



รายงาน ความคืบหน้าของโครงการ

หัวข้อ Quiz Program

จัดทำโดย

รายชื่อสมาชิกกลุ่ม กลุ่มที่ 21

นายกวีพล	ขุนศรี	รหัสนักศึกษา 60070003
นายชยากร	เทียนขาว	รหัสนักศึกษา 60070013

นำเสนอ

ผศ.ดร.ธนิศา นุ่มนนท์

รายงานนี้เป็นส่วนหนึ่งของวิชา OBJECT-ORIENTED PROGRAMMING ( OOP )

## บทนำ

เนื่องจากในภาคการเรียนที่ 1/2561 ได้มีการเรียนในรายวิชา OBJECT-ORIENTED PROGRAMMING ( OOP ) และได้ศึกษาเกี่ยวกับหลักการเขียนโปรแกรมเชิงอ็อบเจกต์ และได้นำความรู้ที่ได้ศึกษามาต่อยอดในการทำโปรแกรมตอบคำถามและเช็คชื่อ ( Quiz Game ) มีวัตถุประสงค์เพื่อที่จะได้ทบทวนความรู้ในวิชาหลังชั่วโมงเรียนและเป็นการเช็คชื่อไปในตัว โปรแกรมนี้มีประโยชน์ต่อนักศึกษาและอาจารย์ที่สอนในรายวิชานั้นๆ

## วัตถุประสงค์

1. ) เพื่อเพิ่มความสะดวกสบายในการเช็คชื่อของอาจารย์
2. ) ลดเวลาในการเช็คชื่อ
3. ) นักศึกษาได้ทบทวนความรู้ที่ได้เรียนไป

## ประโยชน์ที่คาดว่าจะได้รับ และการนำไปใช้งานได้จริง

สำหรับตัวผู้จัดทำโครงการจะได้รับความรู้ความเข้าใจในการพัฒนาโปรแกรมเชิงอ็อบเจกต์ ทำให้ได้ทำงานร่วมกับสมาชิกในทีมได้ฝึกการทำงานเป็นทีม และสำหรับบุคคลที่นำโครงการไปใช้สำหรับอาจารย์ก็จะลดเวลาในการเช็คชื่อและมีการเช็คการเข้าชั้นเรียนของนักศึกษาได้ สำหรับนักศึกษานั้นก็จะได้ทบทวนความรู้ที่ได้ศึกษาไป ในบางครั้งอาจจะมีส่วนที่ไม่เข้าใจก็สามารถถามอาจารย์ได้เลย เพราะหลังจากที่ตอบคำถามในแต่ละข้อเสร็จแล้วนั้นจะมีการแสดงว่าคำตอบที่ตนตอบนั้นถูกหรือผิดรวมทั้งแสดงจำนวนคนที่ตอบในแต่ละข้อ อาจารย์สามารถอธิบายให้นักศึกษาต่อได้ว่าทำไมถึงตอบข้อนั้น

## รายละเอียดโปรแกรมที่จะพัฒนาในเชิงเทคนิค

เป็นโปรแกรมที่จะให้นักศึกษาได้ทบทวนความรู้หลังจากจบชั่วโมงไปแล้วและทำการเช็คชื่อนักศึกษาไปในตัวด้วย ซึ่งจะทำให้ไม่ต้องเช็คชื่อภายในห้องเรียนแต่จะเช็คชื่อจากการทำการทบทวนความรู้ ซึ่งระบบจะแบ่งเป็น 2 ส่วนใหญ่ๆ คือ ส่วนอาจารย์และส่วนนักศึกษา

1.) ส่วนอาจารย์ : โปรแกรมสำหรับอาจารย์นั้นจะมีหน้า Log in เหมือนกับของนักศึกษา แต่หลังจากที่เข้าสู่ระบบแล้ว โปรแกรมสำหรับอาจารย์นั้นจะมีหน้า เพิ่ม/ลบ/แก้ไข แบบทบทวนในแต่ละสัปดาห์และสามารถดูรายชื่อนักศึกษาที่ได้ทำแบบทดสอบในสัปดาห์นั้นๆได้ หลังจากนักศึกษาทำแบบทดสอบในสัปดาห์นั้นและหมดเวลาตามที่กำหนด ในส่วนของอาจารย์นั้นจะ

ปรากฏรายชื่อนักศึกษาที่ได้ทำแบบทดสอบในครั้งนั้นๆ และมีปุ่มในการเช็คชื่อและเก็บข้อมูลลงฐานข้อมูล

