FPT Education
FPT UNIVERSITY

Java

JPA Application Development & CRUD Operations

Objectives

- Data Management
  - Implement CRUD (Create, Read, Update, Delete) operations to manage data effectively.
  - Utilize JPA to map Java objects to database tables, simplifying data persistence and retrieval.
- User Interface Development
  - Design and develop a user-friendly interface using JavaFX, enabling seamless interaction with the application.
  - Implement features such as form input validation, error handling, and feedback mechanisms to enhance user experience.

Student Management System

Build Simple Student Management System

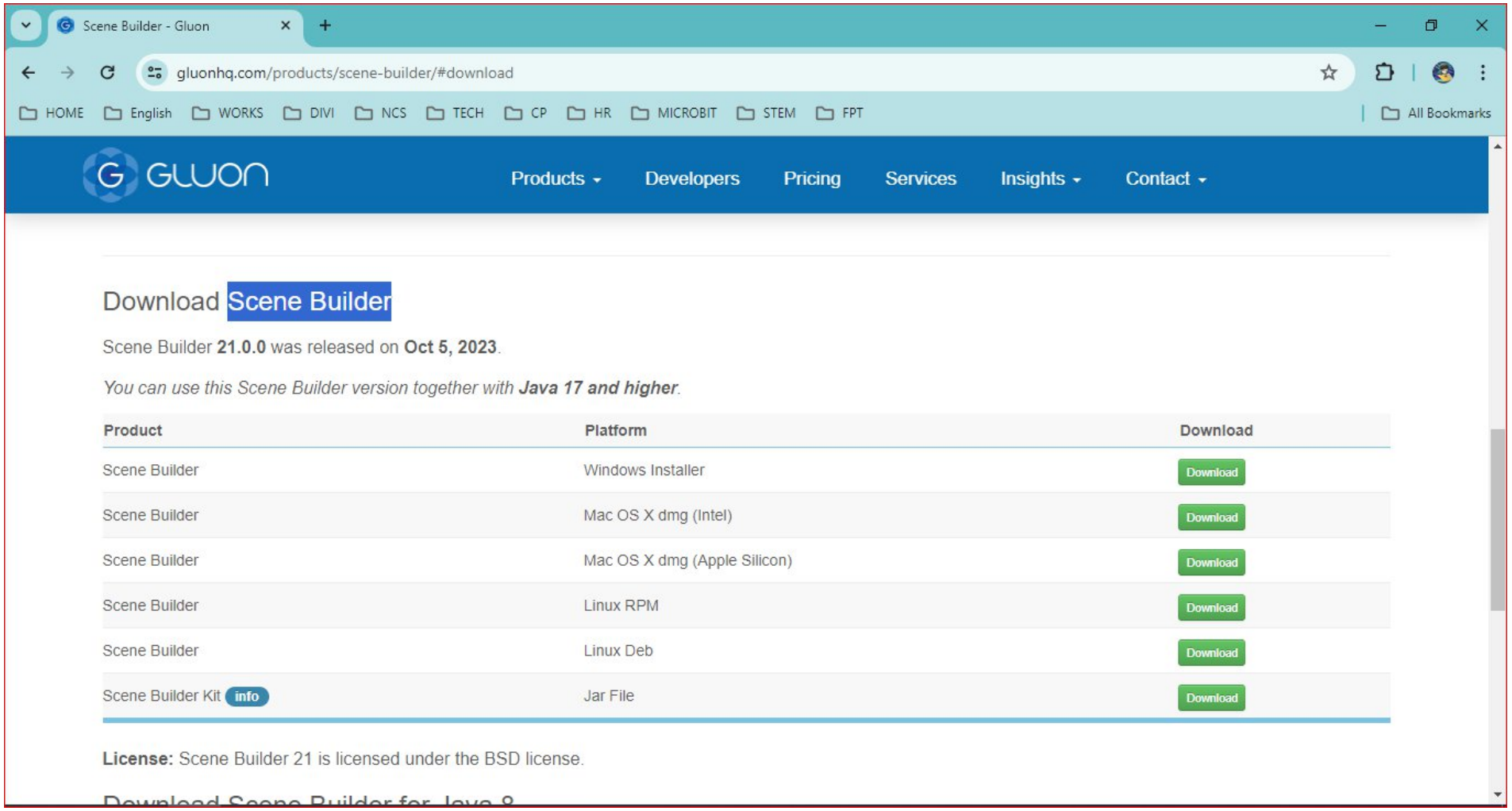
- Step 1. Create Project in IntelliJ
- Step 2. Setup Scene Builder
- Step 3. Build CRUD Application with Repository Pattern
- Step 4. Design GUI Application with JavaFX

