")) {
                    basicDBObject2.put("Description", ProDescription.getText());
                }
                basicDBObject2.put("CategoryName",chicebox);
                if (!Objects.equals(Quantity.getText(), "")) {
                    basicDBObject2.put("Quantity", Quantity.getText());
                }
                BasicDBObject updateObject = new BasicDBObject("$");
                ProductColl.updateOne(doc, updateObject);
                Error.statics("Category Edited");
                NPProduct.setText("");
                ProDescription.setText("");
                Dropbox.setValue("");
                Quantity.setText("");
            } else {
                Error.error("This Category name already exists");
            }
        } else {
            Error.error("Enter the id of the product you want to edit first and press the set button");
        }
    }

    public void Back(ActionEvent event) throws IOException {
        FXMLLoader fxmLoader = new FXMLLoader(Home.class.getResource("Product.fxml"));
        window.newWindow(fxmLoader);
        window.closeWindow(event);
    }

    public void Set() {
        ArrayList<String> num = Find.findall("ID",ProductColl);
        ArrayList<String> name = Find.findall("ProductName",ProductColl);
        if (Objects.equals(ProID.getText(), "")){
            Error.error("Search Field is empty");
        }else if (num.contains(ProID.getText())){
            ArrayList<String> set = new ArrayList<>();
            set.add(Find.find(ProductColl, ProID.getText(), findBy: "ID", findField: "ProductName").get(0));
            set.add(Find.find(ProductColl, ProID.getText(), findBy: "ID", findField: "description").get(0));
            set.add(Find.find(ProductColl, ProID.getText(), findBy: "ID", findField: "CategoryName").get(0));
            set.add(Find.find(ProductColl, ProID.getText(), findBy: "ID", findField: "Quantity").get(0));
        }
    }
}
```

```
ArrayList<String> num = Find.findall("ID",ProductColl);
ArrayList<String> name = Find.findall("ProductName",ProductColl);
if (Objects.equals(ProID.getText(), "")){
    Error.error("Search Field is empty");
} else if (num.contains(ProID.getText())){
    ArrayList<String> set = new ArrayList<>();
    set.add(Find.find(ProductColl, ProID.getText(), findBy: "ID", findField: "ProductName").get(0));
    set.add(Find.find(ProductColl, ProID.getText(), findBy: "ID", findField: "description").get(0));
    set.add(Find.find(ProductColl, ProID.getText(), findBy: "ID", findField: "CategoryName").get(0));
    set.add(Find.find(ProductColl, ProID.getText(), findBy: "ID", findField: "Quantity").get(0));
}
```

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CW [C:\Users\Lucifer Diots\Desktop\CW] - ProductEdit.java

```
set.add((Find.find(ProductColl, ProID.getText(), findBy: "ID", findField: "CategoryName")).get(0));
set.add((Find.find(ProductColl, ProID.getText(), findBy: "ID", findField: "Quantity")).get(0));
NProduct.setText(set.get(0));
ProDescription.setText(set.get(1));
Dropbox.setValue(set.get(2));
Quantity.setText(set.get(3));
} else if (name.contains(ProID.getText().toUpperCase())) {
ArrayList<String> set = new ArrayList<>();
set.add((Find.find(ProductColl, ProID.getText(), findBy: "ProductName", findField: "ProductName")).get(0));
set.add((Find.find(ProductColl, ProID.getText(), findBy: "ProductName", findField: "description")).get(0));
set.add((Find.find(ProductColl, ProID.getText(), findBy: "ProductName", findField: "CategoryName")).get(0));
set.add((Find.find(ProductColl, ProID.getText(), findBy: "ProductName", findField: "Quantity")).get(0));
NProduct.setText(set.get(0));
ProDescription.setText(set.get(1));
Dropbox.setValue(set.get(2));
Quantity.setText(set.get(3));
}
else {
    Error.error("Enter the id of the product you want to edit first and press the set button");
}
}

@Override
public void initialize(URL location, ResourceBundle resources) {
    ArrayList<String> AllCategories3 = Find.findAll( findField: "CategoryName", Categories);
    AllCategories3.add("NONE");
    for (String allcategory : AllCategories3) {
        Dropbox.setValue("NONE");
        Dropbox.getItems().add(allcategory);
    }
}
```

Version Control Run TODO Problems Profiler Terminal Build Dependencies

Build completed successfully in 8 sec, 595 ms (15 minutes ago)

8:46 CRLF UTF-8 4 spaces ENG 1255 AM 12/17/2021

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CW [C:\Users\Lucifer Diots\Desktop\CW] - ProductNew.java

```
public class ProductNew implements Initializable{
    @FXML
    public TextField NProduct;
    public TextArea ProDescription;
    public TextField Quantity;
    public ChoiceBox<String> Dropbox;

    DBConnection dbConnection = DBConnection.getConnection();
    MongoCollection<Document> Categories = dbConnection.getMongoCollection("Categories");

    public void EnterProduct() {

        String NewProduct = NProduct.getText();
        String PrDescription = ProDescription.getText();
        String chicebox = Dropbox.getValue();
        try {
            String IntQua = Quantity.getText();
            MongoCollection<Document> product = dbConnection.getMongoCollection("Product");

            ArrayList<String> ProductList = Find.findAll( findField: "ProductName", product);

            if (!ProductList.contains(NewProduct.toUpperCase())){
                if (NewProduct.equals("") || Objects.equals(PrDescription, " ") || IntQua.equals("")){
                    Error.error("Some fields are empty");
                }
            }
            else {
                Document document = new Document();
                document.append("CategoryName", chicebox);
                document.append("ProductName", NewProduct);
                document.append("description", PrDescription.getText());
                document.append("Quantity", Integer.parseInt(IntQua));
                dbConnection.insertOne(document);
                Error.info("Product added successfully");
            }
        } catch (Exception e) {
            Error.error("An error occurred while adding the product");
        }
    }
}
```

Version Control Run TODO Problems Profiler Terminal Build Dependencies

Build completed successfully in 8 sec, 595 ms (15 minutes ago)

7:16 CRLF UTF-8 4 spaces ENG 1256 AM 12/17/2021

Screenshot of the Java code editor showing the `ProductNew.java` file. The code handles product creation logic, including validation for category existence and quantity being a number.

```
    settext();
}
} else {
    Error.error("Product already Exists");
}
} catch (Exception exc) {
    Error.error("Qty must be number");
}
}

public void Back (ActionEvent event) throws IOException {
FXMLLoader fxmlLoader = new FXMLLoader(Home.class.getResource( name: "Product.fxml"));
window.newWindow(fxmlLoader);
window.closeWindow(event);
}

public void settext() {
    NProduct.setText("");
    ProDescription.setText("");
    Quantity.setText("");
    Dropbox.setValue("NONE");
}

@Override
public void initialize(URL location, ResourceBundle resources) {
    ArrayList<String> AllCategories3 = Find.findAll( findField: "CategoryName", Categories);
    AllCategories3.add("NONE");
    for ( String allCategory : AllCategories3) {
        dropdown.setvalue("NONE");
    }
}
```

Screenshot of the Java code editor showing the `tblPrdt.java` file. This class represents a product entry in a database table, with methods for setting and getting various product details.

