Database Connection

```
public class CustomDbConnection {
   public static Connection getConnection() {
      Connection conn = null;
      try {
   conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/dhaka1", "root",
"1234");
         System.out.println("Connected database dhaka1");
      } catch (SQLException ex) {
         Logger.getLogger(CustomDbConnection.class.getName()).log(Level.SEVERE, null, ex);
      }
      return conn;
   }
   public static void main(String[] args) {
      Connection conn = CustomDbConnection.getConnection();
   }
}
```

```
public class ProductDaoImpl implements ProductDao {
   Connection conn = CustomDbConnection.getConnection();

   public static void main(String[] args) {
      ProductDaoImpl pdi = new ProductDaoImpl();
      pdi.createTable();
```

```
}

    @Override
    public void createTable() {
        String sql = "create table if not exists product(id int(11) auto_increment primary key,"
                + " product_name varchar(50), product_code varchar(50), qty int(11), unite_price double, "
                + "total_price double)";
        try {
            PreparedStatement pstmt = conn.prepareStatement(sql);
            pstmt.execute();
            System.out.println("Product table created");
        } catch (SQLException ex) {
            Logger.getLogger(ProductDaoImpl.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    @Override
    public void save(Product p) {
        String sql = "insert into product(product_name,product_code,qty,unite_price,total_price) values(?,?,?,?,?)";
        try {
            PreparedStatement pstmt = conn.prepareStatement(sql);
            pstmt.setString(1, p.getProductName());
            pstmt.setString(2, p.getProcutCode());
            pstmt.setInt(3, p.getQty());
            pstmt.setDouble(4, p.getUnitePrice());
            pstmt.setDouble(5, p.getTotalPrice());
```

```
            pstmt.executeUpdate();
            System.out.println("insert success");
        } catch (SQLException ex) {
            Logger.getLogger(ProductDaoImpl.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    @Override
    public void update(Product p) {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
    }

    @Override
    public void delete(Product p) {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
    }

    @Override
    public Product getProductById(int id) {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
    }

    @Override
    public Product getProductByProductCode(String code) {
        Product product = null;
```

```
    String sql = "select * from product where product_code=?";
    try {
      PreparedStatement pstmt = conn.prepareStatement(sql);
      pstmt.setString(1, code);
      ResultSet rs = pstmt.executeQuery();
      while (rs.next()) {
        product = new Product(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getInt(4),
            rs.getDouble(5), rs.getDouble(6));
      }
    } catch (SQLException ex) {
      Logger.getLogger(ProductDaoImpl.class.getName()).log(Level.SEVERE, null, ex);
    }
    return product;
  }

  @Override
  public List<Product> getList() {
    List<Product> list = new ArrayList();
    String sql = "select * from product";
    try {
      PreparedStatement pstmt = conn.prepareStatement(sql);
      ResultSet rs = pstmt.executeQuery();
      while (rs.next()) {
        Product product = new Product(rs.getInt(1), rs.getString(2), rs.getString(3),
rs.getInt(4),
            rs.getDouble(5), rs.getDouble(6));
        list.add(product);
```

```
            }
        }catch (SQLException ex) {
            Logger.getLogger(ProductDaoImpl.class.getName()).log(Level.SEVERE, null, ex);
        }
    return list;
    }
}


public class ProdcutSalesDaoImpl implements ProductSalesDao {
    Connection conn = CustomDbConnection.getConnection();
    public static void main(String[] args) {
        ProductSalesDaoImpl pdi = new ProductSalesDaoImpl();
        pdi.createTable();


    @Override
    public void createTable() {
        String sql = "create table if not exists sales(id int(11) auto_increment primary key,"
                + " product_name varchar(50), product_code varchar(50), qty int(11), unite_price
double, "
                + "total_price double, product_id int(11), FOREIGN KEY (product_id) REFERENCES
product(id))";
        try {
            PreparedStatement pstmt = conn.prepareStatement(sql);
            pstmt.execute();
            System.out.println("Sales table created");
        } catch (SQLException ex) {
```

```
            Logger.getLogger(ProdcutSalesDaoImpl.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    @Override
    public void save(ProductSales ps) {
        String sql = "insert into
sales(product_name,product_code,qty,unite_price,total_price,product_id) "
                + "values(?,?,?,?,?,?)";
        try {
            PreparedStatement pstmt = conn.prepareStatement(sql);
            pstmt.setString(1, ps.getProductName());
            pstmt.setString(2, ps.getProcutCode());
            pstmt.setInt(3, ps.getQty());
            pstmt.setDouble(4, ps.getUnitePrice());
            pstmt.setDouble(5, ps.getTotalPrice());
            pstmt.setInt(6, ps.getProduct().getId());
            pstmt.executeUpdate();
            System.out.println("insert success");
        } catch (SQLException ex) {
            Logger.getLogger(ProdcutSalesDaoImpl.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    @Override
    public void update(ProductSales ps) {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
```

```
    }

    @Override
    public void delete(ProductSales ps) {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