FPT Education
FPT UNIVERSITY

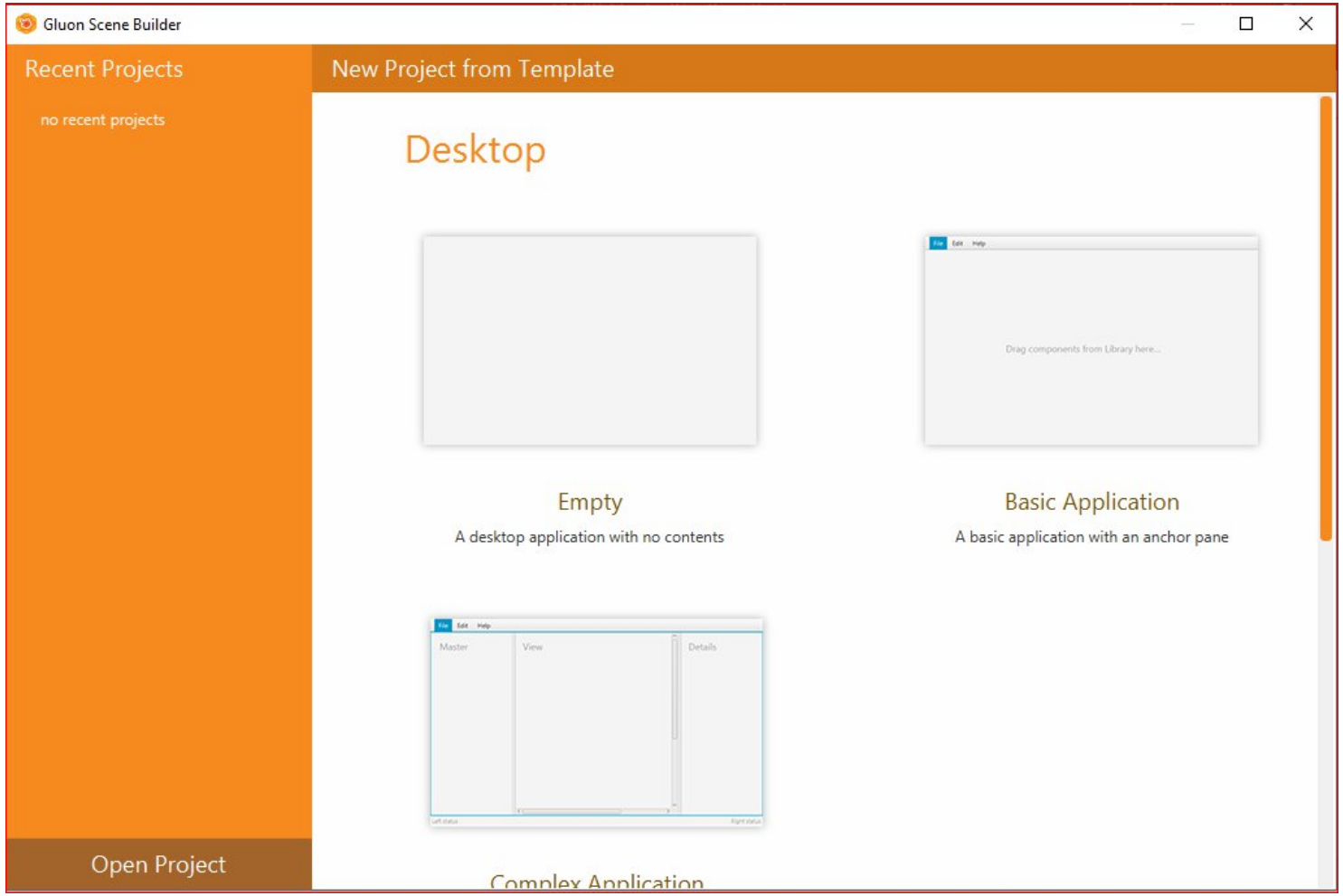
Java

Setup Scene Builder

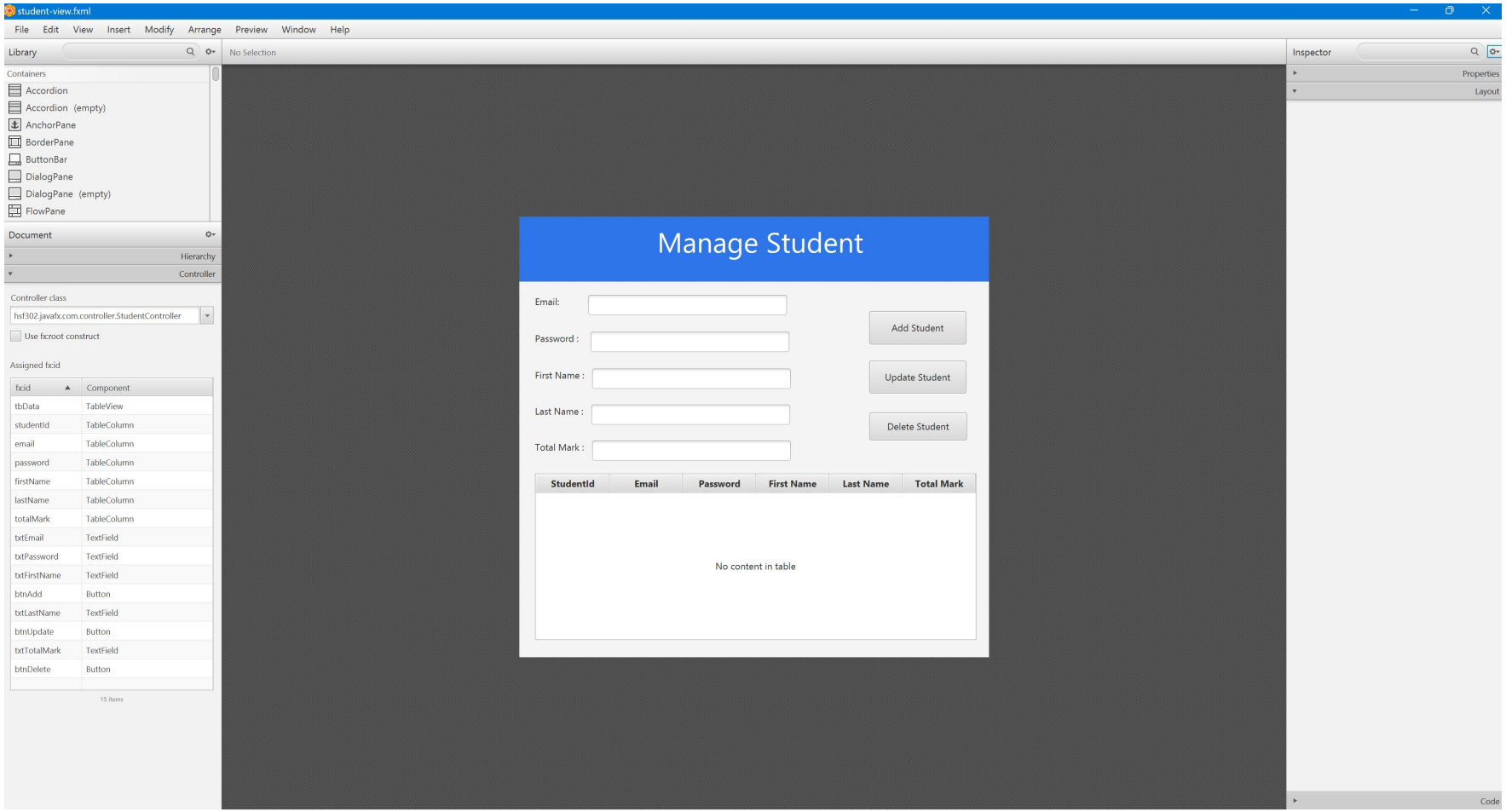
Go to website



2. Open Gluon Scene Builder



Open JavaFX File

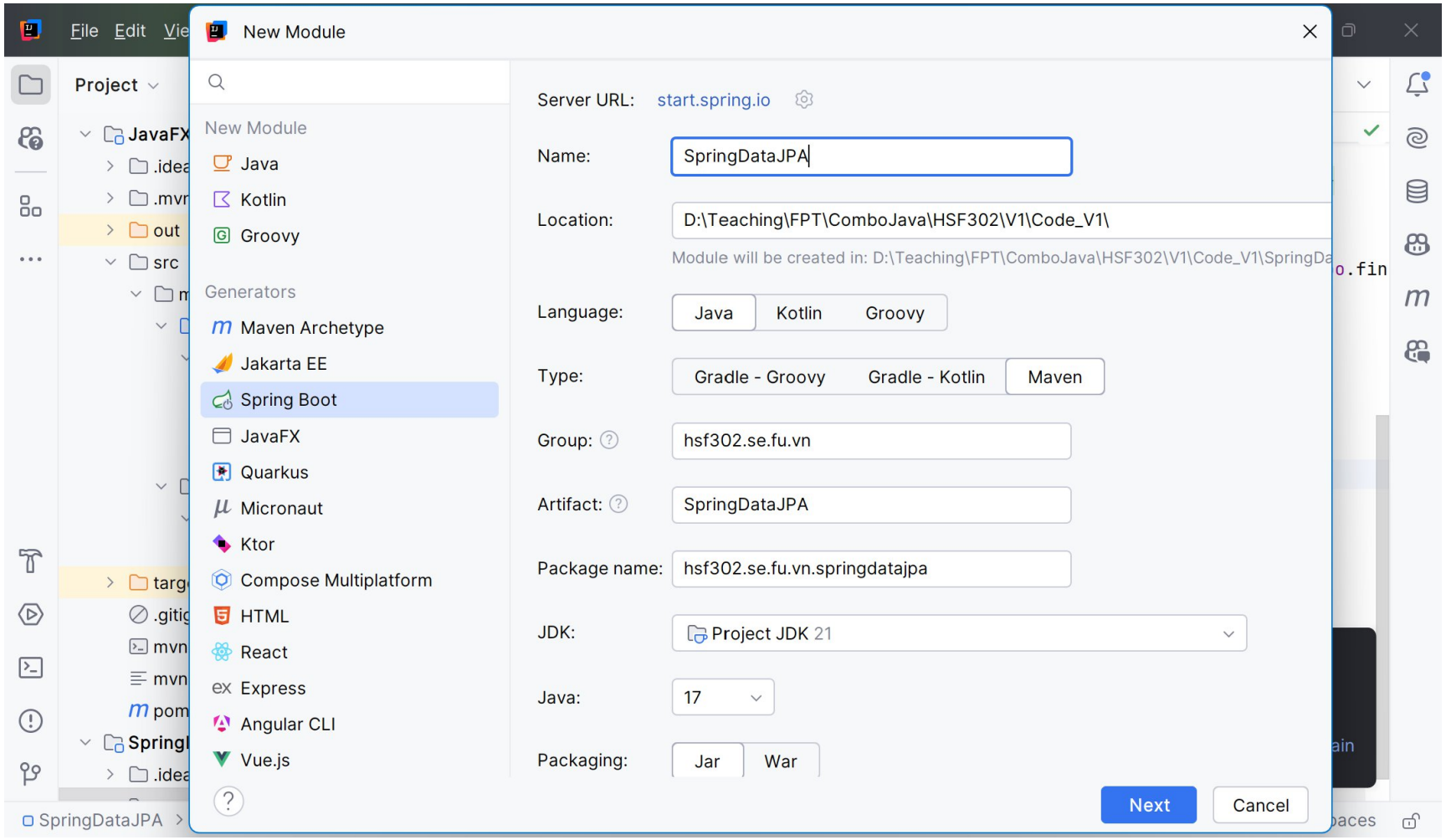


FPT Education
FPT UNIVERSITY

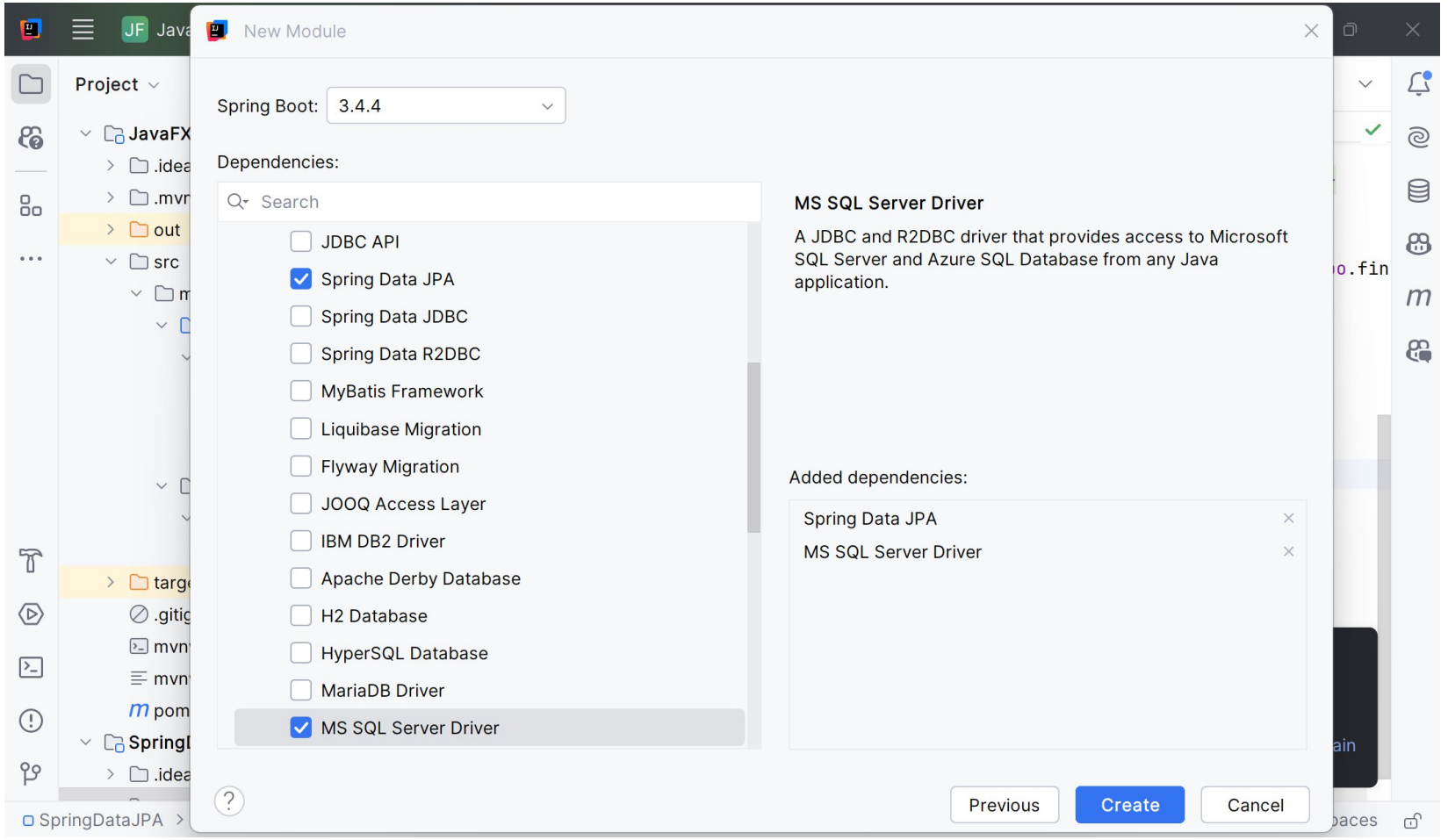
Java

Build CRUD Application with Repository Pattern

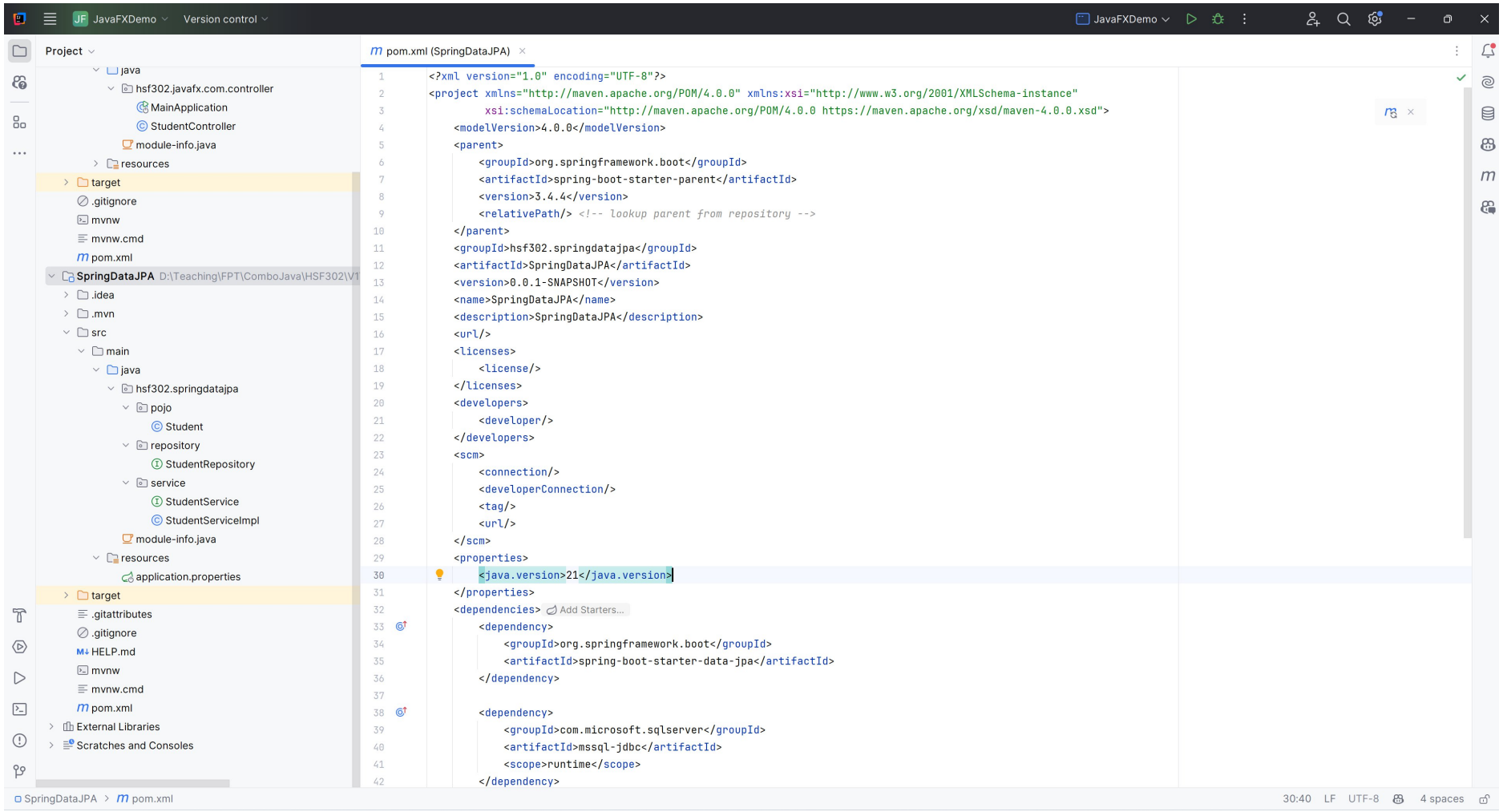
File -> New Module



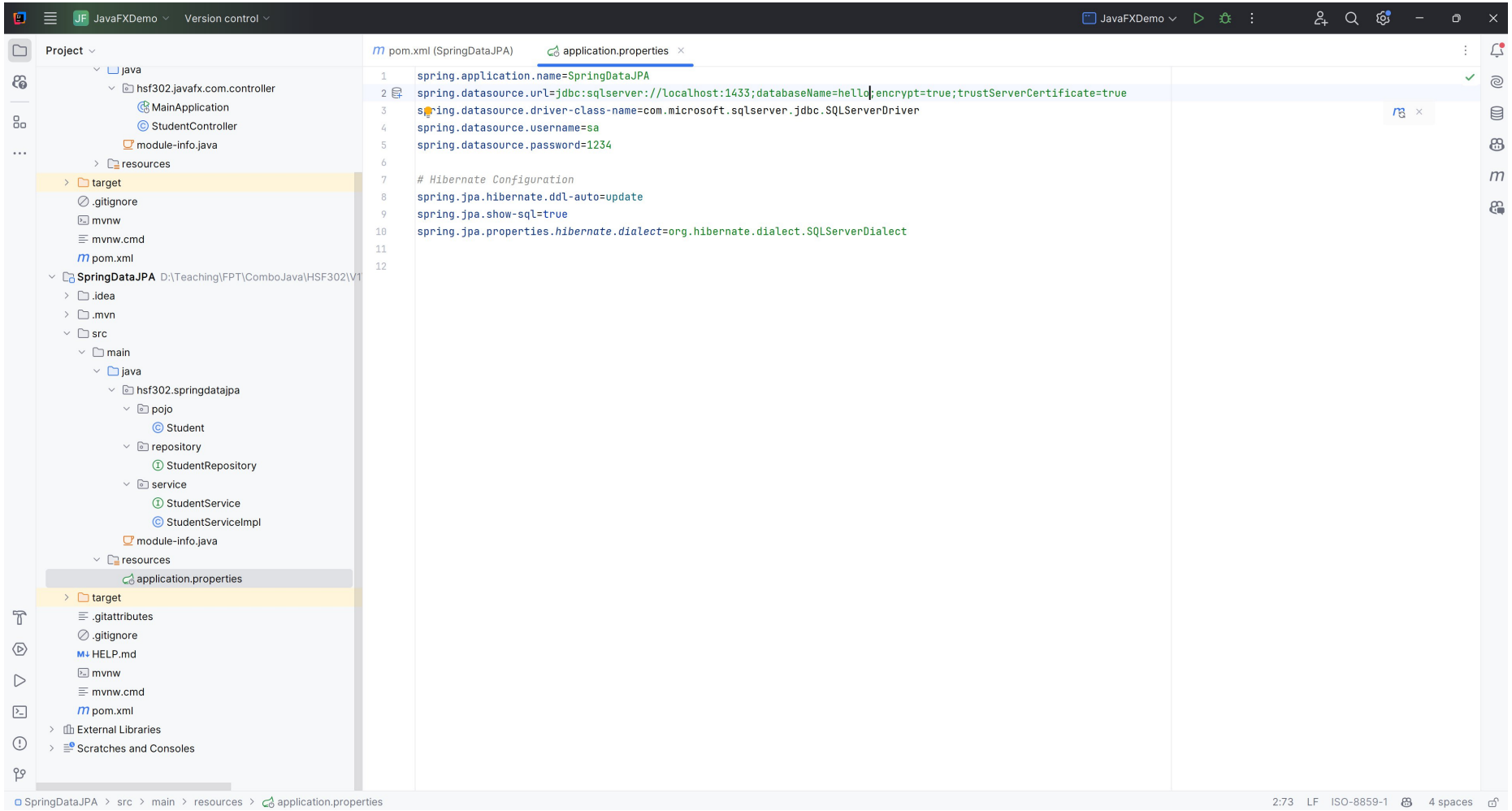
File -> New Module



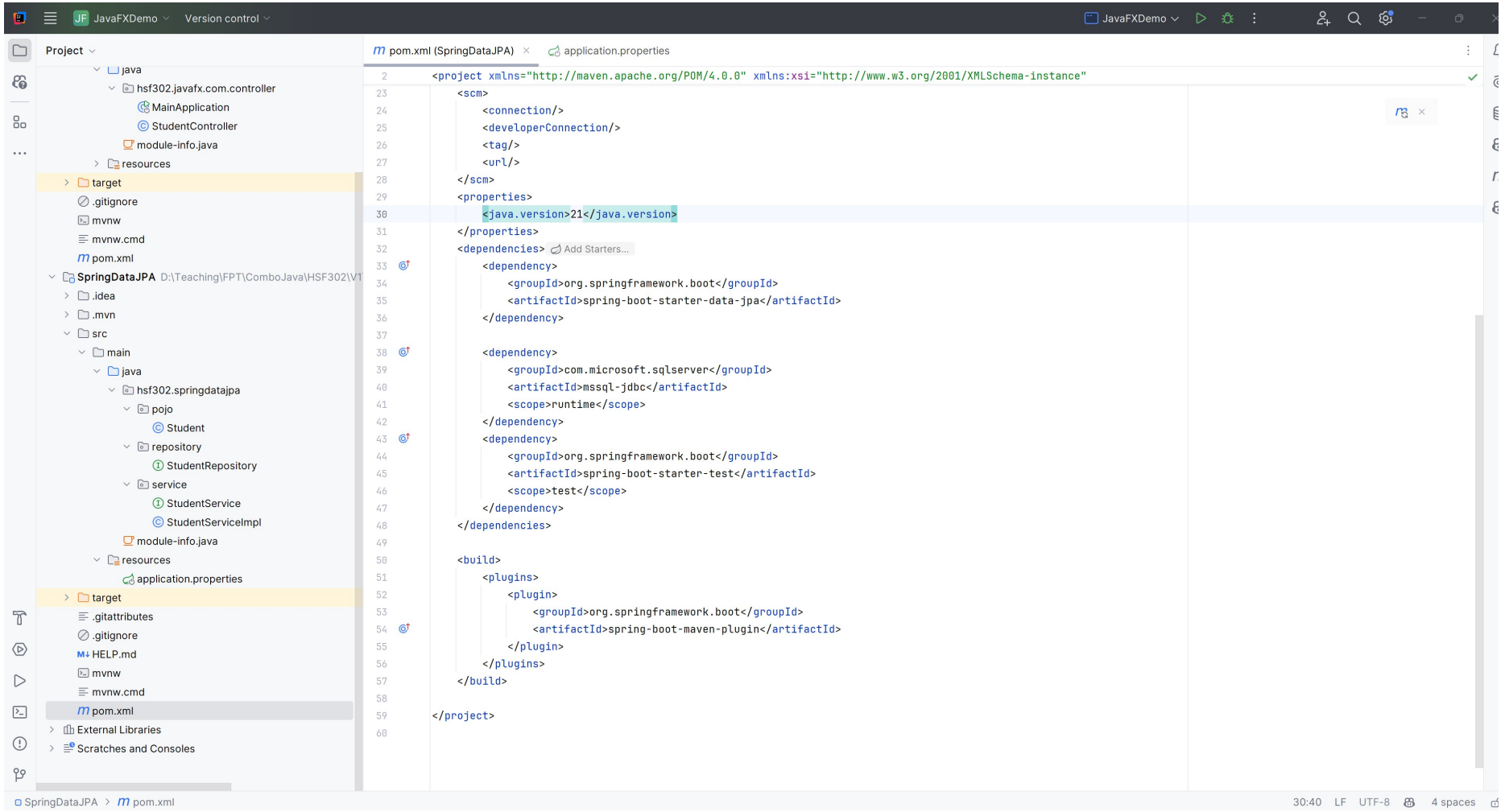
Create the Structure



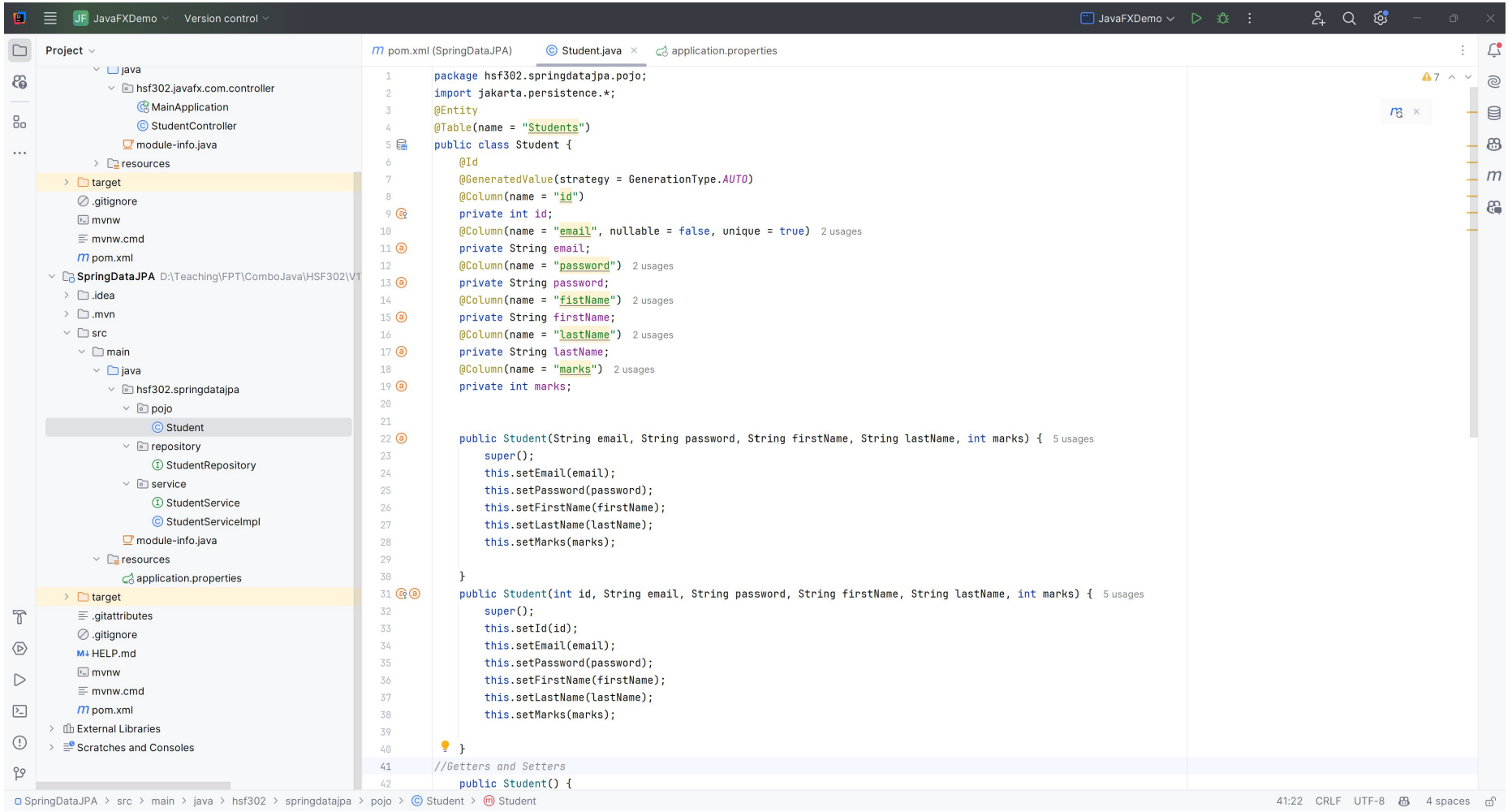