```
public tblPrdt(String clmid, String qty, String clmname, String clmnCtgy, String clmDes) {
    this.clmid = clmid;
    Qty = qty;
    this.clmname = clmname;
    this.clmnCtgy = clmnCtgy;
    this.clmDes = clmDes;
}

public String getClmid() { return clmid; }

public void setClmid(String clmid) { this.clmid = clmid; }

public String getQty() {
    return Qty;
}

public void setQty(String qty) {
    Qty = qty;
}

public String getClmname() { return clmname; }

public void setClmname(String clmname) { this.clmname = clmname; }

public String getClmnCtgy() { return clmnCtgy; }

public void setClmnCtgy(String clmnCtgy) { this.clmnCtgy = clmnCtgy; }

public String getClmDes() { return clmDes; }
```

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CW (C:\Users\Lucifer Diots\Desktop\CW) - Stock.java

Project: CW C:\Users\Lucifer Diots\Desktop\CW

src

- main
- java
- com.example.cw
- Categories
- Product
- Stock
- functions

Stock.java X

```
import java.util.ResourceBundle;
public class Stock implements Initializable {
    public TableColumn<tblStock, String> Supplier;
    public TableColumn<tblStock, String> Updated;
    public TableColumn<tblStock, String> ProdName;
    public TableColumn<tblStock, String> Qty;
    public TableColumn<tblStock, String> UnitPrice;
    public TableColumn<tblStock, String> TotalPrice;
    public TableView<tblStock> tbl;
    public TextField ProNameTxt;
    public TextField UnitPriceTxt;
    public TextField QtyText;
    public TextField SupplierTxt;
    public TextField DeleteTxt;

    DBConnection dbConnection = DBConnection.getConnection();
    MongoCollection<Document> StockColl = dbConnection.getMongoCollection("Stock");
    ObservableList<tblStock> iterate() {
        MongoCollection<Document> getMongoCollection(String collection)
        FindIterable<Document> Find = StockColl.find();
        ObservableList<tblStock> List = FXCollections.observableArrayList();
        FindIterable<Document> Find = StockColl.find();
        for (Document DOC : Find) {
            tblStock StockDoc = new tblStock();
            StockDoc.setUpdated(DOC.getString(key: "ID"));
            StockDoc.setProdName(DOC.getString(key: "ProductName"));
            StockDoc.setQty(DOC.getString(key: "Quantity"));
            StockDoc.setUnitPrice(DOC.getString(key: "UnitPrice"));
            StockDoc.setSupplier(DOC.getString(key: "Supplier"));
            StockDoc.setTotalPrice(DOC.getString(key: "TotalPrice"));
        }
        return List;
    }

    public void home(ActionEvent event) throws IOException {
        window.closeWindow(event);
        FXMLLoader fxmlLoader = new FXMLLoader(Home.class.getResource(name: "home.fxml"));
        window.newWindow(fxmlLoader);
    }

    public void logout(ActionEvent event) throws IOException {
        window.closeWindow(event);
        FXMLLoader fxmlLoader = new FXMLLoader(Home.class.getResource(name: "LoginPage.fxml"));
        window.newWindow(fxmlLoader);
    }

    ArrayList<String> ProductList = Find.findAll(findField: "ProductName", StockColl);

    public void AddStock(ActionEvent event) {
        try {
            String ProductName = ProNameTxt.getText().toUpperCase();
            String UnitPrice = String.valueOf(Integer.parseInt(UnitPriceTxt.getText()));
            String Quantity = String.valueOf(Integer.parseInt(QtyText.getText()));
            String SupplierVari = SupplierTxt.getText();
            String TotalPrice = String.valueOf(Integer.parseInt(UnitPrice)*Integer.parseInt(Quantity));
            if (!ProductList.contains(ProductName.toUpperCase())) {
                if (!(ProductName.equals("") || Objects.equals(SupplierVari, "")) || Quantity.equals("") || UnitPrice.equals("")))
                    Error.statics("Product Stock Added");
            }
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}
```

Build completed successfully in 8 sec, 595 ms (16 minutes ago)

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CW (C:\Users\Lucifer Diots\Desktop\CW) - Stock.java

Project: CW C:\Users\Lucifer Diots\Desktop\CW

src

- main
- java
- com.example.cw
- Categories
- Product
- Stock
- functions

Stock.java X

```
}

    return List;
}

public void home(ActionEvent event) throws IOException {
    window.closeWindow(event);
    FXMLLoader fxmlLoader = new FXMLLoader(Home.class.getResource(name: "home.fxml"));
    window.newWindow(fxmlLoader);
}

public void logout(ActionEvent event) throws IOException {
    window.closeWindow(event);
    FXMLLoader fxmlLoader = new FXMLLoader(Home.class.getResource(name: "LoginPage.fxml"));
    window.newWindow(fxmlLoader);
}

ArrayList<String> ProductList = Find.findAll(findField: "ProductName", StockColl);

public void AddStock(ActionEvent event) {
    try {
        String ProductName = ProNameTxt.getText().toUpperCase();
        String UnitPrice = String.valueOf(Integer.parseInt(UnitPriceTxt.getText()));
        String Quantity = String.valueOf(Integer.parseInt(QtyText.getText()));
        String SupplierVari = SupplierTxt.getText();
        String TotalPrice = String.valueOf(Integer.parseInt(UnitPrice)*Integer.parseInt(Quantity));
        if (!ProductList.contains(ProductName.toUpperCase())) {
            if (!(ProductName.equals("") || Objects.equals(SupplierVari, "")) || Quantity.equals("") || UnitPrice.equals("")))
                Error.statics("Product Stock Added");
        }
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}
```

Build completed successfully in 8 sec, 595 ms (17 minutes ago)

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CW [C:\Users\Lucifer Dots\Desktop\CW] - Stock.java

Project: CW C:\Users\Lucifer Dots\Desktop\CW

src

- main
- java
- com.example.cw
- Stock
- AddStock

Stock.java

```
document.put("ID", String.valueOf(productList.size() + i));
document.put("ProductName", ProductName);
document.put("UnitPrize", UnitPrize);
document.put("Quantity", Quantity);
document.put("TotalPrice", TotalPrice);
document.put("Supplier", SupplierVari);
StockColl.insertOne(document);
window.closeWindow(event);

reArrange();
FXMLLoader fxmlLoader = new FXMLLoader(Home.class.getResource( name: "Stock.fxml"));
window.newWindow(fxmlLoader);

} else {
    Error.error("Some fields are empty");
}
} else {
    Error.error("Product Name is already exists");
}
} catch (NumberFormatException e) {
    Error.error("Quantity and UnitPrize must be Integer");
} catch (IOException e) {
    e.printStackTrace();
}

public void Delete(ActionEvent event) throws IOException {

    String ID = DeleteTxt.getText();

    ArrayList<String> productName = Find.findAll( findField: "ID", StockColl);
```

Version Control Run TODO Profiler Terminal Build Dependencies

Build completed successfully in 8 sec, 595 ms (17 minutes ago)

100:43 CRLF UTF-8 4 spaces ENG 12:57 AM 12/17/2021

Type here to search

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CW [C:\Users\Lucifer Dots\Desktop\CW] - Stock.java

Project: CW C:\Users\Lucifer Dots\Desktop\CW

src

- main
- java
- com.example.cw
- Stock
- AddStock

Stock.java

```
if (!ID.equals("")){
    if (productName.contains(ID.toUpperCase())){
        Document document = new Document();
        document.put("ID", ID);
        StockColl.deleteOne(document);
        reArrange();
        window.closeWindow(event);
        Error.statics("Stock Deleted");
        FXMLLoader fxmlLoader = new FXMLLoader(Home.class.getResource( name: "Stock.fxml"));
        window.newWindow(fxmlLoader);
    }
    else {
        Error.error("Product not Exists");
    }
} else {
    Error.error("ID field is empty");
}

public void reArrange(){
    ArrayList<String> count =Find.findAll( findField: "ID", StockColl);
    for (int i=0;i<count.size();i++){
        BasicDBObject basicDBObject = new BasicDBObject("ID",count.get(i));
        BasicDBObject basicDBObject2 = new BasicDBObject();
        basicDBObject2.put("ID",String.valueOf(i+1));
        BasicDBObject updateObject = new BasicDBObject();
        updateObject.put("$set", basicDBObject2); // ($)
        StockColl.updateOne(basicDBObject, updateObject);
    }
}
```

Version Control Run TODO Profiler Terminal Build Dependencies

Build completed successfully in 8 sec, 595 ms (17 minutes ago)

100:43 CRLF UTF-8 4 spaces ENG 12:57 AM 12/17/2021

Type here to search

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CW [C:\Users\Lucifer Diotics\Desktop\CW] - Stock.java

Project: CW src main java com example cw Stock Stock AddStock

Stock.java