    }

    @Override
    public ProductSales getProductSalesById(int id) {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
    }

    @Override
    public ProductSales getProductSalesByProductCode(String code) {
        ProductSales productSales = null;
        String sql = "select * from sales where product_code=?";
        try {
            PreparedStatement pstmt = conn.prepareStatement(sql);
            pstmt.setString(1, code);
            ResultSet rs = pstmt.executeQuery();
            while (rs.next()) {
                productSales = new ProductSales(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getInt(4),
                        rs.getDouble(5), rs.getDouble(6), new Product(rs.getInt(7)));
            }
        } catch (SQLException ex) {
```

```
            Logger.getLogger(ProdcutSalesDaoImpl.class.getName()).log(Level.SEVERE, null, ex);
        }
        return productSales;
    }

    @Override
    public List<ProductSales> getList() {
        List<ProductSales> list = new ArrayList();
        String sql = "select * from product";
        try {
            PreparedStatement pstmt = conn.prepareStatement(sql);
            ResultSet rs = pstmt.executeQuery();
            while (rs.next()) {
                ProductSales productSales = new ProductSales(rs.getInt(1), rs.getString(2),
rs.getString(3), rs.getInt(4),
                        rs.getDouble(5), rs.getDouble(6), new Product(rs.getInt(7)));
                list.add(productSales);
            }
        } catch (SQLException ex) {
            Logger.getLogger(ProdcutSalesDaoImpl.class.getName()).log(Level.SEVERE, null, ex);
        }
        return list;
    }
}
```

```
public class SummaryDaoImpl implements SummaryDao {
   Connection conn = CustomDbConnection.getConnection();

   public static void main(String[] args) {
      SummaryDaoImpl pdi = new SummaryDaoImpl();
      pdi.createTable();
   }

   @Override
   public void createTable() {
      String sql = "create table if not exists summary(id int(11) auto_increment primary key,"
            + " product_name varchar(50), product_code varchar(50), total_qty int(11), sold_qty
int(11), "
            + "available_qty int(11), product_id int(11), FOREIGN KEY (product_id) REFERENCES
product(id))";
      try {
         PreparedStatement pstmt = conn.prepareStatement(sql);
         pstmt.execute();
         System.out.println("Summary table created");
      } catch (SQLException ex) {
         Logger.getLogger(SummaryDaoImpl.class.getName()).log(Level.SEVERE, null, ex);
      }
   }

   @Override
   public void save(Summary s) {
      String sql = "insert into
summary(product_name,product_code,total_qty,sold_qty,available_qty,product_id) "
```

```
            + "values(?,?,?,?,?,?)";
        try {
            PreparedStatement pstmt = conn.prepareStatement(sql);
            pstmt.setString(1, s.getProductName());
            pstmt.setString(2, s.getProcutCode());
            pstmt.setInt(3, s.getTotalQty());
            pstmt.setInt(4, s.getSoldQty());
            pstmt.setInt(5, s.getAvailableQty());
            pstmt.setInt(6, s.getProduct().getId());
            pstmt.executeUpdate();
            System.out.println("insert success");
        } catch (SQLException ex) {
            Logger.getLogger(SummaryDaoImpl.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    @Override
    public void update(Summary s) {
        String sql = "update summary set(total_qty=?,sold_qty=?,available_qty=? where
product_code=?)";
        try {
            PreparedStatement pstmt = conn.prepareStatement(sql);
            pstmt.setInt(1, s.getTotalQty());
            pstmt.setInt(2, s.getSoldQty());
            pstmt.setInt(3, s.getAvailableQty());
            pstmt.setString(4, s.getProcutCode());
        } catch (SQLException ex) {
```

```
            Logger.getLogger(SummaryDaoImpl.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    @Override
    public void delete(Summary s) {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
    }

    @Override
    public Summary getSummaryById(int id) {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
    }

    @Override
    public Summary getSummaryByProductCode(String code) {
        Summary summary = null;
        String sql = "select * from summary where product_code=?";
        try {
            PreparedStatement pstmt = conn.prepareStatement(sql);
            pstmt.setString(1, code);
            ResultSet rs = pstmt.executeQuery();
            while (rs.next()) {
                summary = new Summary(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getInt(4),
                        rs.getInt(5), rs.getInt(6), new Product(rs.getInt(7)));
            }
```

```
        } catch (SQLException ex) {
            Logger.getLogger(SummaryDaoImpl.class.getName()).log(Level.SEVERE, null, ex);
        }
        return summary;
    }

    @Override
    public List<Summary> getList() {
        List<Summary> list = new ArrayList();
        String sql = "select * from summary";
        try {
            PreparedStatement pstmt = conn.prepareStatement(sql);
            ResultSet rs = pstmt.executeQuery();
            while (rs.next()) {
                Summary summary = new Summary(rs.getInt(1), rs.getString(2), rs.getString(3),
rs.getInt(4),
                        rs.getInt(5), rs.getInt(6), new Product(rs.getInt(7)));
                list.add(summary);
            }
        } catch (SQLException ex) {
            Logger.getLogger(SummaryDaoImpl.class.getName()).log(Level.SEVERE, null, ex);
        }
        return list;
    }
}
```

