Natalia Palej

A00279259@student.tus.ie

Year 3 Semester 1

Software Design with Artificial Intelligence for Cloud Computing

Software Dev Project

Project 3

Table of Contents

[Project Brief 2](#_Toc151753741)

[Code 3](#_Toc151753742)

[StudentModel.java 3](#_Toc151753743)

[StudentInterface.java (interface) 6](#_Toc151753744)

[StudentService.java (interface) 6](#_Toc151753745)

[StudentController.java 7](#_Toc151753746)

[StudentGUI.java 10](#_Toc151753747)

[StudentServer.java 21](#_Toc151753748)

[StudentClient.java 22](#_Toc151753749)

[Project Screencast 23](#_Toc151753750)

[References 23](#_Toc151753751)

# Project Brief

For my project I have designed a Students System allowing user ton view, add, update and delete students. The application uses GUI interface which is user friendly and easy to navigate. To create this project, I have chosen two design patterns: MVC (Model-View-Controller) and Singleton.

**MVC** design pattern separates logic and data retrieval, making it more maintainable and readable. Since my project is a Students “database” (simulated one, not the real database), I thought it would align with app needs. The classes are broken down into the following:

* Student - represents the data structure for student information (constructor, getters, setters, printDetails method)
* StudentController - contains all the methods responsible for handling data manipulation (CRUD) as well as serialization/deserialization.
* StudentGUI – graphical user interface displaying student information, allowing user to perform various actions via buttons.
* StudentInterface (interface) – defines methods that update the view based on user manipulations in the model.

**Singleton** – initially including it wasn’t a part of the plan, although later in implementation stage I ran into issues with displaying data. Using the Singleton pattern ensures that only one instance of the student data exists throughout the application session. This prevents data duplication and ensures the object’s state remains current.

The other three classes handle RMI (Remote Method Invocation).

* StudentService (interface) – defines the remote methods accessible by client.
* StudentServer – initializes the remote StudentController object and binds it to registry, so it becomes accessible to the client.
* StudentClient – acts as the client-side component responsible for interacting with the server. It establishes the communication between GUI and the server, where user can perform various operations.

# Code

## Student.java

/\*\*

\* author **@NataliaPalej** A00279259

\*/

**import** java**.**io**.**Serializable**;**

**public** **class** Student **implements** Serializable **{**

**private** String id**,** name**,** surname**,** address**,** course**;**

**private** **float** grade1**,** grade2**,** grade3**,** grade4**;**

**private** **static** **final** **long** serialVersionUID **=** 1L**;**

Student**(**String id**,** String name**,** String surname**,** String address**,** String course**,** **float** grade1**,** **float** grade2**,** **float** grade3**,** **float** grade4**){**

**this.**id **=** id**;**

**this.**name **=** name**;**

**this.**surname **=** surname**;**

**this.**address **=** address**;**

**this.**course **=** course**;**

**this.**grade1 **=** grade1**;**

**this.**grade2 **=** grade2**;**

**this.**grade3 **=** grade3**;**

**this.**grade4 **=** grade4**;**

**}**

// Empty Constructor

Student**(){**

**}**

**public** String getId**()** **{**

**return** id**;**

**}**

**public** **void** setId**(**String id**)** **{**

**this.**id **=** id**;**

**}**

**public** String getName**()** **{**

**return** name**;**

**}**

**public** **void** setName**(**String name**)** **{**

**this.**name **=** name**;**

**}**

**public** String getSurname**()** **{**

**return** surname**;**

**}**

**public** **void** setSurname**(**String surname**)** **{**

**this.**surname **=** surname**;**

**}**

**public** String getAddress**()** **{**

**return** address**;**

**}**

**public** **void** setAddress**(**String address**)** **{**

**this.**address **=** address**;**

**}**

**public** String getCourse**()** **{**

**return** course**;**

**}**

**public** **void** setCourse**(**String course**)** **{**

**this.**course **=** course**;**

**}**

**public** **float** getGrade1**()** **{**

**return** grade1**;**

**}**

**public** **void** setGrade1**(float** grade1**)** **{**

**this.**grade1 **=** grade1**;**

**}**

**public** **float** getGrade2**()** **{**

**return** grade2**;**

**}**

**public** **void** setGrade2**(float** grade2**)** **{**

**this.**grade2 **=** grade2**;**

**}**

**public** **float** getGrade3**()** **{**

**return** grade3**;**

**}**

**public** **void** setGrade3**(float** grade3**)** **{**

**this.**grade3 **=** grade3**;**

**}**

**public** **float** getGrade4**()** **{**

**return** grade4**;**

**}**

**public** **void** setGrade4**(float** grade4**)** **{**

**this.**grade4 **=** grade4**;**

**}**

**public** **float** getAverage**(float** grade**,** **float** grade2**,** **float** grade3**,** **float** grade4**)** **{**

**float** average **=** **(**grade**+**grade2**+**grade3**+**grade4**)/**4**;**

average **=** Math**.**round**(**average**\***100**)/**100**;**

**return** average**;**

**}**

**public** String printDetails**()** **{**

**return** "\nID: " **+** **this.**id **+**

"\nName: " **+** **this.**name **+**

"\tSurname: " **+** **this.**surname **+**

"\tAddress: " **+** **this.**address **+**

"\tCourse: " **+** **this.**course **+**

"\nGrade1: " **+** **this.**grade1 **+**

"\tGrade2: " **+** **this.**grade2 **+**

"\tGrade3: " **+** **this.**grade3 **+**

"\tGrade4: " **+** **this.**grade4 **+**

"\t\tGPA: " **+** getAverage**(this.**grade1**,** **this.**grade2**,** **this.**grade3**,** **this.**grade4**)** **+** "%"**;**