Update application.properties



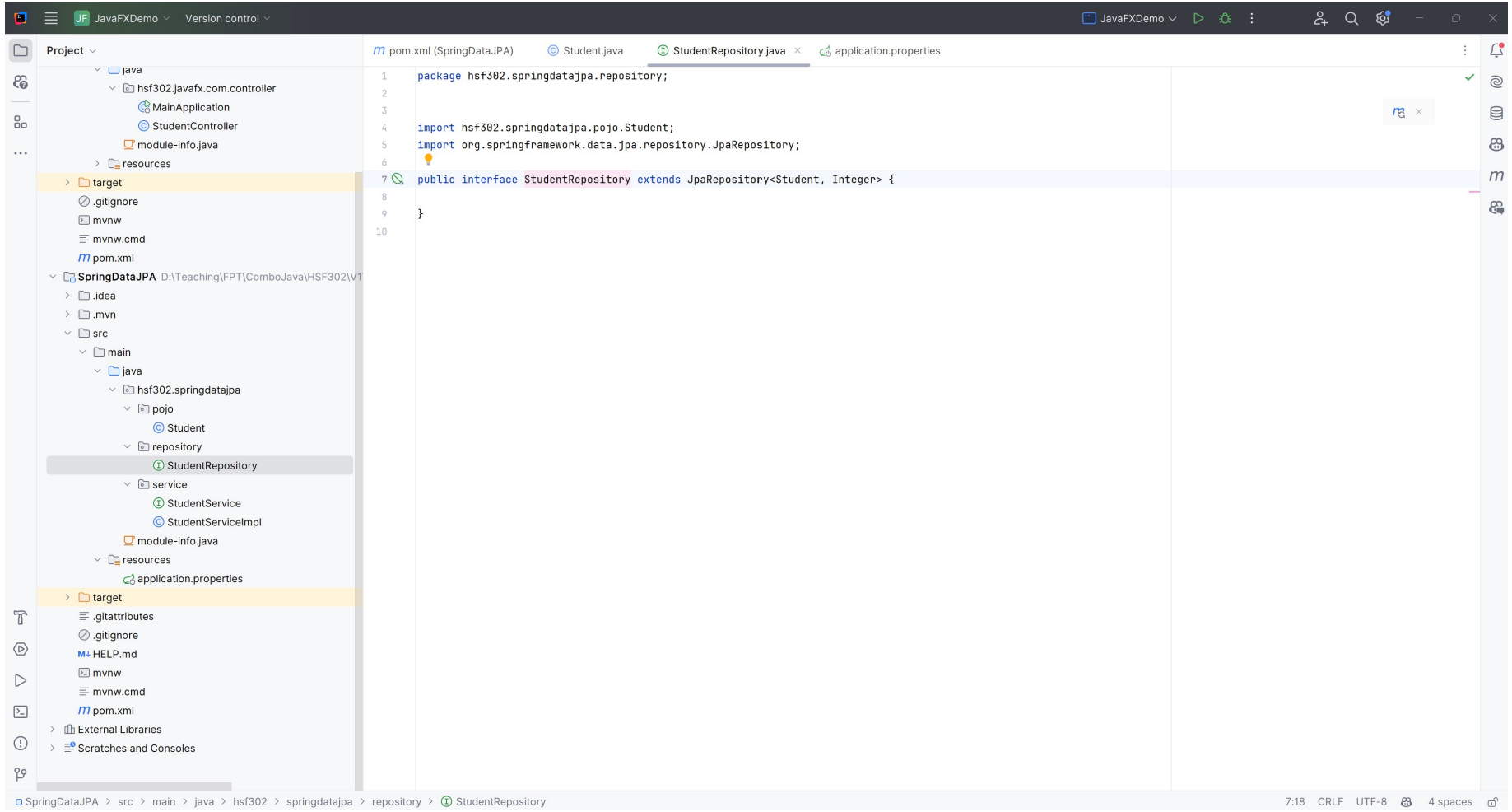
Update pom.xml



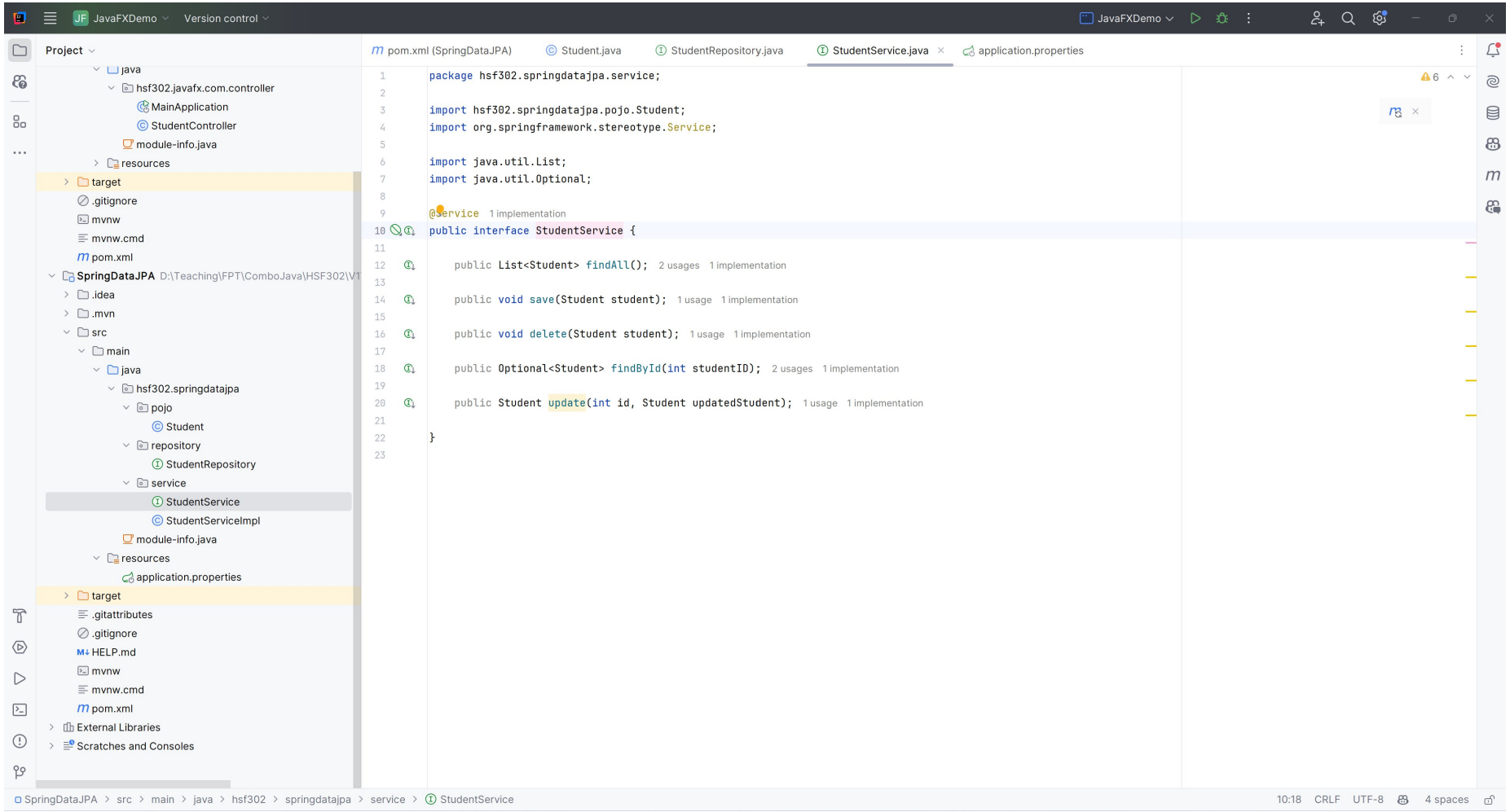
Create Student.java



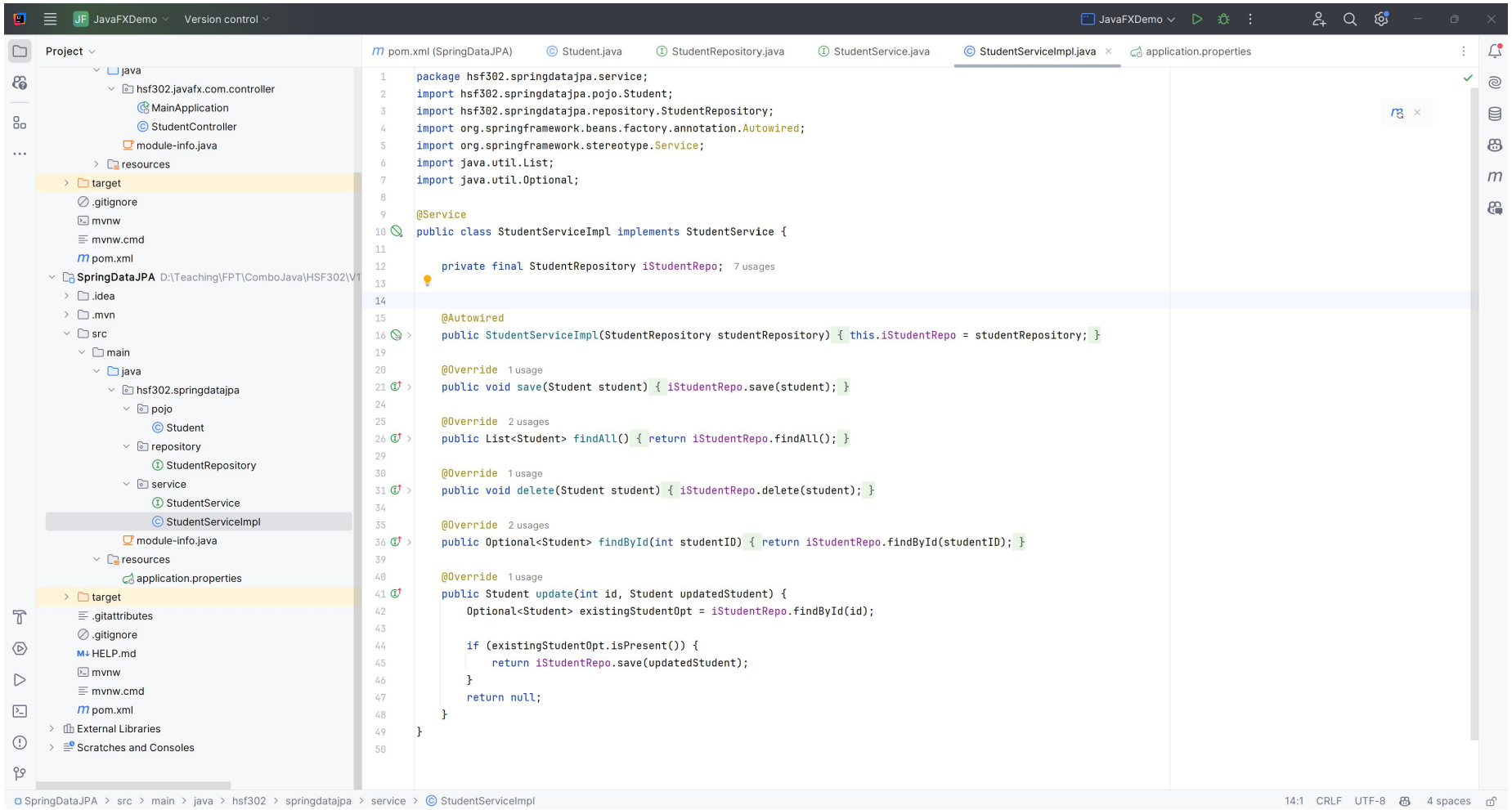
Create StudentRepository.java



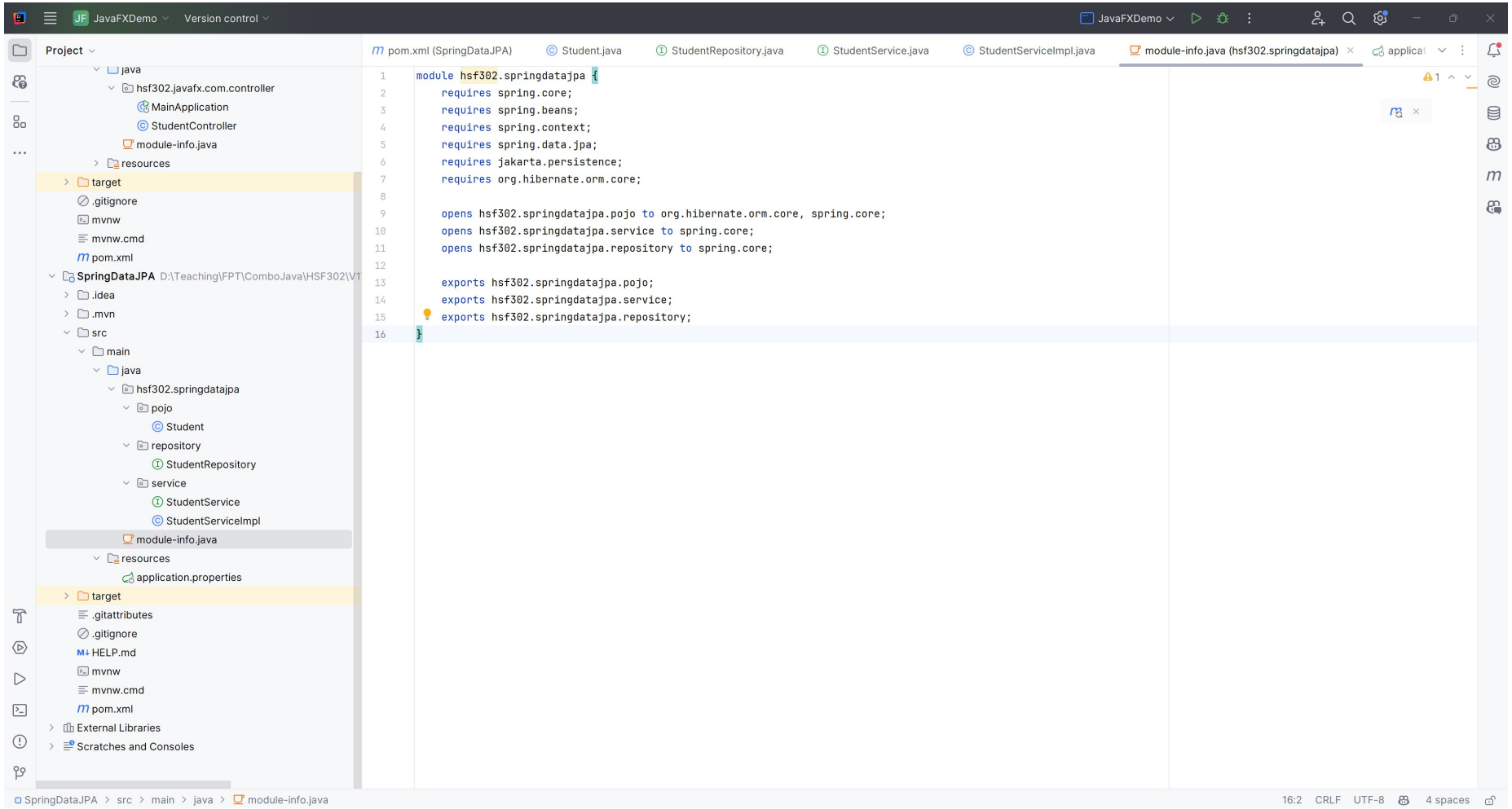
Create StudentService.java



Create StudentServiceImpl.java



Create module-info.java

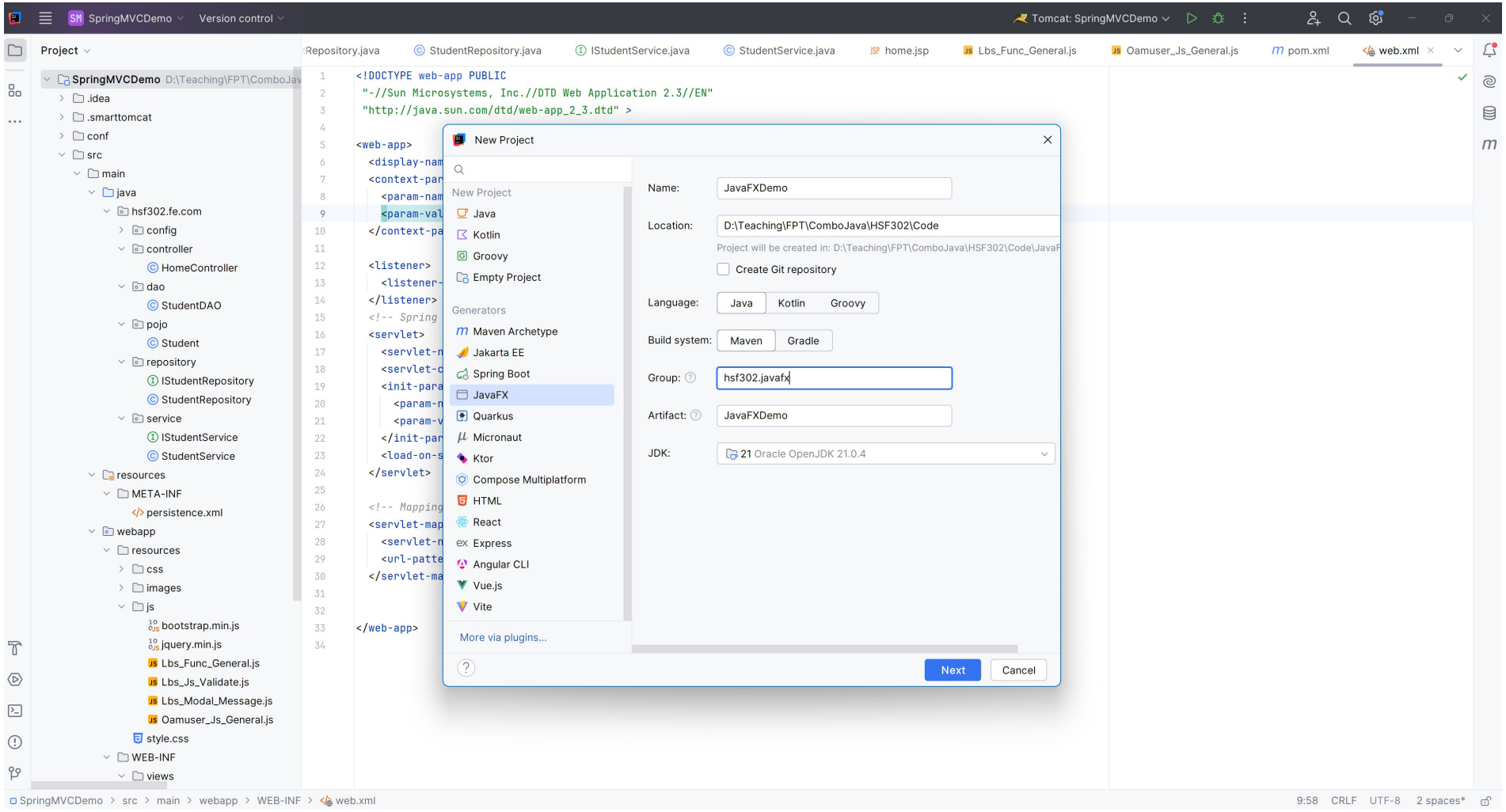


FPT Education
FPT UNIVERSITY

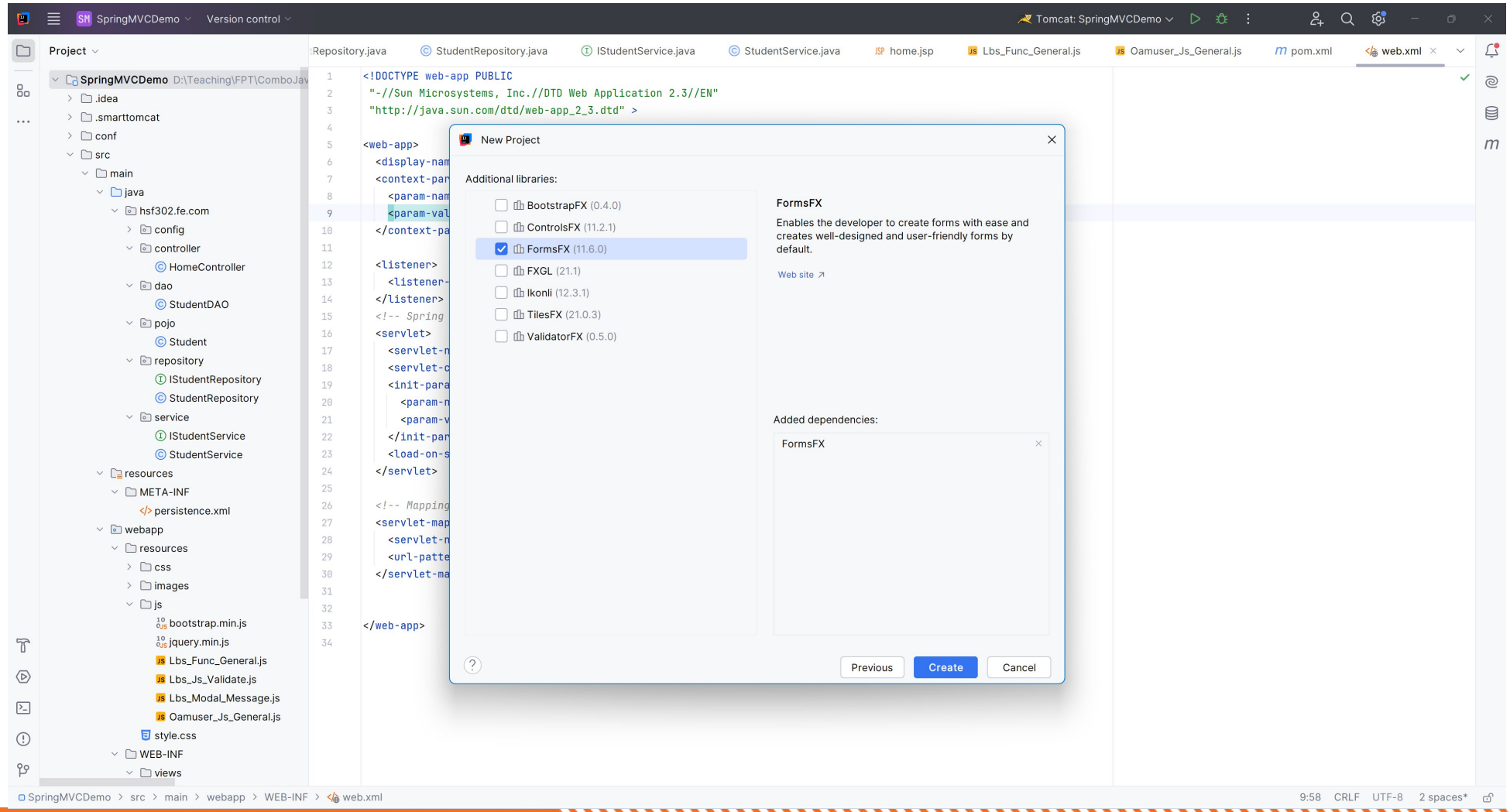
Java

Design GUI Application with JavaFX

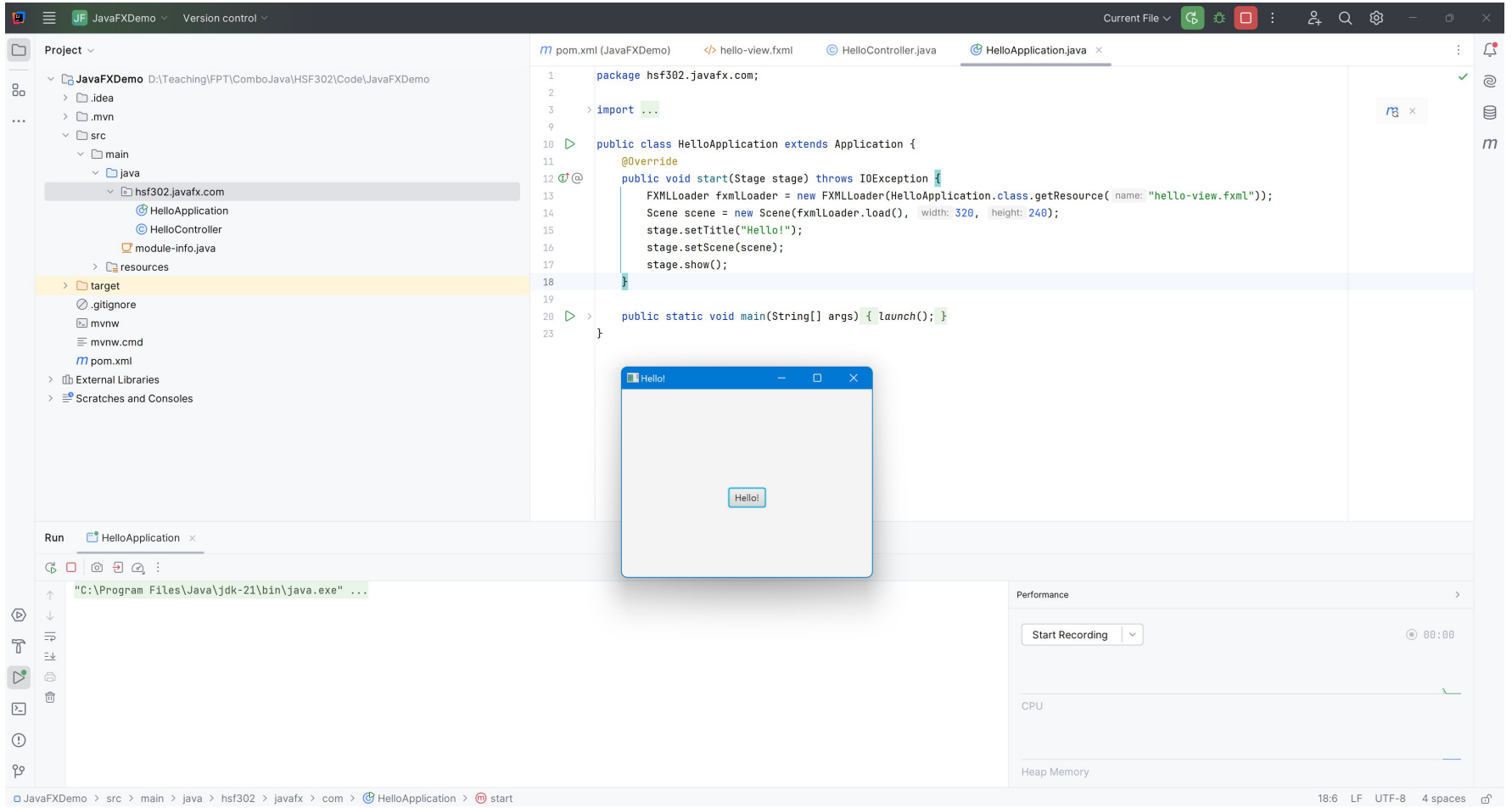
