

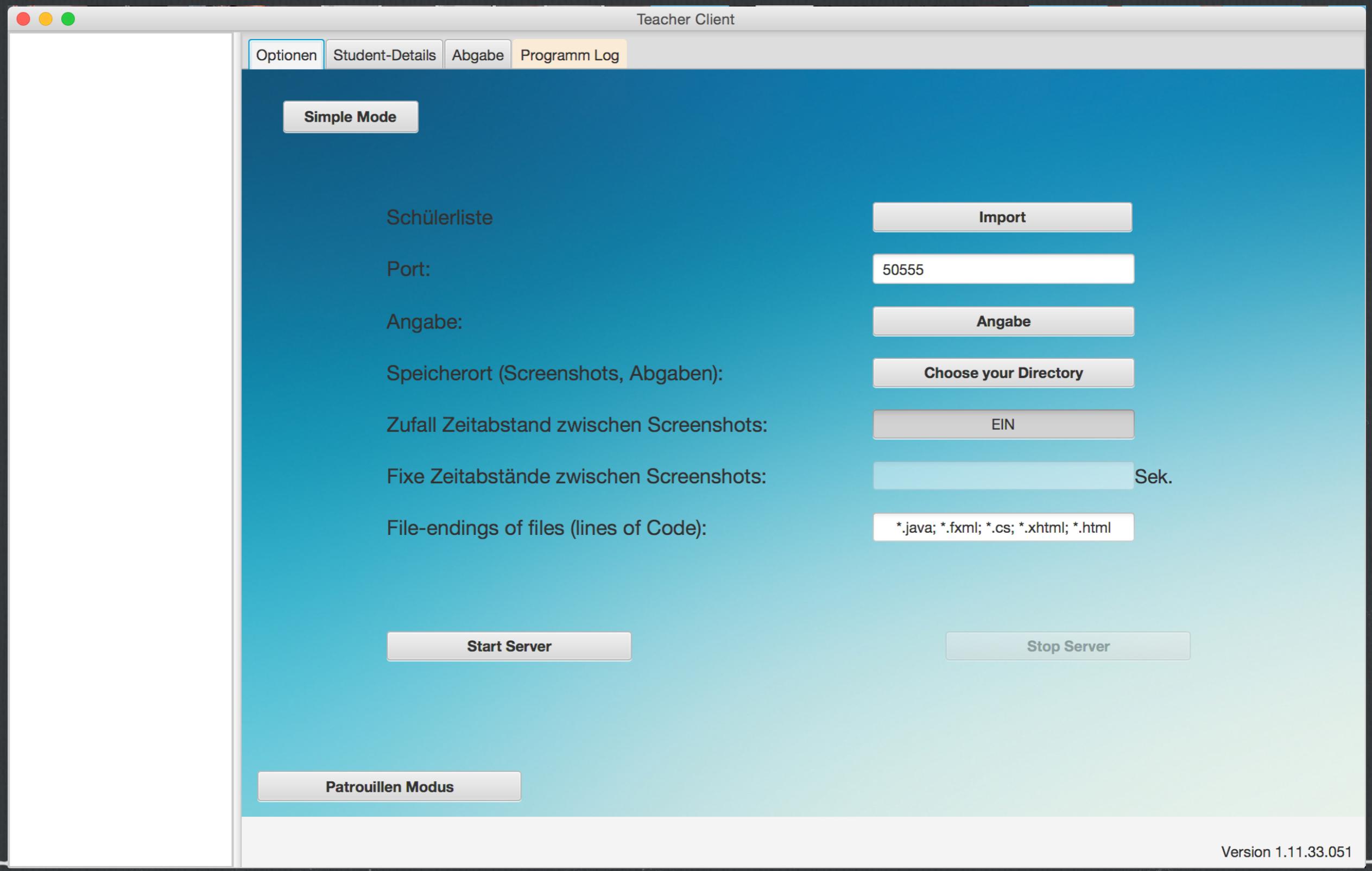
Testumgebung

Melhorn Tobias, Hermüller Philipp, Pohn Patrick, Gnadlinger Johannes

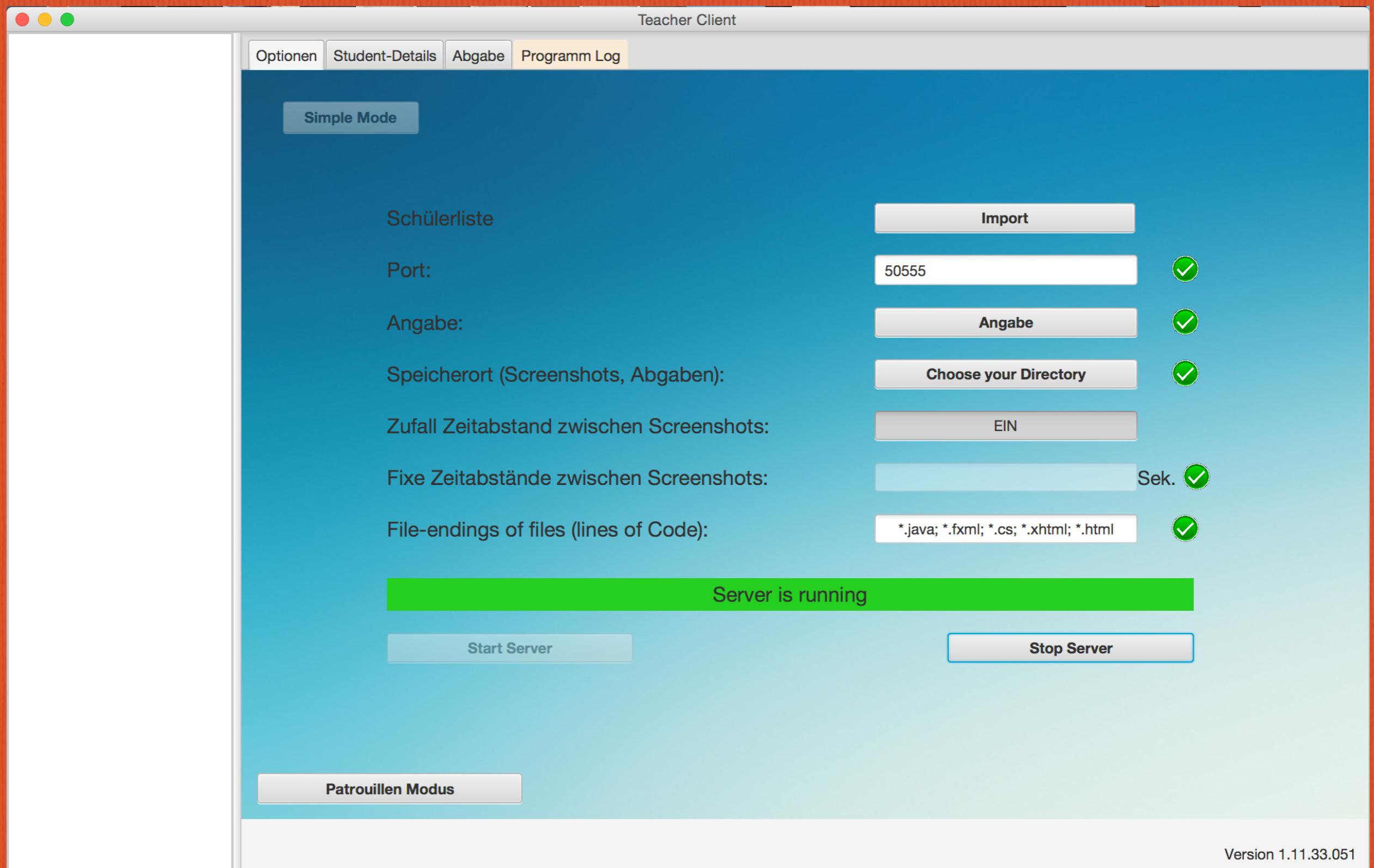
Grundsätzliche Erklärung des Projektes

- Schüler überwachen**
 - Screenshot anzeigen
 - Zeilen im Code zählen
- Benachrichtigung bei möglichen Schummelversuch**
- ...

