企業內的系統都是由資料庫以及一大堆的資料欄位、功能按鈕所構成，這就是 ERP(Enterprise Resource Planning)系統

學生資料管理系統:Client-Server 架構

簡介

學生資料管理系統是所有各級學校最重要的校務資訊系統功能之一，請用 Client-Server 架構實作一個簡易的雛型系統。

Lab: 準備資料庫

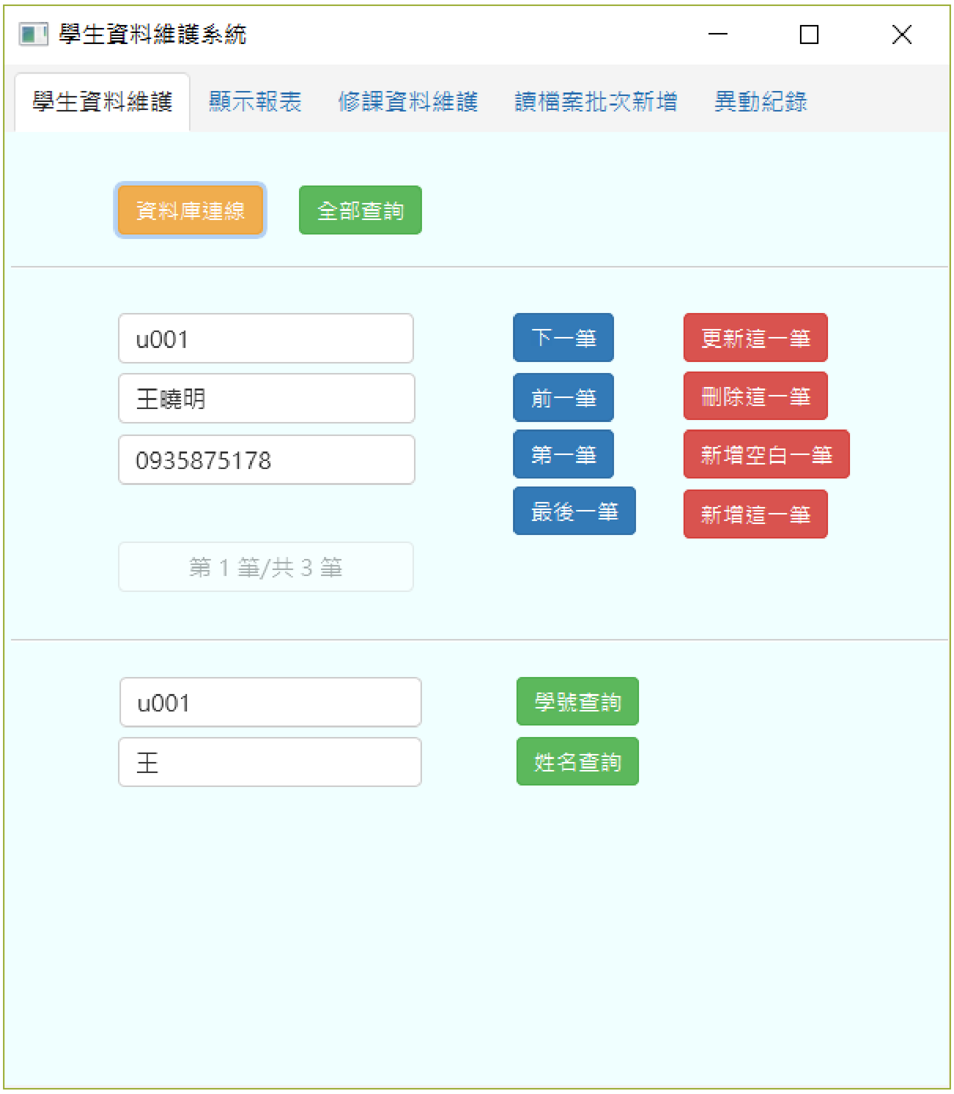
參看前一個講義。

Lab1: 學生資料維護

GUI 畫面整合提供許多操作功能
跟 GUI 畫面整合一起提供瀏覽、insert, delete, update 之功能與搜尋之功能
CRUD: create, retrieve, update, delete

學生資料 GUI 畫面整合，提供瀏覽、insert, delete, update 之功能與搜尋之功能，參考畫面如下:

JavaFX Theme Builder: CrudStudent.fxml



u001

王曉明

0935875178

第 1 筆/共 3 筆

u001

王

CrudStudent.fxml

```
<?xml version="1.0" encoding="UTF-8"?>



<?import javafx.scene.text.*?>

<?import javafx.geometry.*?>

<?import java.lang.*?>

<?import java.util.*?>

<?import javafx.scene.*?>

<?import javafx.scene.control.*?>

<?import javafx.scene.layout.*?>



<AnchorPane     id="AnchorPane"     fx:id="baseAnchorPane"     prefHeight="650.0"     prefWidth="600.0"     stylesheets="@bootstrap3.css"     xmlns="http://javafx.com/javafx/8"
```

```
xmlns:fx="http://javafx.com/fxml/1" fx:controller="javafx_studentdb.CrudStudentController">
  <children>
    <TabPane layoutX="-1.0" layoutY="-2.0" prefHeight="650.0" prefWidth="600.0" tabClosingPolicy="UNAVAILABLE">
      <tabs>
        <Tab text="學生資料維護">
          <content>
            <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="429.0" prefWidth="620.0" style="-fx-background-color: Azure;">
                <children>
                  <TextField fx:id="stName" layoutX="73.0" layoutY="153.0" prefHeight="32.0" prefWidth="189.0" promptText="姓名" />
                  <TextField fx:id="stID" layoutX="73.0" layoutY="115.0" prefHeight="32.0" prefWidth="188.0" promptText="學號" />
                  <TextField fx:id="stPhone" layoutX="73.0" layoutY="192.0" prefHeight="32.0" prefWidth="189.0" promptText="電話" />
                  <TextField fx:id="recordNo" alignment="CENTER" disable="true" editable="false" layoutX="73.0" layoutY="260.0" prefHeight="30.0" prefWidth="188.0" />
                  <Button layoutX="324.0" layoutY="115.0" mnemonicParsing="false" onAction="#nextRecord" styleClass="primary" text="下一筆" />
                  <Button layoutX="324.0" layoutY="153.0" mnemonicParsing="false" onAction="#previousRecord" styleClass="primary" text="前一筆" />
                  <Button layoutX="432.0" layoutY="115.0" mnemonicParsing="false" onAction="#update" styleClass="danger" text="更新這一筆" />
                  <Button layoutX="432.0" layoutY="152.0" mnemonicParsing="false" onAction="#delete" styleClass="danger" text="刪除這一筆" />
                  <Button layoutX="432.0" layoutY="227.0" mnemonicParsing="false" onAction="#insert" styleClass="danger" text="新增這一筆" />
                  <Button layoutX="432.0" layoutY="189.0" mnemonicParsing="false" onAction="#blankRecord" styleClass="danger" text="新增空白一筆" />
                  <Button layoutX="324.0" layoutY="189.0" mnemonicParsing="false" onAction="#firstRecord" styleClass="primary" text="第一筆" />
                  <Button layoutX="324.0" layoutY="225.0" mnemonicParsing="false" onAction="#lastRecord" styleClass="primary" text="最後一筆" />
                  <Separator layoutX="5.0" layoutY="317.0" prefHeight="0.0" prefWidth="610.0" />
                  <TextField fx:id="queryID" layoutX="74.0" layoutY="346.0" prefHeight="32.0" prefWidth="192.0" promptText="查詢學號" text="u001" />
                  <TextField fx:id="queryName" layoutX="73.0" layoutY="384.0" prefHeight="32.0" prefWidth="193.0" promptText="查詢姓名" text="王" />
                  <Button layoutX="326.0" layoutY="346.0" mnemonicParsing="false" onAction="#findID" styleClass="success" text="學號查詢" />
                  <Button layoutX="326.0" layoutY="384.0" mnemonicParsing="false" onAction="#findName" styleClass="success" text="姓名查詢" />
                  <Button layoutX="188.0" layoutY="34.0" mnemonicParsing="false" onAction="#findAll" styleClass="success" text="全部查詢" />
                  <Button layoutX="73.0" layoutY="34.0" mnemonicParsing="false" onAction="#connectToDB" styleClass="warning" text="資料庫連線" />
                  <Separator layoutX="5.0" layoutY="80.0" prefHeight="0.0" prefWidth="610.0" />
                </children>
              </AnchorPane>
          </content>
        </Tab>
          <Tab text="顯示報表">
            <content>
              <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="475.0" prefWidth="590.0">
                <children>
                  <VBox prefHeight="604.0" prefWidth="607.0">
                    <children>
```

```
             <HBox prefHeight="69.0" prefWidth="600.0" spacing="10.0">
               <children>
                 <Button mnemonicParsing="false" onAction="#report_all_student" text="學生基本資料報表">
                   <styleClass>
                     <String fx:value="action-btn" />
                     <String fx:value="success" />
                   </styleClass>
                   <HBox.margin>
                     <Insets left="20.0" top="15.0" />
                   </HBox.margin>
                 </Button>
                 <Button layoutX="276.0" layoutY="25.0" mnemonicParsing="false" text="報表輸出至檔案">
                   <styleClass>
                     <String fx:value="btn" />
                     <String fx:value="info" />
                   </styleClass>
                   <HBox.margin>
                     <Insets top="15.0" />
                   </HBox.margin>
                 </Button>
               </children>
               <VBox.margin>
                 <Insets />
               </VBox.margin>
             </HBox>
             <TextArea fx:id="display" maxWidth="-Infinity" minWidth="-Infinity" prefHeight="500.0" prefWidth="564.0">
               <VBox.margin>
                 <Insets left="20.0" />
               </VBox.margin></TextArea>
           </children>
         </VBox>
       </children>
     </AnchorPane>
   </content>
 </Tab>
<Tab text="修課資料維護">
 <content>
   <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="459.0" prefWidth="620.0" />
```

```
                </content>
              </Tab>
                <Tab text="讀檔案批次新增">
                    <content>
                        <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="180.0" prefWidth="200.0" />
                    </content>
                </Tab>
                <Tab text="異動紀錄">
                    <content>
                        <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="180.0" prefWidth="200.0">
                            <children>
                                <TextArea fx:id="status" layoutX="24.0" layoutY="18.0" prefHeight="403.0" prefWidth="572.0" />
                            </children>
                        </AnchorPane>
                    </content>
                </Tab>
           </tabs>
        </TabPane>
    </children>
</AnchorPane>
```