Open IntelliJ, File | New | Maven Project



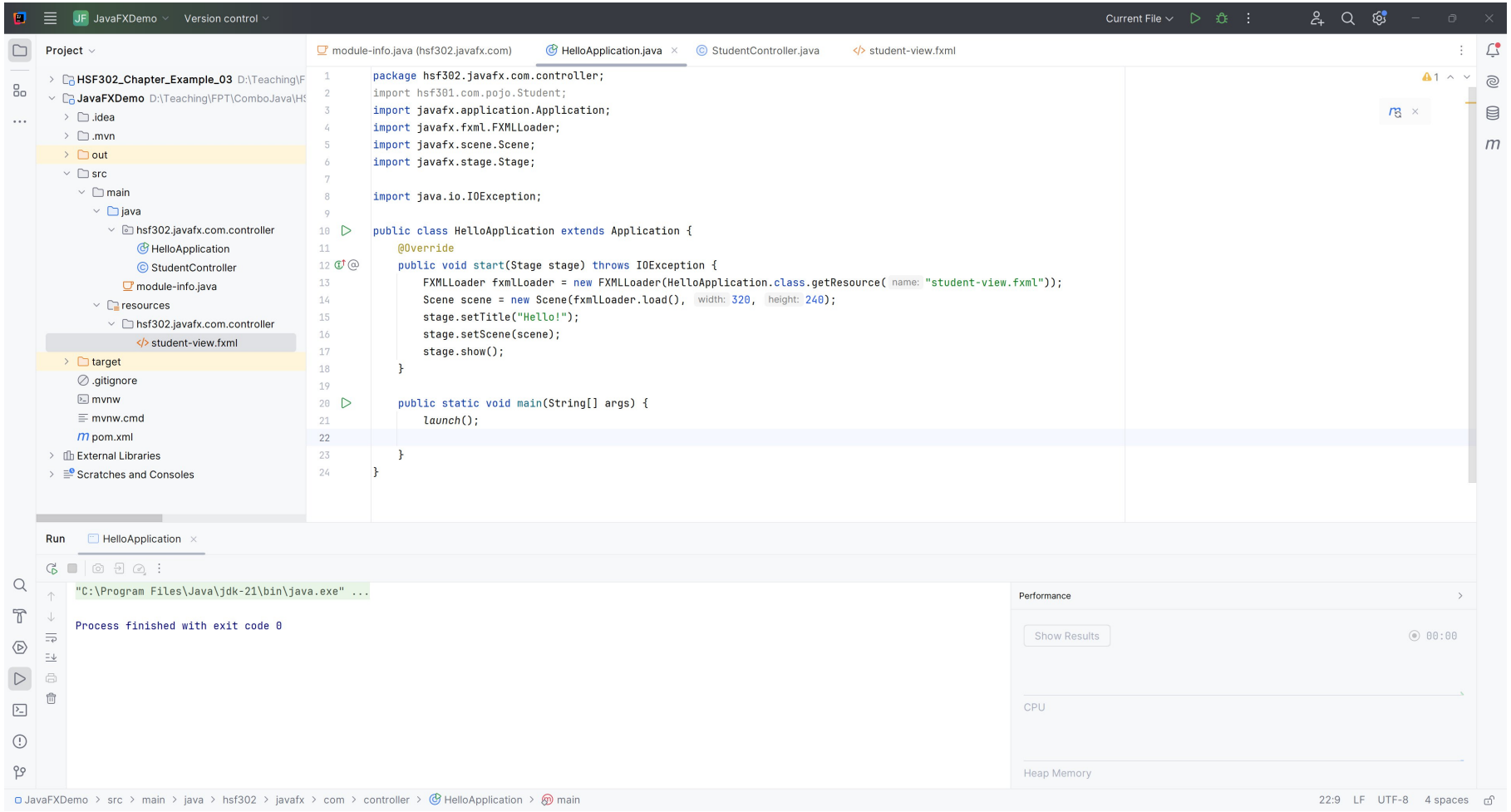
Check FormFX



Run Project

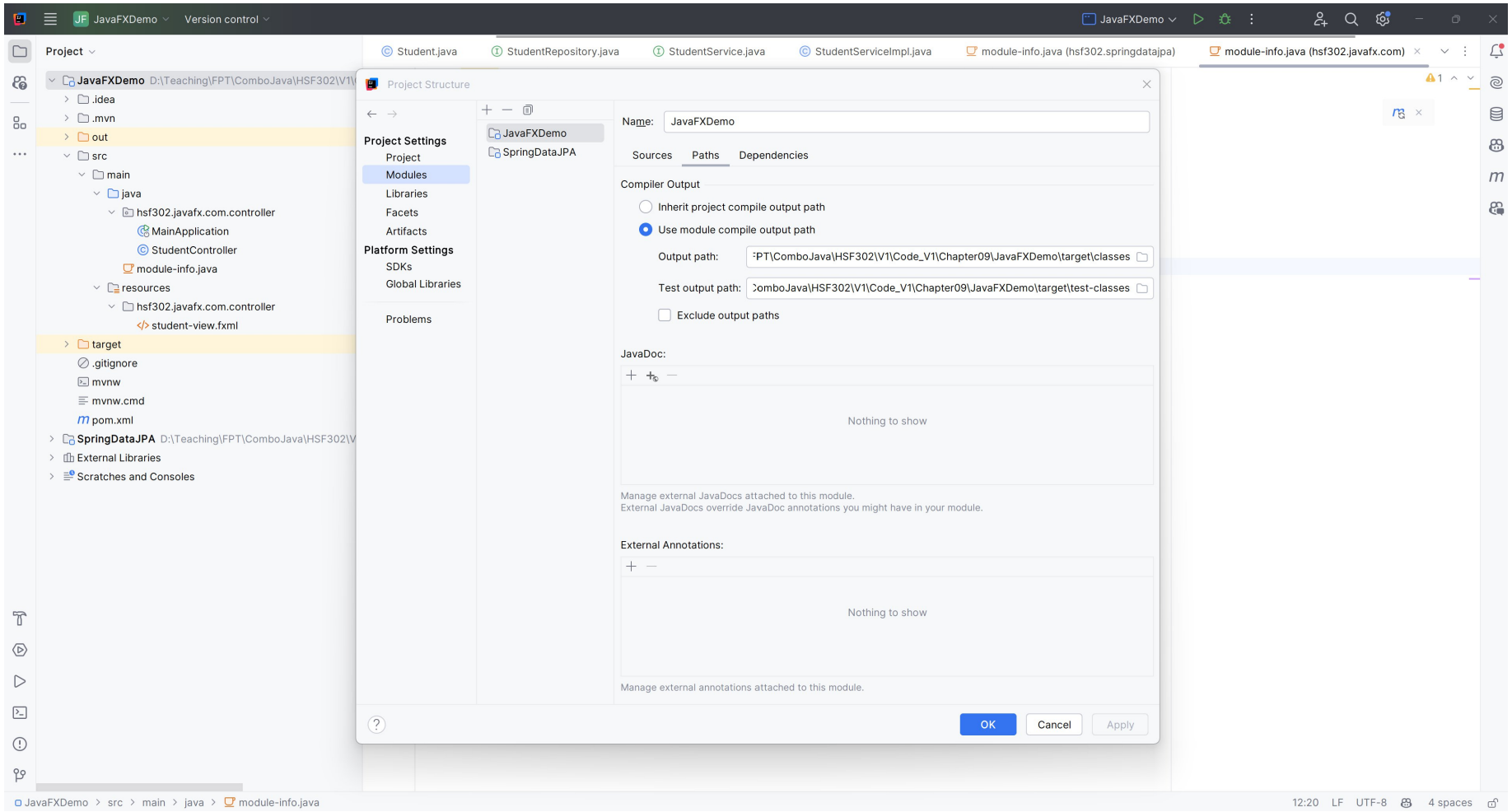

JavaFXDemo
Version control
Current File
Project
JavaFXDemo D:\Teaching\FPT\ComboJava\HSF302\Code\JavaFXDemo
.idea
.mvn
src
main
java
hsf302.javafx.com
HelloApplication
HelloController
module-info.java
resources
target
.gitignore
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
pom.xml (JavaFXDemo)
hello-view.fxml
HelloController.java
HelloApplication.java
package hsf302.javafx.com;
import ...
public class HelloApplication extends Application {
@Override
public void start(Stage stage) throws IOException {
FXMLLoader fxmlLoader = new FXMLLoader(HelloApplication.class.getResource(name: "hello-view.fxml"));
Scene scene = new Scene(fxmlLoader.load(), width: 320, height: 240);
stage.setTitle("Hello!");
stage.setScene(scene);
stage.show();
}
public static void main(String[] args) { launch(); }
}
Hello!
Hello!
Run
HelloApplication
"C:\Program Files\Java\jdk-21\bin\java.exe" ...
Performance
Start Recording
00:00
CPU
Heap Memory
JavaFXDemo > src > main > java > hsf302 > javafx > com > HelloApplication > start
18:6 LF UTF-8 4 spaces