**}**

**public** String getStudentDetails**()** **{**

**return** "\nID: " **+** **this.**id **+**

"\nName: " **+** **this.**name **+**

"\nSurname: " **+** **this.**surname **+**

"\nAddress: " **+** **this.**address **+**

"\nCourse: " **+** **this.**course **+**

"\nGrade1: " **+** **this.**grade1 **+**

"\nGrade2: " **+** **this.**grade2 **+**

"\nGrade3: " **+** **this.**grade3 **+**

"\nGrade4: " **+** **this.**grade4 **+**

"\nGPA: " **+** getAverage**(this.**grade1**,** **this.**grade2**,** **this.**grade3**,** **this.**grade4**)** **+** "%"**;**

**}**

**}**

## StudentInterface.java (interface)

/\*\*

\* author **@NataliaPalej** A00279259

\*/

**import** java**.**rmi**.**RemoteException**;**

**public** **interface** StudentInterface **{**

**void** showAllStudents**()** **throws** RemoteException**;**

**void** showStudentDetails**(**Student student**)** **throws** RemoteException**;**

**void** showMessage**(**String message**);**

**void** showSuccessMessage**(**String message**);**

**void** showErrorMessage**(**String message**);**

**}**

## StudentService.java (interface)

/\*\*

\* author **@NataliaPalej** A00279259

\*/

**import** java**.**rmi**.**Remote**;**

**import** java**.**rmi**.**RemoteException**;**

**public** **interface** StudentService **extends** Remote **{**

String getAll**()** **throws** RemoteException**;**

Student getByID**(**String id**)** **throws** RemoteException**;**

Student getByName**(**String name**)** **throws** RemoteException**;**

**void** add**(**Student student**)** **throws** RemoteException**;**

**void** update**(**Student studentToUpdate**,** Student updatedStudent**)** **throws** RemoteException**;**

**void** delete**(**String id**)** **throws** RemoteException**;**

**}**

## StudentController.java

**import** java**.**io**.**\***;**

**import** java**.**rmi**.**RemoteException**;**

**import** java**.**rmi**.**server**.**UnicastRemoteObject**;**

**import** java**.**util**.**\***;**

@SuppressWarnings**(**"serial"**)**

**public** **class** StudentController **extends** UnicastRemoteObject **implements** StudentService **{**

**private** List**<**Student**>** studentList**;**

// Declare singleton instance

**private** **static** StudentController instance**;**

**public** StudentController**()** **throws** RemoteException **{**

**super();**

File file **=** **new** File**(**"students.ser"**);**

**if** **(**file**.**exists**())** **{**

**this.**studentList **=** deserialiseStudents**();**

**}** **else** **{**

**this.**studentList **=** studentList**();**

serialiseStudents**(**studentList**);**

**}**

**}**

**public** StudentController**(**List**<**Student**>** studentList**)** **throws** RemoteException **{**

**this.**studentList **=** studentList**;**

**}**

@Override

**public** **void** add**(**Student student**)** **throws** RemoteException **{**

studentList**.**add**(**student**);**

instance**.**serialiseStudents**(**studentList**);**

**}**

@Override

**public** **void** delete**(**String name**)** **throws** RemoteException **{**

Iterator**<**Student**>** iterator **=** studentList**.**iterator**();**

**while** **(**iterator**.**hasNext**())** **{**

Student student **=** iterator**.**next**();**

**if** **(**student**.**getName**().**equalsIgnoreCase**(**name**)** **||** student**.**getId**().**equalsIgnoreCase**(**name**))** **{**

iterator**.**remove**();**

instance**.**serialiseStudents**(**studentList**);**

System**.**out**.**println**(**"StudentController delete(): Student " **+** student**.**getName**()** **+** " was deleted."**);**

**}**

**}**

**if** **(**studentList**.**isEmpty**())** **{**

System**.**out**.**println**(**"StudentController delete(): List is empty"**);**

**}**

**}**

@Override

**public** **void** update**(**Student studentToUpdate**,** Student updatedStudent**)** **throws** RemoteException **{**

**for** **(int** i**=**0**;** i**<**studentList**.**size**();** i**++)** **{**

**if** **(**studentToUpdate**.**getId**().**equals**(**studentList**.**get**(**i**).**getId**())** **||** studentToUpdate**.**getName**().**equals**(**studentList**.**get**(**i**).**getName**()))** **{**

studentList**.**set**(**i**,** updatedStudent**);**

instance**.**serialiseStudents**(**studentList**);**

**return;**

**}**

**}**

System**.**out**.**println**(**"update(): Student not found in the list."**);**

**}**

@Override

**public** String getAll**()** **throws** RemoteException **{**

StringBuilder result **=** **new** StringBuilder**();**

**for** **(**Student student **:** studentList**)** **{**

result**.**append**(**student**.**printDetails**());**

**}**

**return** result**.**toString**();**

**}**

**public** List**<**Student**>** getAllStudents**()** **{**

**return** studentList**;**

**}**

@Override

**public** Student getByName**(**String name**)** **throws** RemoteException **{**

**for** **(**Student student **:** studentList**)** **{**

**if** **(**student**.**getName**().**equals**(**name**))** **{**

**return** student**;**

**}**

**}**

**return** **null;**

**}**

@Override

**public** Student getByID**(**String id**)** **throws** RemoteException **{**

**for** **(**Student student **:** studentList**)** **{**

**if** **(**student**.**getId**().**equals**(**id**))** **{**

**return** student**;**

**}**

**}**

**return** **null;**

**}**

// Initial data for the studentList

**public** **static** List**<**Student**>** studentList**()** **{**

List**<**Student**>** students **=** **new** ArrayList**<>();**

students**.**add**(new** Student**(**"1"**,** "Kasia"**,** "Pretty"**,** "Tullamore"**,** "Bachelor of Beauty"**,** 89**,** 93**,** 99**,** 91**));**