**2.) ส่วนนักศึกษา :** โปรแกรมสำหรับนักศึกษานั้นจะมีหน้า Log in เหมือนกับของอาจารย์แต่จะไม่สามารถ ลบ/แก้ไข ส่วนต่างๆได้เหมือนกับส่วนของอาจารย์ แต่จะปรากฏแค่แบบทดสอบนั้นสัปดาห์ที่ทางได้กำหนดไว้แล้ว และจะต้องทำแบบทดสอบนั้นตามเวลาที่อาจารย์กำหนด หลังจากทำแบบทดสอบเสร็จแล้ว จะปรากฏหน้าคะแนนอันดับ 1 - 5 ในครั้งนั้นๆ และปรากฏคะแนนตัวเองด้วย

### Input/Output Specification

**Input :** รหัสสำหรับนักศึกษา/อาจารย์, คำตอบในแต่ละข้อ

**Output :** คะแนนอันดับ 1 - 3 และคะแนนตนเอง, รายชื่อนักศึกษาที่ได้ทำแบบทดสอบนั้นตามเวลาที่กำหนด

### Functional Specification

จะแบ่งส่วนต่างๆออกเป็น 2 ส่วน ซึ่งส่วนของอาจารย์จะมีระบบที่สามารถ เพิ่ม/แก้ไข/ลบ ส่วนต่างๆของโปรแกรมได้ ส่วนนักศึกษาสามารถทำแบบทดสอบได้เท่านั้นเพื่อเป็นการเช็คชื่อเข้าชั้นเรียน สำหรับหน้าแสดงผลหลังจากจบแบบทดสอบจะมีส่วนที่แตกต่างกัน ส่วนของอาจารย์จะแสดงรายชื่อทั้งหมดพร้อมคะแนนนักศึกษาที่ได้ทำแบบทดสอบและมีปุ่มเก็บข้อมูลลงฐานข้อมูล ส่วนนักศึกษามีแค่ส่วนแสดงว่าตนได้คะแนนเท่าไรและใครบ้างที่ได้อันดับ 1 - 5

### ขอบเขตและข้อจำกัดของโปรแกรมที่จะพัฒนา

- สำหรับนักศึกษาจะต้องเข้าสู่ระบบและเข้าแบบทดสอบในสัปดาห์นั้นก่อนที่อาจารย์จะกดเริ่ม
- นักศึกษาสามารถเห็นคะแนนแค่ส่วนของตัวเองและคะแนน 1 - 5 ได้เท่านั้น
- ระยะเวลาในแต่ละข้อจะมีทั้งหมด 20 วิ
- ระบบจะทำการเก็บคะแนนผ่านทางฐานข้อมูล จำเป็นต้องเชื่อมต่ออินเทอร์เน็ต
- หลังจบคำถามในแต่ละข้อจะมีการแสดงข้อที่ถูกและข้อที่ผิด รวมทั้งแสดงจำนวนที่ตอบในแต่ละข้อ อาจารย์สามารถอธิบายต่อได้ว่าทำไมถึงตอบข้อนี้สำหรับนักศึกษาที่สงสัย

ตารางระบุว่า ใครจะรับผิดชอบส่วนใดและกำหนดระยะเวลาเท่าใด

กิจกรรม	ช่วงเวลา	ผู้รับผิดชอบ
ออกแบบและวางแผนโครงการ	2 - 5 ตุลาคม 2561	กวีพล/ชยากร/นฤเทพ
พัฒนาพื้นฐานของโปรแกรม	6 - 15 ตุลาคม 2561	กวีพล/ชยากร/นฤเทพ
พัฒนาระบบเข้าสู่ระบบ	16 - 20 ตุลาคม 2561	กวีพล/ชยากร/นฤเทพ
พัฒนาการตอบคำถาม	21 - 31 ตุลาคม 2561	กวีพล/ชยากร/นฤเทพ
พัฒนาระบบเก็บคะแนน	1 - 2 พฤษภาคม 2561	กวีพล/ชยากร/นฤเทพ
พัฒนาระบบฐานข้อมูล	3 - 10 พฤษภาคม 2561	กวีพล/ชยากร/นฤเทพ
พัฒนาส่วนต่อประสานกราฟิกกับผู้ใช้	11 - 15 พฤษภาคม 2561	กวีพล/ชยากร/นฤเทพ
พัฒนาการแสดงผลสำหรับอาจารย์ และแก้ไขข้อผิดพลาดต่างๆของโปรแกรม	15 - 31 พฤษภาคม	กวีพล/ชยากร/นฤเทพ
ทดสอบและปรับปรุงแอปพลิเคชัน	1 - 10 ธันวาคม 2561	กวีพล/ชยากร/นฤเทพ