```
159     window.closeWindow(event);  
160  
161     Error.warning("Enter the relevant ID or Product Name that you want to edit and press the Set button search");  
162  
163     FXMLLoader fxmlLoader = new FXMLLoader(User.class.getResource( name: "StockEdit2.fxml"));  
164     window.newWindow(fxmlLoader);  
165 }  
166  
167 @Override  
168 public void initialize(URL location, ResourceBundle resources) {  
169     try {  
170         reArrange();  
171  
172         iterate();  
173  
174         Updated.setCellValueFactory(new PropertyValueFactory<>("Updated"));  
175         ProdName.setCellValueFactory(new PropertyValueFactory<>("ProdName"));  
176         Qty.setCellValueFactory(new PropertyValueFactory<>("Qty"));  
177         UnitPrice.setCellValueFactory(new PropertyValueFactory<>("UnitPrice"));  
178         TotalPrice.setCellValueFactory(new PropertyValueFactory<>("TotalPrice"));  
179         Supplier.setCellValueFactory(new PropertyValueFactory<>("Supplier"));  
180  
181         tbl.setItems(iterate());  
182     } catch (Exception ignored) {  
183     }  
184 }  
185  
186 }  
187 }
```

Version Control Run TODO Problems Profiler Terminal Build Dependencies

Build completed successfully in 8 sec, 595 ms (17 minutes ago)

100:43 CRLF UTF-8 4 spaces ENG 1258 AM 12/17/2021

Type here to search

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CW [C:\Users\Lucifer Diotics\Desktop\CW] - StockEdit2.java

Project: CW src main java com example cw Stock StockEdit2 SetStock

StockEdit2.java

```
17 import java.util.Objects;  
18  
19 public class StockEdit2 extends Stock{  
20     @FXML  
21     public TextField UnitPrz;  
22     @FXML  
23     public TextField EditQTY;  
24     @FXML  
25     public TextField PrdctNm;  
26     @FXML  
27     public TextField EditSupplier;  
28     @FXML  
29     public TextField StockID;  
30  
31     public void EditStock() {  
32         BasicDBObject doc = new BasicDBObject("ID", StockID.getText());  
33         FindIterable<Document> findIterable = StockColl.find(doc);  
34         for (Document DOC : findIterable){  
35             BasicDBObject basicDBObject2 = new BasicDBObject();  
36             if (!Objects.equals(PrdctNm.getText(), "")) {  
37                 basicDBObject2.put("ProductName", PrdctNm.getText().toUpperCase());  
38             }if (!Objects.equals(UnitPrz.getText(), "")) {  
39                 basicDBObject2.put("UnitPrice", UnitPrz.getText());  
40             }if (!Objects.equals(EditQTY.getText(), "")) {  
41                 basicDBObject2.put("Quantity", EditQTY.getText());  
42             }if (!Objects.equals(EditSupplier.getText(), "")) {  
43                 basicDBObject2.put("Supplier", EditSupplier.getText());  
44             }if (!Objects.equals(EditQTY.getText(), "") || !Objects.equals(UnitPrz.getText(), "")) {  
45                 if (!Objects.equals(EditQTY.getText(), "")) {  
46                     String Total = String.valueOf(DOC.getInteger(key: "UnitPrice")*Integer.parseInt(EditQTY.getText()));  
47                     basicDBObject2.put("TotalPrice", Total);  
48             }  
49         }  
50     }  
51 }
```

Version Control Run TODO Problems Profiler Terminal Build Dependencies

Build completed successfully in 8 sec, 595 ms (18 minutes ago)

69:72 (125 chars, 1 line break) CRLF UTF-8 4 spaces ENG 1258 AM 12/17/2021

Type here to search

The screenshot shows the IntelliJ IDEA interface with the StockEdit2.java file open. The code implements a method to update a stock item based on its ID. It uses a BasicDBObject to build an update query, setting the quantity and total price. It then performs a find-and-update operation on the StockColl collection. After the update, it sets various UI components (PrdtNm, UnitPrz, EditQTY, EditSupplier) to empty strings. Finally, it calls a SetStock() method.

```
String Total = String.valueOf(DOC.getInteger("Quantity") * Integer.parseInt(UnitPrz.getText()));
basicDBObject.put("TotalPrice", Total);

BasicDBObject updateObject = new BasicDBObject();
updateObject.put("$_set", basicDBObject);
StockColl.updateOne(DOC, updateObject);

Error.statics("Stock Edited");
PrdtNm.setText("");
UnitPrz.setText("");
EditQTY.setText("");
EditSupplier.setText("");

}

public void SetStock() {
    ArrayList<String> num = Find.findall("ID", StockColl);
    ArrayList<String> name = Find.findall("ProductName", StockColl);
    if (!Objects.equals(StockID.getText(), "")) {
        Error.error("Enter the id of the product you want to edit first and press the set button");
    }
    else if (num.contains(StockID.getText())) {
        ArrayList<String> set = new ArrayList<>();
        set.add(Find.find(StockColl, StockID.getText(), findBy: "ID", findField: "ProductName").get(0));
        set.add(Find.find(StockColl, StockID.getText(), findBy: "ID", findField: "UnitPrice").get(0));
        set.add(Find.find(StockColl, StockID.getText(), findBy: "ID", findField: "Quantity").get(0));
        set.add(Find.find(StockColl, StockID.getText(), findBy: "ID", findField: "Supplier").get(0));
        PrdtNm.setText(set.get(0));
        UnitPrz.setText(set.get(1));
        EditQTY.setText(set.get(2));
        EditSupplier.setText(set.get(3));
    }
    else if (name.contains(StockID.getText().toUpperCase())){
        ArrayList<String> set = new ArrayList<>();
        set.add(Find.find(StockColl, StockID.getText(), findBy: "ProductName", findField: "ProductName").get(0));
        set.add(Find.find(StockColl, StockID.getText(), findBy: "ProductName", findField: "UnitPrice").get(0));
        set.add(Find.find(StockColl, StockID.getText(), findBy: "ProductName", findField: "Quantity").get(0));
        set.add(Find.find(StockColl, StockID.getText(), findBy: "ProductName", findField: "Supplier").get(0));
        PrdtNm.setText(set.get(0));
        UnitPrz.setText(set.get(1));
        EditQTY.setText(set.get(2));
        EditSupplier.setText(set.get(3));
    }
    else {
        Error.error("This ID or ProductName doesn't exists");
    }
}
```

This screenshot shows the same IntelliJ IDEA interface with the StockEdit2.java file open. The code has been modified to include a Back() method. This method handles an ActionEvent, closes the current window, iterates through a loop, and then creates a new window using FXMLLoader to load the Stock.fxml file.