Student.java (學生 Class)

```
package javafx_studentdb;

/*
** 學生類別 學生基本屬性
*/
public class Student {

    public Student(String id, String name, String phone) {
        this.id = id;
        this.name = name;
        this.phone = phone;
    }
    public String id;     //private 比較好
    public String name;
    public String phone;

    public String getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public String getPhone() {
        return phone;
    }

    public void setId(String id) {
        this.id = id;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setPhone(String phone) {
        this.phone = phone;
    }

}
```

DBStudent.java

```
package javafx_studentdb;


import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

public class DBStudent {

    private static final String DB_URL = "jdbc:mariadb://localhost:3306/studentdb";
    private static final String USER = "mis";
    private static final String PASSWORD = "mis123";

    //全域變數操作資料庫比較方便
    private static Connection conn = null;
    private static Statement state = null;
    private static ResultSet resultSet = null;

    //連線 connect 資料庫
    public static boolean connect() {
        boolean sucess = false;
        try {
            conn = DriverManager.getConnection(DB_URL, USER, PASSWORD);
            state = conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);
            resultSet = state.executeQuery("SELECT * FROM student");
            sucess = true;
        } catch (SQLException ex) {
            System.out.println("資料庫 connect 異常:" + ex.toString());
        }
        return sucess;
    }
    //取得目前 resultSet 的所有紀錄並回傳 Student 物件的 List
    public static List< Student> getAllStudent() {
        List< Student> results = new ArrayList();
        try {
            while (resultSet.next()) {
                results.add(new Student(
                        resultSet.getString("student_id"),
                        resultSet.getString("name"),
                        resultSet.getString("phone")));
        } // end while
    } // end try
    catch (SQLException ex) {
        System.out.println("資料庫 getAllStudent 操作異常:" + ex.toString());
    } // end catch
    return results;
}

//選擇全部的資料
public static void selectAll() {
    try {
        resultSet = state.executeQuery("select * from student");
    } catch (SQLException ex) {
        System.out.println("selectAll 操作異常:" + ex.toString());
    }
}

//選擇特定姓名
public static boolean selectByName(String name) {
    boolean sucess = false;
    try {
        String sql = String.format("select * from student where name like '%s%s'", name, "%");
        resultSet = state.executeQuery(sql);
        sucess = resultSet.next();
        resultSet.beforeFirst();
    } catch (SQLException ex) {
        System.out.println("資料庫 selectByName 操作異常:" + ex.toString());
    }
    return sucess;
}

//選擇某個 student_id
public static boolean selectByID(String id) {
    boolean sucess = false;
    try {
        String sql = String.format("select * from student where student_id = '%s'", id);
        resultSet = state.executeQuery(sql);
        if (resultSet.next()) {
            sucess = true;
        }
        resultSet.beforeFirst();
    } catch (SQLException ex) {
        System.out.println("資料庫 selectByID 操作異常:" + ex.toString());
    }
    return sucess;
}

//新增一筆
public static boolean insert(String id, String name, String phone) {
    boolean sucess = false;
    String sql = String.format("Insert into student(student_id,name,phone) "
            + "values ('%s','%s','%s')", id, name, phone);
    try {
        state.execute(sql);
        resultSet = state.executeQuery("SELECT * FROM student");
        sucess = true;
    } catch (SQLException ex) {
        System.out.println("資料庫 insert 操作異常:" + ex.toString());
    }
    return sucess;
}

//insert 另一種寫法:參數可以是 Student 物件  引數可以比較精簡
public static boolean insert2(Student st) {
    boolean sucess = false;
    String sql = String.format("Insert into student(student_id,name,phone) "
            + "values ('%s','%s','%s')", st.getId(), st.getName(), st.getPhone());
    try {
        state.execute(sql);
        resultSet = state.executeQuery("SELECT * FROM student");
        sucess = true;
    } catch (SQLException ex) {
        System.out.println("資料庫 insert 操作異常:" + ex.toString());
    }
    return sucess;
}
```

```
    //刪除這一筆
    public static boolean delete(String id) {
        String sql = String.format("delete from student where student_id='%s'", id);
        boolean sucess = false;
        try {
            sucess = state.execute(sql);
            resultSet = state.executeQuery("SELECT * FROM student");
        } catch (SQLException ex) {
            System.out.println("資料庫 delete 操作異常:" + ex.toString());
        }
        return sucess;
    }

    //更新這一筆
    public static boolean update(String id, String name, String phone) {
        boolean sucess = false;
        String sql = String.format("update  student SET name='%s',phone='%s' where
student_id='%s'", name, phone, id);
        try {
            state.execute(sql);
            resultSet = state.executeQuery("SELECT * FROM student");//最好是撈新的資料
            sucess = true;
        } catch (SQLException ex) {
            System.out.println("資料庫 update 操作異常:" + ex.toString());
        }
        return sucess;
    }

    //關閉連線 暫時沒用到
    public static boolean closeConnection() {
        boolean sucess = false;
        try {
            conn.close();
            sucess = true;
        } catch (SQLException ex) {
            System.out.println("資料庫 close 操作異常:" + ex.toString());
        }
```

```
            return sucess;
        }

        //Terminal 印出目前的資料
        public static void print(List<Student> students) {
            for (Student student : students) {
                System.out.printf("%s %s %s\n", student.getId(), student.getName(),
student.getPhone());
            }
        }

        //Terminal 印出目前的資料  寫法 2 lambda 寫法
        public static void print2(List<Student> students) {
            students.forEach((student) -> {
                System.out.printf("%s %s %s\n", student.getId(), student.getName(),
student.getPhone());
            });
        }

        //測試一下資料庫操作是否能成功
        public static void main(String[] args) {
            DBStudent.connect();
            List<Student> results;
            print(DBStudent.getAllStudent());
            //Studentdb.insert("u127", "馬大牌 4", "0012");
            //results = DBStudent.getAllStudent();
            //print(DBStudent.getAllStudent());
        }
}
```

