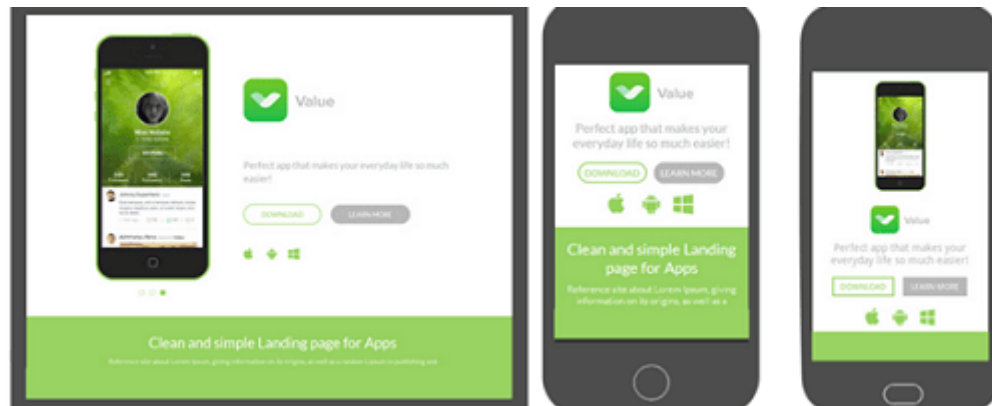


# שיטות הנדסיות לפיתוח מערכות תוכנה

Working with GUI:  
Client Server -JavaFX

# הגדרה: החלק של התוכנה שהמשתמש בא איתו באינטרקציה



**הגדרה:** קבוצה חדשה של חבילות גרפיות ומדיה  
המאפשרת למפתח תוכנה לתכנן, לכתוב ולבדוק אפליקציות  
עם ממשק משתמש נוח וידידותי שיכול לרוץ על פלטפורמות  
שונות.

ל-JavaFX מספר יתרונות:

✓ קלה לשימוש: API אחד עבור גרפיקה ומולטימדיה  
(תמונות, אנימציה, אודיו ווידאו)

✓ כוללת אפשרות עבודה עם Scene Builder

כתיבת קוד ליצירת ממשק גרפי כוללת שלושה מרכיבים:

1. בחירת אלמנטים מסוגים שונים (כפתורים, רשימות, שדות טקסט, כותרות) שיופיעו על המסך/מסכי התוכנית
2. הארגון הדו-מימדי של האלמנטים ופריסתם על המסך (בתחילת התוכנית ולאחר שינוי גודל החלון ע"י המשתמש/סגירתו/פתיחתו מחדש)
3. ההתנהגות הדינמית של האלמנטים בתגובה לפעולות של המשתמש/ת ("**אירועים**": הקלדה, הקלקה, גרירה)

- כדי לעבוד עם JavaFX יש צורך בגרסת Java מתאימה ובגרסת אקליפס מתאימה
  - ✓ מומלץ להשתמש בגרסאות המומלצות לעבודה בקורס וההנחיות להתקנתן נמצאות באתר הקורס
- בנוסף יש להתקין תוכנת Scene Builder
- למניעת בעיות אפשריות בפעולת ה-GUI **נדרשת תאימות** של גרסת ה-Scene Builder לגרסת Java שמותקנת במחשבי המכללה ובמחשבי הפיתוח שלכם



להלן מסמך מקורס קדם JAVA שמתאר את עיקרי העבודה  
של eclipse עם scene builder:

<https://docs.google.com/document/d/1pNF8wXRG10YcJ1fKbDJwLX23eu4GA8XPG1BJqKAEmWI/edit#>



כתיבת קוד בעזרת JavaFX ו-Scene Builder:

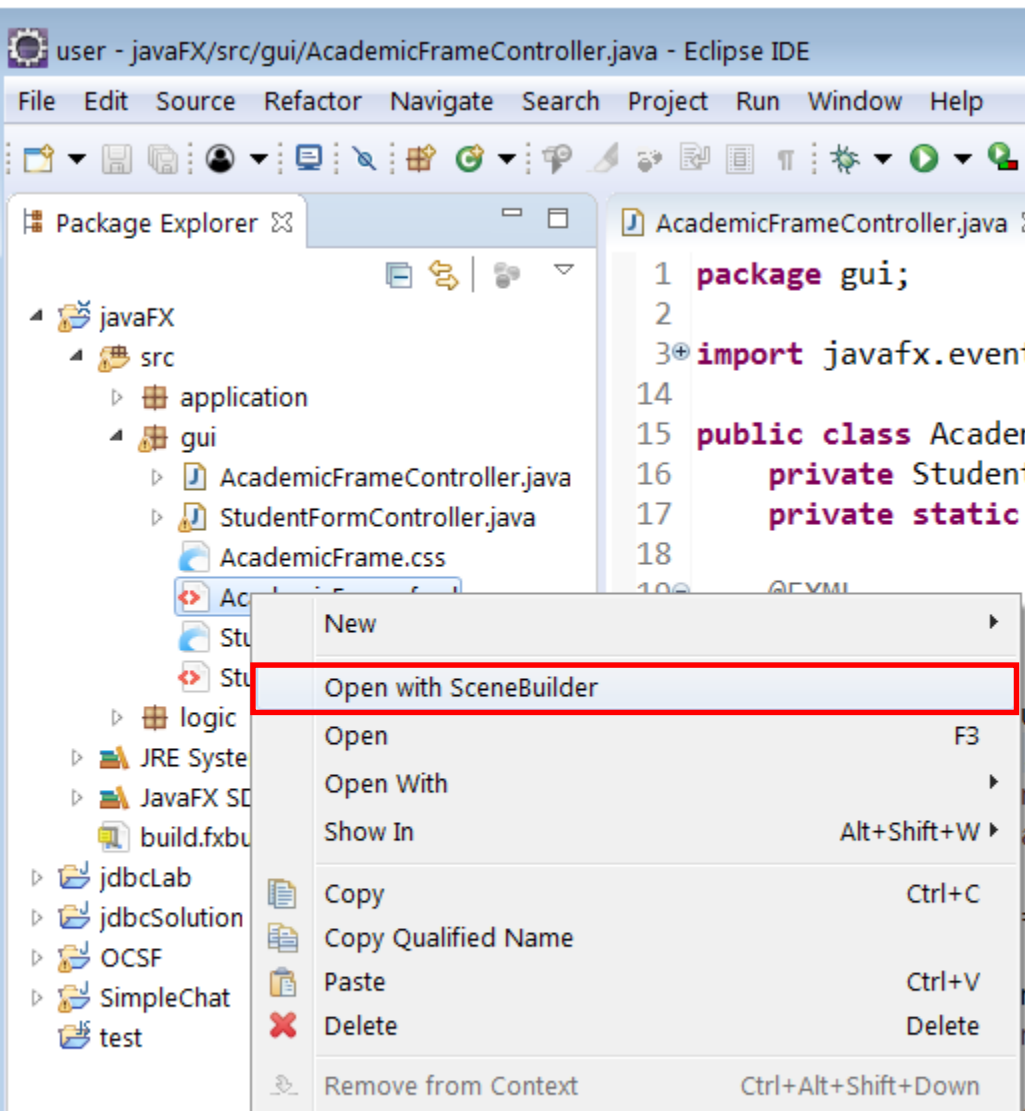
1. יש ליצור פרויקט JavaFX

2. יש ליצור קובץ FXML document

✓ קובץ המבוסס על שפת XML המאפשר בניית ממשק משתמש  
בנפרד מהחלק הלוגי של מערכת התוכנה

במקרה שלנו (של התרגול) – יש לעשות יבוא לפרויקט קיים.