## Hooza ( Quiz Game )

Hooza คือ โปรแกรมที่ไว้ใช้สำหรับการสร้างคำถามต่าง ๆ เพื่อใช้กับเรื่องต่าง ๆ ไม่ว่าจะเป็น การทบทวนคำถามหลังเรียน, การใช้ในการสอบ ฯลฯ โดยตัวโปรแกรมจะแบ่งออกเป็น 2 ส่วนคือ Teacher ในส่วนของ Teacher จะมีตัวเลือกสำหรับการสร้างคำถามและการสร้าง Lobby ซึ่งส่วนของ Student จะมีแค่ Join เข้าสู่ Lobby ของ Teacher เท่านั้น

\*\*\* สำหรับรหัสในการใช้งานช่วงพัฒนานั้น \*\*\*

Teacher คือ Username : 0001 Password : 1234

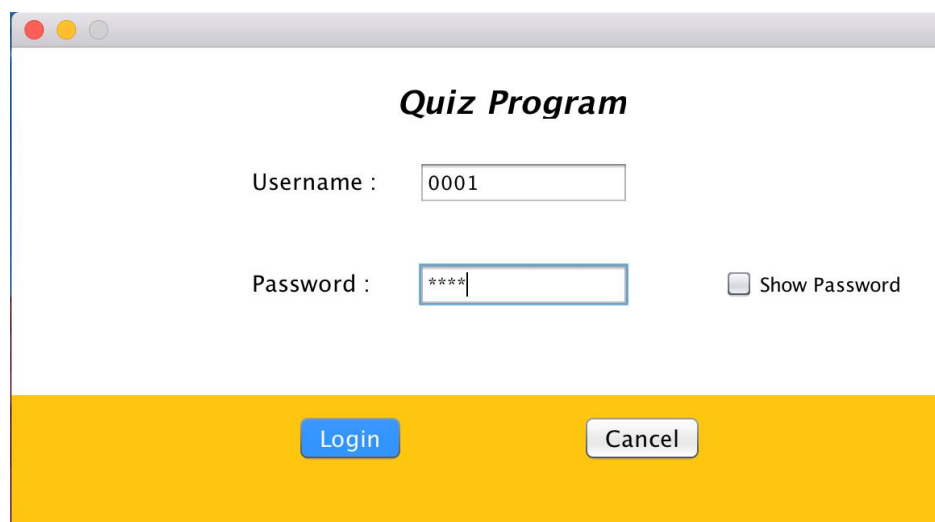
Student คือ Username : 60070003 Password : 1234

## ภาษาที่ใช้ในการพัฒนา Hooza



## ขั้นตอนการทำงานของ Hooza

- 1.) เมื่อเปิดโปรแกรมขึ้นมาแล้วจะพบกับหน้าเข้าสู่ระบบ ซึ่งข้อมูลของผู้ใช้ต่าง ๆ ไม่ว่าจะเป็น ชื่อ, นามสกุล, รหัสนักศึกษา, คณะ จะถูกเก็บอยู่บน Database Online ซึ่งจะต้องเพิ่ม Library สำหรับการเชื่อมต่อ Database ซึ่ง Library เมื่อ Clone Project มาแล้วจะมาพร้อมกับ mysql-connector-java-8.0.13.jar ซึ่งเป็น Library สำหรับการเชื่อมต่อ Database

A screenshot of a Java Swing window titled "Quiz Program". It has a white background and a yellow footer bar. The window contains two text input fields: "Username :" with the value "0001" and "Password :" with the value "\*\*\*\*". To the right of the password field is a checkbox labeled "Show Password". At the bottom of the window, there are two buttons: "Login" and "Cancel".