students**.**add**(new** Student**(**"2"**,** "Dominik"**,** "Dominos"**,** "Tullamore"**,** "Bachelor of Computing"**,** 72**,** 68**,** 88**,** 84**));**

students**.**add**(new** Student**(**"3"**,** "Adrian"**,** "Diablo"**,** "Athlone"**,** "Bachelor of Computing"**,** 51**,** 92**,** 90**,** 87**));**

students**.**add**(new** Student**(**"4"**,** "Natalia"**,** "Normanes"**,** "Athlone"**,** "Bachelor of Computing"**,** 91**,** 93**,** 94**,** 89**));**

students**.**add**(new** Student**(**"5"**,** "Slowak"**,** "Golias"**,** "Trim"**,** "Bachelor of Social"**,** 41**,** 38**,** 68**,** 51**));**

students**.**add**(new** Student**(**"6"**,** "Monika"**,** "Monikowska"**,** "Monksland"**,** "Bachelor of Hairdressing"**,** 31**,** 48**,** 67**,** 36**));**

students**.**add**(new** Student**(**"7"**,** "Kuba"**,** "Kubowski"**,** "Monksland"**,** "Bachelor of Mechanics"**,** 71**,** 68**,** 78**,** 83**));**

**return** students**;**

**}**

// Serialise Method

**public** **void** serialiseStudents**(**List**<**Student**>** students**)** **{**

**try** **{**

// Serialise to the file

FileOutputStream fileOut **=** **new** FileOutputStream**(**"students.ser"**);**

// Assign what object to serialise

ObjectOutputStream objOut **=** **new** ObjectOutputStream**(**fileOut**);**

// Loop through phones array

**for** **(**Student s **:** students**)** **{**

objOut**.**writeObject**(**s**);**

**}**

objOut**.**close**();**

fileOut**.**close**();**

System**.**out**.**println**(**"serialiseStudents(): File successfully serialized"**);**

**}** **catch** **(**Exception e**)** **{**

System**.**out**.**println**(**"Exception cought.\n"**);**

e**.**printStackTrace**();**

**}**

**}**

// Deserialise method

**public** List**<**Student**>** deserialiseStudents**()** **{**

List**<**Student**>** students **=** **new** ArrayList**<>();**

**try** **{**

FileInputStream fileIn **=** **new** FileInputStream**(**"students.ser"**);**

ObjectInputStream objIn **=** **new** ObjectInputStream**(**fileIn**);**

// Loop until end of file is reached

**while** **(true)** **{**

**try** **{**

Student s **=** **(**Student**)** objIn**.**readObject**();**

students**.**add**(**s**);**

**}** **catch** **(**Exception e**)** **{**

//System.out.println("\n# End of File Reached#\n");

**break;**

**}**

**}**

objIn**.**close**();**

fileIn**.**close**();**

System**.**out**.**println**(**"deserialiseStudents(): File successfully deserialized"**);**

**}** **catch** **(**Exception e**)** **{**

System**.**out**.**println**(**"Couldn't deserialize the file!"**);**

e**.**printStackTrace**();**

**}**

**return** students**;**

**}**

// Singleton pattern ensuring only one sintance is in use

**public** **static** StudentController getInstance**()** **throws** RemoteException **{**

**if** **(**instance **==** **null)** **{**

instance **=** **new** StudentController**();**

System**.**out**.**println**(**"getInstance(): Created instance of StudentController."**);**

**}**

**return** instance**;**

**}**

**}**

## StudentGUI.java

/\*\*

\* author **@NataliaPalej** A00279259

\*/

**import** javax**.**swing**.\*;**

**import** javax**.**swing**.**table**.**DefaultTableModel**;**

**import** java**.**awt**.\*;**

**import** java**.**awt**.**event**.**ActionEvent**;**

**import** java**.**awt**.**event**.**ActionListener**;**

**import** java**.**rmi**.**RemoteException**;**

**import** java**.**util**.**List**;**

@SuppressWarnings**(**"serial"**)**

**public** **class** StudentGUI **extends** JFrame **implements** ActionListener**,** StudentInterface **{**

Container cp**;**

GridBagLayout gridBag **=** **new** GridBagLayout**();**

GridBagConstraints c **=** **new** GridBagConstraints**();**

JButton getAllBtn **=** **new** JButton**(**"GET \*all\*"**);**

JButton getByIDBtn **=** **new** JButton**(**"GET by ID"**);**

JButton getByNameBtn **=** **new** JButton**(**"GET by name"**);**

JButton addBtn **=** **new** JButton**(**"ADD"**);**

JButton updateBtn **=** **new** JButton**(**"UPDATE"**);**

JButton deleteBtn **=** **new** JButton**(**"DELETE"**);**

JButton exitBtn **=** **new** JButton**(**"EXIT"**);**

JLabel emptyField **=** **new** JLabel**(**" "**,** SwingConstants**.**CENTER**);**

JLabel title **=** **new** JLabel**(**"Students"**,** SwingConstants**.**CENTER**);**

JLabel emptyLine **=** **new** JLabel**(**""**,** SwingConstants**.**CENTER**);**

JLabel searchLabel **=** **new** JLabel**(**"Search:"**,** SwingConstants**.**CENTER**);**

JTextField searchInput **=** **new** JTextField**(**" "**,** 20**);**

JTable studentTable**;**

DefaultTableModel tableModel**;**

**public** StudentGUI**(**StudentController studentController**)** **throws** RemoteException **{**

setTitle**(**"Natalia Palej A0027959"**);**

setDefaultCloseOperation**(**JFrame**.**EXIT\_ON\_CLOSE**);**

// All the GUI layouts, buttons etc in a separate method to make it more readable

setupGUI**();**

pack**();**

// Put frame in the center

setLocationRelativeTo**(null);**

setSize**(**1000**,** 1000**);**

setVisible**(true);**

// Run the application

run**();**

**}**

@Override

**public** **void** actionPerformed**(**ActionEvent e**)** **{**

**if** **(**e**.**getSource**().**equals**(**getAllBtn**))** **{**

System**.**out**.**println**(**"\nactionPerformed(): GET ALL button clicked."**);**