Build the Architecture

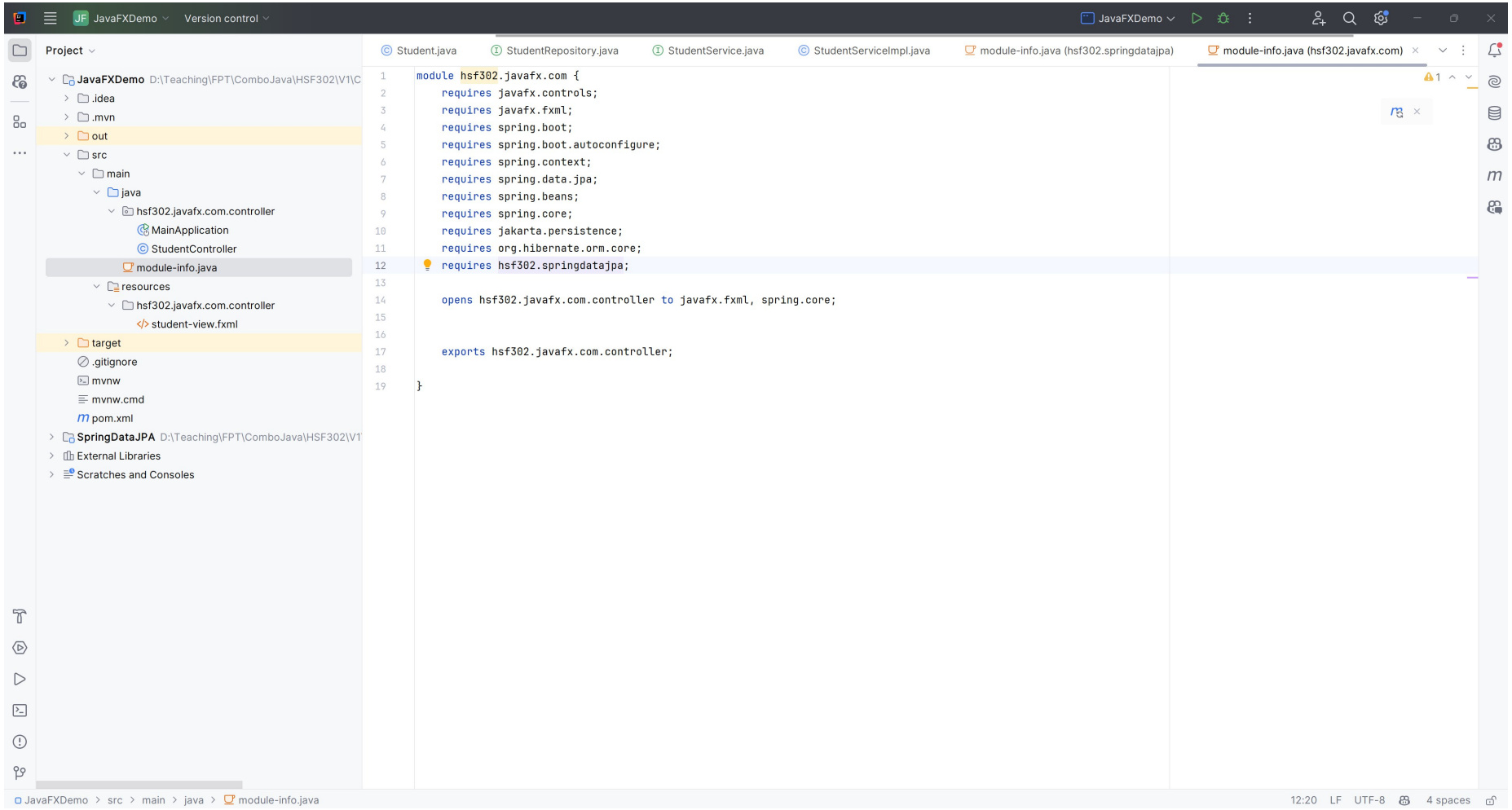


FPT Education
FPT UNIVERSITY

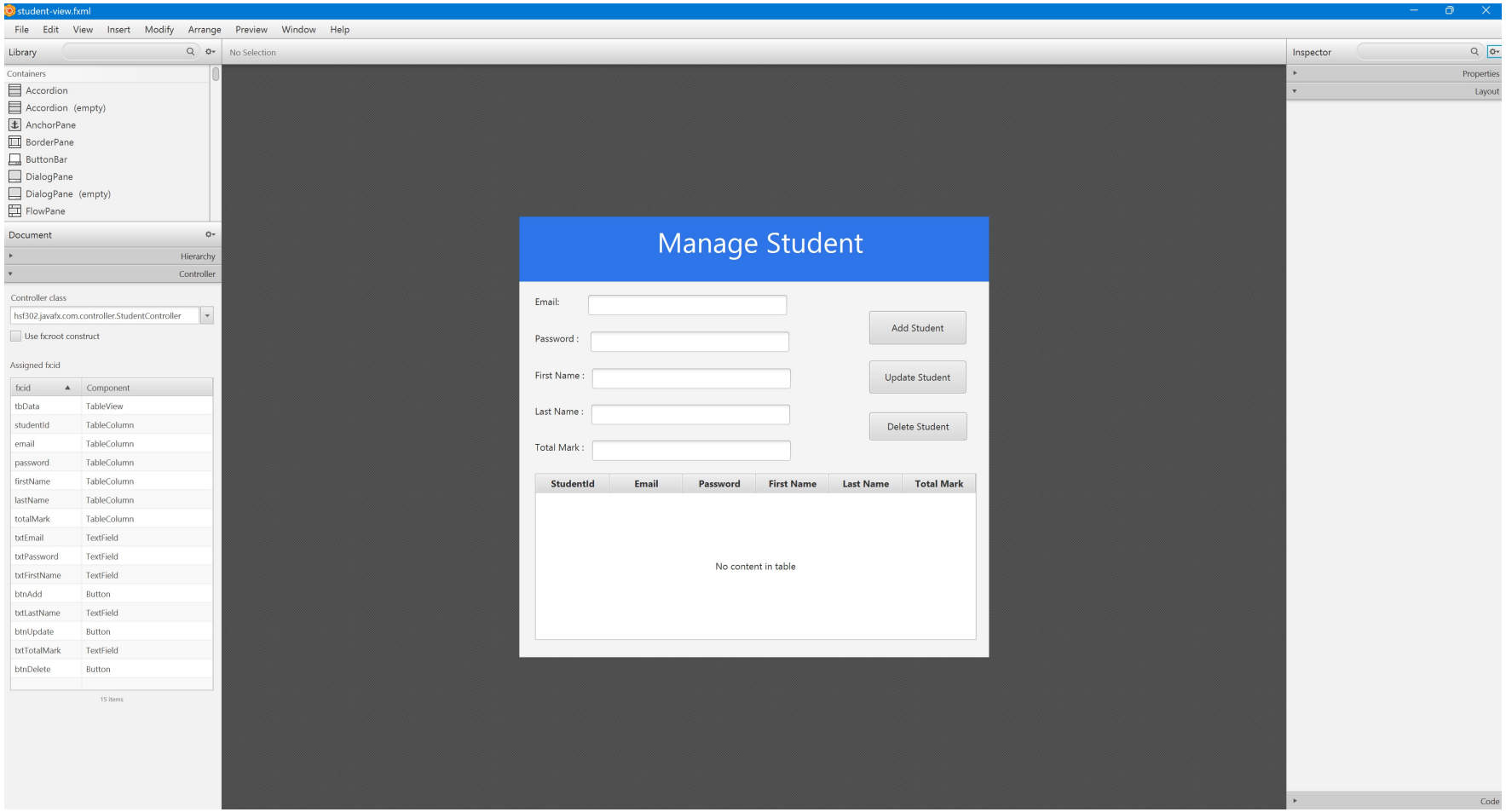
Add Reference Another Project



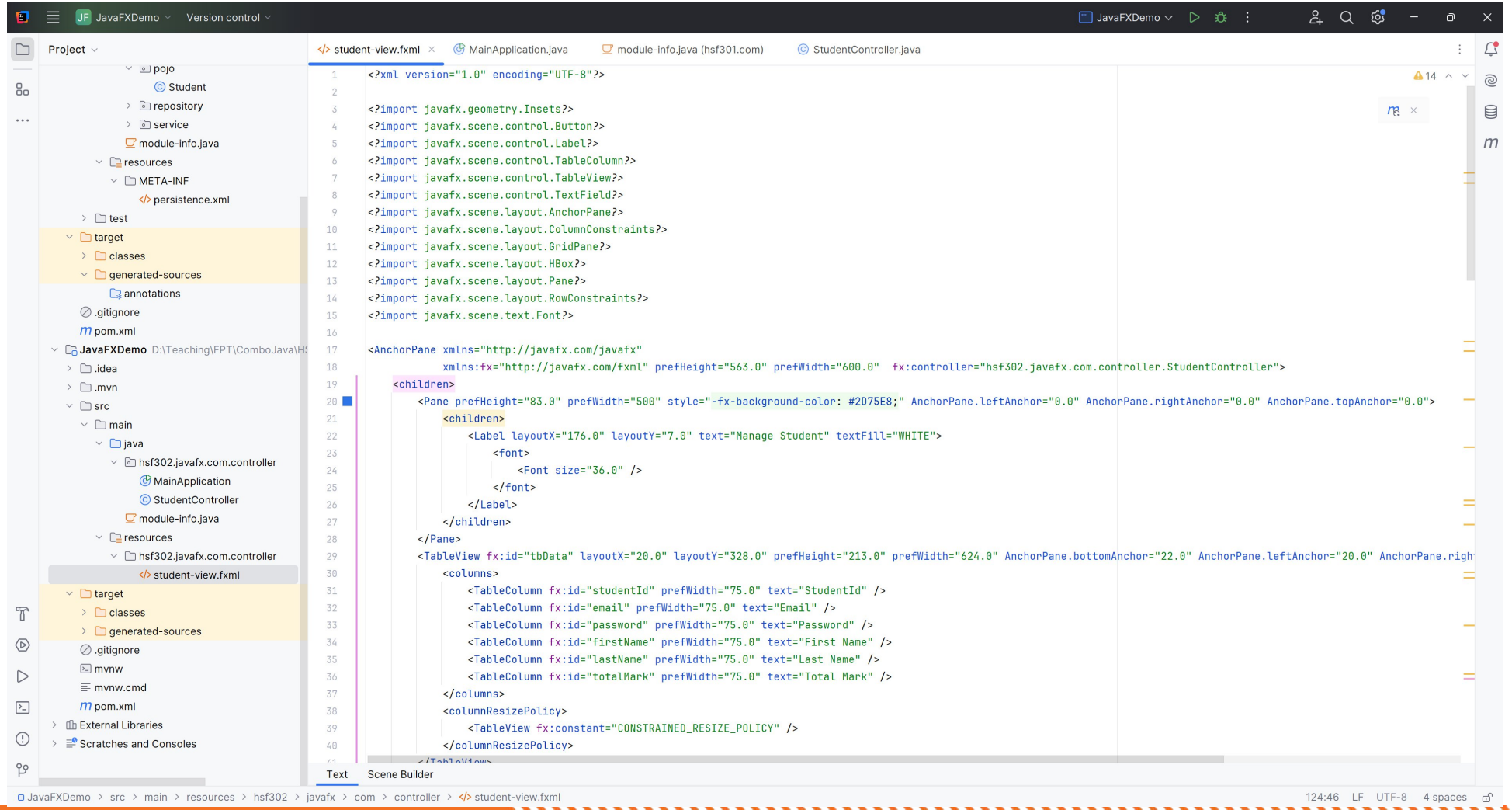
Edit module-info.java in JavaFXDemo



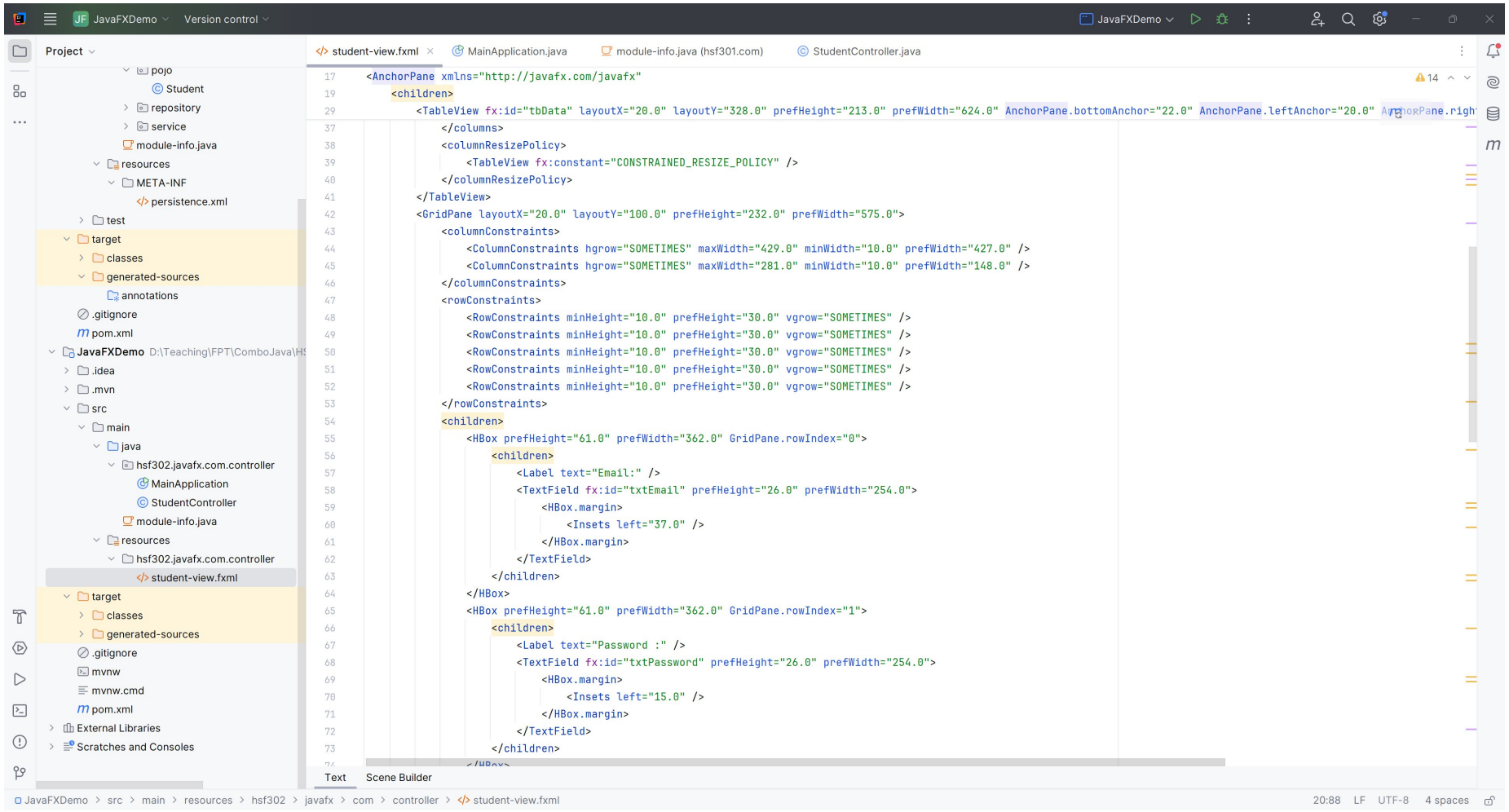
Open Student-view.fxml in SceneBuilder



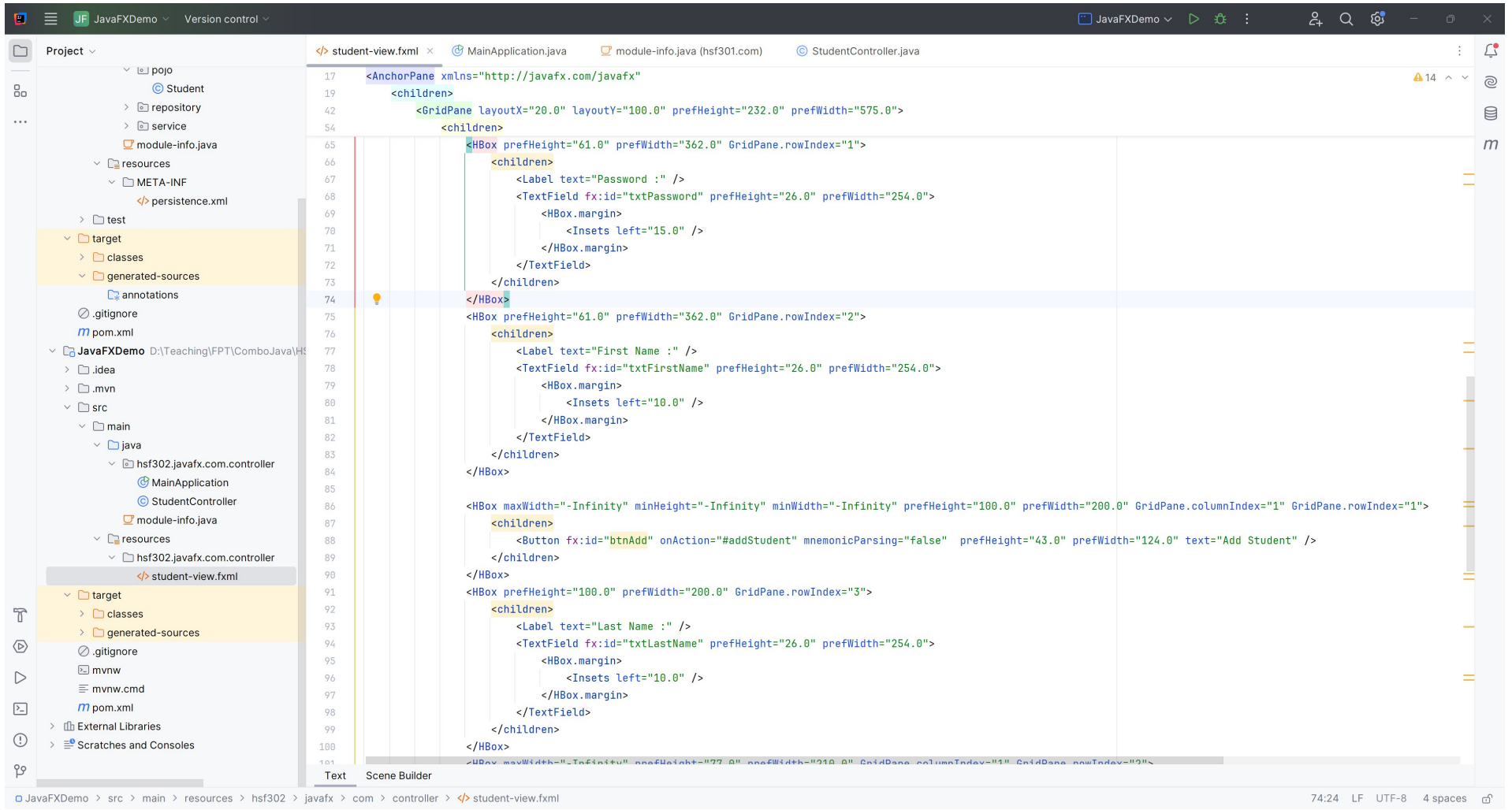
Edit student-view.fxml



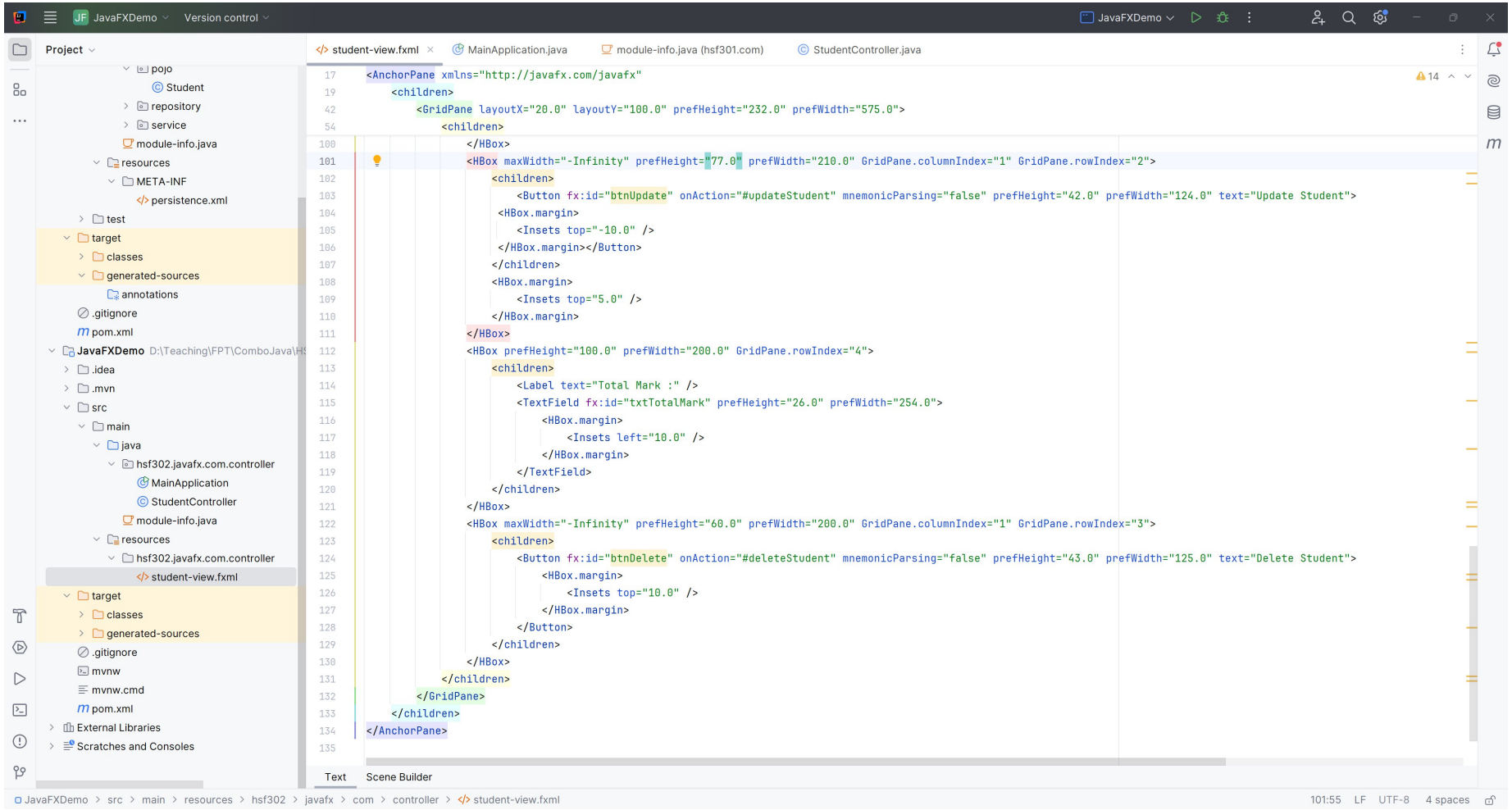
Edit student-view.fxml



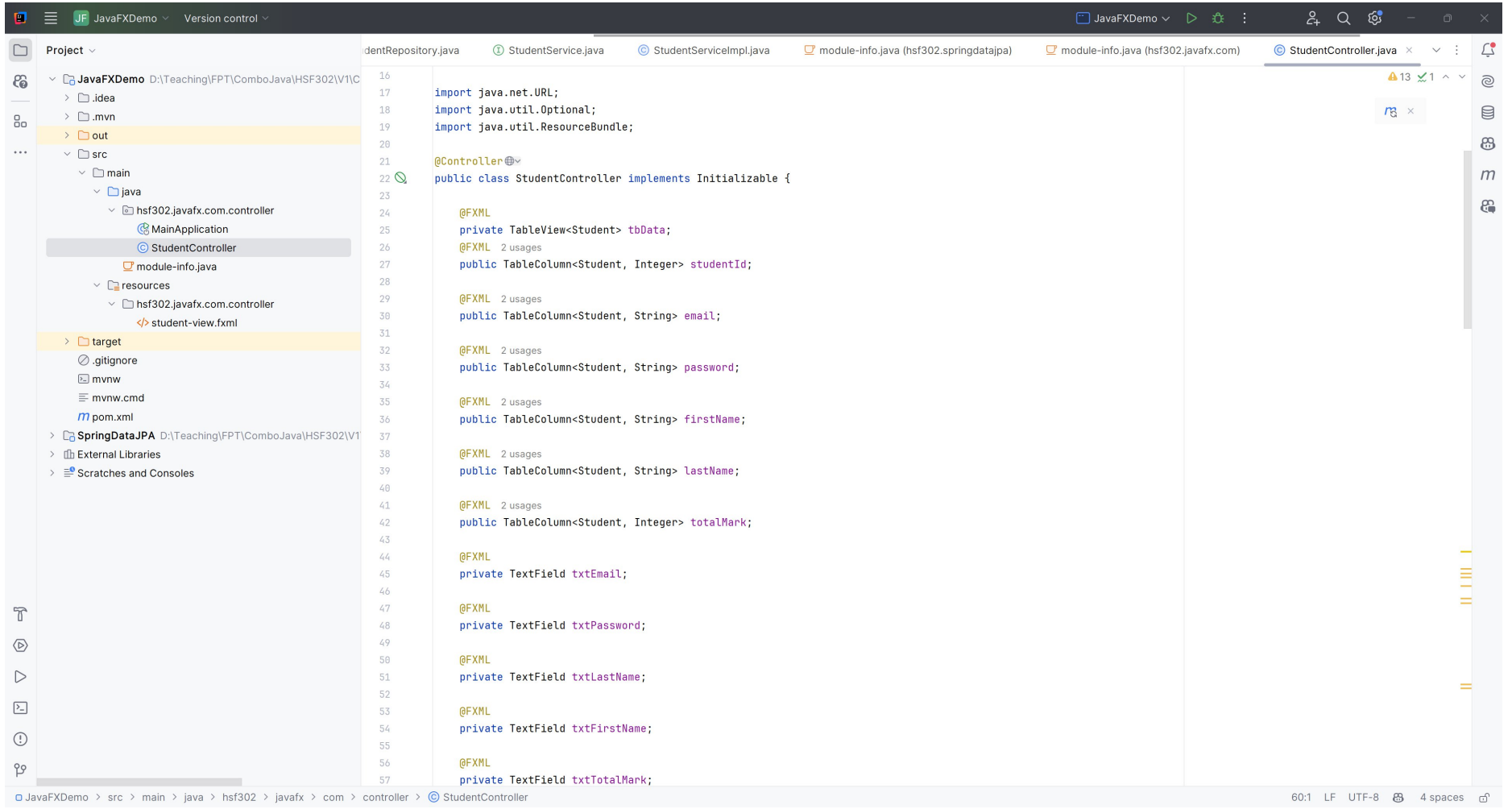
Edit student-view.fxml



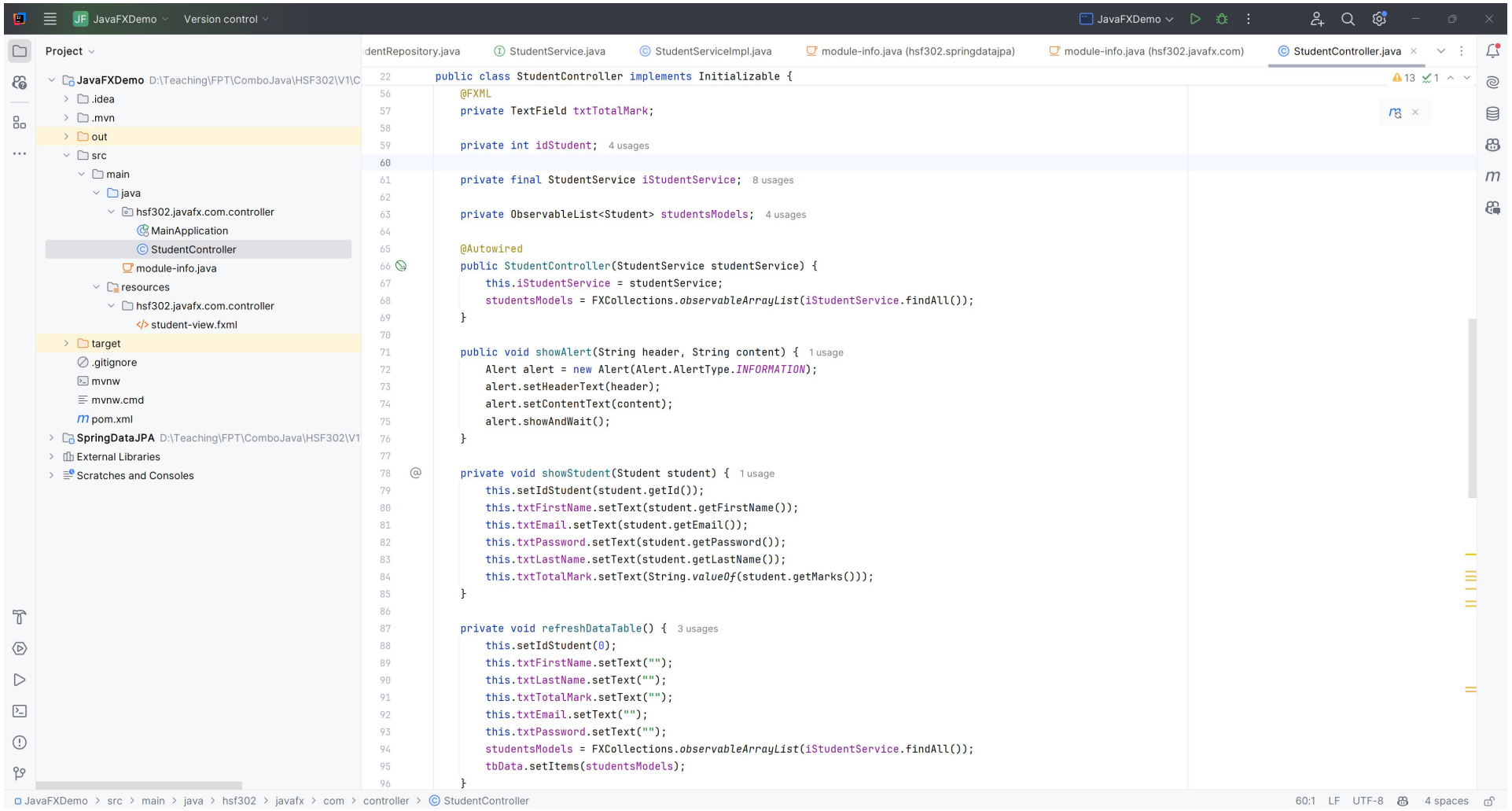
Edit student-view.fxml



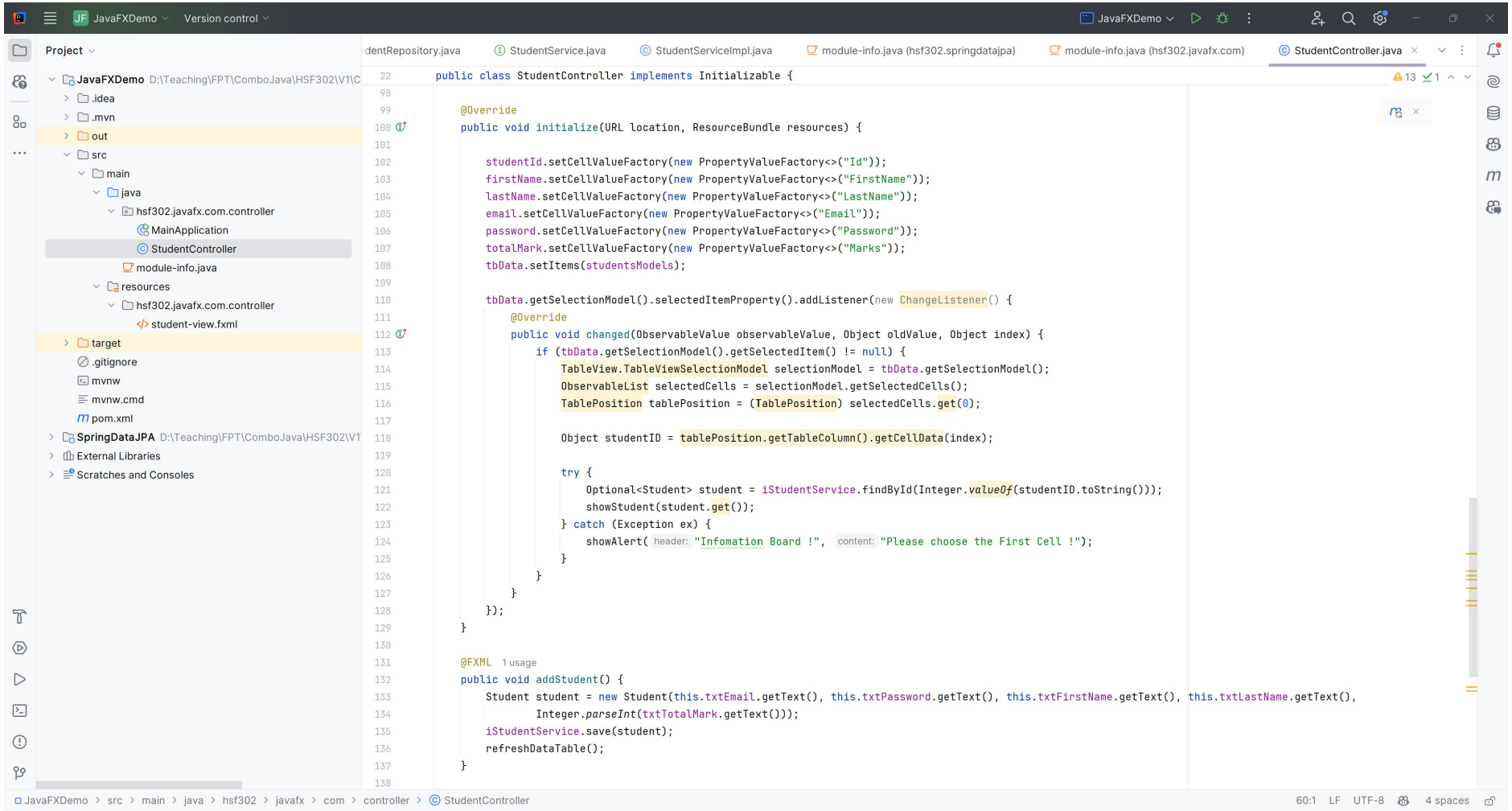
Edit the StudentController.java



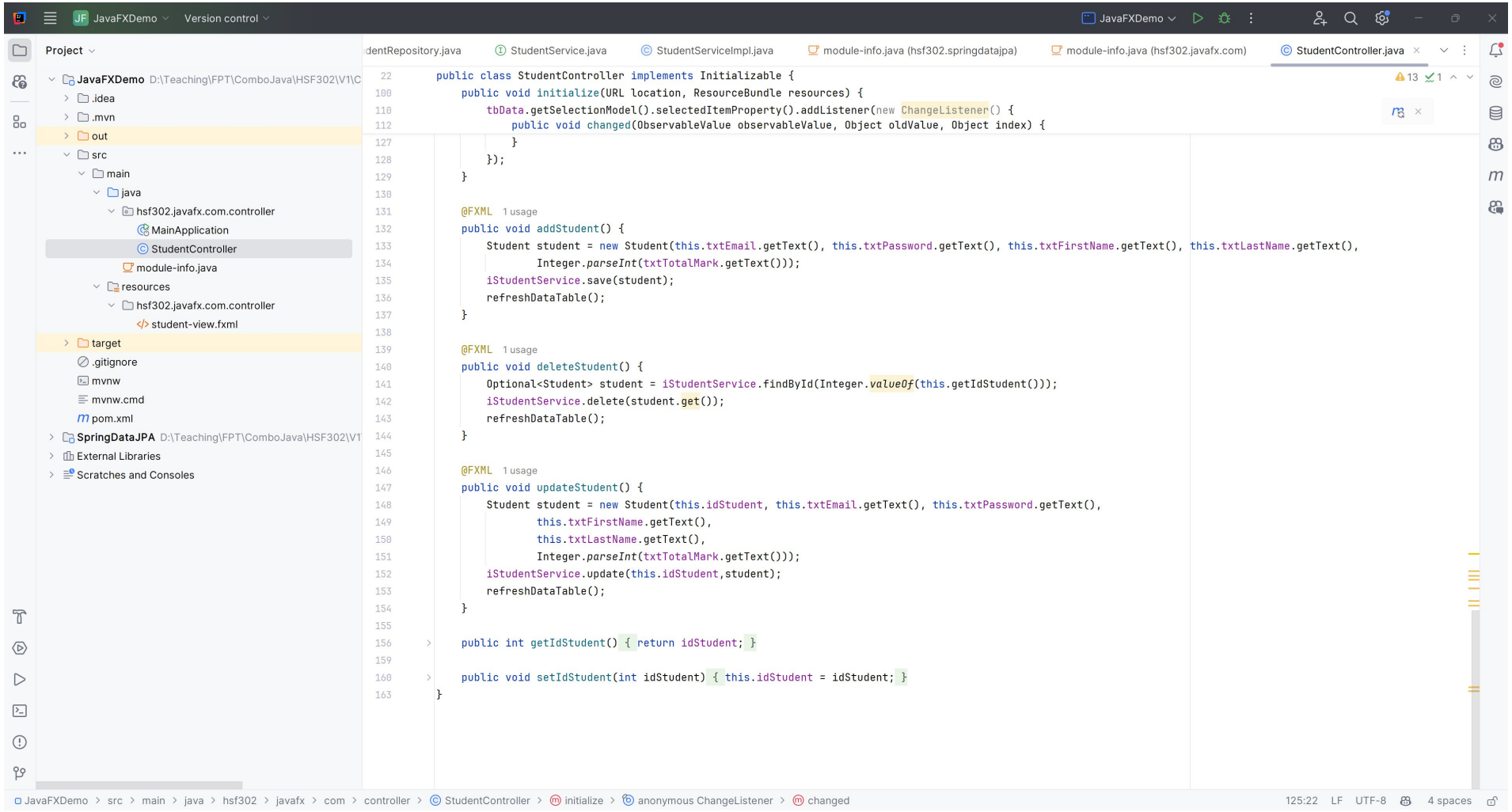
Edit the StudentController.java



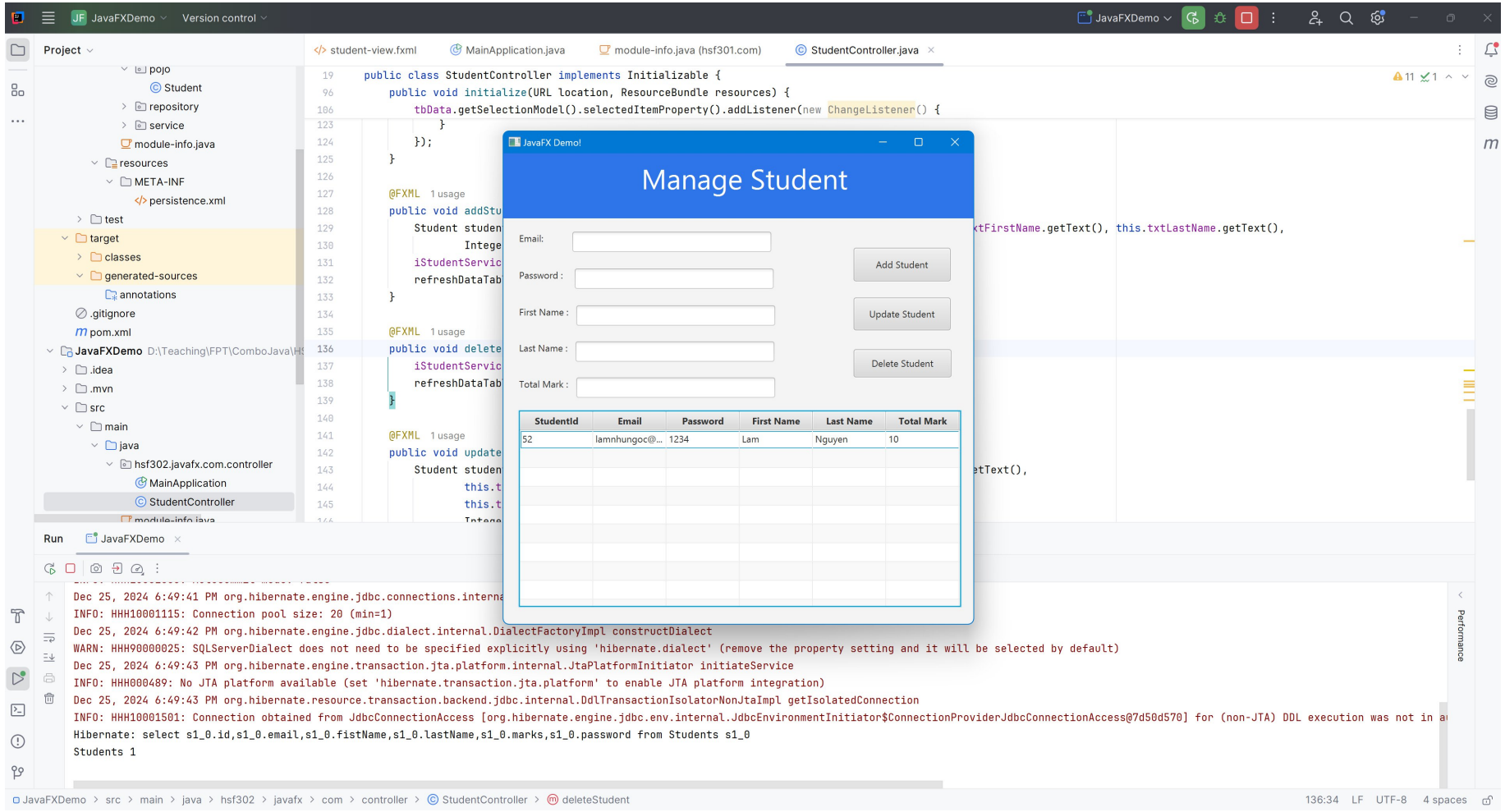
Edit the StudentController.java



Edit the StudentController.java



Run Project