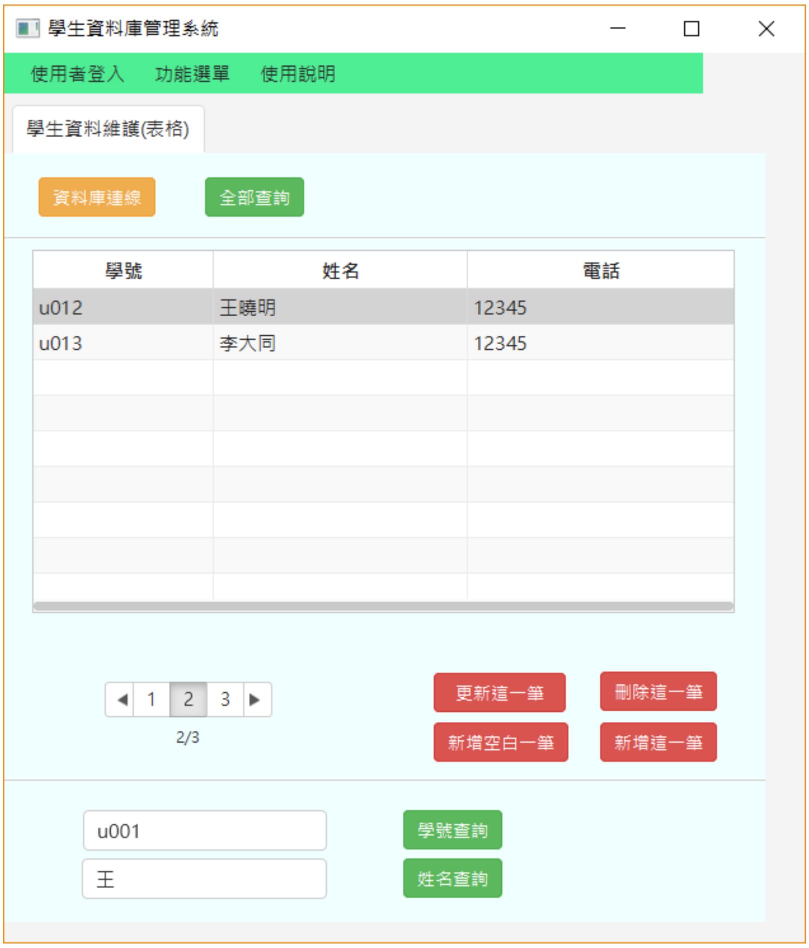
CrudStudentController.java

```
package javafx_studentdb;


import java.net.URL;
import java.util.Calendar;
import java.util.List;
import java.util.ResourceBundle;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.TextArea;
import javafx.scene.control.TextField;

public class CrudStudentController implements Initializable {

    @FXML
    private TextField stName;
    @FXML
    private TextField stID;
    @FXML
    private TextField stPhone;
    @FXML
    private TextField recordNo;
    @FXML
    private TextField queryID;
    @FXML
    private TextField queryName;
    @FXML
    private TextArea display;
    @FXML
    private TextArea status;

    Calendar time = Calendar.getInstance();

    List<Student> students = null;
    int current = 0;
    int student_size = 0;

    @Override
    public void initialize(URL url, ResourceBundle rb) {

        /* 連線也可以寫在這裡
        if (DBStudent.connect()) {
            students = DBStudent.getAllStudent();
            showRecord();
            status.appendText(time.getTime() + "連線成功\n");
        } else {
            status.appendText(time.getTime() + "連線失敗\n");
        }
        */
    }

    private void showRecord() {
        if (!students.isEmpty()) {
            stID.setText(students.get(current).getId());
            stName.setText(students.get(current).getName());
            stPhone.setText(students.get(current).getPhone());
            String msg = String.format("第 %d 筆/共 %d 筆", current + 1, students.size());
            recordNo.setText(msg);
        } else {
            //return;
        }
    }

    @FXML
    private void nextRecord(ActionEvent event) {

        if (current < students.size() - 1) {
            current += 1;
        } else {
            current = 0;
        }
        showRecord();
    }

    @FXML
    private void previousRecord(ActionEvent event) {
        if (current > 0) {
            current -= 1;
        } else {
            current = students.size() - 1;
        }
        showRecord();
```

```
}

@FXML
private void update(ActionEvent event) {
    String id = stID.getText();
    String name = stName.getText();
    String phone = stPhone.getText();
    boolean sucess = DBStudent.update(id, name, phone);

    if (sucess) {
        status.appendText(time.getTime() + "更新成功\n");
        students = DBStudent.getAllStudent();
        showRecord();
    } else {
        status.appendText(time.getTime() + "更新失敗\n");
    }
}

@FXML
private void delete(ActionEvent event) {
    String id = students.get(current).getId();
    boolean sucess = DBStudent.delete(id);
    showlog("刪除一筆資料", sucess);
    students = DBStudent.getAllStudent();
    current = 0;
    showRecord();
}

@FXML
private void insert(ActionEvent event) {
    String id = stID.getText();
    String name = stName.getText();
    String phone = stPhone.getText();
    boolean sucess = DBStudent.insert(id, name, phone);
    showlog("新增一筆資料", sucess);

    students = DBStudent.getAllStudent();
    showRecord();
}

@FXML
private void blankRecord(ActionEvent event) {
    stID.setText("");
    stName.setText("");
    stPhone.setText("");
    recordNo.setText("");
}

@FXML
private void firstRecord(ActionEvent event) {
    current = 0;
    showRecord();
}

@FXML
private void lastRecord(ActionEvent event) {
    current = students.size() - 1;
    showRecord();
}

@FXML
private void findID(ActionEvent event) {
    boolean sucess = DBStudent.selectByID(queryID.getText());
    if (sucess) {
        students = DBStudent.getAllStudent();
        current = 0;
        showRecord();
        //System.out.println(students.size());
        status.appendText(time.getTime() + "查找學號成功\n");
    } else {
        status.appendText(time.getTime() + "查找學號失敗\n");
    }
}

@FXML
private void findName(ActionEvent event) {
```

```
    boolean sucess = DBStudent.selectByName(queryName.getText());
    showlog("查找姓名", sucess);
    students = DBStudent.getAllStudent();
    current = 0;
    showRecord();
}

@FXML
private void findAll(ActionEvent event) {
    DBStudent.selectAll();
    students = DBStudent.getAllStudent();
    showRecord();
}

@FXML
private void report_all_student(ActionEvent event) {

    DBStudent.selectAll();
    List<Student> all_stud = DBStudent.getAllStudent();
    display.setText("學生基本資料報表\n");
    String row = String.format("%s\t%s\t %s\n", "學號", "姓名", "電話");
    display.appendText(row);
    for (Student stud : all_stud) {
        row = String.format("%s\t%s\t%s\n", stud.getId(), stud.getName(), stud.getPhone());
        display.appendText(row);
    }
}

@FXML
private void connectToDB(ActionEvent event) {
    if (DBStudent.connect()) {
        students = DBStudent.getAllStudent();
        showRecord();
        status.appendText(time.getTime() + "連線成功\n");
    } else {
        status.appendText(time.getTime() + "連線失敗\n");
    }
    }

    @FXML
    private void closeConnection(ActionEvent event) {
        boolean sucess = DBStudent.closeConnection();
        showlog("關閉連線", sucess);
        students = null;
        stID.setText("");
        stName.setText("");
        stPhone.setText("");
        recordNo.setText("");
    }

    public void showlog(String msg, boolean sucess) {
        if (sucess) {
            status.appendText(time.getTime() + msg + " 成功\n");
        } else {
            status.appendText(time.getTime() + msg + "失敗\n");
        }
    }
}
```