Teacher-GUI



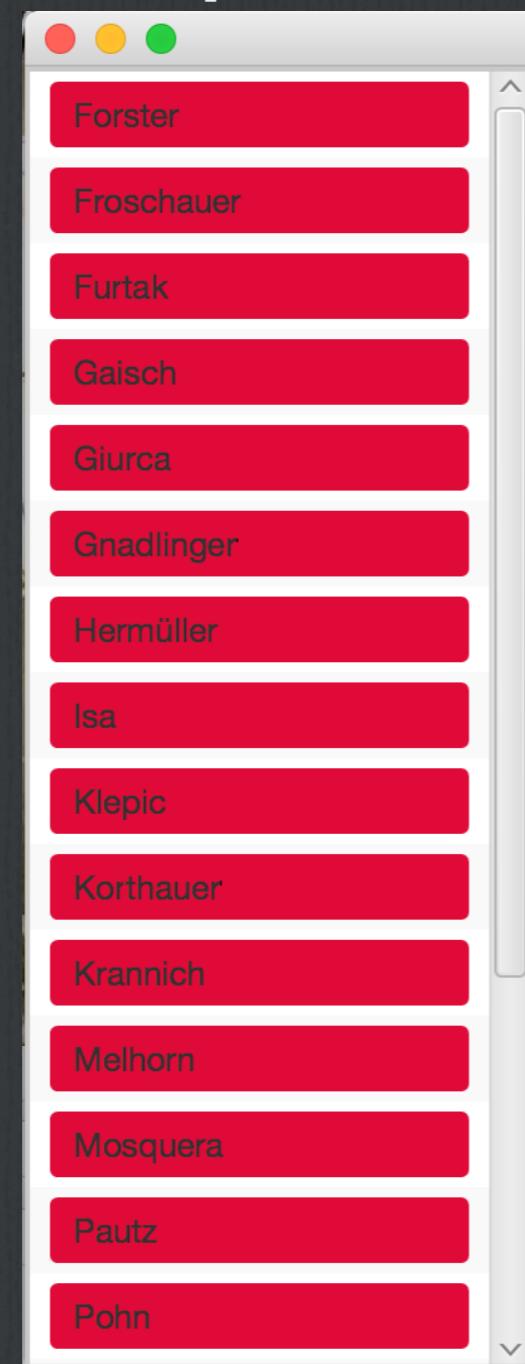
Teacher-GUI



importiert



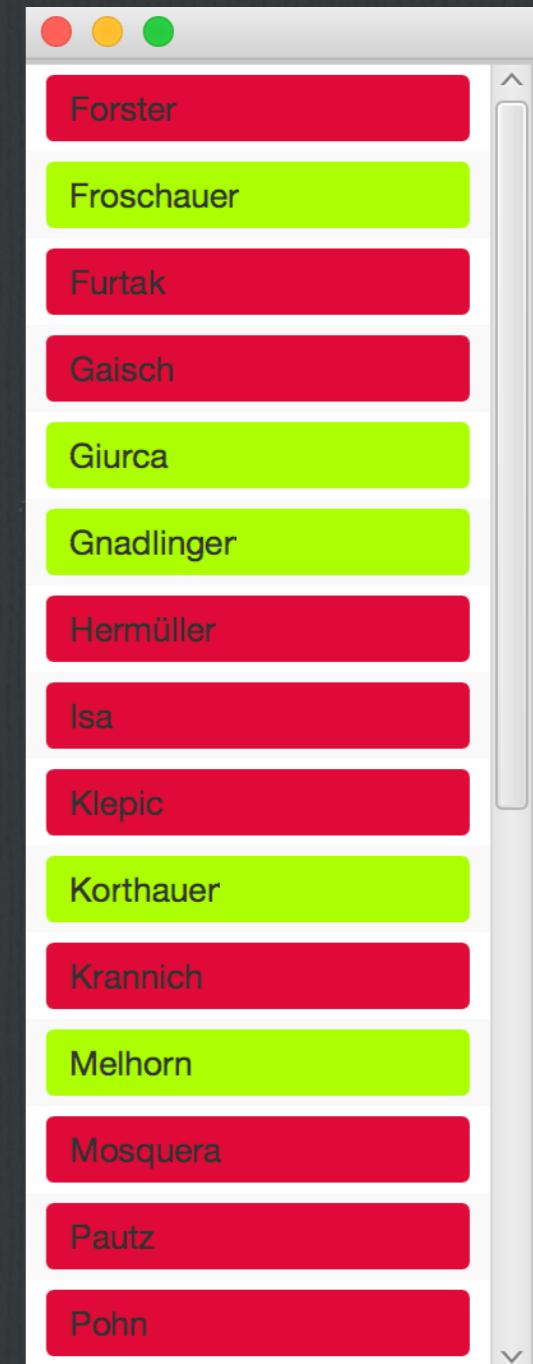
leer