JavaFXDemo
Version control
Project
student-view.fxml
MainApplication.java
module-info.java (hsf301.com)
StudentController.java
JavaFX Demo!
Manage Student
Email:
Password :
First Name :
Last Name :
Total Mark :
Add Student
Update Student
Delete Student
StudentId
Email
Password
First Name
Last Name
Total Mark
52
lamnhungoc@...
1234
Lam
Nguyen
10
Run
JavaFXDemo
Dec 25, 2024 6:49:41 PM org.hibernate.engine.jdbc.connections.internal
INFO: HHH10001115: Connection pool size: 20 (min=1)
Dec 25, 2024 6:49:42 PM org.hibernate.engine.jdbc.dialect.internal.DialectFactoryImpl constructDialect
WARN: HHH90000025: SQLServerDialect does not need to be specified explicitly using 'hibernate.dialect' (remove the property setting and it will be selected by default)
Dec 25, 2024 6:49:43 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000489: No JTA platform available (set 'hibernate.transaction.jta.platform' to enable JTA platform integration)
Dec 25, 2024 6:49:43 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConnection
Hibernate: select s1_0.id,s1_0.email,s1_0.fistName,s1_0.lastName,s1_0.marks,s1_0.password from Students s1_0
Students 1
Performance
JavaFXDemo > src > main > java > hsf302 > javafx > com > controller > StudentController > deleteStudent
136:34 LF UTF-8 4 spaces

Summary

Concepts were introduced:

- Implement CRUD (Create, Read, Update, Delete) operations to manage data effectively.
- Utilize Hibernate to map Java objects to database tables, simplifying data persistence and retrieval.
- Design and develop a user-friendly interface using JavaFX, enabling seamless interaction with the application.
- Implement features such as form input validation, error handling, and feedback mechanisms to enhance user experience.