Lab2: 學生資料維護-TableView 表格操作



學號	姓名	電話
u012	王曉明	12345
u013	李大同	12345

u001

王

CrudStudentTableView.fxml

```
<?xml version="1.0" encoding="UTF-8"?>



<?import javafx.scene.text.*?>

<?import javafx.geometry.*?>

<?import java.lang.*?>

<?import java.util.*?>

<?import javafx.scene.*?>

<?import javafx.scene.control.*?>

<?import javafx.scene.layout.*?>



<AnchorPane id="AnchorPane" prefHeight="650.0" prefWidth="600.0" stylesheets="@bootstrap3.css" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1" fx:controller="javafx_studentdb.CrudStudentTableViewController">

    <children>

        <TabPane layoutX="-1.0" layoutY="-2.0" prefHeight="650.0" prefWidth="600.0" tabClosingPolicy="UNAVAILABLE">

            <tabs>
```

```
          <Tab text="學生資料維護(表格)">
            <content>
              <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="429.0" prefWidth="620.0" style="-fx-background-color: Azure;">
                <children>
                  <Button layoutX="339.0" layoutY="409.0" mnemonicParsing="false" onAction="#update" prefHeight="31.0" prefWidth="104.0" styleClass="danger" text="更新這一筆" />
                  <Button layoutX="470.0" layoutY="408.0" mnemonicParsing="false" onAction="#delete" styleClass="danger" text="刪除這一筆" />
                  <Button layoutX="470.0" layoutY="448.0" mnemonicParsing="false" onAction="#insert" styleClass="danger" text="新增這一筆" />
                  <Button layoutX="339.0" layoutY="448.0" mnemonicParsing="false" onAction="#blankRecord" styleClass="danger" text="新增空白一筆" />
                  <Separator layoutX="-5.0" layoutY="488.0" prefHeight="0.0" prefWidth="610.0" />
                  <TextField fx:id="queryID" layoutX="63.0" layoutY="517.0" prefHeight="32.0" prefWidth="192.0" promptText="查詢學號" text="u001" />
                  <TextField fx:id="queryName" layoutX="62.0" layoutY="555.0" prefHeight="32.0" prefWidth="193.0" promptText="查詢姓名" text="王" />
                  <Button layoutX="315.0" layoutY="517.0" mnemonicParsing="false" onAction="#findID" styleClass="success" text="學號查詢" />
                  <Button layoutX="315.0" layoutY="555.0" mnemonicParsing="false" onAction="#findName" styleClass="success" text="姓名查詢" />
                  <Button layoutX="159.0" layoutY="19.0" mnemonicParsing="false" onAction="#findAll" styleClass="success" text="全部查詢" />
                  <Button layoutX="28.0" layoutY="19.0" mnemonicParsing="false" onAction="#connectToDB" styleClass="warning" text="資料庫連線" />
                  <Separator layoutX="-3.0" layoutY="61.0" prefHeight="0.0" prefWidth="610.0" />
                  <TableView fx:id="table_student" layoutX="23.0" layoutY="76.0" prefHeight="286.0" prefWidth="553.0">
                    <columns>
                      <TableColumn fx:id="col_id" onEditCommit="#onIdEditCommit" prefWidth="142.0" text="學號" />
                      <TableColumn fx:id="col_name" onEditCommit="#onNameEditCommit" prefWidth="200.0" text="姓名" />
                        <TableColumn fx:id="col_phone" onEditCommit="#onPhoneEditCommit" prefWidth="210.0" text="電話" />
                    </columns>
                  </TableView>
                  <Pagination fx:id="pagination" layoutX="28.0" layoutY="381.0" prefHeight="86.0" prefWidth="236.0" />
                </children>
              </AnchorPane>
            </content>
          </Tab>
      </tabs>
    </TabPane>
  </children>
</AnchorPane>
```