```
set.add(Find.find(StockColl, StockID.getText(), findBy: "ID", findField: "Quantity").get(0));
set.add(Find.find(StockColl, StockID.getText(), findBy: "ID", findField: "Supplier").get(0));
PrdtNm.setText(set.get(0));
UnitPrz.setText(set.get(1));
EditQTY.setText(set.get(2));
EditSupplier.setText(set.get(3));

}

public void Back(ActionEvent event) throws IOException {
    window.closeWindow(event);
    iterate();
    FXMLLoader fxmlLoader = new FXMLLoader(Home.class.getResource( name: "Stock.fxml"));
    window.newWindow(fxmlLoader);
}
```

The screenshot shows the IntelliJ IDEA interface with the project navigation bar at the top. The current file is `tbStock.java`, which contains the following Java code:

```
package com.example.cw.Stock;

public class tbStock {
    private String Supplier;
    private String Updated;
    private String ProdName;
    private String Qty;
    private String UnitPrice;
    private String TotalPrice;

    public String getSupplier() {
        return Supplier;
    }

    public void setSupplier(String supplier) {
        Supplier = supplier;
    }

    public String getUpdated() {
        return Updated;
    }

    public void setUpdated(String updated) {
        Updated = updated;
    }

    public String getProdName() {
        return ProdName;
    }

    public void setProdName(String prodName) {
        ProdName = prodName;
    }
}
```

The project structure on the left shows packages like `com.example.cw` containing `Categories`, `Product`, and `Stock`. The `Stock` package contains `Stock`, `StockEdit2`, and `tbStock`.

The screenshot shows the IntelliJ IDEA interface with the project navigation bar at the top. The current file is `User.java`, which contains the following Java code:

```
import java.util.ArrayList;
import java.util.Objects;

public class User {
    @FXML
    public TextField NUsername;
    public TextField NPassword;
    public TextField CPassword;

    @FXML //labels
    public Label PasswordID1;
    public Label IncUserName;
    public Label PasswordID2;

    DBConnection dbConnection = DBConnection.getConnection();
    MongoCollection<Document> collection = dbConnection.getMongoCollection("Users");

    ListDoc add = new ListDoc();

    ArrayList<String> UsernameList = new ArrayList<>();
    ArrayList<String> PasswordList = new ArrayList<>();

    public void Back(ActionEvent event) throws IOException {
        FXMLLoader fxmlLoader = new FXMLLoader(User.class.getResource( name: "home.fxml" ));
        window.newWindow(fxmlLoader);

        window.closeWindow(event);
    }

    public void EnterUser() {
        label( Username: "", PassField1: "", PassField2: "" );
        String NewUser = NUsername.getText();
    }
}
```

The project structure on the left shows packages like `com.example.cw` containing `Categories`, `Product`, and `Stock`. The `Stock` package contains `Stock`, `StockEdit2`, and `tbStock`.

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CW [C:\Users\Lucifer Diotics\Desktop\CW] - User.java

Project: CW C:\Users\Lucifer Diotics\Desktop\CW

src

main

java

com.example.cw

Categories

CategoriesDelete

CategoriesEdit

CategoriesNew

tblCategory

Product

ProductDelete

ProductEdit

ProductNew

tblPdt

Stock

StockEdit2

tblStock

Home

LoginController

Open

User

functions

DBConnection

EditValue

Error

Find

ListDoc

Update

window

module-info.java

OpenJava x User.java

```
label(Username: "", Passfield1: "", Passfield2: "");  
String NewUser = Username.getText();  
String NewPass = NPassword.getText();  
String ConPass = CNPassword.getText();  
  
BasicDBObject doc = new BasicDBObject("Name", NewUser.toUpperCase());  
find(UserNameList, PasswordList, doc, collection);  
if (!UserNameList.contains(NewUser.toUpperCase())){  
    if ((NewUser.toUpperCase()).contains(NewPass.toUpperCase()))  
        if ((ConPass.toUpperCase()).contains(NewUser.toUpperCase())){  
            if (NewPass.equals("") || ConPass.equals("")){  
                Error.error("Password Field is empty");  
            } else {  
                label(Username: "", Passfield1: "This can be easily guessed", Passfield2: "");  
            }  
        } else if (!Objects.equals(ConPass, NewPass)){  
            label(Username: "", Passfield1: "", Passfield2: "Don't match Passwords");  
            text(NewUser, Passfield1: "", Passfield2: "");  
        } else if (NewUser.length()<=5 || NewPass.length()<=5){  
            if (NewUser.length()<=5){  
                label(Username: "Username must be at least 6 letters", Passfield1: "", Passfield2: "");  
                text(Username: "", NewPass, ConPass);  
            } else {  
                label(Username: "Weak Password", Passfield1: "", Passfield2: "");  
            }  
        } else {  
            //connect to mongoDB  
            add.AddOrRemoveCollection(NewUser.toUpperCase(), ConPass, collection, Display: "User Added Successfully", call: 1);  
        }  
    }  
}  
  
label(Username: "username already EXISTS", Passfield1: "", Passfield2: "");  
text(Username: "", Passfield1: "", Passfield2: "");  
  
public void delete() {  
    String NewUser = Username.getText();  
    String NewPass = NPassword.getText();  
    String ConPass = CNPassword.getText();  
    label(Username: "", Passfield1: "", Passfield2: "");  
  
    BasicDBObject doc = new BasicDBObject("Name", NewUser.toUpperCase());  
    find(UserNameList, PasswordList, doc, collection);  
  
    if (UserNameList.contains(NewUser.toUpperCase())){  
        if (Objects.equals(NewPass, ConPass)){  
            if (PasswordList.contains(ConPass)){  
                add.AddOrRemoveCollection(NewUser.toUpperCase(), ConPass, collection, Display: "User's password DELETED successfully");  
                text(Username: "", Passfield1: "", Passfield2: "");  
            } else {  
                Error.error("password not equal to user's password");  
                text(NewUser, Passfield1: "", Passfield2: "");  
            }  
        } else {  
            label(Username: "", Passfield1: "", Passfield2: "Don't match Passwords");  
            text(NewUser, Passfield1: "", Passfield2: "");  
        } else {  
            Error.error("Username can't find");  
            text(Username: "", Passfield1: "", Passfield2: "");  
        }  
    }  
}
```

Build completed successfully in 8 sec, 595 ms (23 minutes ago)

12657 CRLF UTF-8 4 spaces ENG 103 AM 12/17/2021

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CW [C:\Users\Lucifer Diotics\Desktop\CW] - User.java

Project: CW C:\Users\Lucifer Diotics\Desktop\CW

src

main

java

com.example.cw

Categories

CategoriesDelete

CategoriesEdit

CategoriesNew

tblCategory

Product

ProductDelete

ProductEdit

ProductNew

tblPdt

Stock

StockEdit2

tblStock

Home

LoginController

Open

User

functions

DBConnection

EditValue

Error

Find

ListDoc

Update

window

module-info.java

OpenJava x User.java

```
label(Username: "username already EXISTS", Passfield1: "", Passfield2: "");  
text(Username: "", Passfield1: "", Passfield2: "");  
  
public void delete() {  
    String NewUser = Username.getText();  
    String NewPass = NPassword.getText();  
    String ConPass = CNPassword.getText();  
    label(Username: "", Passfield1: "", Passfield2: "");  
  
    BasicDBObject doc = new BasicDBObject("Name", NewUser.toUpperCase());  
    find(UserNameList, PasswordList, doc, collection);  
  
    if (UserNameList.contains(NewUser.toUpperCase())){  
        if (Objects.equals(NewPass, ConPass)){  
            if (PasswordList.contains(ConPass)){  
                add.AddOrRemoveCollection(NewUser.toUpperCase(), ConPass, collection, Display: "User's password DELETED successfully");  
                text(Username: "", Passfield1: "", Passfield2: "");  
            } else {  
                Error.error("password not equal to user's password");  
                text(NewUser, Passfield1: "", Passfield2: "");  
            }  
        } else {  
            label(Username: "", Passfield1: "", Passfield2: "Don't match Passwords");  
            text(NewUser, Passfield1: "", Passfield2: "");  
        } else {  
            Error.error("Username can't find");  
            text(Username: "", Passfield1: "", Passfield2: "");  
        }  
    }  
}
```

Build completed successfully in 8 sec, 595 ms (23 minutes ago)

12657 CRLF UTF-8 4 spaces ENG 103 AM 12/17/2021

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CW [C:\Users\Lucifer Dots\Desktop\CW] - User.java

CW | src | main | java | com | example | cw | User | find

Project | OpenJava | User.java | Maven | Diff