**try** **{**

getAll**();**

**}** **catch** **(**RemoteException e1**)** **{**

e1**.**printStackTrace**();**

**}**

**}** **else** **if** **(**e**.**getSource**().**equals**(**getByIDBtn**))** **{**

System**.**out**.**println**(**"\nactionPerformed(): GET BY ID button clicked."**);**

getByID**();**

**}** **else** **if** **(**e**.**getSource**().**equals**(**getByNameBtn**))** **{**

System**.**out**.**println**(**"\nactionPerformed(): GET BY NAME button clicked."**);**

getByName**();**

**}** **else** **if** **(**e**.**getSource**().**equals**(**addBtn**))** **{**

System**.**out**.**println**(**"\nactionPerformed(): ADD button clicked."**);**

**try** **{**

add**();**

**}** **catch** **(**RemoteException e1**)** **{**

e1**.**printStackTrace**();**

**}**

**}** **else** **if** **(**e**.**getSource**().**equals**(**updateBtn**))** **{**

System**.**out**.**println**(**"\nactionPerformed(): UPDATE button clicked."**);**

**try** **{**

update**();**

**}** **catch** **(**RemoteException e1**)** **{**

e1**.**printStackTrace**();**

**}**

**}** **else** **if** **(**e**.**getSource**().**equals**(**deleteBtn**))** **{**

System**.**out**.**println**(**"\nactionPerformed(): DELETE button clicked."**);**

delete**();**

**}** **else** **if** **(**e**.**getSource**().**equals**(**exitBtn**))** **{**

System**.**out**.**println**(**"\nactionPerformed(): EXIT button clicked."**);**

// Close the app

System**.**exit**(**0**);**

**}** **else** **{**

System**.**out**.**println**(**"\nactionPerformed(): No action available"**);**

**}**

**}**

// GUI setup - buttons/labels/table/panel

**private** **void** setupGUI**()** **{**

cp **=** getContentPane**();**

cp**.**setLayout**(**gridBag**);**

cp**.**setBackground**(**Color**.**white**);**

c**.**fill **=** GridBagConstraints**.**HORIZONTAL**;**

// Picutre

JLabel iconLabel **=** **new** JLabel**();**

ImageIcon pic **=** **new** ImageIcon**(**"\\C:\\Users\\Palej\\OneDrive\\Desktop\\Software Design with AI for Cloud\\Year\_3\\software-dev-labs3\\Assessment\\java-project3\\image.jpg"**);**

iconLabel**.**setIcon**(**pic**);**

c**.**gridx **=** 10**;**

c**.**gridy **=** 0**;**

gridBag**.**setConstraints**(**iconLabel**,** c**);**

cp**.**add**(**iconLabel**);**

// Title

title**.**setFont**(new** Font**(**"Arial"**,** Font**.**PLAIN**,** 30**));**

c**.**gridx **=** 1**;**

c**.**gridy **=** 1**;**

c**.**gridwidth **=** 10**;** // Span over 10 columns

gridBag**.**setConstraints**(**title**,** c**);**

cp**.**add**(**title**);**

// Search Label

c**.**gridx **=** 1**;**

c**.**gridy **=** 3**;**

c**.**gridwidth **=** 5**;**

gridBag**.**setConstraints**(**searchLabel**,** c**);**

cp**.**add**(**searchLabel**);**

// Search Input

c**.**gridx **=** 6**;**

c**.**gridy **=** 3**;**

c**.**gridwidth **=** 5**;**

gridBag**.**setConstraints**(**searchInput**,** c**);**

cp**.**add**(**searchInput**);**

// GET by name button

getByNameBtn**.**setBackground**(**Color**.**darkGray**);**

getByNameBtn**.**setForeground**(**Color**.**white**);**

getByNameBtn**.**addActionListener**(this);**

c**.**gridx **=** 1**;**

c**.**gridy **=** 4**;**

c**.**gridwidth **=** 5**;**

gridBag**.**setConstraints**(**getByNameBtn**,** c**);**

cp**.**add**(**getByNameBtn**);**

// GET by ID button

getByIDBtn**.**setBackground**(**Color**.**darkGray**);**

getByIDBtn**.**setForeground**(**Color**.**white**);**

getByIDBtn**.**addActionListener**(this);**

c**.**gridx **=** 6**;**

c**.**gridy **=** 4**;**

c**.**gridwidth **=** 5**;**

gridBag**.**setConstraints**(**getByIDBtn**,** c**);**

cp**.**add**(**getByIDBtn**);**

// Empty field for messages

c**.**gridx **=** 1**;**

c**.**gridy **=** 5**;**

c**.**gridwidth **=** 10**;**

gridBag**.**setConstraints**(**emptyField**,** c**);**

cp**.**add**(**emptyField**);**

// Create table model with column names

tableModel **=** **new** DefaultTableModel**();**

tableModel**.**setColumnIdentifiers**(new** String**[]{**"ID"**,** "Name"**,** "Surname"**,** "Address"**,** "Course"**,** "Grade1"**,** "Grade2"**,** "Grade3"**,** "Grade4"**,** "GPA"**});**