CrudStudentTableViewController.java

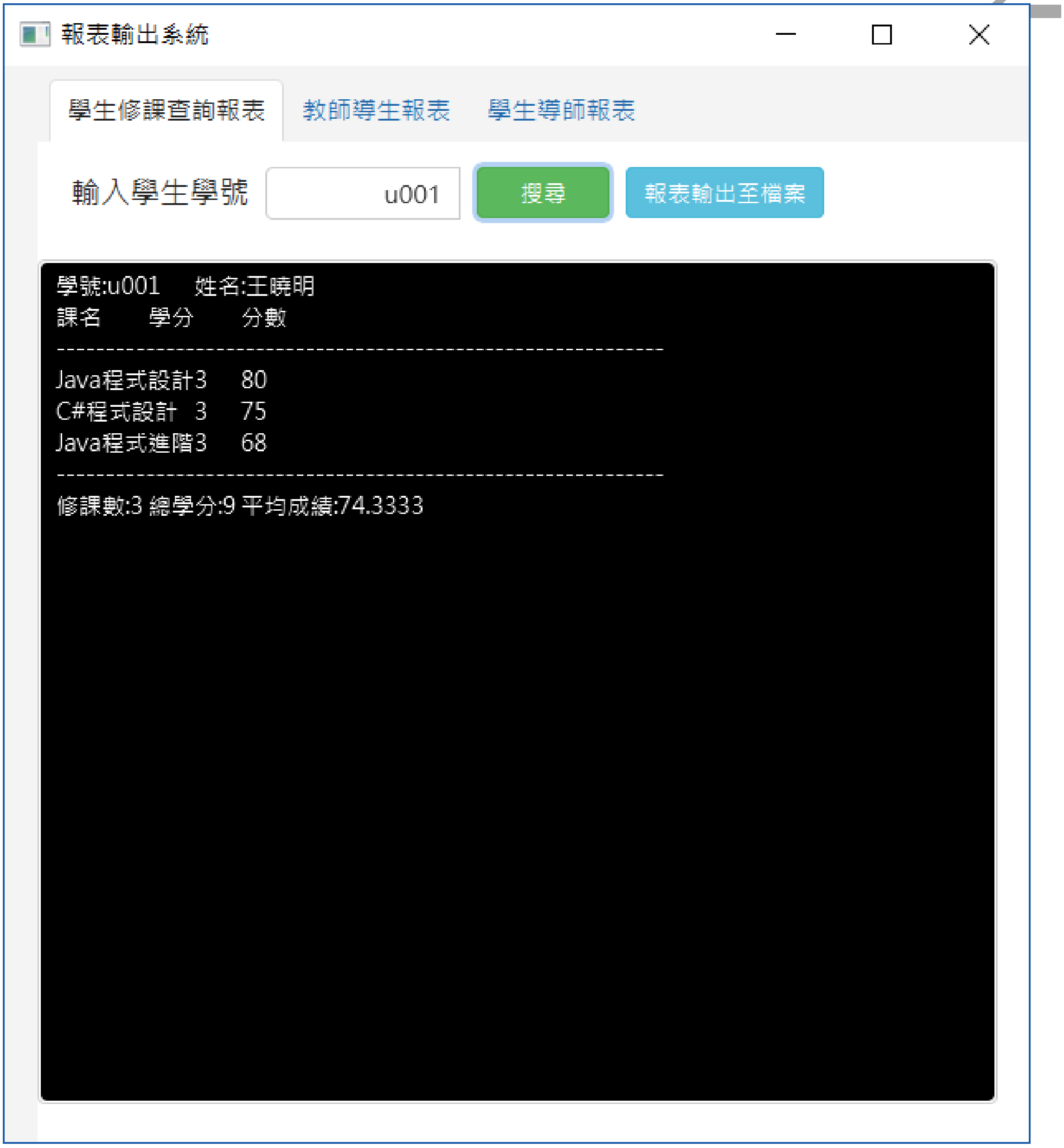
```
package javafx_studentdb;
```

```
import java.net.URL;
import java.util.List;
import java.util.ResourceBundle;
import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.Pagination;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableColumn.CellEditEvent;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.control.cell.TextFieldTableCell;

public class CrudStudentTableViewController implements Initializable {

    List<Student> students = null;

    @FXML
    private TableView<Student> table_student;
    @FXML
    private TableColumn<Student, String> col_id;
    @FXML
    private TableColumn<Student, String> col_name;
    @FXML
    private TableColumn<Student, String> col_phone;
    @FXML
    private Pagination pagination;

    private final int RowsPerPage = 2;

    @FXML
    private TextField queryID;
    @FXML
    private TextField queryName;
    @Override
    public void initialize(URL url, ResourceBundle rb) {
        initTable(); //表格初始化
        connect_db(); //連結到資料庫，並將所有資料顯示到表格
    }

    private void initTable() {
        col_id.setCellValueFactory(new PropertyValueFactory<>("id"));
        col_name.setCellValueFactory(new PropertyValueFactory<>("name"));
        col_phone.setCellValueFactory(new PropertyValueFactory<>("phone"));

        //按下頁次會驅動的事件，寫法格式有點難理解，說明如後:
        //ObservableValue<? extends Number>  是介面，
        // ? extends Number  表示某種型態繼承 Number 類別   ?表示此型態沒被用到所以用?
代替
        // changed  有三個參數: ObservableValue、舊的頁次、新的頁次
        // ObservableValue 是頁次物件的一些屬性  印出如下的結果:
        //IntegerProperty [bean: Pagination[id=pagination, styleClass=pagination], name:
currentPageIndex, value: 1]
        pagination.currentPageIndexProperty().addListener(new ChangeListener<Number>() {
            @Override
            public void changed(ObservableValue<? extends Number> observable, Number
oldValue, Number newValue) {
                showTablePage(newValue.intValue(), RowsPerPage);
                //System.out.println(observable);
            }
        });

        //  表格切換到一下筆，對應的驅動方法，此處暫時沒用到，寫法與前面類似
        table_student.getSelectionModel().selectedIndexProperty().addListener(new
ChangeListener<Number>() {
            @Override
            public void changed(ObservableValue<? extends Number> observable, Number
oldValue, Number newValue) {
                System.out.println(newValue);
            }
        });
        //讓表格內容可以修改
        table_student.setEditable(true);
        //表格欄位設定成可以編輯必須分別塞入一個 TextFieldTableCell 類別元件
        col_id.setCellFactory(TextFieldTableCell.forTableColumn());
        col_name.setCellFactory(TextFieldTableCell.forTableColumn());
        col_phone.setCellFactory(TextFieldTableCell.forTableColumn());
    }

    //表格內容載入
    private void loadTable() {
        int totalPage = (int) (Math.ceil(students.size() * 1.0 / RowsPerPage));
        pagination.setPageCount(totalPage);
        //pagination.setCurrentPageIndex(0);
        int currentpg = pagination.getCurrentPageIndex();
        showTablePage(currentpg, RowsPerPage);
    }

    //顯示某一個頁面的表格內容
    private void showTablePage(int pg, int row_per_pg) {
        table_student.getItems().clear(); //先清除表格內容
        int from = pg * row_per_pg;   //計算在此頁面顯示第幾筆到第幾筆
        int to = Math.min(from + row_per_pg, students.size());
        //students 一筆一筆加到表格中
        for (int i = from; i < to; i++) {
            table_student.getItems().add(students.get(i));
        }
    }

    //學生學號欄位若有修改驅動這個方法
    @FXML
    private void onIdEditCommit(CellEditEvent<Student, String> event) {
        //拿到表格中所在的該筆紀錄(是一筆學生物件)
        Student stud = table_student.getSelectionModel().getSelectedItem();
        stud.setId(event.getNewValue()); //將該筆學生物件修改成新的值
    }

    //學生姓名欄位若有修改驅動這個方法
    @FXML
```

```
    private void onNameEditCommit(CellEditEvent<Student, String> t) {
        //取得該筆紀錄的方式有以下 3 種寫法:
        //Student stud = t.getTableView().getItems().get(t.getTablePosition().getRow());
        //Student stud = (Student) t.getTableView().getItems().get(t.getTablePosition().getRow());
        Student stud = table_student.getSelectionModel().getSelectedItem();
        stud.setName(t.getNewValue());
        //另種寫法 2:
        //((Student)
t.getTableView().getItems().get(t.getTablePosition().getRow())).setName(t.getNewValue());
    }

    //學生電話欄位若有修改驅動這個方法
    @FXML
    private void onPhoneEditCommit(CellEditEvent<Student, String> event) {
        Student stud = table_student.getSelectionModel().getSelectedItem();
        stud.setPhone(event.getNewValue());
    }

    //更新一筆紀錄
    @FXML
    private void update(ActionEvent event) {
        Student stud = table_student.getSelectionModel().getSelectedItem();
        String id = stud.getId();
        String name = stud.getName();
        String phone = stud.getPhone();
        DBStudent.update(id, name, phone);
        students = DBStudent.getAllStudent();
        loadTable();
    }

    // 刪除一筆紀錄
    @FXML
    private void delete(ActionEvent event) {

        Student stud = table_student.getSelectionModel().getSelectedItem();
        String id = stud.getId();

        boolean sucess = DBStudent.delete(id);
        students = DBStudent.getAllStudent();
        loadTable();
    }

    //新增一筆紀錄
    @FXML
    private void insert(ActionEvent event) {
        Student stud = table_student.getSelectionModel().getSelectedItem();
        String id = stud.getId();
        String name = stud.getName();
        String phone = stud.getPhone();
        DBStudent.insert(id, name, phone);
        students = DBStudent.getAllStudent();
        loadTable();
    }

    //新增空白的一筆紀錄
    @FXML
    private void blankRecord(ActionEvent event) {
        table_student.getItems().add(new Student("u012", "李大同 1", "12345"));
    }

    //搜尋特定學生學號
    @FXML
    private void findID(ActionEvent event) {
        boolean sucess = DBStudent.selectByID(queryID.getText());
        students = DBStudent.getAllStudent();
        loadTable();
    }

    //搜尋特定姓名
    @FXML
    private void findName(ActionEvent event) {
        DBStudent.selectByName(queryName.getText());
        students = DBStudent.getAllStudent();
        loadTable();
    }
    //搜尋全部
    @FXML
    private void findAll(ActionEvent event) {
        DBStudent.selectAll();
        students = DBStudent.getAllStudent();
        loadTable();
    }

    //連接到資料庫按鈕呼叫的方法
    @FXML
    private void connectToDB(ActionEvent event) {
        connect_db();
    }

    //連接到資料庫 並將所有資料顯示到表格
    private void connect_db() {
        DBStudent.connect();
        students = DBStudent.getAllStudent();
        loadTable();
    }

}
```