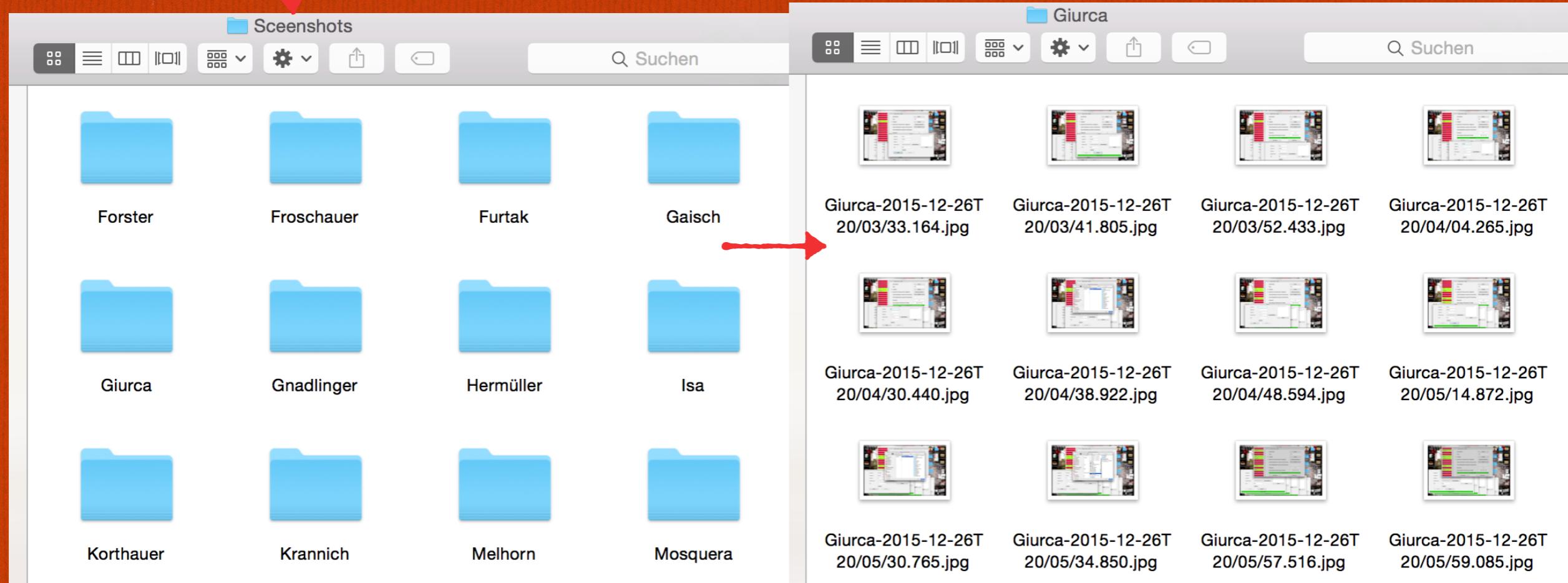
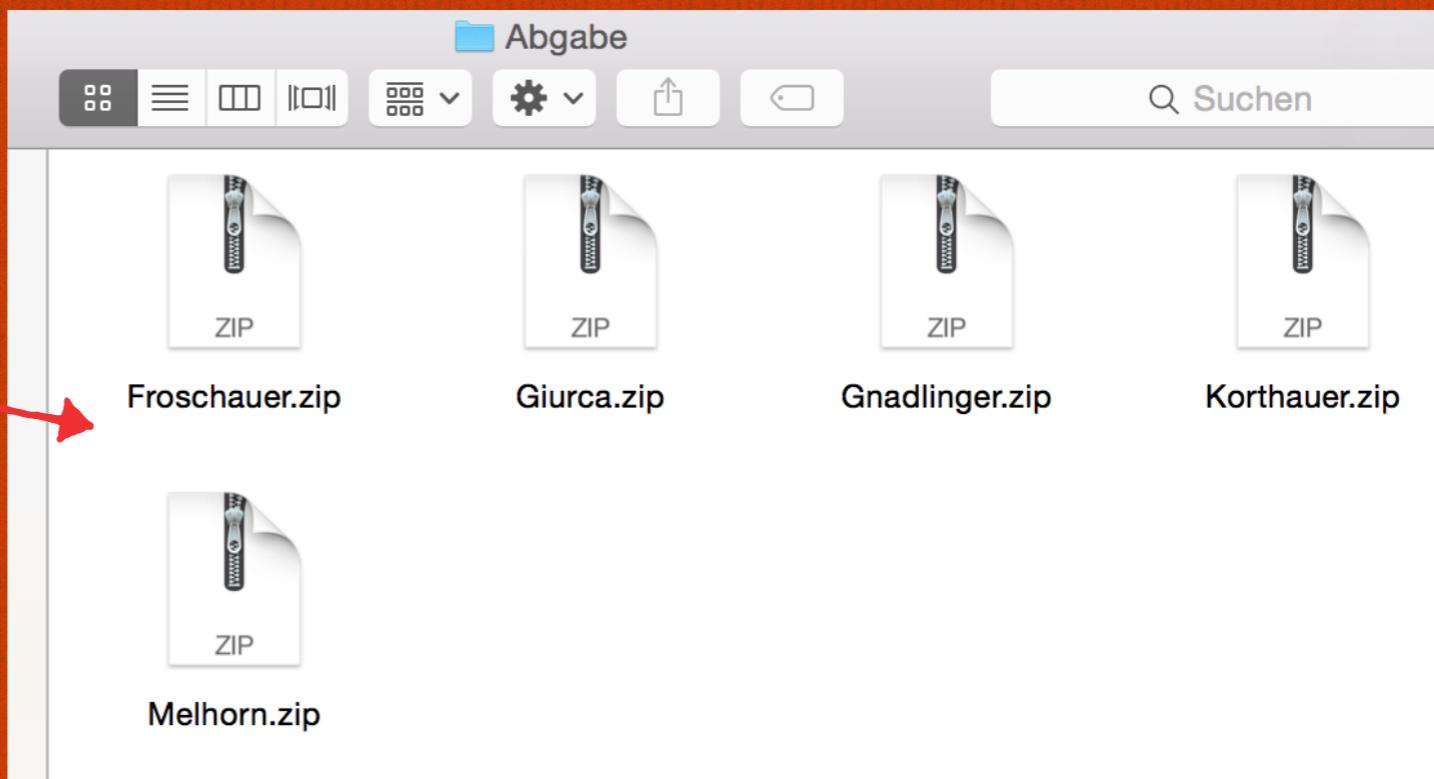
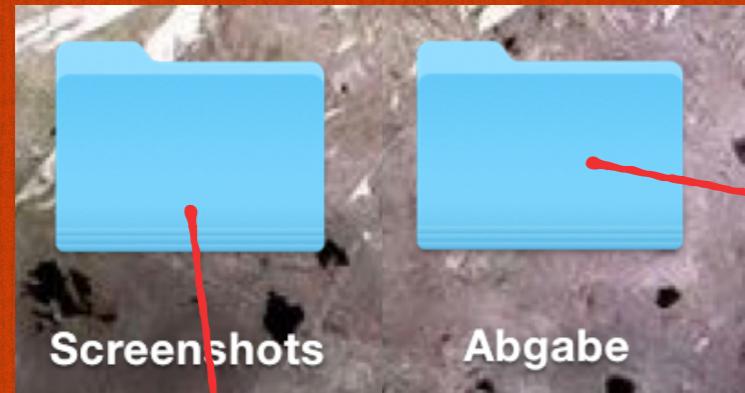
ausgeloggt