// Create JTable with the table model

studentTable **=** **new** JTable**(**tableModel**);**

// Adjust column size

studentTable**.**getColumnModel**().**getColumn**(**0**).**setPreferredWidth**(**5**);**

studentTable**.**getColumnModel**().**getColumn**(**1**).**setPreferredWidth**(**80**);**

studentTable**.**getColumnModel**().**getColumn**(**2**).**setPreferredWidth**(**100**);**

studentTable**.**getColumnModel**().**getColumn**(**3**).**setPreferredWidth**(**100**);**

studentTable**.**getColumnModel**().**getColumn**(**4**).**setPreferredWidth**(**150**);**

studentTable**.**getColumnModel**().**getColumn**(**5**).**setPreferredWidth**(**30**);**

studentTable**.**getColumnModel**().**getColumn**(**6**).**setPreferredWidth**(**30**);**

studentTable**.**getColumnModel**().**getColumn**(**7**).**setPreferredWidth**(**30**);**

studentTable**.**getColumnModel**().**getColumn**(**8**).**setPreferredWidth**(**30**);**

studentTable**.**getColumnModel**().**getColumn**(**9**).**setPreferredWidth**(**40**);**

// Add the table to a scroll pane and adjust size

JScrollPane scrollPane **=** **new** JScrollPane**(**studentTable**);**

scrollPane**.**setPreferredSize**(new** Dimension**(**700**,** 400**));**

c**.**gridx **=** 1**;**

c**.**gridy **=** 6**;**

c**.**gridwidth **=** 10**;**

gridBag**.**setConstraints**(**scrollPane**,** c**);**

cp**.**add**(**scrollPane**);**

// GET \*all\* button

getAllBtn**.**setBackground**(**Color**.**DARK\_GRAY**);**

getAllBtn**.**setForeground**(**Color**.**white**);**

getAllBtn**.**addActionListener**(this);**

c**.**gridx **=** 1**;**

c**.**gridy **=** 7**;**

c**.**gridwidth **=** 10**;**

gridBag**.**setConstraints**(**getAllBtn**,** c**);**

cp**.**add**(**getAllBtn**);**

// ADD button

addBtn**.**setBackground**(**Color**.**green**);**

addBtn**.**addActionListener**(this);**

c**.**gridx **=** 1**;**

c**.**gridy **=** 8**;**

c**.**gridwidth **=** 10**;**

gridBag**.**setConstraints**(**addBtn**,** c**);**

cp**.**add**(**addBtn**);**

// UPDATE button

updateBtn**.**setBackground**(**Color**.**yellow**);**

updateBtn**.**addActionListener**(this);**

c**.**gridx **=** 1**;**

c**.**gridy **=** 9**;**

c**.**gridwidth **=** 5**;**

gridBag**.**setConstraints**(**updateBtn**,** c**);**

cp**.**add**(**updateBtn**);**

// DELETE button

deleteBtn**.**setBackground**(**Color**.**red**);**

deleteBtn**.**addActionListener**(this);**

c**.**gridx **=** 6**;**

c**.**gridy **=** 9**;**

c**.**gridwidth **=** 5**;**

gridBag**.**setConstraints**(**deleteBtn**,** c**);**

cp**.**add**(**deleteBtn**);**

// EXIT button

exitBtn**.**setBackground**(**Color**.**black**);**

exitBtn**.**setForeground**(**Color**.**white**);**

exitBtn**.**addActionListener**(this);**

c**.**gridx **=** 1**;**

c**.**gridy **=** 10**;**

c**.**gridwidth **=** 10**;**

gridBag**.**setConstraints**(**exitBtn**,** c**);**

cp**.**add**(**exitBtn**);**

// Set the same size to all buttons

Dimension buttonSize **=** **new** Dimension**(**150**,** 40**);**

getAllBtn**.**setPreferredSize**(**buttonSize**);**

getByIDBtn**.**setPreferredSize**(**buttonSize**);**

getByNameBtn**.**setPreferredSize**(**buttonSize**);**

addBtn**.**setPreferredSize**(**buttonSize**);**

updateBtn**.**setPreferredSize**(**buttonSize**);**

deleteBtn**.**setPreferredSize**(**buttonSize**);**

exitBtn**.**setPreferredSize**(**buttonSize**);**

// Add margin

JLabel topMargin **=** **new** JLabel**(**" "**);**

topMargin**.**setPreferredSize**(new** Dimension**(**20**,** 50**));**

JLabel bottomMargin **=** **new** JLabel**(**" "**);**

bottomMargin**.**setPreferredSize**(new** Dimension**(**20**,** 50**));**

JLabel rightMargin **=** **new** JLabel**(**" "**);**

rightMargin**.**setPreferredSize**(new** Dimension**(**50**,** 20**));**

JLabel leftMargin **=** **new** JLabel**(**" "**);**

leftMargin**.**setPreferredSize**(new** Dimension**(**50**,** 20**));**

c**.**gridx **=** 0**;**

c**.**gridy **=** 0**;**

c**.**gridwidth **=** 1**;**

gridBag**.**setConstraints**(**topMargin**,** c**);**

cp**.**add**(**topMargin**);**

c**.**gridx **=** 0**;**

c**.**gridy **=** 12**;**

c**.**gridwidth **=** 1**;**

gridBag**.**setConstraints**(**bottomMargin**,** c**);**

cp**.**add**(**bottomMargin**);**

c**.**gridx **=** 0**;**

c**.**gridy **=** 0**;**

c**.**gridwidth **=** 1**;**

gridBag**.**setConstraints**(**leftMargin**,** c**);**

cp**.**add**(**leftMargin**);**

c**.**gridx **=** 5**;**

c**.**gridy **=** 0**;**

c**.**gridwidth **=** 1**;**

gridBag**.**setConstraints**(**rightMargin**,** c**);**

cp**.**add**(**rightMargin**);**

**}**

// Call deserialiseStudents method from StudentController

**public** List**<**Student**>** getStudentList**()** **throws** RemoteException **{**

**return** StudentController**.**getInstance**().**deserialiseStudents**();**

**}**

// Separate methods for CRUD operations handling

**private** **void** getAll**()** **throws** RemoteException **{**

// Retrieve list of students from the controller

List**<**Student**>** studentList **=** getStudentList**();**

**if** **(!**studentList**.**isEmpty**())** **{**

showAllStudents**();**

showMessage**(**"success : List of all students"**);**

System**.**out**.**println**(**"success : getAll()" **+** StudentController**.**getInstance**().**getAll**());**