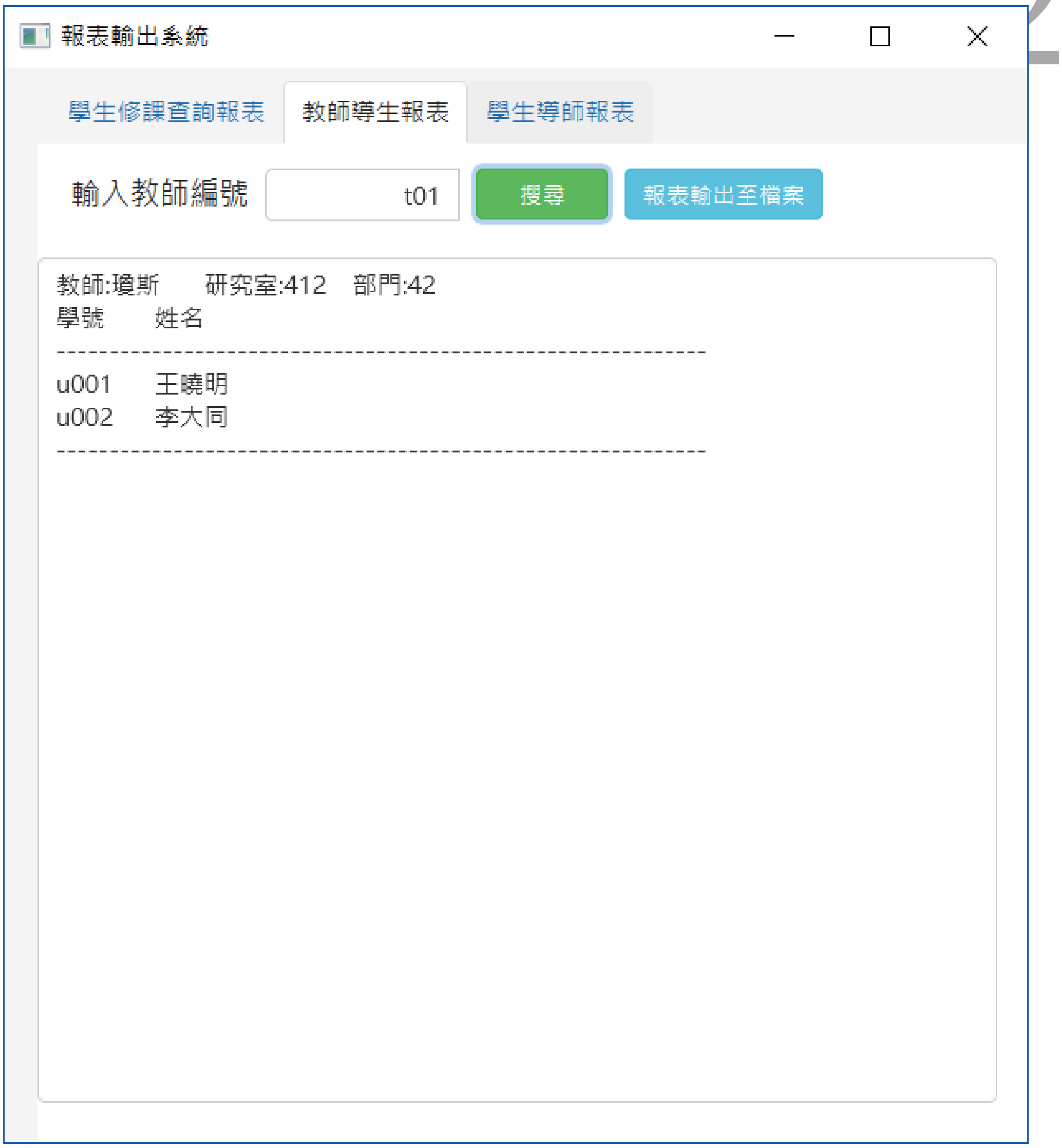
Lab3: 產生報表

Theme Builder: GenerateReport.fxml

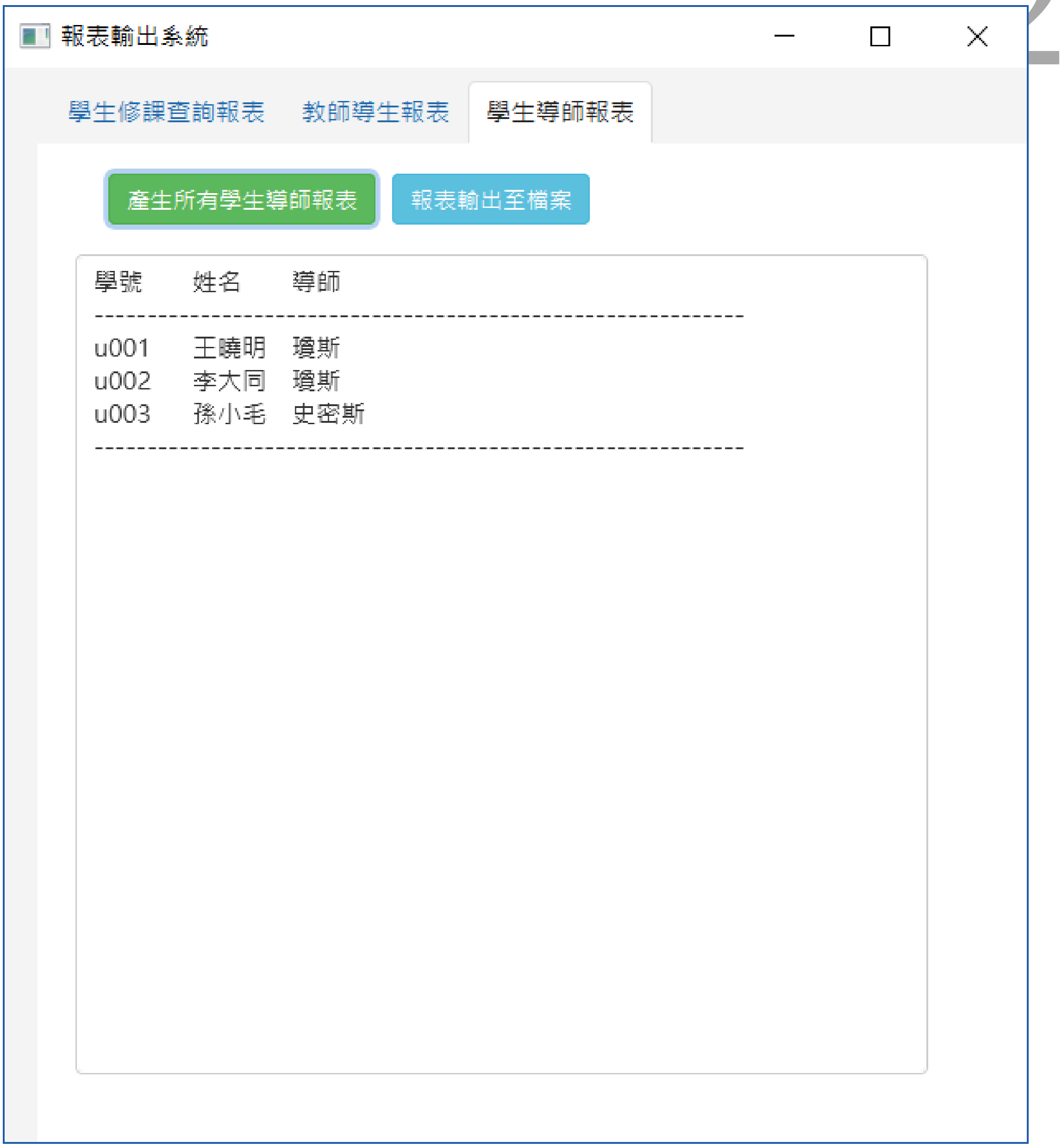


u001

```
學號:u001    姓名:王曉明
課名      學分      分數
------------------------------------------------------------
Java程式設計3    80
C#程式設計  3    75
Java程式進階3    68
------------------------------------------------------------
修課數:3 總學分:9 平均成績:74.3333
```



```
教師:瓊斯     研究室:412   部門:42
學號      姓名
------------------------------------------------------------
u001      王曉明
u002      李大同
------------------------------------------------------------
```



```
學號    姓名    導師
------------------------------------------------------------
u001    王曉明  瓊斯
u002    李大同  瓊斯
u003    孫小毛  史密斯
------------------------------------------------------------
```