Mustermann

arbeitet



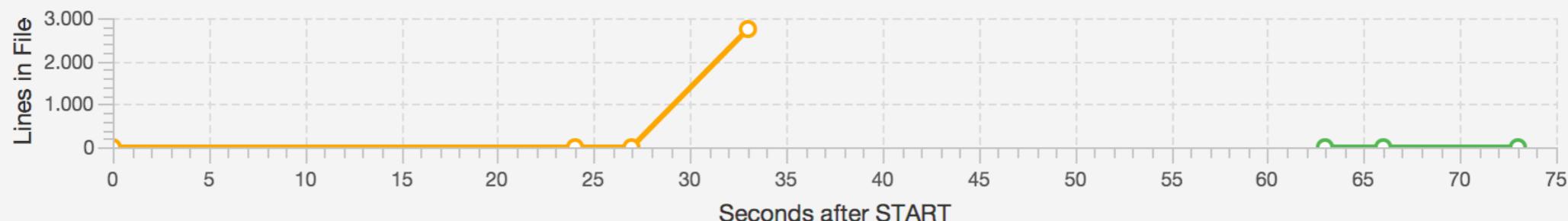
Test abgegeben



Teacher Client

Optionen Student-Details Abgabe

Lines of Code in Project



Project Structure:

```

TestumgebungNeu [RemoteControl] (~/Desktop)
  - .idea
  - out
  - src
    - main
      - java
        - at.htl.remotecontrol
          - actions
            - ClickMouse
            - MoveMouse
            - RobotAction
            - RobotActionQueue
            - ScreenShot
          - datepicker
            - DateTimePicker
            - DateTimePickerSkin
            - MainDateTimePicker
          - entity
            - Cryptography
            - Directory
            - FileStream
            - Image
            - Interval
            - LineCounter
            - Session
            - Student
            - StudentView
  - TeacherServer.java
  - ControllerTeacher.java
  - StudentGui.java
  - Threader.java
  - ControllerStudent.java
  - TeacherServer.java
  - LineCounter.java
  - Client.java