**}** **else** **{**

showErrorMessage**(**"getAll() List not found or empty. Cannot Deserialize!"**);**

**}**

**}**

**private** **void** getByID**()** **{**

// Get the text from the selectStudent text field

String searchStudent **=** searchInput**.**getText**();**

**try** **{**

Student student **=** StudentController**.**getInstance**().**getByID**(**searchStudent**);**

showSuccessMessage**(**"Student " **+** student**.**getId**()** **+** " " **+** student**.**getName**()** **+** " successfully fetched"**);**

showStudentDetails**(**student**);**

**}** **catch** **(**Exception e**)** **{**

showErrorMessage**(**"Student " **+** searchStudent **+** " not in the list."**);**

System**.**out**.**println**(**"Student " **+** searchStudent **+** " not in the list."**);**

**}**

**}**

**private** **void** getByName**()** **{**

// Get the text from the selectStudent text field

String searchStudent **=** searchInput**.**getText**();**

**try** **{**

Student student **=** StudentController**.**getInstance**().**getByName**(**searchStudent**);**

showSuccessMessage**(**"Student " **+** searchStudent **+** " successfully fetched"**);**

showStudentDetails**(**student**);**

**}** **catch** **(**Exception e**)** **{**

showErrorMessage**(**"Student " **+** searchStudent **+** " not in the list."**);**

System**.**out**.**println**(**"Student " **+** searchStudent **+** " not in the list."**);**

**}**

**}**

**private** **void** add**()** **throws** RemoteException **{**

// Create input fields for a new student

JTextField id **=** **new** JTextField**(**10**);**

JTextField name **=** **new** JTextField**(**10**);**

JTextField surname **=** **new** JTextField**(**10**);**

JTextField address **=** **new** JTextField**(**10**);**

JTextField course **=** **new** JTextField**(**10**);**

JTextField grade1 **=** **new** JTextField**(**10**);**

JTextField grade2 **=** **new** JTextField**(**10**);**

JTextField grade3 **=** **new** JTextField**(**10**);**

JTextField grade4 **=** **new** JTextField**(**10**);**

// Create array with labels and their fields

// JOptionPane recognizes Strings and treats them as Labels

Object**[]** fields **=** **{**

"ID: "**,** id**,**

"Name: "**,** name**,**

"Surname: "**,** surname**,**

"Address: "**,** address**,**

"Course: "**,** course**,**

"Grade1: "**,** grade1**,**

"Grade2: "**,** grade2**,**

"Grade3: "**,** grade3**,**

"Grade4: "**,** grade4

**};**

// Show the input window and get the user input

**int** result **=** JOptionPane**.**showConfirmDialog**(null,** fields**,** "Add Student"**,** JOptionPane**.**OK\_CANCEL\_OPTION**);**

// Check if the user clicked OK

**if** **(**result **==** JOptionPane**.**OK\_OPTION**)** **{**

// Get inputs

String idText **=** id**.**getText**();**

String nameText **=** name**.**getText**();**

String surnameText **=** surname**.**getText**();**

String addressText **=** address**.**getText**();**

String courseText **=** course**.**getText**();**

// Parse grades to doubles

// Parsing to doubles throws unexpected error? Parsed to floats instead

**float** grade1Text **=** Float**.**parseFloat**(**grade1**.**getText**());**

**float** grade2Text **=** Float**.**parseFloat**(**grade2**.**getText**());**

**float** grade3Text **=** Float**.**parseFloat**(**grade3**.**getText**());**

**float** grade4Text **=** Float**.**parseFloat**(**grade4**.**getText**());**

// ADd new Student

Student newStudent **=** **new** Student**(**idText**,** nameText**,** surnameText**,** addressText**,** courseText**,** grade1Text**,** grade2Text**,** grade3Text**,** grade4Text**);**

StudentController**.**getInstance**().**add**(**newStudent**);**

// showMessageDialog to notify user

JOptionPane**.**showMessageDialog**(null,** "Student " **+** nameText **+** " added successfully!"**);**

showSuccessMessage**(**"Student " **+** nameText **+** " added successfully!"**);**

showAllStudents**();**

**}** **else** **if** **(**result **==** JOptionPane**.**OK\_CANCEL\_OPTION**)** **{**

System**.**out**.**println**(**"add(): Add new student cancelled."**);**

showSuccessMessage**(**"Add new student successfully cancelled."**);**

**}**

**}**

**private** **void** update**()** **throws** RemoteException **{**

String updateStudent **=** searchInput**.**getText**();**

List**<**Student**>** studentList **=** getStudentList**();**

// Find the student to update in the list

Student studentToUpdate **=** **null;**

**for** **(**Student student **:** studentList**)** **{**

**if** **(**updateStudent**.**equalsIgnoreCase**(**student**.**getName**())** **||** updateStudent**.**equalsIgnoreCase**(**student**.**getId**()))** **{**

studentToUpdate **=** student**;**

**break;**

**}**

**}**

**if** **(**studentToUpdate **!=** **null)** **{**

// Create input fields to update student

JTextField id **=** **new** JTextField**(**10**);**

JTextField name **=** **new** JTextField**(**10**);**

JTextField surname **=** **new** JTextField**(**10**);**

JTextField address **=** **new** JTextField**(**10**);**

JTextField course **=** **new** JTextField**(**10**);**

JTextField grade1 **=** **new** JTextField**(**10**);**

JTextField grade2 **=** **new** JTextField**(**10**);**

JTextField grade3 **=** **new** JTextField**(**10**);**

JTextField grade4 **=** **new** JTextField**(**10**);**

// Set student values in input fields

id**.**setText**(**studentToUpdate**.**getId**());**

name**.**setText**(**studentToUpdate**.**getName**());**

surname**.**setText**(**studentToUpdate**.**getSurname**());**

address**.**setText**(**studentToUpdate**.**getAddress**());**

course**.**setText**(**studentToUpdate**.**getCourse**());**

grade1**.**setText**(**String**.**valueOf**(**studentToUpdate**.**getGrade1**()));**

grade2**.**setText**(**String**.**valueOf**(**studentToUpdate**.**getGrade2**()));**

grade3**.**setText**(**String**.**valueOf**(**studentToUpdate**.**getGrade3**()));**

grade4**.**setText**(**String**.**valueOf**(**studentToUpdate**.**getGrade4**()));**