**Quiz Program**

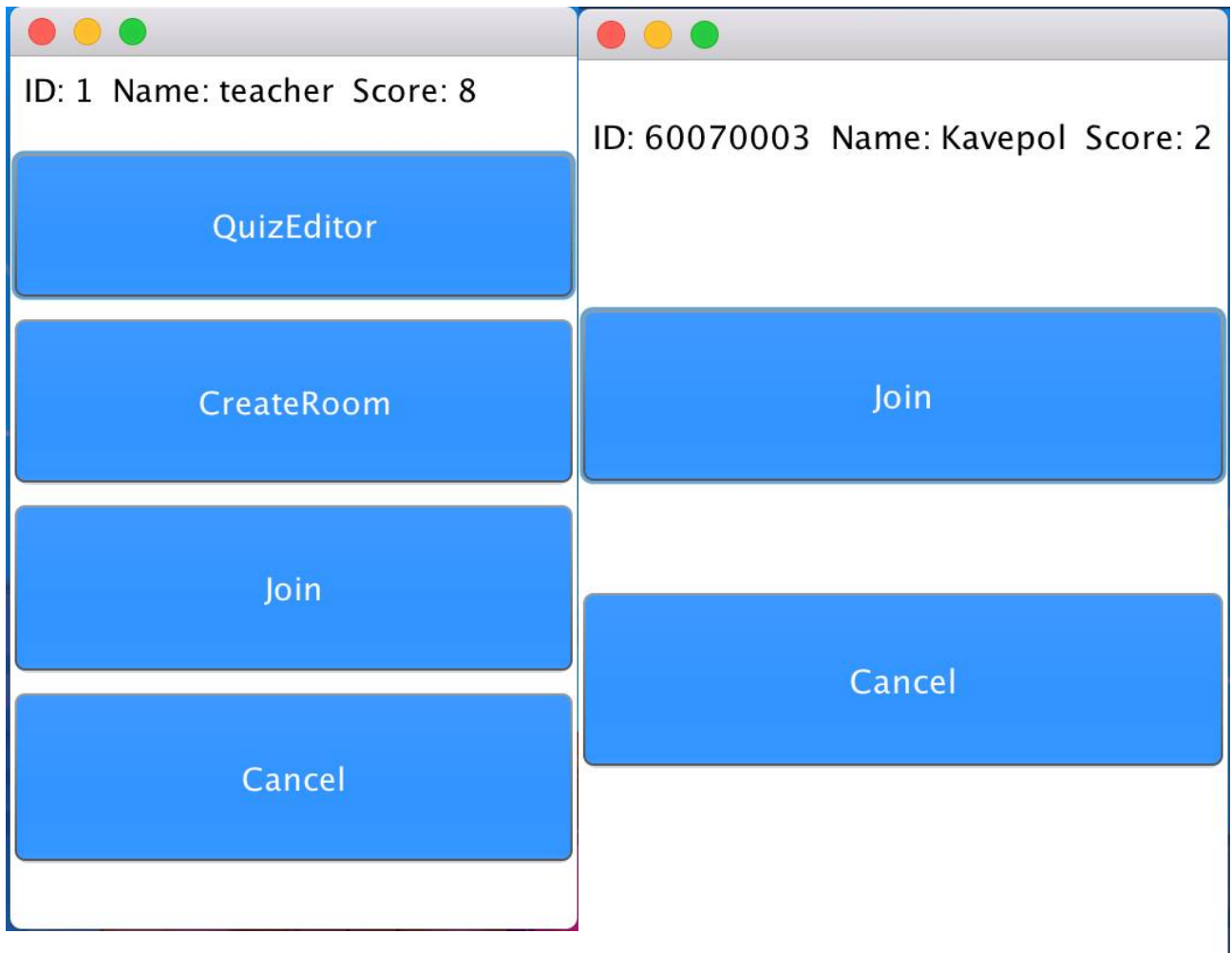
Username : 0001

Password : \*\*\*\* ☐ Show Password

Login Cancel



2.) หลังจากเข้าสู่ระบบเรียบร้อยแล้วหน้าต่างของ Teacher และ Student จะมีความแตกต่างกัน ตรงที่ Teacher จะมีหน้าต่าง Quiz editor กับ Create room เพิ่มขึ้นมา