Ppppppppppppppp

```
public void displayProductListIntoTable() {
        clearTable();
        ProductDao productDao = new ProductDaoImpl();
        List<Product> list = productDao.getList();



        DefaultTableModel model = (DefaultTableModel) tblDisplay.getModel();
        Object[] cols = new Object[6];
        for (int i = 0; i < list.size(); i++) {
            cols[0] = list.get(i).getId();
            cols[1] = list.get(i).getProductName();
            cols[2] = list.get(i).getProcutCode();
            cols[3] = list.get(i).getQty();
            cols[4] = list.get(i).getUnitePrice();
            cols[5] = list.get(i).getTotalPrice();
            model.addRow(cols);
        }

    }

    public void clearTable() {
        DefaultTableModel model = (DefaultTableModel) tblDisplay.getModel();
        model.setRowCount(0);

    }
```

```
////////////////
purchase view
// TODO add your handling code here:
        ProductDao productDao = new ProductDaoImpl();
        Product product = new Product(txtPName.getText().trim(), txtPCode.getText().trim(),
            Integer.parseInt(txtQty.getText().trim()),
Double.parseDouble(txtUPrice.getText().trim()),
            Double.parseDouble(txtTPrice.getText().trim()));
        productDao.save(product);



         SummaryDao summaryDao = new SummaryDaoImpl();
        try {
           Summary summary =
summaryDao.getSummaryByProductCode(txtPCode.getText().trim());


           if (summary.getProcutCode()!= null) {


            int totalQty = summary.getTotalQty() + Integer.parseInt(txtQty.getText().trim());
            int availableqty = summary.getAvailableQty() +
Integer.parseInt(txtQty.getText().trim());


            Summary summaryUp = new Summary(summary.getProcutCode(), totalQty,
summary.getSoldQty(),
                availableqty);
            summaryDao.update(summaryUp);
            JOptionPane.showMessageDialog(null, "Purchase Again success");
           } else {
```

```
            Product product1 =
productDao.getProductByProductCode(product.getProcutCode());

            Summary summary2 = new Summary(txtPName.getText().trim(),
txtPCode.getText().trim(),

                  Integer.parseInt(txtQty.getText().trim()), 0,
Integer.parseInt(txtQty.getText().trim()), product1);

            summaryDao.save(summary2);

            JOptionPane.showMessageDialog(null, "Purchase success");

         }

      } catch (Exception e) {

         Product product1 = productDao.getProductByProductCode(product.getProcutCode());

         Summary summary = new Summary(txtPName.getText().trim(),
txtPCode.getText().trim(),

               Integer.parseInt(txtQty.getText().trim()), 0,
Integer.parseInt(txtQty.getText().trim()),

               product1);

         summaryDao.save(summary);

         JOptionPane.showMessageDialog(null, "Purchase success");

      }


      displayProductListIntoTable();
```