// Array with labels and their fields

Object**[]** fields **=** **{**

"ID: "**,** id**,**

"Name: "**,** name**,**

"Surname: "**,** surname**,**

"Address: "**,** address**,**

"Course: "**,** course**,**

"Grade1: "**,** grade1**,**

"Grade2: "**,** grade2**,**

"Grade3: "**,** grade3**,**

"Grade4: "**,** grade4

**};**

**int** result **=** JOptionPane**.**showConfirmDialog**(null,** fields**,** "Update Student"**,** JOptionPane**.**OK\_CANCEL\_OPTION**);**

**if** **(**result **==** JOptionPane**.**OK\_OPTION**)** **{**

**float** grade1Text **=** Float**.**parseFloat**(**grade1**.**getText**());**

**float** grade2Text **=** Float**.**parseFloat**(**grade2**.**getText**());**

**float** grade3Text **=** Float**.**parseFloat**(**grade3**.**getText**());**

**float** grade4Text **=** Float**.**parseFloat**(**grade4**.**getText**());**

// Create updated student

Student updatedStudent **=** **new** Student**(**

id**.**getText**(),**

name**.**getText**(),**

surname**.**getText**(),**

address**.**getText**(),**

course**.**getText**(),**

grade1Text**,**

grade2Text**,**

grade3Text**,**

grade4Text

**);**

StudentController**.**getInstance**().**update**(**studentToUpdate**,** updatedStudent**);**

// Show message dialog

JOptionPane**.**showMessageDialog**(null,** "Student " **+** updateStudent **+** " was updated successfully!"**);**

showSuccessMessage**(**"Student " **+** updateStudent **+** " updated successfully!"**);**

clearTable**();**

tableModel**.**addRow**(new** Object**[]{**

updatedStudent**.**getId**(),**

updatedStudent**.**getName**(),**

updatedStudent**.**getSurname**(),**

updatedStudent**.**getAddress**(),**

updatedStudent**.**getCourse**(),**

updatedStudent**.**getGrade1**(),**

updatedStudent**.**getGrade2**(),**

updatedStudent**.**getGrade3**(),**

updatedStudent**.**getGrade4**(),**

updatedStudent**.**getAverage**(**updatedStudent**.**getGrade1**(),** updatedStudent**.**getGrade2**(),** updatedStudent**.**getGrade3**(),** updatedStudent**.**getGrade4**())** **+** "%"

**});**

**}** **else** **if** **(**result **==** JOptionPane**.**CANCEL\_OPTION**)** **{**

System**.**out**.**println**(**"update(): Update student cancelled."**);**

showSuccessMessage**(**"Update student cancelled."**);**

**}**

**}** **else** **{**

System**.**out**.**println**(**"update(): Student " **+** updateStudent **+** " not found for update."**);**

showErrorMessage**(**"Student " **+** updateStudent **+** " not found for update."**);**

**}**

**}**

**private** **void** delete**()** **{**

String deleteStudent **=** searchInput**.**getText**();**

**try** **{**

**int** result **=** JOptionPane**.**showConfirmDialog**(null,** "Are you sure you want to delete " **+** deleteStudent **+** "?"**);**

**if** **(**result **==** JOptionPane**.**YES\_OPTION**)** **{**

StudentController**.**getInstance**().**delete**(**deleteStudent**);**

JOptionPane**.**showMessageDialog**(null,** "Student " **+** deleteStudent **+** " deleted successfully!"**);**

showSuccessMessage**(**"Student " **+** deleteStudent **+** " was successfully deleted."**);**

clearTable**();**

showAllStudents**();**

**}** **else** **if** **(**result **==** JOptionPane**.**NO\_OPTION**)** **{**

showSuccessMessage**(**"Delete student cancelled."**);**

**}**

**}** **catch** **(**Exception e**)** **{**

showErrorMessage**(**"Couldn't delete student " **+** deleteStudent **+** "."**);**

**}**

**}**

**private** **void** clearTable**()** **{**

tableModel**.**setRowCount**(**0**);**

**}**

// Initialize student controller

**public** **void** setStudentController**()** **throws** RemoteException **{**

StudentController**.**getInstance**();**

**}**

**public** **void** run**()** **throws** RemoteException **{**

getAll**();**

setVisible**(true);**

**}**

@Override

**public** **void** showAllStudents**()** **throws** RemoteException **{**

List**<**Student**>** studentList **=** StudentController**.**getInstance**().**getAllStudents**();**

// Clear existing rows in the table

clearTable**();**

// Loop through the list and add students to the table

**for** **(**Student student **:** studentList**)** **{**

tableModel**.**addRow**(new** Object**[]{**

student**.**getId**(),**

student**.**getName**(),**

student**.**getSurname**(),**

student**.**getAddress**(),**

student**.**getCourse**(),**

student**.**getGrade1**(),**

student**.**getGrade2**(),**

student**.**getGrade3**(),**

student**.**getGrade4**(),**

student**.**getAverage**(**student**.**getGrade1**(),** student**.**getGrade2**(),** student**.**getGrade3**(),** student**.**getGrade4**())** **+** "%"