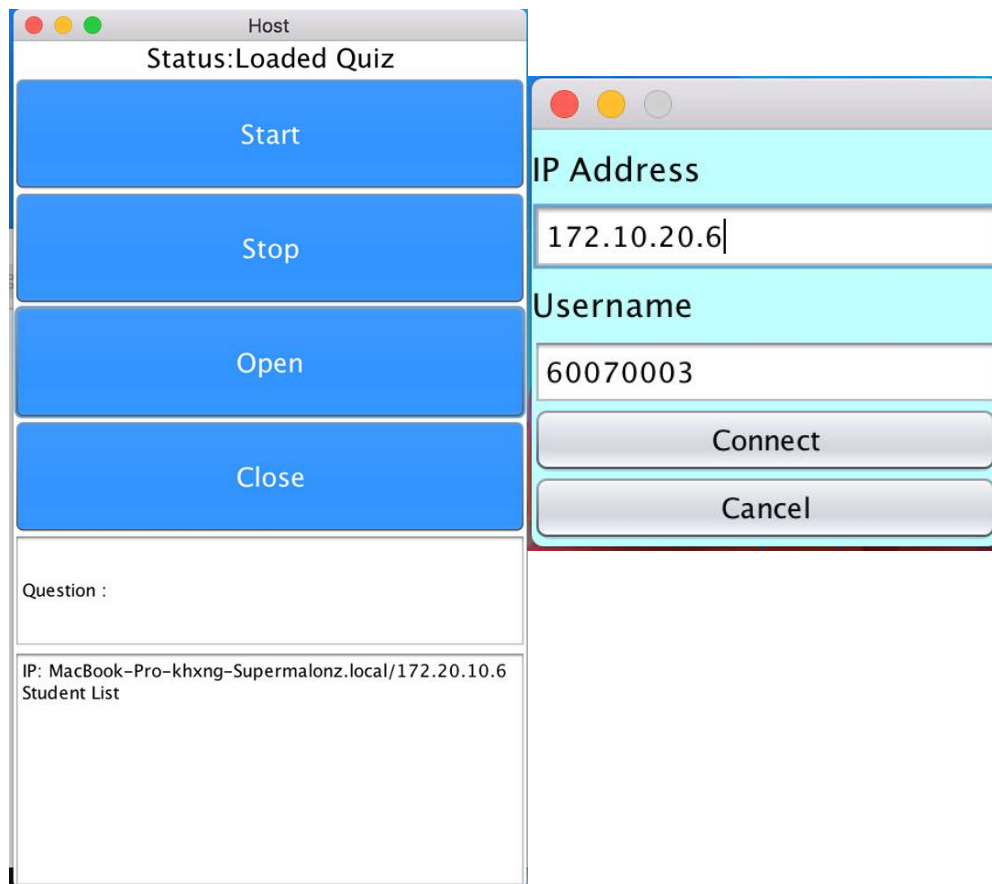


3.) ในส่วนการทำงานใน Quiz editor นั้นจะมีไว้สำหรับการสร้างคำถาม ในส่วนของคำถามนั้นจะมีการบันทึกไฟล์เก็บไว้ในตัวเครื่องเป็น Object ของคำถาม ในคำถาม 1 ข้อ สามารถมีได้หลายคำถามได้

Field	Left Window	Right Window
Quiz Name		OOP
Question		เงื่อนไข Method ที่ยังไม่สมบูรณ์
A		2
B		4
C		1
D		5
Correct	choice correct A-D	C

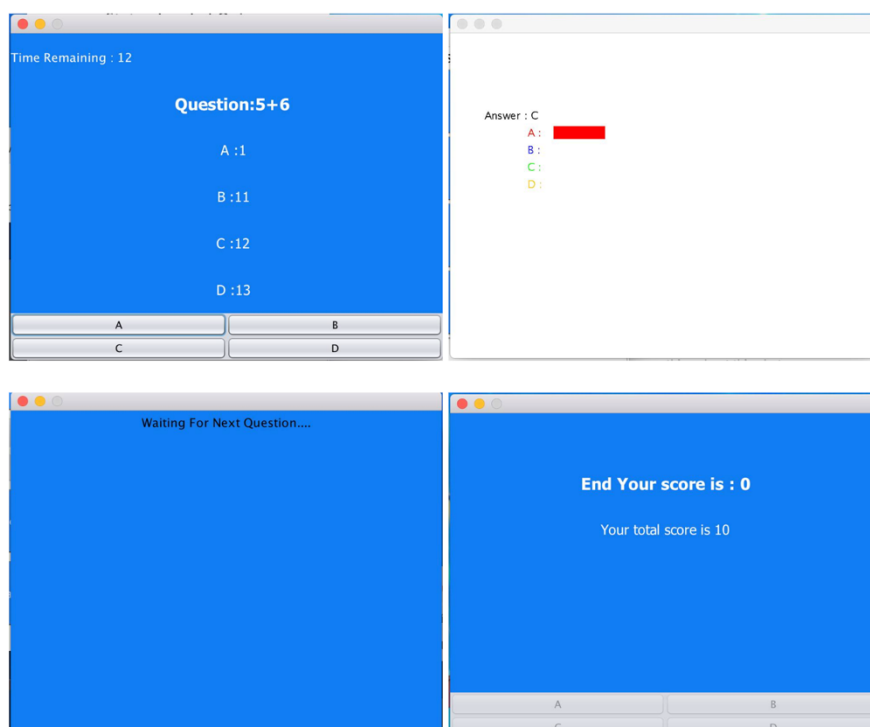
Buttons: Add Problem, Save Quiz, Load Quiz