```

Code Editor (TeacherServer.java):

```

public static int PORT = 5555;

private final SocketWriterThread writer;
private final SocketReaderThread reader;

public TeacherServer(Socket socket) throws IOException, ClassNotFoundException {
    ObjectOutputStream out = new ObjectOutputStream(socket.getOutputStream());
    ObjectInputStream in = new ObjectInputStream(
        new BufferedInputStream(
            socket.getInputStream()));
    System.out.println("waiting for student name ...");

    LoginPacket packet = (LoginPacket) in.readObject();
    Student student = new Student(packet.getUserName(), packet.getDirOfWatch());
    System.out.println("I got the Package: " + packet.getDirOfWatch());
    Session.getInstance().loginStudent(student);

    reader = new SocketReaderThread(student, in, this);
    writer = new SocketWriterThread(student, out);

    reader.setDaemon(true);
    writer.setDaemon(true);

    writer.handOut();

    reader.start();
    writer.start();

    System.out.println("finished connecting to " + socket);
}

```

Run Log:

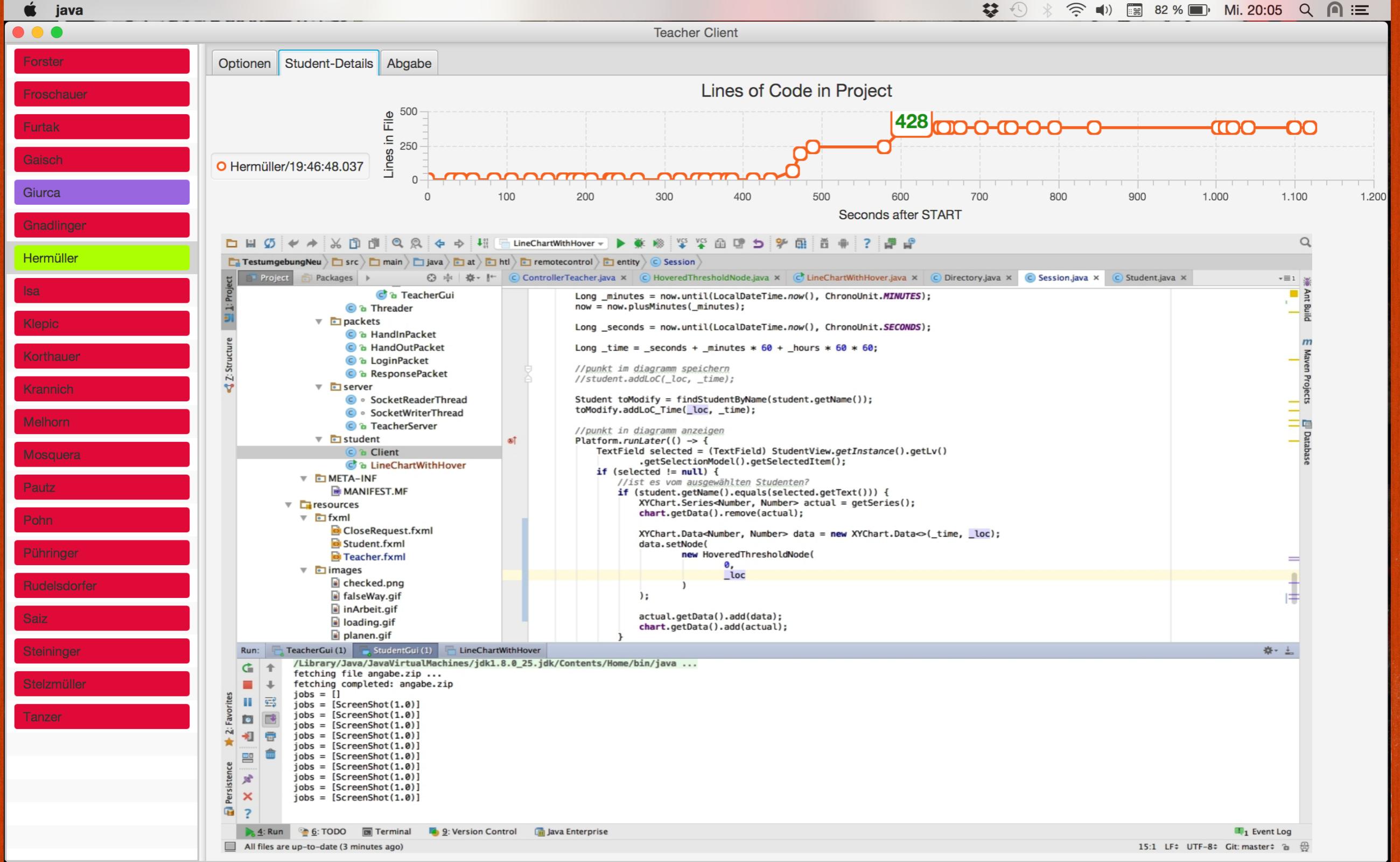
```

2: Favorites
Run: TeacherGui (1) StudentGui (1)
  jobs = [ScreenShot(1.0)]
  jobs = [ScreenShot(1.0)]
  jobs = [ScreenShot(1.0)]
  create zip file handInFile.zip ...
  created zip archive
  sending handInFile.zip ...
  sending completed: handInFile.zip
  ERFOLGREICH ==> true
  Connection closed abruptly: java.net.SocketException: Socket closed
  Directory /Users/Philipp/Desktop/Gurca already exists!
  fetching file angabe.zip ...
  fetching completed: angabe.zip
  jobs = [ScreenShot(1.0)]
  jobs = [ScreenShot(1.0)]
  jobs = [ScreenShot(1.0)]