GenerateReport.fxml

```
<?xml version="1.0" encoding="UTF-8"?>



<?import javafx.scene.text.*?>

<?import javafx.geometry.*?>

<?import java.lang.*?>

<?import java.net.*?>

<?import java.util.*?>

<?import javafx.scene.*?>

<?import javafx.scene.control.*?>
```

```
<?import javafx.scene.layout.*?>


<AnchorPane prefHeight="650.0" prefWidth="600.0" stylesheets="@bootstrap3.css" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1" fx:controller="javafx_studentdb.GenerateReportController">

   <children>

      <TabPane layoutY="1.0" prefHeight="650.0" prefWidth="600.0" tabClosingPolicy="UNAVAILABLE" AnchorPane.leftAnchor="20.0">

        <tabs>

          <Tab text="學生修課查詢報表">

               <content>

                  <AnchorPane prefHeight="665.0" prefWidth="789.0">

                     <children>

                        <TextArea fx:id="display" layoutY="71.0" maxWidth="-Infinity" minWidth="-Infinity" prefHeight="511.0" prefWidth="581.0" stylesheets="@myTextArea.css" />

                        <TextField fx:id="field_student_id" alignment="CENTER_RIGHT" layoutX="138.0" layoutY="15.0" prefHeight="32.0" prefWidth="118.0" styleClass="first" text="u001">

                           <HBox.margin>

                              <Insets top="15.0" />

                           </HBox.margin>

                        </TextField>

                        <Button layoutX="266.0" layoutY="15.0" mnemonicParsing="false" onAction="#generate_stud_report" text="搜尋">

                           <HBox.margin>

                              <Insets top="15.0" />

                           </HBox.margin>

                           <styleClass>

                              <String fx:value="action-btn" />

                              <String fx:value="success" />

                           </styleClass>

                        </Button>

                        <Label layoutX="20.0" layoutY="15.0" prefHeight="30.0" prefWidth="108.0" text="輸入學生學號">

                           <HBox.margin>

                              <Insets left="20.0" top="15.0" />

                           </HBox.margin>

                           <font>

                              <Font size="18.0" />

                           </font>

                        </Label>

                        <Button layoutX="356.0" layoutY="15.0" mnemonicParsing="false" onAction="#generate_stud_report" text="報表輸出至檔案">

                           <styleClass>

                              <String fx:value="btn" />

                              <String fx:value="info" />

                           </styleClass>
```

```
                       <HBox.margin>
                           <Insets top="15.0" />
                       </HBox.margin>
                   </Button>
               </children>
           </AnchorPane>
       </content>
</Tab>
<Tab text="教師導生報表">
   <content>
      <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="180.0" prefWidth="200.0">
               <children>
                   <HBox prefHeight="75.0" prefWidth="582.0" spacing="10.0">
                       <children>
                           <Label prefHeight="30.0" prefWidth="108.0" text="輸入教師編號">
                               <font>
                                   <Font size="18.0" />
                               </font>
                               <HBox.margin>
                                   <Insets left="20.0" top="15.0" />
                               </HBox.margin>
                           </Label>
                           <TextField fx:id="field_teacher_id" alignment="CENTER_RIGHT" prefHeight="32.0" prefWidth="118.0" styleClass="first" text="t01">
                               <HBox.margin>
                                   <Insets top="15.0" />
                               </HBox.margin>
                           </TextField>
                           <Button mnemonicParsing="false" onAction="#generate_teacher_report" text="搜尋">
                               <styleClass>
                                   <String fx:value="action-btn" />
                                   <String fx:value="success" />
                               </styleClass>
                               <HBox.margin>
                                   <Insets top="15.0" />
                               </HBox.margin>
                           </Button>
                           <Button layoutX="276.0" layoutY="25.0" mnemonicParsing="false" onAction="#generate_stud_report" text="報表輸出至檔案">
                               <styleClass>
```

```
                  <String fx:value="btn" />
                  <String fx:value="info" />
                </styleClass>
                <HBox.margin>
                  <Insets top="15.0" />
                </HBox.margin>
              </Button>
            </children>
            <VBox.margin>
              <Insets />
            </VBox.margin>
          </HBox>
          <TextArea fx:id="display_teacher" editable="false" layoutY="69.0" prefHeight="512.0" prefWidth="582.0" />
        </children></AnchorPane>
  </content>
</Tab>
  <Tab text="學生導師報表">
    <content>
      <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="180.0" prefWidth="200.0">
        <children>
          <TextArea fx:id="display_studAdvisor" layoutX="23.0" layoutY="67.0" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="497.0" prefWidth="517.0" />
          <Button layoutX="43.0" layoutY="18.0" mnemonicParsing="false" onAction="#generate_studAdvisor_report" text="產生所有學生導師報表">
            <styleClass>
              <String fx:value="action-btn" />
              <String fx:value="success" />
            </styleClass>
            <HBox.margin>
              <Insets left="20.0" top="15.0" />
            </HBox.margin>
          </Button>
          <Button layoutX="215.0" layoutY="18.0" mnemonicParsing="false" onAction="#generate_stud_report" text="報表輸出至檔案">
            <styleClass>
              <String fx:value="btn" />
              <String fx:value="info" />
            </styleClass>
            <HBox.margin>
              <Insets top="15.0" />
            </HBox.margin>
```

```
                              </Button>

                          </children>

                      </AnchorPane>

                  </content>

              </Tab>

         </tabs>

      </TabPane>

  </children>

</AnchorPane>
```

GenerateReportController.java

```
package javafx_studentdb;

import java.net.URL;
import java.util.ResourceBundle;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.TextArea;
import javafx.scene.control.TextField;



public class GenerateReportController implements Initializable {

    @FXML
    private TextField field_student_id;
    @FXML
    private TextArea display;
    @FXML
    private TextField field_teacher_id;
    @FXML
    private TextArea display_teacher;
    @FXML
    private TextArea display_studAdvisor;

    @Override
    public void initialize(URL url, ResourceBundle rb) {
        DBReport.connect(); //DBReport 資料庫連線 只要連線一次即可
    }

    @FXML
    private void generate_stud_report(ActionEvent event) {
        String student_id = field_student_id.getText();
        display.setText(DBReport.reportStudClass(student_id));
    }

    @FXML
    private void generate_teacher_report(ActionEvent event) {
        String teacher_id = field_teacher_id.getText();
        display_teacher.setText(DBReport.reportTeacherStud(teacher_id));
    }

    @FXML
    private void generate_studAdvisor_report(ActionEvent event) {

        display_studAdvisor.setText(DBReport.reportStudAdvisor());
    }
}
```

DBReport.java

```
package javafx_studentdb;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class DBReport {

    private static final String DB_URL = "jdbc:mariadb://localhost:3306/studentdb2";
    private static final String USER = "mis";
    private static final String PASS = "mis123";
    private static Connection conn = null;
    private static Statement state = null;

    //必須連線一次　使用全域變數 conn, state
    public static boolean connect() {
        boolean sucess = false;
        try {
            conn = DriverManager.getConnection(DB_URL, USER, PASS);
            state = conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);
            sucess = true;
        } catch (SQLException ex) {
            System.out.println("資料庫操作異常:" + ex.toString());
        }
        return sucess;
    }

    //學生修課報表-----
    public static String reportStudClass(String student_id) {
        String msg;
        String report = "";
        ResultSet rs;

        String sql = String.format("SELECT student_class.student_id, student.name, class_name, class.credit, score "
                + "FROM student_class Inner join student on student_class.student_id = student.student_id "
                + "INNER JOIN class ON student_class.class_id = class.class_id "
                + "WHERE student_class.student_id='%s'", student_id);

        try {
           //Connection conn = DriverManager.getConnection(DB_URL, USER , PASS); //使用全域變數，不必再連一次
            //state = conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);
            rs = state.executeQuery(sql);
            if (rs.next() == false) {
                return "查無資料!";
            } else {
                msg = String.format("學號:%s\t姓名:%s\n", rs.getString("student_id"), rs.getString("name"));
                report += msg;
                report += "課名\t學分\t分數\n";
                report += "-----------------------------------------------------------\n";
                do {
                    msg = String.format("%s\t%s\t%s\n", rs.getString("class_name"), rs.getString("credit"), rs.getString("score"));
                    report += msg;
                } while (rs.next());
                report += "--------------------------------------------------------------\n";

                String sql2 = String.format("SELECT COUNT(*) as  修課數, SUM(credit) as  總學分, AVG(score) as  平均  "
                        + "FROM student_class "
                        + "INNER JOIN class on student_class.class_id = class.class_id "
                        + "WHERE student_id='%s'", student_id);
                rs = state.executeQuery(sql2);
                rs.first();

                msg = String.format("修課數:%s\t 總學分:%s\t 平均成績:%s\n", rs.getString("修課數"), rs.getString("總學分"), rs.getString("平均"));
                report += msg;
            }
        } catch (SQLException ex) {
            System.out.println("資料庫操作異常:" + ex.toString());
        }
        return report;
    }
    //導師的導生資料列印
    public static String reportTeacherStud(String teacher_id) {

        String msg;
        String report = "";
        ResultSet rs;

        String sql = String.format("SELECT student_advisor.student_id, student.name "
                + "FROM    student_advisor    INNER JOIN "
                + "student ON student_advisor.student_id = student.student_id "
                + "WHERE student_advisor.teacher_id = '%s'", teacher_id);
        String sql_teacher = String.format("SELECT name, room, dept "
                + "FROM    teacher "
                + "WHERE teacher_id = '%s'", teacher_id);

        try {
            rs = state.executeQuery(sql_teacher);
            if (rs.next() == false) {
                return "查無資料!";
            } else {
                rs.first();
            msg = String.format("教師:%s\t 研究室:%s\t 部門:%s\n", rs.getString("name"), rs.getString("room"), rs.getString("dept"));
                report += msg;

                rs = state.executeQuery(sql);
                rs.first();
                report += "學號\t 姓名\n";
                report += "--------------------------------------------------------------\n";
                do {
                    msg = String.format("%s\t%s\n", rs.getString("student_id"), rs.getString("name"));
                    report += msg;
                } while (rs.next());
                report += "--------------------------------------------------------------\n";
            }
        } catch (SQLException ex) {
            System.out.println("資料庫操作異常:" + ex.toString());
        }
        return report;
    }
     //列印所有學生與其導師資料
    public static String reportStudAdvisor() {

        String msg;
        String report = "";
        ResultSet rs;
        String sql = String.format("SELECT student.student_id AS '學號', student.name AS '姓名', teacher.Name AS '老師'\n"
                + "FROM student_advisor\n"
                + "INNER JOIN student ON student_advisor.Student_id = student.student_id \n"
                + "INNER JOIN teacher ON student_advisor.Teacher_id = teacher.Teacher_id;");

        try {
            rs = state.executeQuery(sql);
            rs.first();
            report += "學號\t 姓名\t 導師\n";
```