4.) หลังจากที่เราสร้างคำถามเสร็จเรียบร้อยแล้ว ในส่วนของ Teacher ต้องทำการสร้าง Lobby เพื่อให้ Student เชื่อมต่อเข้ามา การทำงานของ Hooza นั้นจะมีการทำงานผ่านทาง Localhost เมื่อ Teacher สร้าง Lobby จะเป็นการสร้างตัว Server เพื่อไว้ใช้สำหรับการเชื่อมต่อระหว่าง Student และ Teacher ซึ่งตัว Server จะแตก Thread ให้แก่ตัว Student ทุกตัวที่เชื่อมต่อเข้ามาที่ Lobby หลังจาก Teacher เข้าสู่หน้าต่าง Create room แล้วจะต้องทำการกดปุ่ม Open เพื่อเป็นโหลดคำถามที่ได้สร้างขึ้นจากตัวเครื่องถึงจะทำการเริ่มเล่นได้ แล้วจะมีปุ่ม Start สำหรับการเริ่ม Quiz หลังจากกดปุ่ม Start แล้วปุ่มจะเปลี่ยนเป็นปุ่ม Next เพื่อไว้ใช้สำหรับกดไปในข้อถัดไป, ปุ่ม Stop สำหรับการหยุด Quiz ในข้อนั้น



5.) หลังจากที่ Student Join เข้าสู่ Lobby โดยใช้ Ip ที่แสดงอยู่ของ Server แล้วจะมีส่วนที่ขึ้นให้ Server ว่ามีใครเข้ามาแล้วบ้าง หลังจากที่ Teacher กด Start แล้วจะมีการส่งคำถามไปที่ละข้อให้กับ Student แต่ละคน เมื่อ Student ทุกคนตอบคำถามเสร็จแล้วในข้อนั้นหรือเวลาหมด ฝั่ง Teacher จะทำการกด Stop เพื่อเป็นการสิ้นสุดคำถามในข้อนั้นแล้วจะแสดงหน้าต่างกราฟแสดงว่า มีใครตอบข้อไหนบ้างแล้วข้อที่ถูกคือข้อไหน เมื่อตอบคำถามครบทุกข้อแล้ว จะมีหน้าต่างแสดงว่า ได้คะแนนทั้งหมดเท่าไรและมีคะแนนรวมทั้งหมดเท่าไรแล้ว จะทำการเก็บคะแนนนั้นไว้ที่

Database





## ตัวอย่าง Code ของ การสร้าง Socket ของ Server

### ตัวอย่าง Code ของ การสร้าง Socket ของ Server

```
```java
public void run(){
    try {
        ss = new ServerSocket(5555);
        System.out.println("Server is running..");
        analysis = new QuestionAnalysis();
        while (!exit){
            client = new ClientHandler(ss.accept());
            client.setQuestionAna(analysis);
            tr = new Thread(client); //tr stand for Thread
            tr.start();
            list.add(client);
            try{
                Thread.sleep(10);
                ta.append(client.getID() + "\n");
            }catch (Exception ex){
                System.out.println("Thread sleep error :" + ex);
            }
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```
```

ตัวอย่าง Code ของการจัดการกับ Student แต่ละคน

```
### ตัวอย่าง Code ของการจัดการกับ Student แต่ละคน
```java
public void run() {

    String run = "";
    //read input only
    try {
        while(running){
            run = in.readLine();
            System.out.println(run);
            String[] data = run.split(" ");
            if(data[0].equals("id")){
                ID = data[1];
            }
            else if(data[0].equals("ch")){

                if(data[1].equals("A")){
                    analysis.SetA();
                }
                else if(data[1].equals("B")){
                    analysis.SetB();
                }
                else if(data[1].equals("C")){
                    analysis.SetC();
                }
                else if(data[1].equals("D")){
                    analysis.SetD();
                }
            }

        }

    }

    catch (IOException i){
        System.out.println(i);
    }finally {
        try{
            sk.close();
            running = false;
        }catch (IOException i){}
    }

}
```
```