## עריכת קובץ FXML document בעזרת Scene Builder:



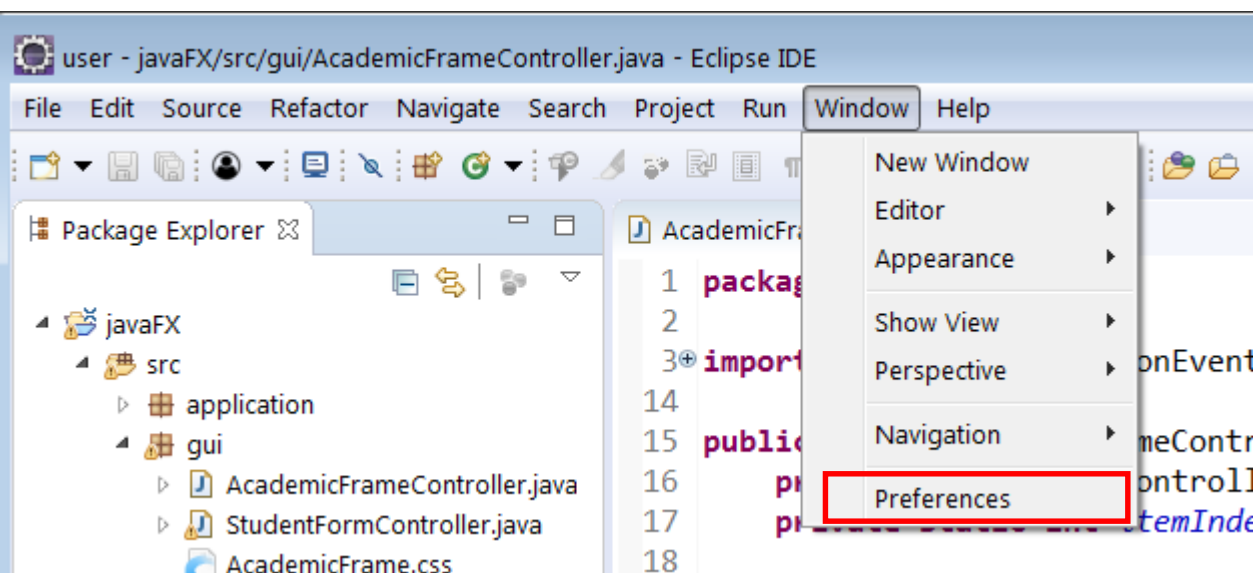
- ב-Package Explorer יש לבחור את הקובץ וללחוץ על הכפתור הימני בעכבר

- יש לבחור באפשרות של "Open with SceneBuilder"