```
116     }
117
118     // functions
119     @
120     public void find(ArrayList<String> userNameList,
121                     ArrayList<String> passwordList,
122                     BasicDBObject doc,
123                     MongoCollection<Document> collection) {
124         FindIterable<Document> findIterable = collection.find();
125         FindIterable<Document> findIterable = collection.find();
126         for (Document doc : findIterable) {
127             userNameList.add((String) doc.get("Name"));
128
129             for (Document document : findIterable) {
130                 passwordList.add((String) document.get("Password"));
131             }
132         }
133     }
134
135     public void text(String Username, String PassField1, String PassField2) {
136         NUsername.setText(Username);
137         NPassword.setText(PassField1);
138         CNPassword.setText(PassField2);
139     }
140
141     public void label(String Username, String PassField1, String PassField2) {
142         IncUserName.setText(Username);
143         PasswordID1.setText(PassField1);
144         PasswordID2.setText(PassField2);
145     }
146 }
```

Version Control | Run | TODO | Problems | Profiler | Terminal | Build | Dependencies | Event Log

Build completed successfully in 8 sec, 595 ms (23 minutes ago)

126:57 CRLF UTF-8 4 spaces 104 AM ENG 12/17/2021

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CW [C:\Users\Lucifer Dots\Desktop\CW] - window.java

CW | src | main | java | functions | window

Project | ListDoc.java | window.java | Maven | Diff

```
10     import java.io.IOException;
11
12     public class window {
13         @
14         public static void newWindow(FXMLLoader fxmlLoader) throws IOException {
15             Stage addContactStage = new Stage();
16             Scene scene = new Scene(fxmlLoader.load());
17             addContactStage.setScene(scene);
18             addContactStage.resizableProperty().setValue(false);
19             addContactStage.initStyle(StageStyle.UNDECORATED);
20             addContactStage.show();
21         }
22         @
23         public static void closeWindow(ActionEvent event){
24             Stage firstStage = (Stage) ((Button) event.getSource()).getScene().getWindow();
25             firstStage.close();
26         }
27 }
```

Problems | Current File | Project Errors

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CW [C:\Users\Lucifer Diots\Desktop\CW] - Categories.java
Project src main java com example cw Categories.java
18 import javafx.scene.control.cell.PropertyValueFactory;
19 import org.bson.Document;
20
21 import java.io.IOException;
22 import java.net.URL;
23 import java.util.ResourceBundle;
24
25 public class Categories implements Initializable {
26     @FXML
27     public TableView<tblCtgry> table;
28
29     @FXML
30     public TableColumn<tblCtgry, String> clmid;
31
32     @FXML
33     public TableColumn<tblCtgry, String> clmname;
34
35     @FXML
36     public TableColumn<tblCtgry, String> clmDes;
37
38     CategoriesDelete categoriesDelete = new CategoriesDelete();
39     DBConnection dbConnection = DBConnection.getConnection();
40     MongoCollection<Document> CategoriesCollec = dbConnection.getMongoCollection("Categories");
41
42     ObservableList<tblCtgry> iterate() {
43
44         ObservableList<tblCtgry> List = FXCollections.observableArrayList();
45         FindIterable<Document> FindIterable = CategoriesCollec.find();
46
47         for (Document DOC : FindIterable) {
48             tblCtgry tblCtgry = new tblCtgry();
49             tblCtgry.setClmid(DOC.getString(key: "ID"));
50             tblCtgry.setClmname(DOC.getString(key: "CategoryName"));
51             if (!DOC.getString(key: "Description").equals("")){
52                 List.add(tblCtgry);
53             }
54         }
55         return List;
56     }
57
58     @Override
59     public void initialize(URL location, ResourceBundle resources) {
60         iterate();
61         clmid.setCellValueFactory(new PropertyValueFactory<>("clmid"));
62         clmname.setCellValueFactory(new PropertyValueFactory<>("clmname"));
63         clmDes.setCellValueFactory(new PropertyValueFactory<>("clmDes"));
64
65         table.setItems(iterate());
66
67         categoriesDelete.IDset(CategoriesCollec);
68     }
69
70     public void logout(ActionEvent event) throws IOException {
71
72         FXMLLoader fxmlLoader = new FXMLLoader(Home.class.getResource(name: "LoginPage.fxml"));
73         window.newWindow(fxmlLoader);
74         window.closeWindow(event);
75     }
76
77     public void home(ActionEvent event) throws IOException {
78
79         window.closeWindow(event);
80         FXMLLoader fxmlLoader = new FXMLLoader(Home.class.getResource(name: "home.fxml"));
81         window.newWindow(fxmlLoader);
82     }
83
84     public void NewCategorie(ActionEvent event) throws IOException {
85
86         FXMLLoader fxmlLoader = new FXMLLoader(User.class.getResource(name: "CategoriesNew.fxml"));
87         window.newWindow(fxmlLoader);
88         window.closeWindow(event);
89     }
90 }
```

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CW [C:\Users\Lucifer Diots\Desktop\CW] - Categories.java
Project src main java com example cw Categories.java
57
58     @Override
59     public void initialize(URL location, ResourceBundle resources) {
60         iterate();
61         clmid.setCellValueFactory(new PropertyValueFactory<>("clmid"));
62         clmname.setCellValueFactory(new PropertyValueFactory<>("clmname"));
63         clmDes.setCellValueFactory(new PropertyValueFactory<>("clmDes"));
64
65         table.setItems(iterate());
66
67         categoriesDelete.IDset(CategoriesCollec);
68     }
69
70     public void logout(ActionEvent event) throws IOException {
71
72         FXMLLoader fxmlLoader = new FXMLLoader(Home.class.getResource(name: "LoginPage.fxml"));
73         window.newWindow(fxmlLoader);
74         window.closeWindow(event);
75     }
76
77     public void home(ActionEvent event) throws IOException {
78
79         window.closeWindow(event);
80         FXMLLoader fxmlLoader = new FXMLLoader(Home.class.getResource(name: "home.fxml"));
81         window.newWindow(fxmlLoader);
82     }
83
84     public void NewCategorie(ActionEvent event) throws IOException {
85
86         FXMLLoader fxmlLoader = new FXMLLoader(User.class.getResource(name: "CategoriesNew.fxml"));
87         window.newWindow(fxmlLoader);
88         window.closeWindow(event);
89     }
90 }
```

The screenshot shows the IntelliJ IDEA interface with the project navigation bar at the top. The current file is `CategoriesDelete.java`. The code implements a class `CategoriesDelete` that interacts with MongoDB collections for `Categories` and `Product`. It includes FXML annotations for `DCategory` and `CategoryID`, and handles a `Back` action event. The code uses `ArrayList` to find categories by name and `EditValue` to update the `CategoryName` field in the `Product` collection.

```
public class CategoriesDelete implements Initializable {
    DBConnection dbConnection = DBConnection.getConnection();
    MongoCollection<Document> CategoriesCollec = dbConnection.getMongoCollection("Categories");

    MongoCollection<Document> Product = dbConnection.getMongoCollection("Product");

    @FXML
    public TextField DCategories;

    @FXML
    public TextField CategoryID;

    public void Back(ActionEvent event) throws IOException {
        FXMLLoader fxmlLoader = new FXMLLoader(Home.class.getResource("Categories.fxml"));
        window=newWindow(fxmlLoader);
        IDset(CategoriesCollec);

        window.closeWindow(event);
    }

    public void DeleteCategory() {
        String DltCategory = DCategories.getText();
        String ID = CategoryID.getText();

        ArrayList<String> CategoryList = Find.find(CategoriesCollec,DltCategory, findBy: "CategoryName", findField: "ID");
        ArrayList<String> CategoryList2 = Find.findAll( findField: "CategoryName", CategoriesCollec);

        EditValue.Edit(DltCategory.toUpperCase(),Product, ChangeField: "CategoryName", to: "NONE");
    }
}
```

This screenshot shows the continuation of the `CategoriesDelete.java` code. It includes logic to handle category deletion based on the ID provided. If the category exists, it is deleted from the database, and a success message is displayed. If the category does not exist, an error message is shown. If any fields are empty, an error message is also displayed. The code then proceeds to update the `Product` collection using `IDset` and `BasicDBObject`.