**});**

**}**

**}**

@Override

**public** **void** showStudentDetails**(**Student student**)** **throws** RemoteException **{**

List**<**Student**>** studentList **=** getStudentList**();**

**if** **(**studentList**.**isEmpty**())** **{**

System**.**out**.**println**(**"showStudentDetails(): Student list is empty."**);**

showMessage**(**"List of Students is empty!"**);**

**}** **else** **{**

**if** **(**student **!=** **null)** **{**

System**.**out**.**println**(**"showStudentDetails(): Student " **+** student**.**getId**()** **+** " " **+** student**.**getName**()** **+** " in the list"**);**

showSuccessMessage**(**"Student " **+** student**.**getId**()** **+** " was successfully found."**);**

clearTable**();**

System**.**out**.**println**(**student**.**getStudentDetails**());**

tableModel**.**addRow**(new** Object**[]{**

student**.**getId**(),**

student**.**getName**(),**

student**.**getSurname**(),**

student**.**getAddress**(),**

student**.**getCourse**(),**

student**.**getGrade1**(),**

student**.**getGrade2**(),**

student**.**getGrade3**(),**

student**.**getGrade4**(),**

student**.**getAverage**(**student**.**getGrade1**(),** student**.**getGrade2**(),** student**.**getGrade3**(),** student**.**getGrade4**())** **+** "%"

**});**

**}** **else** **{**

showErrorMessage**(**"Student not found in studentsList."**);**

System**.**out**.**println**(**"showStudentDetails(): Student not found in the studentList."**);**

**}**

**}**

**}**

@Override

**public** **void** showMessage**(**String message**)** **{**

emptyField**.**setText**(**message**);**

**}**

@Override

**public** **void** showSuccessMessage**(**String message**)** **{**

showMessage**(**"\"success\" : " **+** message**);**

**}**

@Override

**public** **void** showErrorMessage**(**String message**)** **{**

showMessage**(**"\"error\" : " **+** message**);**

**}**

**}**

## StudentServer.java

/\*\*

\* author **@NataliaPalej** A00279259

\*/

**import** java**.**rmi**.**registry**.**LocateRegistry**;**

**import** java**.**rmi**.**registry**.**Registry**;**

**public** **class** StudentServer **{**

**public** **static** **void** main**(**String**[]** args**)** **{**

System**.**out**.**println**(**"StudentServer: starting\n"**);**

**try** **{**

// Creating instance of remote StudentController object

StudentController studentController **=** **new** StudentController**();**

// Creating registry on localhost and default RMI port of 1099

Registry registry **=** LocateRegistry**.**createRegistry**(**1099**);**

// Bind remote into the registry

registry**.**bind**(**"StudentService"**,** studentController**);**

System**.**out**.**println**(**"\nStudentServer: ready\n"**);**

**}** **catch** **(**Exception e**)** **{**

System**.**out**.**println**(**"StudentServer: Couldn't start server"**);**

e**.**printStackTrace**();**

**}**

**}**

**}**

## StudentClient.java

/\*\*

\* author **@NataliaPalej** A00279259

\*/

**import** java**.**rmi**.**RemoteException**;**

**import** java**.**rmi**.**registry**.**LocateRegistry**;**

**import** java**.**rmi**.**registry**.**Registry**;**

**public** **class** StudentClient **{**

**static** Student studentClass **=** **new** Student**();**

**public** **static** **void** main**(**String**[]** args**)** **throws** RemoteException **{**

System**.**out**.**println**(**"StudentClient: starting\n"**);**

**try** **{**

System**.**out**.**println**(**"\n------------------------------------------------------------------------------------\n"**);**

// Get registry]

Registry registry **=** LocateRegistry**.**getRegistry**(**"localhost"**,** 1099**);**

// getInstance() to get singleton instance of StudentController

StudentController studentController **=** StudentController**.**getInstance**();**

StudentGUI studentGUI **=** **new** StudentGUI**(**studentController**);**

System**.**out**.**println**(**"\n-----------------------------------------------------------------------------------\n"**);**

System**.**out**.**println**(**"\nStudentClient(): ready"**);**

**}** **catch** **(**Exception e**)** **{**

System**.**out**.**println**(**"\nStudentClient: Couldn't start client"**);**

e**.**printStackTrace**();**

**}**

**}**

**}**

# Project Screencast

[NPalej\_A00279259 Screencast.mp4](https://tusmm-my.sharepoint.com/:v:/g/personal/a00279259_student_tus_ie/ET_BLDyAA9NFkhGQpC7tmfYBnDI6tWqilzIJfWj7Z59u_g?nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJPbmVEcml2ZUZvckJ1c2luZXNzIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsTW9kZSI6InZpZXciLCJyZWZlcnJhbFZpZXciOiJNeUZpbGVzTGlua0RpcmVjdCJ9fQ&e=2uhjYk)

# References

1. Java Swing Layouts: <https://examples.javacodegeeks.com/java-swing-layouts-example/>
2. Swing – Layouts: <https://www.tutorialspoint.com/swing/swing_layouts.htm>
3. Java Layout Manager: <https://www.javatpoint.com/java-layout-manager>
4. JOptionPane: <https://docs.oracle.com/javase/8/docs/api/javax/swing/JOptionPane.html>
5. Java JOptionPane: <https://www.javatpoint.com/java-joptionpane>
6. User input with Java JOptionPane Example: <https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/Java-user-input-with-a-Swing-JOptionPane-example>
7. Singleton in Java: <https://refactoring.guru/design-patterns/singleton/java/example>
8. Getting Started Using Java RMI: <https://docs.oracle.com/javase/8/docs/technotes/guides/rmi/hello/hello-world.html>
9. Displaying Image in Java Swing <https://www.codingninjas.com/studio/library/displaying-image-in-java-swing>