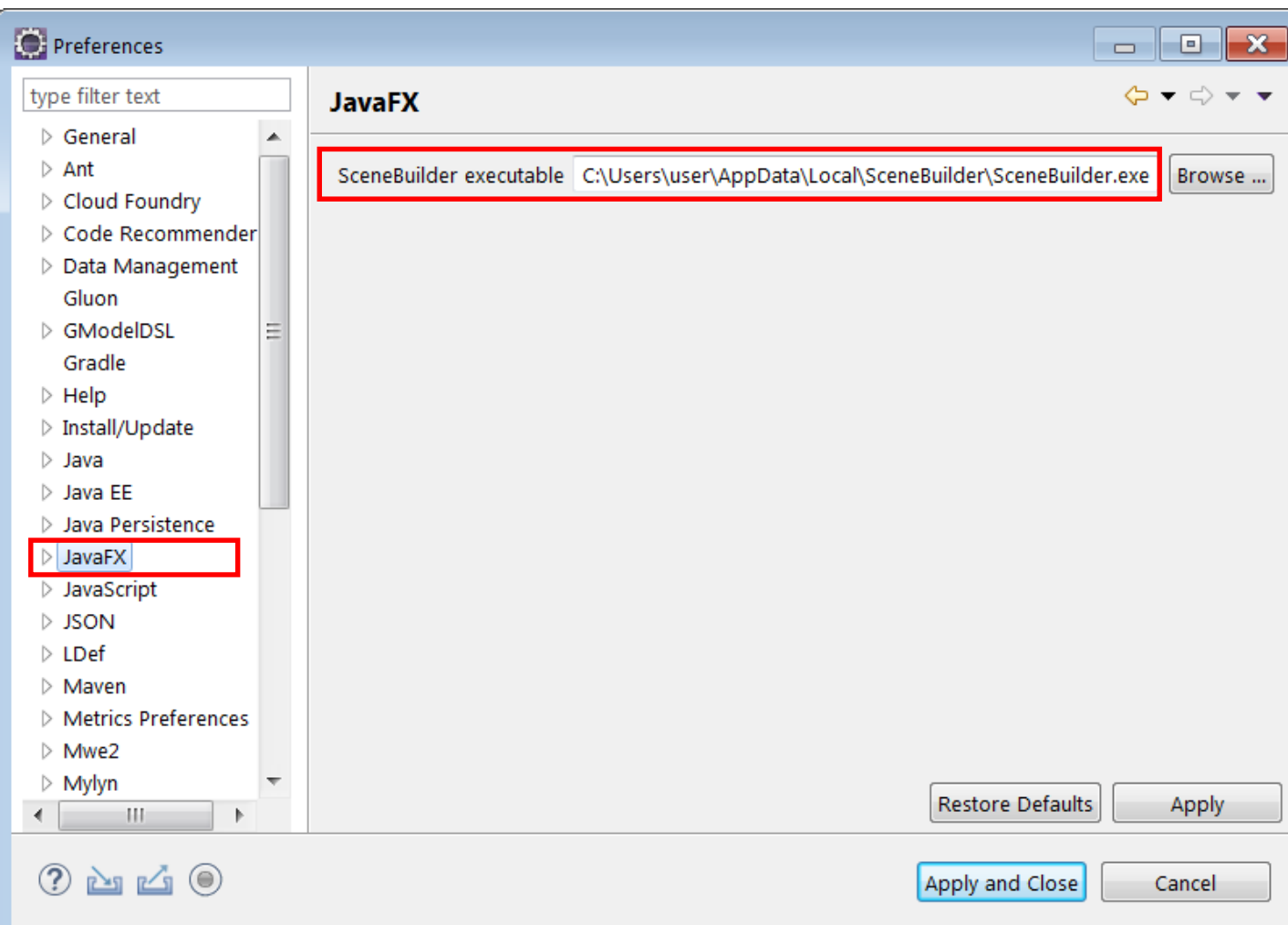


במידה וה-Scene Builder לא מוגדר, יש לבצע:

- לבחור window
- לבחור preferences



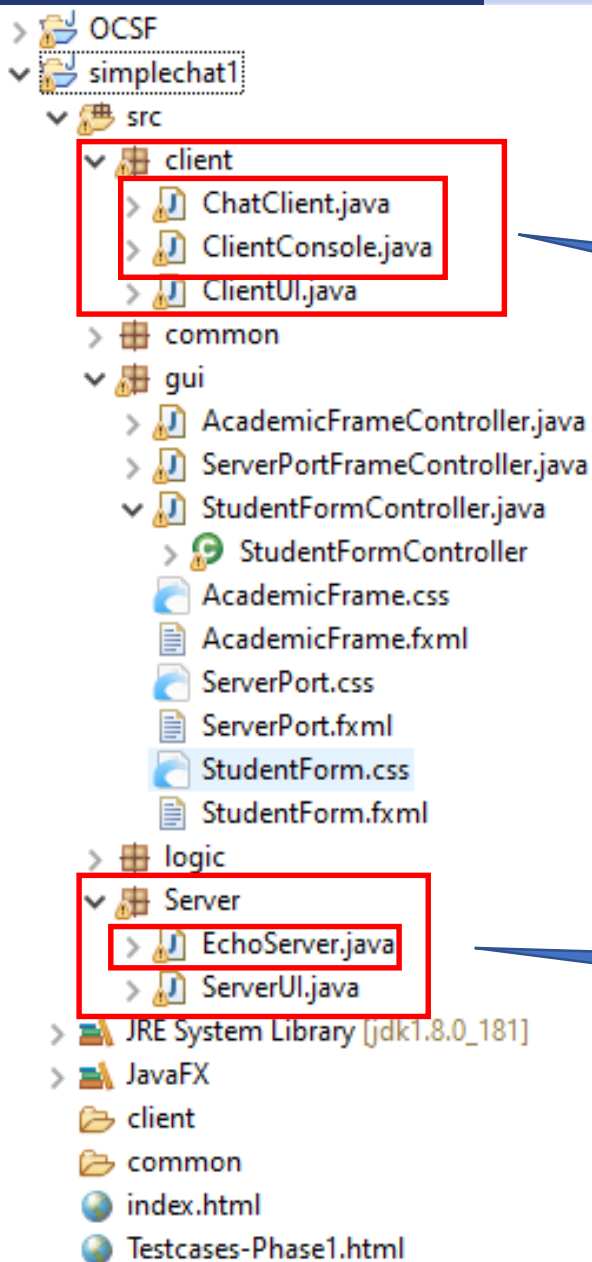
בחלון שנפתח, יש לבחור בJavaFX ולציין את המסלול לScene Builder executable:



החלק של ה Client - Server:

נלמדו בתרגיל 5

נלמד בתרגיל 5



נסתכל על תכונ המחלקה ClientUI :

```
package client;
import javafx.application.Application;

public class ClientUI extends Application {

    public static void main( String args[] ) throws Exception
    {
        launch(args);
    } // end main

    @Override
    public void start(Stage primaryStage) throws Exception {
        // TODO Auto-generated method stub

        AcademicFrameController aFrame = new AcademicFrameController(); // create StudentFrame

        aFrame.start(primaryStage);
    }
}
```