```
if (CategoryList.contains(ID)){
    Document document = new Document();
    document.put("CategoryName", DCategories.getText().toUpperCase());
    document.put("ID", CategoryID.getText());
    CategoriesCollec.deleteOne(document);
    Error.statics("Category Deleted");
    DCategories.setText("");
    CategoryID.setText("");
    IDset(CategoriesCollec);
}

else {
    Error.error("Category not Exists");
}

else {
    Error.error("Some fields are empty");
}

public void IDset(MongoCollection<Document> collection) {
    ArrayList<String> count=Find.findAll( findField: "ID",collection);
    for (int i=0;i<count.size();i++) {
        BasicDBObject basicDBObject = new BasicDBObject("ID",count.get(i));
        BasicDBObject basicDBObject2 = new BasicDBObject();
        basicDBObject2.put("ID",String.valueOf(i+1));
        BasicDBObject updateDBObject = new BasicDBObject();
        updateDBObject.put("$set", basicDBObject2);
        CategoriesCollec.updateOne(basicDBObject, updateDBObject);
    }
}
```

The screenshot shows the IntelliJ IDEA interface with the file `CategoriesEdit.java` open. The code implements a class `CategoriesEdit` with a method `EditCategory()`. The method performs a database update operation using MongoDB. It finds all documents where the `CategoryName` field matches the search term. For each found document, it creates a copy with the `ID` field set to the value from the `EditID` text field. It then updates the original document with the new values from the `BasicDBObject2` copy. Finally, it saves the changes and sets the `Category`, `Description`, and `EditID` fields to empty strings.

```
public class CategoriesEdit {
    public TextField description;
    public TextField NCcategory;
    public TextField EditID;

    DBConnection dbConnection = DBConnection.getConnection();
    MongoCollection<Document> Ctgry = dbConnection.getMongoCollection("Categories");

    public void EditCategory() {

        ArrayList<String> Name = Find.findall( findField: "CategoryName", Ctgry);
        if (!Name.contains(NCategory.getText().toUpperCase())){

            BasicDBObject doc = new BasicDBObject("ID", EditID.getText());
            BasicDBObject basicDBObject2 = new BasicDBObject();
            if (!Objects.equals(NCategory.getText(), "")) {
                basicDBObject2.put("CategoryName", NCategory.getText().toUpperCase());
            } if (!Objects.equals(description.getText(), "")) {
                basicDBObject2.put("Description", description.getText());
            }
            BasicDBObject updateObject = new BasicDBObject();
            updateObject.put("$set", basicDBObject2);
            Ctgry.updateOne(doc, updateObject);
            Error.statics("Category Edited");
            description.setText("");
            NCategory.setText("");
            EditID.setText("");
        } else {
            Error.error("This Category name already exists");
        }
    }
}
```

The screenshot shows the IntelliJ IDEA interface with the file `CategoriesEdit.java` open. The code implements a method `Back(ActionEvent event)` which handles the back button press. It creates a new window for the `Categories.fxml` FXML file and closes the current window. It also implements a `Set()` method which performs a search operation. It finds all documents where the `ID` field matches the value in the `EditID` text field. It then adds these documents to a list. If no documents are found, it adds all documents where the `CategoryName` field matches the search term. For each found document, it creates a copy with the `ID` field set to the value from the `EditID` text field. It then updates the original document with the new values from the `BasicDBObject2` copy. Finally, it saves the changes and sets the `Category`, `Description`, and `EditID` fields to empty strings.

```
public void Back(ActionEvent event) throws IOException {
    FXMLLoader fxmlLoader = new FXMLLoader(Home.class.getResource( name: "Categories.fxml"));
    window.newWindow(fxmlLoader);
    window.closeWindow(event);
}

public void Set() {
    ArrayList<String> num = Find.findall( findField: "ID", Ctgry);
    ArrayList<String> name = Find.findall( findField: "CategoryName", Ctgry);
    if (num.contains(EditID.getText())){
        ArrayList<String> set = new ArrayList<>();
        set.add((Find.find(Ctgry, EditID.getText(), findby: "ID", findField: "CategoryName")).get(0));
        set.add((Find.find(Ctgry, EditID.getText(), findby: "ID", findField: "Description")).get(0));
        NCategory.setText(set.get());
        description.setText(set.get());
    } else if (name.contains(EditID.getText().toUpperCase())){
        ArrayList<String> set = new ArrayList<>();
        set.add((Find.find(Ctgry, EditID.getText(), findby: "CategoryName", findField: "CategoryName")).get(0));
        set.add((Find.find(Ctgry, EditID.getText(), findby: "CategoryName", findField: "Description")).get(0));
        NCategory.setText(set.get());
        description.setText(set.get());
    } else {
        Error.error("Search field is empty");
    }
}
```

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CW [C:\Users\Lucifer Diots\Desktop\CW] - CategoriesNew.java

File src main java com example cw Categories CategoriesNew Back

Project CW C:\Users\Lucifer Diots\Desktop\CW .idea src main java com.example.cw Categories CategoriesDelete CategoriesEdit CategoriesNew CategoriesNew.java CategoriesDelete.java CategoriesEdit.java CategoriesNew.java Stock StockEdit2 StockStock Home LoginController Open User functions DBCollection EditValue Error Find ListDoc Update window module-info.java

```
import java.util.Objects;
```

```
public class CategoriesNew {
```

```
    @FXML
```

```
    public TextField NCategory;
```

```
    public TextField description;
```

```
    DBConnection dbConnection = DBConnection.getConnection();
```

```
    public void EnterCategory() {
```

```
        String NewCategory = NCategory.getText();
```

```
        String Description = description.getText();
```

```
        MongoCollection<Document> Categories = dbConnection.getMongoCollection("Categories");
```

```
        ArrayList<String> CategoryList = Find.findall("CategoryName", Categories);
```

```
        if (!CategoryList.contains(NewCategory.toUpperCase())){
```

```
            if (NewCategory.equals("") || Objects.equals(Description, "")) {
```

```
                Error.error("Some fields are empty");
```

```
            }
```

```
            else if (NewCategory.length()<3){
```

```
                Error.error("Category Name must be at least 4 letters");
```

```
            }
```

```
            else {
```

```
                Document document = new Document();
```

```
                document.put("CategoryName", NCategory.getText().toUpperCase());
```

```
                document.put("Description", description.getText());
```

```
                document.put("ID", String.valueOf(CategoryList.size()+1));
```

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CW [C:\Users\Lucifer Diots\Desktop\CW] - tbICtgy.java

File src main java com example cw Categories CategoriesNew CategoriesNew.java CategoriesDelete CategoriesEdit CategoriesNew CategoriesNew.java CategoriesDelete.java CategoriesEdit.java CategoriesNew.java Stock StockEdit2 StockStock Home LoginController Open User functions DBCollection EditValue Error Find ListDoc Update window module-info.java

```
package com.example.cw.Categories;
```

```
public class tbICtgy {
```

```
    private String clmid;
```

```
    private String clmname;
```

```
    private String clmDes;
```

```
    public String getClmid() {
```

```
        return clmid;
```

```
    }
```

```
    public void setClmid(String clmid) {
```

```
        this.clmid = clmid;
```

```
    }
```

```
    public String getClmname() {
```

```
        return clmname;
```

```
    }
```

```
    public void setClmname(String clmname) {
```

```
        this.clmname = clmname;
```

```
    }
```

```
    public String getClmDes() {
```

```
        return clmDes;
```

```
    }
```

```
    public void setClmDes(String clmDes) {
```

```
        this.clmDes = clmDes;
```

```
    }
```

The image shows two side-by-side Java code editors within an IDE interface. Both editors have a dark theme and display code in white text.

**Editor 1 (Left):**

- Title:** CW [C:\Users\Lucifer Diots\Desktop\CW] - EditValue.java
- Content:** A Java class named `EditValue` with a static method `edit`. The method takes parameters: `MongoCollection<Document> collection`, `String ChangeField`, and `String to`. It uses `BasicDBObject` to build update operations and performs a `updateOne` operation on the collection.
- Project Structure:** The project structure shows packages `com.example.cw` and `com.mongodb`.

**Editor 2 (Right):**

- Title:** CW [C:\Users\Lucifer Diots\Desktop\CW] - Error.java
- Content:** A Java class named `Error` with static methods `error`, `statics`, and `warning`. Each method creates an `Alert` object with specific titles and content based on the input string.
- Project Structure:** The project structure shows packages `com.example.cw` and `javafx.scene.control`.

Both editors include standard IDE features like toolbars, status bars, and search bars at the bottom.

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CW [C:\Users\Lucifer Dots\Desktop\CW] - Find.java

Project: CW C:\Users\Lucifer Dots\Desktop\CW .idea src main java com.example.cw Categories Product Stock Home functions module-info.java