```

Bottom Status Bar:

All files are up-to-date (2 minutes ago)



Screenshot Statistik

- 20 Schüler * 3 Stunden**
-
- 12.300 Screenshots**
- 8.49 GB**
- 1 Stunde**
- ~205 Screenshots**
- 141,5 MB für alle Screenshots**
- 782KB pro Screenshot.**

Programm Log

Teacher Client

Mustermann

Optionen Student-Details Abgabe Programm Log

```
WARN - Directory /Users/Philipp/Desktop/Screenshots is already exist!
WARN - Directory /Users/Philipp/Desktop/Abgabe is already exist!
INFO - class at.htl.timemonitoring.server.Server - waiting for client name ...
INFO - created directory /Users/Philipp/Desktop/Screenshots/Mustermann
INFO - class at.htl.timemonitoring.server.Server - I got the Package: /Users/Philipp/Desktop/Mustermann
INFO - class java.lang.Class - sending Angabe.zip ...
INFO - class java.lang.Class - sending completed: Angabe.zip
INFO - class at.htl.timemonitoring.server.Server - finished connecting to Socket[addr=/192.168.0.107, port=64722, localport=50555]
INFO - class java.lang.Class - sending Angabe.zip ...
INFO - class java.lang.Class - sending completed: Angabe.zip
```

< >

Create JAR File Export ErrorLog

< >

Version 1.11.33.051

Programm Log

```
WARN - Directory /Users/Philipp/Desktop/Screenshots is already exist!
WARN - Directory /Users/Philipp/Desktop/Abgabe is already exist!
INFO - class at.htl.timemonitoring.server.Server - waiting for client name ...
INFO - created directory /Users/Philipp/Desktop/Screenshots/Mustermann
INFO - class at.htl.timemonitoring.server.Server - I got the Package: /Users/Philipp/Desktop/Mustermann
INFO - class java.lang.Class - sending Angabe.zip ...
INFO - class java.lang.Class - sending completed: Angabe.zip
INFO - class at.htl.timemonitoring.server.Server - finished connecting to Socket[addr=/192.168.0.107,port=64722,localport=50555]
INFO - class java.lang.Class - sending Angabe.zip ...
INFO - class java.lang.Class - sending completed: Angabe.zip
```

- INFO's sind in weiß dargestellt**
- WARNING's sind in gelb dargestellt**
- ERROR's sind in rot dargestellt**