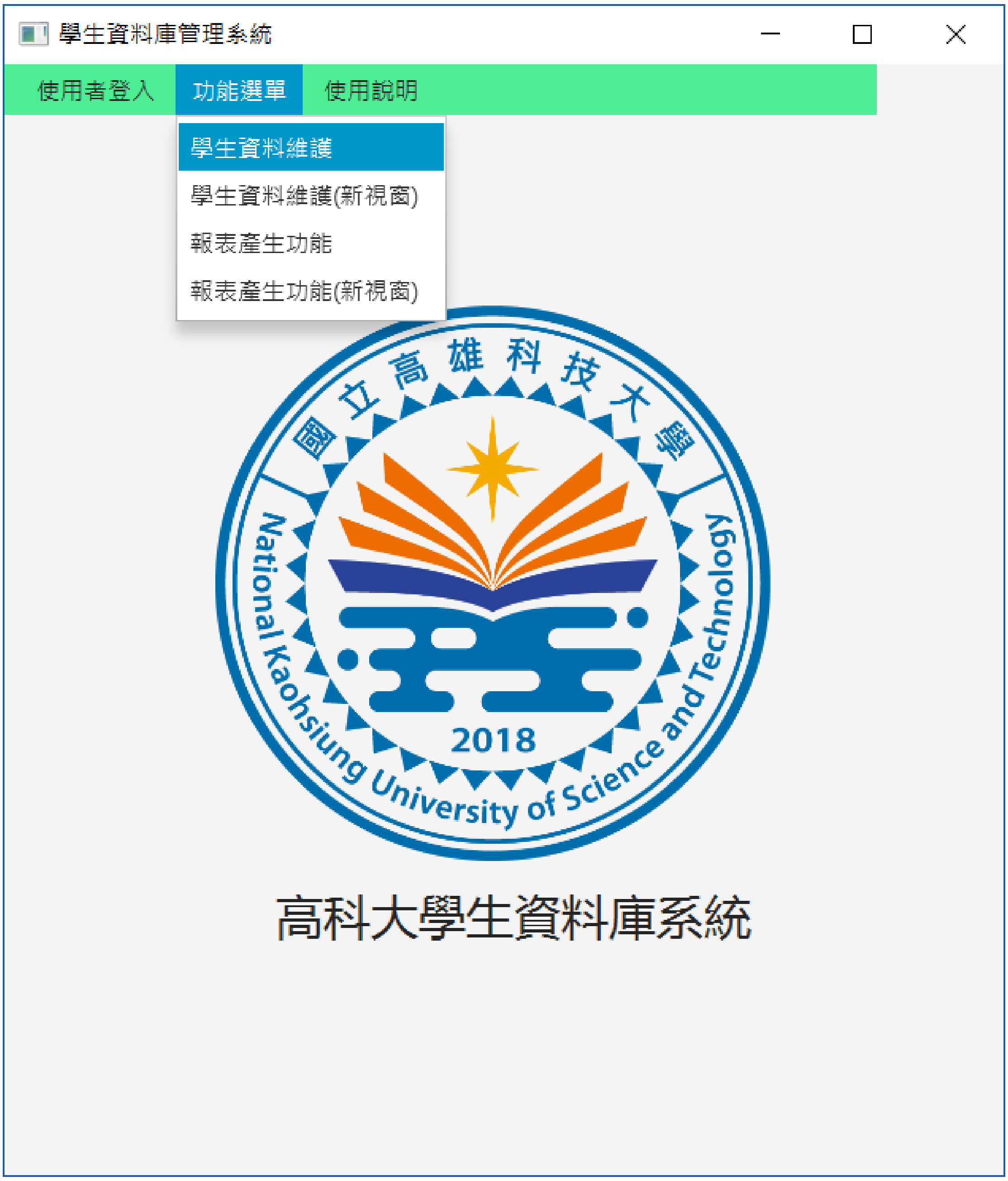
```
            report += "------------------------------------------------------------\n";
            do {
                msg = String.format("%s\t%s\t%s\n", rs.getString("student.student_id"), rs.getString("姓名"), rs.getString("老師"));
                report += msg;
            } while (rs.next());
            report += "------------------------------------------------------------\n";

        } catch (SQLException ex) {
            System.out.println("資料庫操作異常:" + ex.toString());
        }
        return report;
    }
}
```

myTextArea.css

```
.text-area {
    -fx-font-family: "Microsoft JhengHei";
    -fx-font-size: 14;
    -fx-text-fill: #ffffff;
    -fx-display-caret: true;
}

.text-area .scroll-pane .content, .text-area .content{
    -fx-background-color: black;
}

/*
    -fx-font-family: "標楷體";
    -fx-font-family: "KaiTi";
      -fx-font-family: "Microsoft JhengHei";
    -fx-font-family: "Courier New";
*/
```

Lab4: 整合畫面

StudentManagement.fxml



StudentManagementController.java

```
package javafx_studentdb;

import java.io.IOException;
import java.net.URL;
import java.util.ResourceBundle;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.layout.Pane;

import javafx.stage.Stage;

public class StudentManagementController implements Initializable {

    @FXML
    private Pane pane_container;

    @Override
    public void initialize(URL url, ResourceBundle rb) {
    }

    @FXML
    private void gotoReport(ActionEvent event) throws IOException {
        //此處用 nodes 命名
        Parent nodes = FXMLLoader.load(getClass().getResource("GenerateReport.fxml"));
        //取得 pane_container 內所有的元件，並重新設定為新的 nodes
        pane_container.getChildren().setAll(nodes);
        //pane_container.getChildren().set(0,root); //另一種寫法
    }

    @FXML
    private void gotoReportStage(ActionEvent event) throws IOException {
        //產生新的容器，裝一大堆的元件
        Parent root = FXMLLoader.load(getClass().getResource("GenerateReport.fxml"));
        //產生新的布景
        Scene scene = new Scene(root);
        //產生新的一個視窗
        Stage stage = new Stage();
        stage.setScene(scene);
        stage.setTitle("報表輸出系統");
        stage.show();
    }

    @FXML
    private void gotoCrud(ActionEvent event) {
        try {
            Parent root = FXMLLoader.load(getClass().getResource("CrudStudent.fxml"));
            pane_container.getChildren().setAll(root); //設定 pane 窗格底下所有元件
        } catch (IOException ex) {
            System.out.println("發生異常:載入 CrudStudent.fxml");
        }
    }

    @FXML
    private void gotoCrudStage(ActionEvent event) {
        Parent root = null;
        try {
            root = FXMLLoader.load(getClass().getResource("CrudStudent.fxml"));
        } catch (IOException ex) {
            System.out.println("發生異常:載入 CrudStudent.fxml");
        }
        Scene scene = new Scene(root);
        Stage stage = new Stage();
        stage.setScene(scene);
        stage.setTitle("學生資料維護系統");
        stage.show();
    }



     @FXML
    private void gotoCrudTableView(ActionEvent event) {
        try {
            Parent root = FXMLLoader.load(getClass().getResource("CrudStudentTableView.fxml"));
            pane_container.getChildren().setAll(root);
        } catch (IOException ex) {
            System.out.println("發生異常:載入 CrudStudentTableView.fxml");
        }
    }


}
```

StudentManagementMain.java

```
package javafx_studentdb;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class StudentManagementMain extends Application {

    @Override
    public void start(Stage stage) throws Exception {
        Parent root = FXMLLoader.load(getClass().getResource("StudentManagement.fxml"));
        //各別去測試不同的模組功能
        //Parent root = FXMLLoader.load(getClass().getResource("CrudStudent.fxml"));
        //Parent root = FXMLLoader.load(getClass().getResource("GenerateReport.fxml"));
        //Parent root = FXMLLoader.load(getClass().getResource("CrudStudentTableView.fxml"));
        Scene scene = new Scene(root);

        stage.setScene(scene);
        stage.setTitle("學生資料庫管理系統");
        stage.show();
    }

    public static void main(String[] args) {
        launch(args);
        }


}
```