```
import org.bson.Document;
import java.util.ArrayList;

public class Find {
    public static ArrayList<String> Find (MongoCollection<Document> Collection, String gettest, String findBy, String findField) {
        ArrayList<String> CategoryList = new ArrayList<>();

        BasicDBObject doc = new BasicDBObject(findBy, gettest.toUpperCase());
        FindIterable<Document> FindIterable = Collection.find(doc);
        for (Document DOC : FindIterable) {
            CategoryList.add((String) DOC.get(findField));
        }
        return CategoryList;
    }

    public static ArrayList<String> findAll(String findField,MongoCollection<Document> Categories) {
        ArrayList<String> CategoryList2 = new ArrayList<>();

        FindIterable<Document> findIterable = Categories.find();
        for (Document DOC : findIterable) {
            CategoryList2.add((String) DOC.get(findField));
        }
        return CategoryList2;
    }
}
```

Version Control Run TODO Problems Profiler Terminal Build Dependencies

Build completed successfully in 8 sec, 595 ms (2 minutes ago)

12:37 CRLF UTF-8 4 spaces ENG 105 AM 12/17/2021

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CW [C:\Users\Lucifer Dots\Desktop\CW] - Home.java

Project: CW C:\Users\Lucifer Dots\Desktop\CW .idea src main java com.example.cw Categories Product Stock Home LoginController Open User functions module-info.java

```
public class Home {
    @FXML

    public void log_out(ActionEvent event) throws IOException {
        FXMLLoader fxmlLoader = new FXMLLoader(Home.class.getResource( name: "LoginPage.fxml"));
        window.newWindow(fxmlLoader);
        window.closeWindow(event);
    }

    public void stock(ActionEvent event) throws IOException {
        FXMLLoader fxmlLoader = new FXMLLoader(Home.class.getResource( name: "Stock.fxml"));
        window.newWindow(fxmlLoader);

        window.closeWindow(event);
    }

    public void category(ActionEvent event) throws IOException {
        FXMLLoader fxmlLoader = new FXMLLoader(Home.class.getResource( name: "Categories.fxml"));
        window.newWindow(fxmlLoader);

        window.closeWindow(event);
    }

    public void product(ActionEvent event) throws IOException {
        FXMLLoader fxmlLoader = new FXMLLoader(Home.class.getResource( name: "Product.fxml"));
        window.newWindow(fxmlLoader);

        window.closeWindow(event);
    }
}
```

Version Control Run TODO Problems Profiler Terminal Build Dependencies

Build completed successfully in 8 sec, 595 ms (2 minutes ago)

3548 (4 chars) CRLF UTF-8 4 spaces ENG 101 AM 12/17/2021

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CW [C:\Users\Lucifer Dots\Desktop\CW] - ListDoc.java

Project ▾

- Product
  - Product
  - ProductDelete
  - ProductEdit
  - ProductNew
  - tblPdt
- Stock
  - Stock
  - StockEdit2
  - tblStock
- Home
- LoginController
- Open
- User

functions

- DBConnection
- EditValue
- Error
- Find
- ListDoc
- Update
- window

resources

- com.example.cw
  - images
  - new
    - Categories.xml
    - CategoriesDelete.fxml
    - CategoriesEdit.fxml
    - CategoriesNew.fxml
    - home.fxml
    - LoginPage.fxml
    - Product.fxml
    - ProductDelete.fxml

module-info.java

Build completed successfully in 8 sec, 595 ms (25 minutes ago)

21:10 CRLF UTF-8 4 spaces ENG 105 AM 12/17/2021

```
public void AddOrRemoveCollection(String Name, String Password, MongoCollection<Document> collection, String Display, int call) {
    Error.statics(Display);
    Document NEWUser = new Document();
    NEWUser.put("Name", Name);
    NEWUser.put("Password", Password);
    System.out.println(NEWUser);
    if (call == 1) {
        collection.insertOne(NEWUser);
    } else if (call == 2) {
        collection.deleteOne(NEWUser);
    }
}

public void AddOrRemoveCategory(String Category, String Description, int ball, String ID, MongoCollection<Document> collection, String Display, int call) {
    Error.statics(Display);
    Document NEWUser = new Document();
    NEWUser.put("CategoryName", Category);
    if (ball == 1) {
        NEWUser.put("Description", Description);
    } else {
        NEWUser.put("ID", ID);
        System.out.println(NEWUser);
        if (call == 1) {
            collection.insertOne(NEWUser);
        } else if (call == 2) {
            collection.deleteOne(NEWUser);
        }
}
```

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CW [C:\Users\Lucifer Dots\Desktop\CW] - LoginController.java

Project ▾

- CVW [C:\Users\Lucifer Dots\Desktop\CW]
- idea
- src
  - main
    - java
      - com.example.cw
        - Categories
        - CategoriesDelete
        - CategoriesEdit
        - CategoriesNew
        - tblCategory
      - Product
        - Product
        - ProductDelete
        - ProductEdit
        - ProductNew
        - tblPdt
      - Stock
        - Stock
        - StockEdit2
        - tblStock
      - Home
      - LoginController
      - Open
      - User
    - functions
      - DBConnection
      - EditValue
      - Error
      - Find
      - ListDoc
      - Update
      - window
  - module-info.java

Build completed successfully in 8 sec, 595 ms (21 minutes ago)

559 LF UTF-8 4 spaces ENG 130 AM 12/17/2021

```
public class LoginController {
    @FXML
    public PasswordField password;

    @FXML
    public TextField username;

    @FXML
    public Button close_id;

    @FXML
    public Label wrong;
    public String user;
    public String passw;
    DBConnection dbConnection = DBConnection.getConnection();
    MongoCollection<Document> collection = dbConnection.getMongoCollection("Users");

    public void close(ActionEvent event) {
        // get a handle to the stage
        Stage stage = (Stage) ((Button) event.getSource()).getScene().getWindow();
        stage.close();
    }

    public void Singin(ActionEvent event) throws IOException {
        user = username.getText();
        passw = password.getText();
        ArrayList<String> UserNameList = new ArrayList<>();
        ArrayList<String> PasswordList = new ArrayList<>();
        BasicDBObject doc = new BasicDBObject("Name", user.toUpperCase());
```

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CW [C:\Users\Lucifer Dots\Desktop\CW] - LoginController.java

CW | src | main | java | com | example | cw | LoginController | Singin

Project | LoginController.java | X