נסתכל על תכונ המחלקה ServerUI :

```
package Server;

import javafx.application.Application;

public class ServerUI extends Application {
    final public static int DEFAULT_PORT = 5555;
    public static Vector<Student> students=new Vector<Student>();

    public static void main( String args[] ) throws Exception
    {
        launch(args);
    } // end main

    @Override
    public void start(Stage primaryStage) throws Exception {
        // TODO Auto-generated method stub
        ServerPortFrameController aFrame = new ServerPortFrameController(); // create StudentFrame

        aFrame.start(primaryStage);
    }

    public static void runServer(String p)
    {
        int port = 0; //Port to listen on

        try
        {
            port = Integer.parseInt(p); //Set port to 5555
        }
        catch(Throwable t)
        {
            System.out.println("ERROR - Could not connect!");
        }

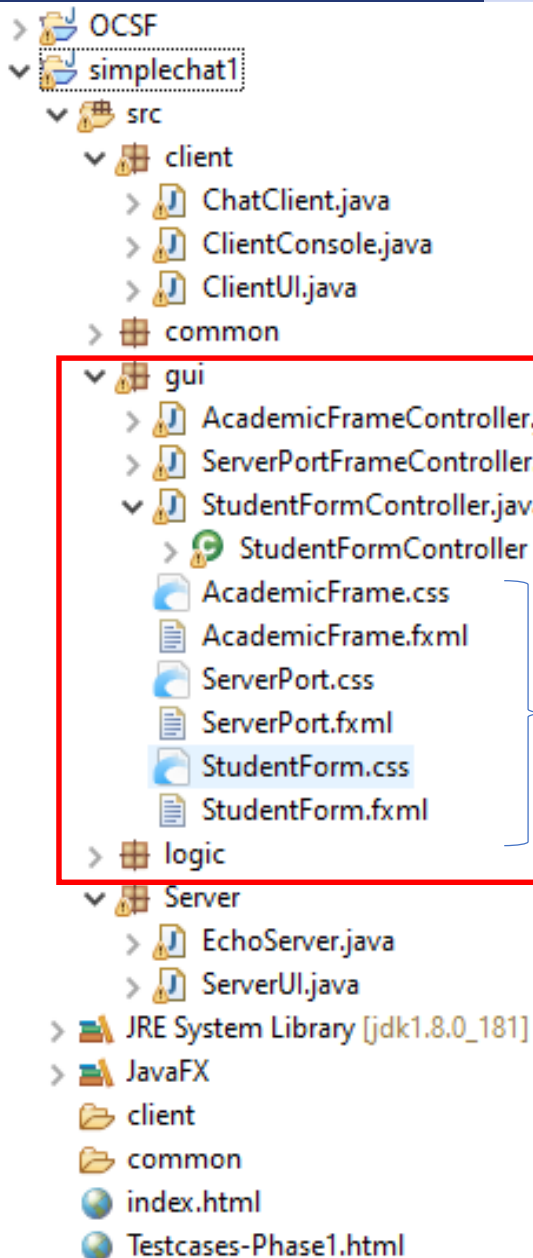
        EchoServer sv = new EchoServer(port);

        try
        {
            sv.listen(); //Start listening for connections
        }
    }
}
```

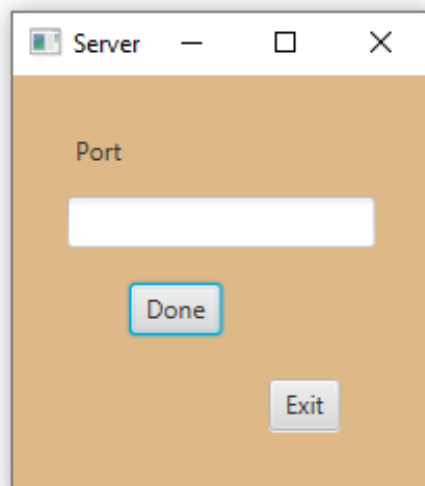
## החלק של ה javaFX:

כל אחד מהם  
אחראי על מסך

אפשר לראות  
שקיימים 3 מסכים



כעת בואו נסתכל על המסכים של ה Server



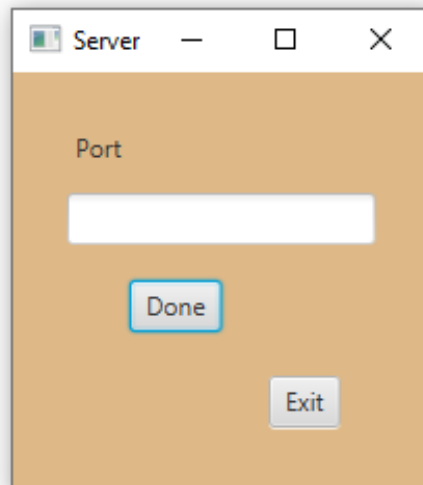
המתודה `start()` היא  
 נקודת הכניסה לכל  
 האפליקציות של  
 JavaFX

```

public void start(Stage primaryStage) throws Exception {
    Parent root = FXMLLoader.load(getClass().getResource("/gui/ServerPort.fxml"));

    Scene scene = new Scene(root);
    scene.getStylesheets().add(getClass().getResource("/gui/ServerPort.css").toExternalForm());
    primaryStage.setTitle("Server");
    primaryStage.setScene(scene);

    primaryStage.show();
}
  