==========

Sales View

```
public void displayProductSalesListIntoTable() {

      clearTable();

      SummaryDao summaryDao = new SummaryDaoImpl();
```

```
    ProductDao productDao = new ProductDaoImpl();

    List<Summary> list = summaryDao.getList();

    DefaultTableModel model = (DefaultTableModel) tblDisplay.getModel();
    Object[] cols = new Object[7];
    for (int i = 0; i < list.size(); i++) {
       cols[0] = list.get(i).getId();
       cols[1] = list.get(i).getProductName();
       cols[2] = list.get(i).getProcutCode();
       cols[3] = list.get(i).getTotalQty();
       cols[4] = list.get(i).getSoldQty();
       cols[5] = list.get(i).getAvailableQty();
       cols[5] = list.get(i).getAvailableQty();

       Product product = productDao.getProductByProductCode(list.get(i).getProcutCode());

       model.addRow(cols);
    }

  }

  public void clearTable() {
    DefaultTableModel model = (DefaultTableModel) tblDisplay.getModel();
    model.setRowCount(0);

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

```
// TODO add your handling code here:
    ProductDao pDao = new ProductDaoImpl();

    ProductSalesDao salesDao = new ProdcutSalesDaoImpl();

    Product product = pDao.getProductByProductCode(txtPCode.getText().trim());
    ProductSales sales = new ProductSales(txtPName.getText().trim(),
txtPCode.getText().trim(),
        Integer.parseInt(txtSoldQty.getText().trim()),
Double.parseDouble(txtUPrice.getText().trim()),
        Double.parseDouble(txtTPrice.getText().trim()), product);
    salesDao.save(sales);

    SummaryDao summaryDao = new SummaryDaoImpl();

    Summary summary = summaryDao.getSummaryByProductCode(sales.getProcutCode());

    if (summary.getProcutCode() != null) {

      int totalQty = summary.getTotalQty();
      int soldQty = summary.getSoldQty() + Integer.parseInt(txtSoldQty.getText().trim());
      int availableqty = summary.getAvailableQty() -
Integer.parseInt(txtSoldQty.getText().trim());

      Summary summaryUp = new Summary(summary.getProcutCode(), totalQty, soldQty,
availableqty);
      summaryDao.update(summaryUp);
```

```
            JOptionPane.showMessageDialog(null, "Sales success");

            displayProductSalesListIntoTable();
        }
    }

    private void tblDisplayMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        int i = tblDisplay.getSelectedRow();
        DefaultTableModel model = (DefaultTableModel) tblDisplay.getModel();
        txtPName.setText(model.getValueAt(i, 1).toString());
        txtPCode.setText(model.getValueAt(i, 2).toString());
        lblAvailQty.setText(model.getValueAt(i, 5).toString());


    }
```

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

Summary view

```
public void displayListIntoTable() {
        clearTable();
        SummaryDao summaryDao = new SummaryDaoImpl();
        ProductDao productDao = new ProductDaoImpl();


        List<Summary> list = summaryDao.getList();
        DefaultTableModel model = (DefaultTableModel) tblDisplay.getModel();
```

```
Object[] cols = new Object[7];


for (int i = 0; i < list.size(); i++) {
   cols[0] = list.get(i).getId();
   cols[1] = list.get(i).getProductName();
   cols[2] = list.get(i).getProcutCode();
   cols[3] = list.get(i).getTotalQty();
   cols[4] = list.get(i).getSoldQty();
   cols[5] = list.get(i).getAvailableQty();
   cols[6] = list.get(i).getAvailableQty();


   Product product = productDao.getProductByProductCode(list.get(i).getProcutCode());


   model.addRow(cols);
}
```