Student-GUI

Student

Test

Server-IP: 192.168.0.107 / 50555 Test Connection

Matrikel-Nr.: he120016

Schüler: 01 Max Mustermann

Verzeichnis: /Users/Philipp/Desktop/Mustermann Auswählen

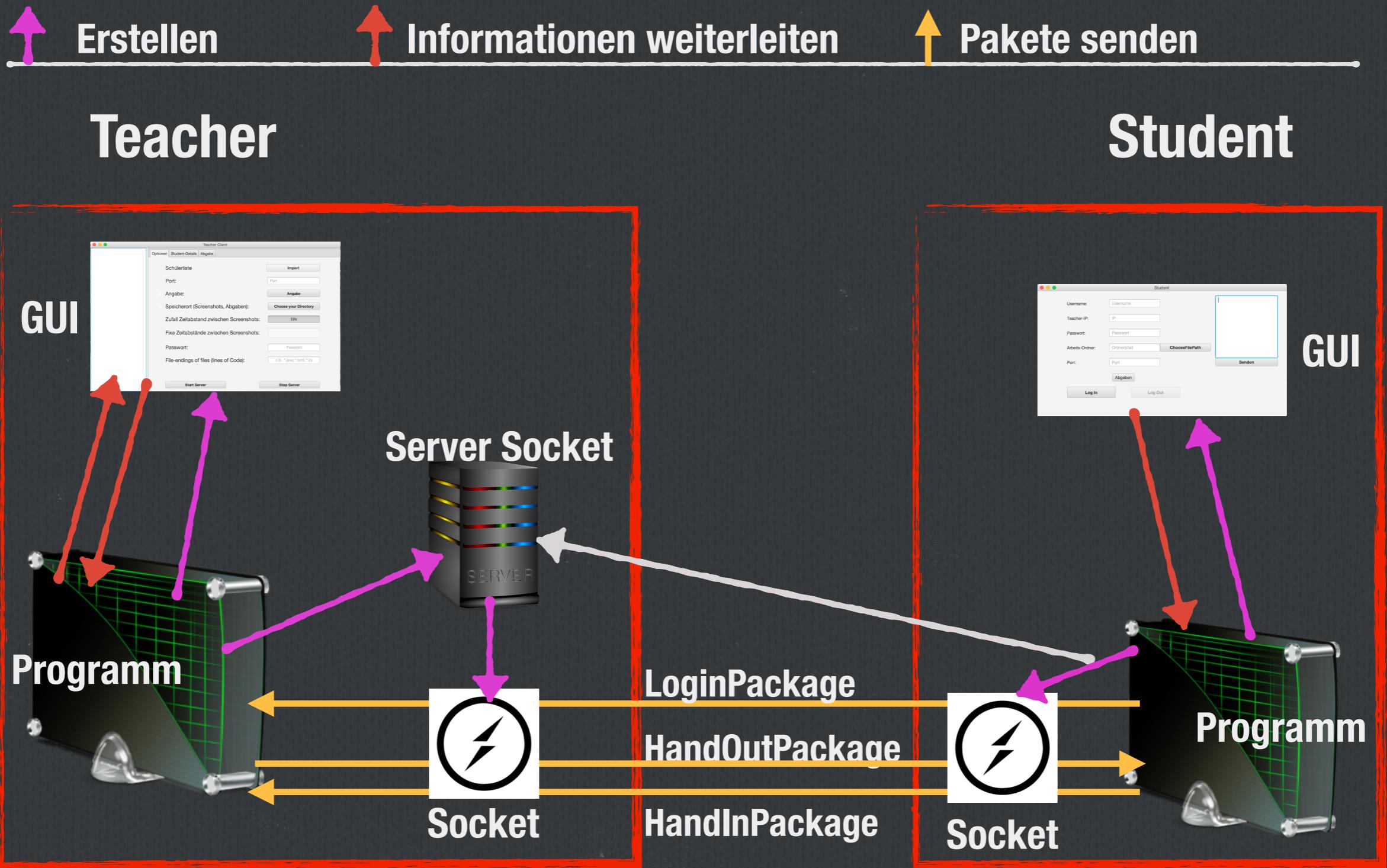
Anmelden Abmelden

Zusatz: Dauer (+ Endzeit), Chat mit Lehrer

Signed in!

The screenshot shows a Mac OS X style window titled "Student". Inside, there's a "Test" button. Below it, a "Server-IP:" field contains "192.168.0.107" and a port field contains "50555", with a "Test Connection" button next to them. A "Matrikel-Nr." field contains "he120016". Under "Schüler:", three buttons show "01", "Max", and "Mustermann". A "Verzeichnis:" field contains "/Users/Philipp/Desktop/Mustermann" with an "Auswählen" button. At the bottom, "Anmelden" and "Abmelden" buttons are shown, with "Abmelden" being highlighted. A note "Zusatz: Dauer (+ Endzeit), Chat mit Lehrer" is at the bottom. A green bar at the very bottom says "Signed in!".

Umsetzung (Prinzip)



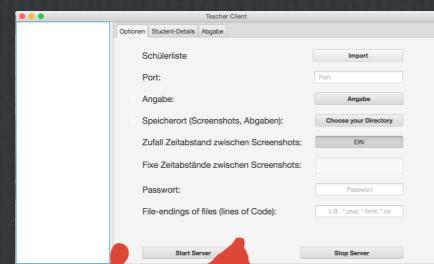
Umsetzung (Harvester)

Teacher

*Screenshot erstellen
und Lines of Code zählen

Student

GUI



Inhalt der Ernte anzeigen

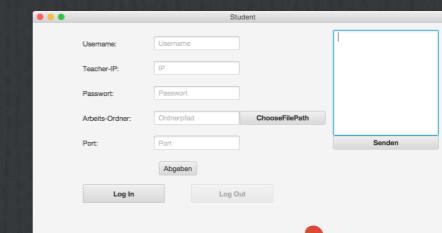
Programm



Socket

sendet Harvester

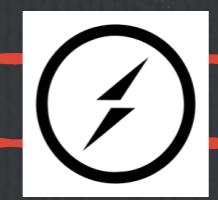
sendet
harvestedPackage



harvested*



Programm



Socket