```



כעת בואו נסתכל על המסכים של ה Server

- מכיל שדה טקסט, ו 2 כפתורים

```
@FXML
private Button btnExit = null;
@FXML
private Button btnDone = null;
@FXML
private Label lblList;

@FXML
private TextField porttxt;
```

- לכל כפתור נבצע פעולות שונות:

```
public void getExitBtn(ActionEvent event) throws Exception {
    System.out.println("exit Academic Tool");
    System.exit(0);
}
```



Get port from GUI

• במידה ונלחץ Done:

```
private String getport()
{
    return porttxt.getText();
}
```

```
public void Done(ActionEvent event) throws Exception {
    String p;
    ((Node)event.getSource()).getScene().getWindow().hide(); //hiding primary window
    Stage primaryStage = new Stage();
    FXMLLoader loader = new FXMLLoader();

    p=getport();
    if(p.trim().isEmpty())
    {
        System.out.println("You must enter a port number");
        Parent root1 = FXMLLoader.load(getClass().getResource("/gui/ServerPort.fxml"));

        Scene scene = new Scene(root1);
        scene.getStylesheets().add(getClass().getResource("/gui/ServerPort.css").toExternalForm());
        primaryStage.setTitle("Server");
        primaryStage.setScene(scene);

        primaryStage.show();
    }
    else
    {
        ServerUI.runServer(p);
    }
}
```

במקרה ולא הוכנס נתון  
נשארים באותו מסך

אם הנתון קיים מריצים  
את השרת

## כעת בואו נסתכל על המסכים של ה Client:

```
@FXML
private Button btnSend = null;

@FXML
private TextField idtxt;

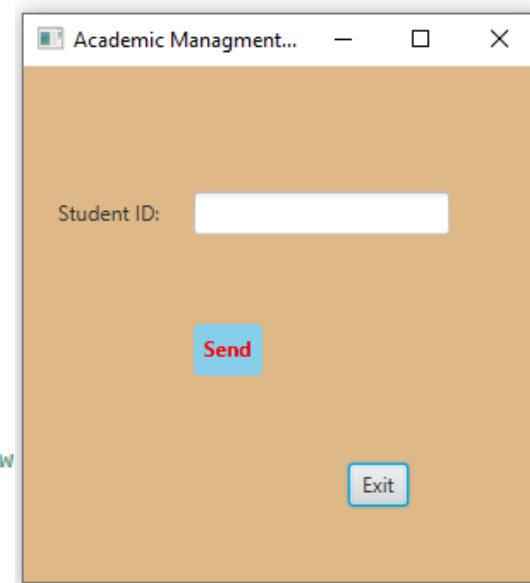
public ClientConsole chat= new ClientConsole("localhost", 5555);
private String getID()
{
    return idtxt.getText();
}

public void Send(ActionEvent event) throws Exception {
    String id;
    ((Node)event.getSource()).getScene().getWindow().hide(); //hiding primary window
    Stage primaryStage = new Stage();
    FXMLLoader loader = new FXMLLoader();

    id=getID();
    if(id.trim().isEmpty())
    {
        System.out.println("You must enter a port number");
        Parent root1 = FXMLLoader.load(getClass().getResource("/gui/AcademicFrame.fxml"));

        Scene scene = new Scene(root1);
        scene.getStylesheets().add(getClass().getResource("/gui/StudentForm.css").toExternalForm());
        primaryStage.setTitle("Academic Managment Tool");
        primaryStage.setScene(scene);
        primaryStage.show();
    }
    else
    {
        chat.accept(id);
    }
}
```

Client Connect



שליחת ההודעה ל Server

אחרי לחיצה על Send, במידה וID קיים  
נקבל מסך הבא:

- fx:id is the identity associated to component in fxml to build a controller,
- id is used for css.

```

AcademicFrame.fxml  StudentForm.fxml
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.scene.control.*?>
4 <?import java.lang.*?>
5 <?import javafx.scene.layout.*?>
6 <?import javafx.scene.layout.AnchorPane?>
7
8 <AnchorPane prefHeight="349.0" prefWidth="312.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1">
9   <children>
10    <Label id="lblName" layoutX="34.0" layoutY="107.0" text="Name:" />
11    <Label id="lblSurname" layoutX="34.0" layoutY="165.0" text="Surname:" />
12    <ComboBox id="cmbFaculty" fx:id="cmbFaculty" layoutX="99.0" layoutY="253.0" prefWidth="150.0" />
13    <Label id="lblFaculty" layoutX="34.0" layoutY="258.0" text="Faculty:" />
14    <TextField id="lblName" fx:id="txtName" layoutX="104.0" layoutY="102.0" />
15    <TextField id="lblSurname" fx:id="txtSurname" layoutX="104.0" layoutY="160.0" />
16  </children>
17 </AnchorPane>
18
  
```

## בואו נסתכל על הקוד ב-JAVA: StudentFormController

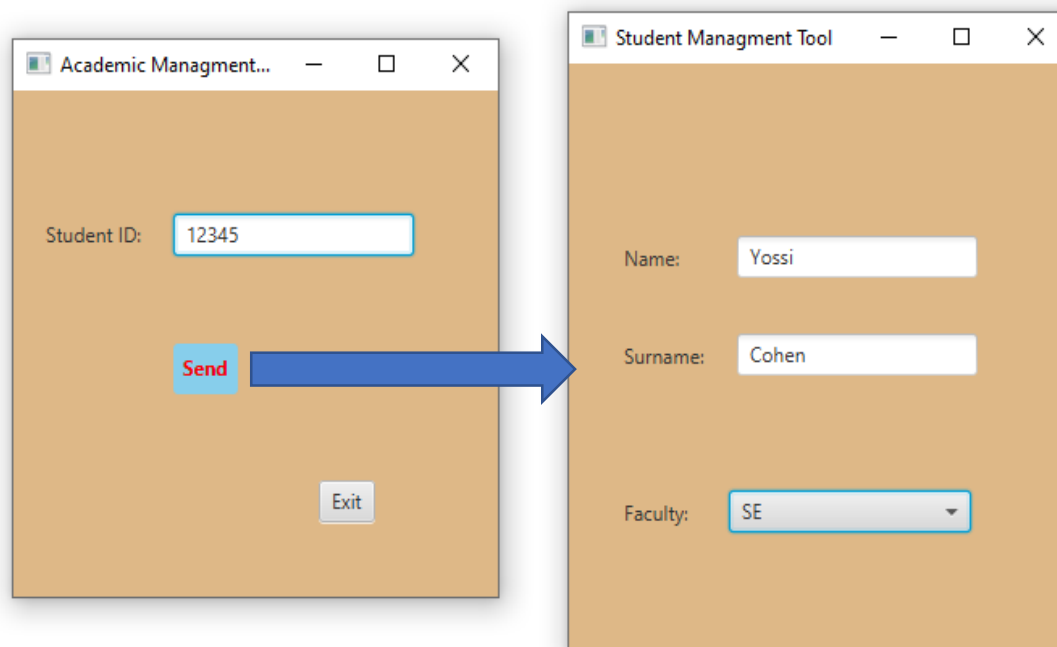
```
public void loadStudent(Student s1){
    this.s=s1;
    this.txtName.setText(s.getPName());
    this.txtSurname.setText(s.getLName());
    this.cmbFaculty.setValue(s.getFc().getFName());
}

// creating list of Faculties
private void setFacultyComboBox() {
    ArrayList<String> al = new ArrayList<String>();
    al.add("ME");
    al.add("IE");
    al.add("SE");

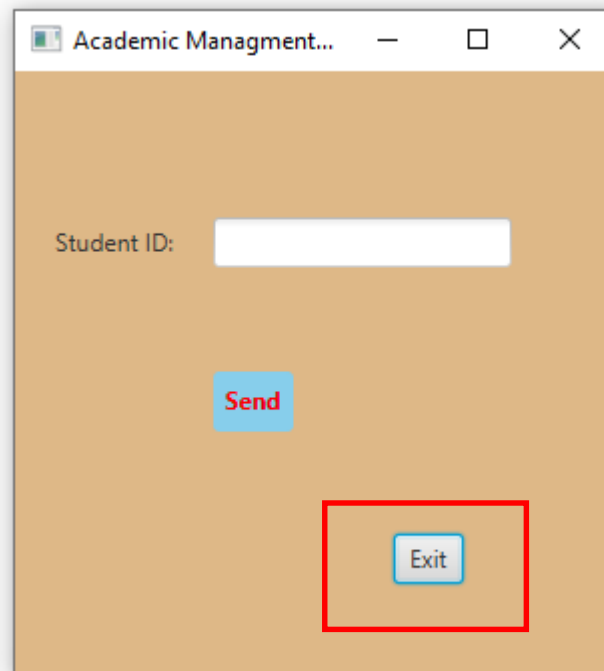
    list = FXCollections.observableArrayList(al);
    cmbFaculty.setItems(list);
}

@Override
public void initialize(URL arg0, ResourceBundle arg1) {
    setFacultyComboBox();
}
```