```
02 package com.example.cw;
03
04 import com.mongodb.BasicDBObject;
05 import com.mongodb.DBObject;
06 import com.mongodb.client.FindIterable;
07 import com.mongodb.client.MongoCollection;
08 import com.mongodb.client.MongoDatabase;
09 import com.mongodb.client.model.Filters;
10 import com.mongodb.client.model.Updates;
11 import com.mongodb.client.result.DeleteResult;
12 import com.mongodb.client.result.UpdateResult;
13 import com.mongodb.client.result.InsertOneResult;
14 import com.mongodb.client.result.InsertManyResult;
15 import com.mongodb.client.result.DeleteResult;
16 import com.mongodb.client.result.UpdateResult;
17 import com.mongodb.client.result.InsertOneResult;
18 import com.mongodb.client.result.InsertManyResult;
19 import com.mongodb.client.result.DeleteResult;
20 import com.mongodb.client.result.UpdateResult;
21 import com.mongodb.client.result.InsertOneResult;
22 import com.mongodb.client.result.InsertManyResult;
23 import com.mongodb.client.result.DeleteResult;
24 import com.mongodb.client.result.UpdateResult;
25 import com.mongodb.client.result.InsertOneResult;
26 import com.mongodb.client.result.InsertManyResult;
27 import com.mongodb.client.result.DeleteResult;
28 import com.mongodb.client.result.UpdateResult;
29 import com.mongodb.client.result.InsertOneResult;
30 import com.mongodb.client.result.InsertManyResult;
31 import com.mongodb.client.result.DeleteResult;
32 import com.mongodb.client.result.UpdateResult;
33 import com.mongodb.client.result.InsertOneResult;
34 import com.mongodb.client.result.InsertManyResult;
35 import com.mongodb.client.result.DeleteResult;
36 import com.mongodb.client.result.UpdateResult;
37 import com.mongodb.client.result.InsertOneResult;
38 import com.mongodb.client.result.InsertManyResult;
39 import com.mongodb.client.result.DeleteResult;
40 import com.mongodb.client.result.UpdateResult;
41 import com.mongodb.client.result.InsertOneResult;
42 import com.mongodb.client.result.InsertManyResult;
43 import com.mongodb.client.result.DeleteResult;
44 import com.mongodb.client.result.UpdateResult;
45 import com.mongodb.client.result.InsertOneResult;
46 import com.mongodb.client.result.InsertManyResult;
47 import com.mongodb.client.result.DeleteResult;
48 import com.mongodb.client.result.UpdateResult;
49 import com.mongodb.client.result.InsertOneResult;
50 import com.mongodb.client.result.InsertManyResult;
51 import com.mongodb.client.result.DeleteResult;
52 import com.mongodb.client.result.UpdateResult;
53 import com.mongodb.client.result.InsertOneResult;
54 import com.mongodb.client.result.InsertManyResult;
55 import com.mongodb.client.result.DeleteResult;
56 import com.mongodb.client.result.UpdateResult;
57 import com.mongodb.client.result.InsertOneResult;
58 import com.mongodb.client.result.InsertManyResult;
59 import com.mongodb.client.result.DeleteResult;
60 import com.mongodb.client.result.UpdateResult;
61 import com.mongodb.client.result.InsertOneResult;
62 import com.mongodb.client.result.InsertManyResult;
63 import com.mongodb.client.result.DeleteResult;
64 import com.mongodb.client.result.UpdateResult;
65 import com.mongodb.client.result.InsertOneResult;
66 import com.mongodb.client.result.InsertManyResult;
67 import com.mongodb.client.result.DeleteResult;
68 import com.mongodb.client.result.UpdateResult;
69 import com.mongodb.client.result.InsertOneResult;
70 import com.mongodb.client.result.InsertManyResult;
71 import com.mongodb.client.result.DeleteResult;
72 import com.mongodb.client.result.UpdateResult;
73 import com.mongodb.client.result.InsertOneResult;
74 import com.mongodb.client.result.InsertManyResult;
75 import com.mongodb.client.result.DeleteResult;
76 import com.mongodb.client.result.UpdateResult;
77 import com.mongodb.client.result.InsertOneResult;
78 import com.mongodb.client.result.InsertManyResult;
```

Build completed successfully in 8 sec, 595 ms (22 minutes ago)

Type here to search

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CW [C:\Users\Lucifer Dots\Desktop\CW] - DBConnection.java

CW | src | main | java | functions | DBConnection

Project | User.java | DBConnection.java | X

```
7 package com.example.cw;
8
9 public class DBConnection {
10     private final MongoDatabase mongoDatabase;
11     private MongoCollection<Document> mongoCollection;
12
13     //static variables
14     private static DBConnection dbConnection;
15
16     //constance
17     private static final String HOSTNAME = "Localhost";
18     private static final int PORT = 27017;
19     private static final String DB_NAME = "DBMS";
20
21     private DBConnection() {
22         //instance variables
23         MongoClient mongoClient = new MongoClient(HOSTNAME, PORT);
24         mongoDatabase = mongoClient.getDatabase(DB_NAME);
25     }
26
27     public static DBConnection getConnection(){
28         if(dbConnection == null){
29             dbConnection =new DBConnection();
30         }
31
32         return dbConnection;
33     }
34
35     public MongoDatabase getMongoDatabase() { return mongoDatabase; }
36
37     public MongoCollection<Document> getMongoCollection(String collection) {
38         mongoCollection = mongoDatabase.getCollection(collection);
39         return mongoCollection;
40     }
41
42     public void closeConnection() {
43         mongoDatabase.close();
44     }
45 }
```

Build completed successfully in 8 sec, 595 ms (22 minutes ago)

Type here to search

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CW [C:\Users\Lucifer Diotics\Desktop\CW] - Open Java

CW src main java com example cw Open start

Project Structure Bookmarks

OpenJava.java

```
1 package com.example.cw;
2 import javafx.application.Application;
3 import javafx.fxml.FXMLLoader;
4 import javafx.scene.Scene;
5 import javafx.stage.Stage;
6 import javafx.stage.StageStyle;
7 import java.io.IOException;
8
9 public class Open extends Application {
10     @Override
11     public void start(Stage stage) throws IOException {
12         FXMLLoader fxmlLoader = new FXMLLoader(Open.class.getResource( name: "LoginPage.fxml"));
13         Scene scene = new Scene(fxmlloader.load());
14         stage.setScene(scene);
15         stage.resizableProperty().setValue(false);
16         stage.initStyle(StageStyle.UNDECORATED);
17         stage.show();
18     }
19
20     public static void main(String[] args) { launch(); }
21 }
22
23
24
```

Version Control Run TODO Problems Profiler Terminal Build Dependencies

Build completed successfully in 8 sec, 595 ms (22 minutes ago)

Event Log 12:51 LF UTF-8 4 spaces ENG 12/17/2021

Type here to search

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CW [C:\Users\Lucifer Diotics\Desktop\CW] - Product.java

CW src main java com example cw Product Product EditProduct

Project Structure Bookmarks

Product.java

```
30     @FXML
31     public TableView<tblPrdkt> table;
32
33     @FXML
34     public TableColumn<tblPrdkt, String> clmid;
35
36     @FXML
37     public TableColumn<tblPrdkt, String> qty;
38
39     @FXML
40     public TableColumn<tblPrdkt, String> clmname;
41
42     @FXML
43     public TableColumn<tblPrdkt, String> clmCtgy;
44
45     @FXML
46     public TableColumn<tblPrdkt, String> clmDes;
47
48     public TextField searchID;
49
50
51     ObservableList<tblPrdkt> All = FXCollections.observableArrayList();
52     DBConnection dbConnection = DBConnection.getConnection();
53     MongoCollection<Document> product = dbConnection.getMongoCollection("Product");
54
55     public ObservableList<tblPrdkt> iterate() {
56
57         ObservableList<tblPrdkt> List = FXCollections.observableArrayList();
58         FindIterable<Document> Find = product.find();
59
60         for (Document DOC : Find) {
61             tblPrdkt tblPrdkt1 = new tblPrdkt(DOC.getString( key: "ID"),DOC.getString( key: "Quantity"),DOC.getString( key: "Prod
62             All.add(tblPrdkt1);
63             tblPrdkt1.setClmid(DOC.getString( key: "ID"));
64             tblPrdkt1.setClmname(DOC.getString( key: "ProductName"));
65             tblPrdkt1.setClmDes(DOC.getString( key: "description"));
66             +tblPrdkt1.setClmCtgy(DOC.getString( key: "CategoryName"));
67         }
68     }
69 }
```

Version Control Run TODO Problems Profiler Terminal Build Dependencies

Build completed successfully in 8 sec, 595 ms (11 minutes ago)

Event Log 107:1 CRLF UTF-8 4 spaces ENG 12/17/2021

Type here to search