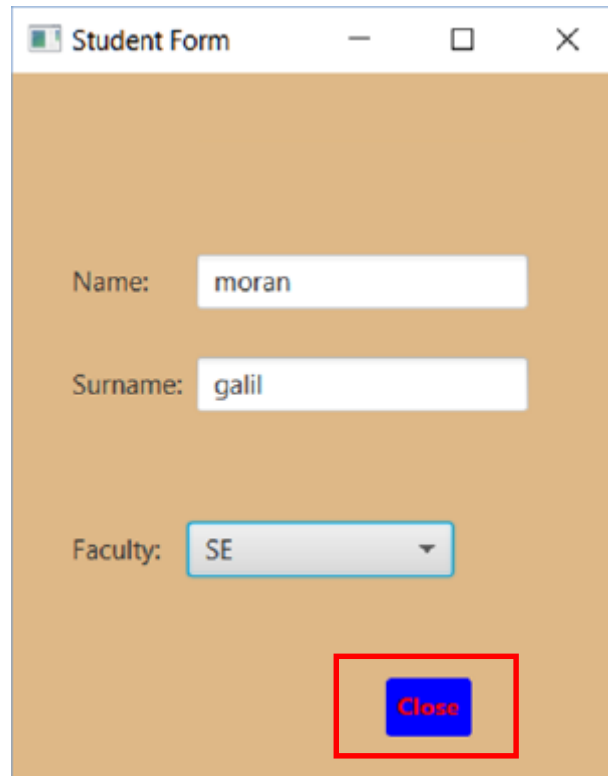
There is a project ClientServerGUI\_FX. The application presents Server that Contain some information for a student . Client that get information from server By insert id and pressing on “Send” button .the application shows details of a student (see pictures below).



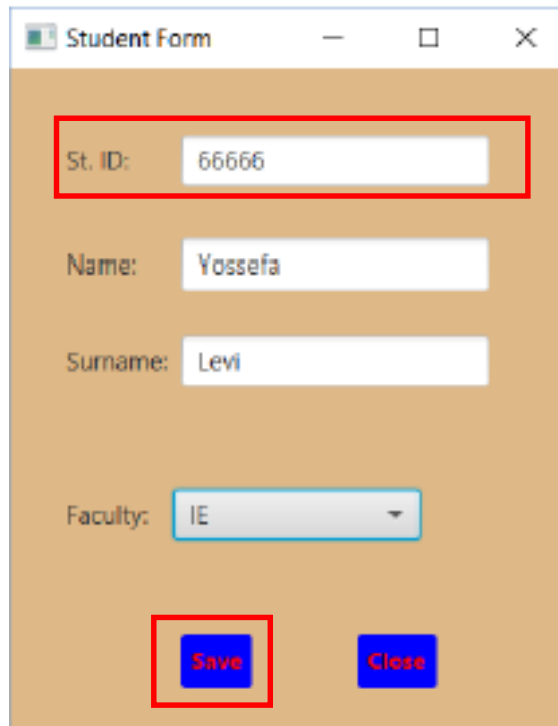
1. Make the button “Exit” is working.



2. Add a button 'Close' to the StudentForm (Button for returning to the previous window).



3. Add the label “**St. ID:**” and *TextField* for presenting a student’s ID (see “Student Form” above).
4. Add a button “**Save**” and its functionality - saving the data in the Server.



The screenshot shows a window titled "Student Form" with a tan background. It contains four input fields: "St. ID:" with the value "66666", "Name:" with the value "Yossefa", "Surname:" with the value "Levi", and "Faculty:" with a dropdown menu showing "IE". At the bottom, there are two buttons: "Save" and "Close". Red rectangles highlight the "St. ID:" field and the "Save" button.



## תהיה הגשת האב-טיפוס...

