

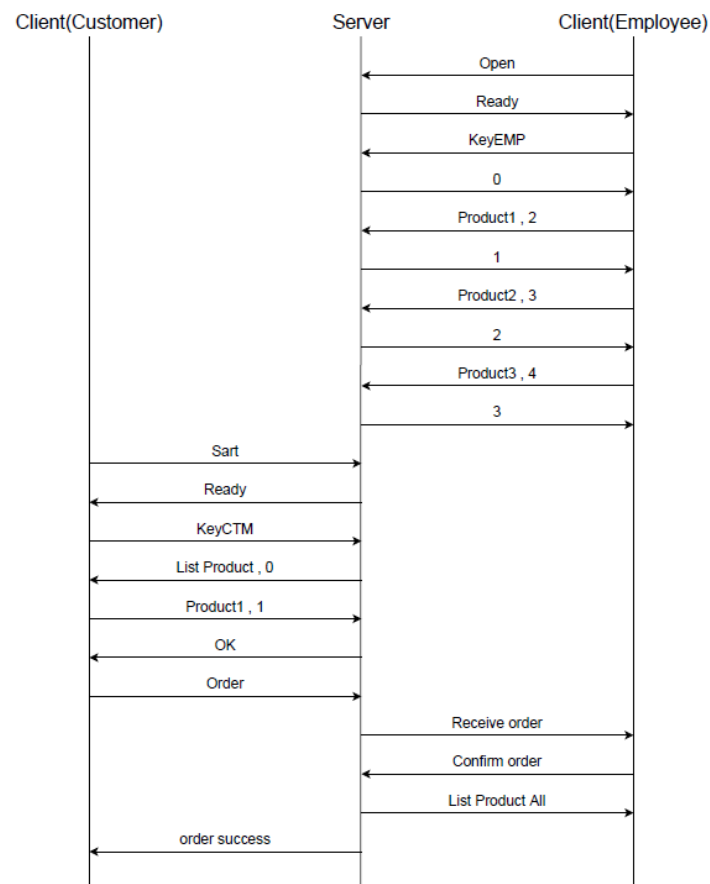
## Socket Assignment ( Part 2)

### Shopping

หลักการทำงานของ Shopping

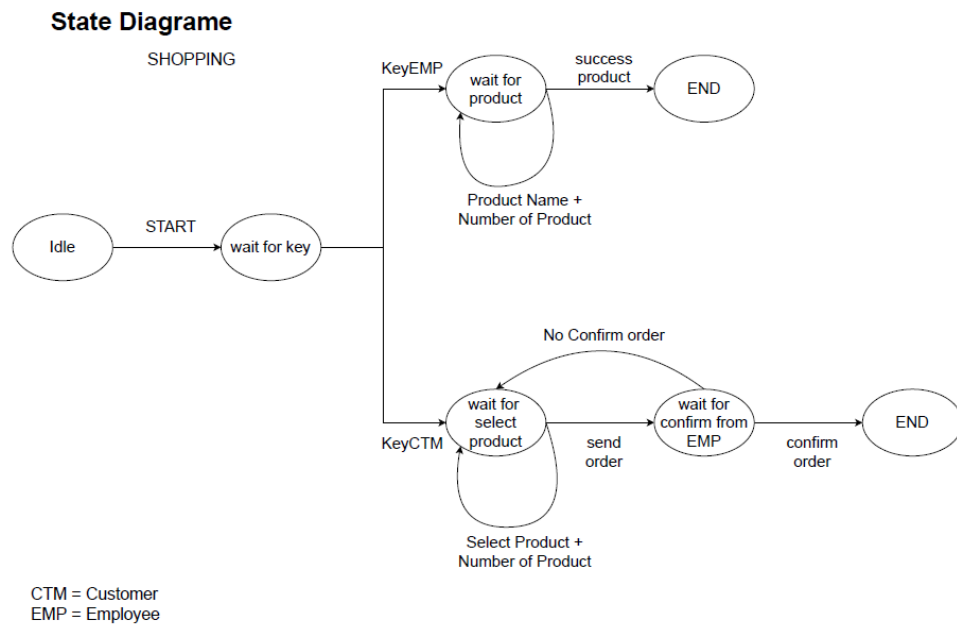
ขั้นแรกเริ่มจากฝั่ง Employee จะทำการสร้าง Product หรือผลิตภัณฑ์ จากนั้นทำการเพิ่มข้อมูลไปยัง Server หลังจากนั้นเมื่อ Customer หรือลูกค้าทำการเริ่มใช้งานทาง Server ก็จะส่ง list รายการสินค้าที่มีมายัง Customer และ Customer ก็จะทำการเลือกสินค้าที่ต้องการ เมื่อเลือกเสร็จก็ใช้คำสั่ง order เพื่อทำการบอกไปยัง Server ว่าตอนนี้ลูกค้าเลือกสินค้าเสร็จแล้ว ทาง Server จะแจ้งเตือนไปยัง Employee ว่าตอนนี้ลูกค้าทำการเลือกเสร็จแล้ว ให้ Employee ทำการ Confirm Order เมื่อทาง Employee Confirm เรียบร้อยแล้ว Server ก็ จะทำการตอบกลับไปยัง Customer ว่า Order success

Sequence Diagram :



CTM = Customer  
EMP = Employee

## State Diagram :



อธิบาย Code Server.js

- กำหนดค่า HOST และ PORT

- ประกาศตัวแปรเพื่อเก็บข้อมูล

```
6210110334, a minute ago | 1 author (6210110334)
1  var net = require("net");
2  var HOST = "127.0.0.1";
3  var PORT = 6969;
4
5  var productList = [];
6  var selectProductList = [];
7  var isHasProduct = false;
8  var stateEMP = 0;
9  var stateCTM = 0;
10 var client = null;
11 var txt;
```

-ทำการสร้าง Server

เมื่อ Client ทำการส่งข้อมูลเข้ามาครั้งแรกก็จะทำการ Check สถานะของ Client

```
net
    .createServer(function (sock) {
        sock.on("data", function (data) {
            txt = data.toString();
            if (txt == "employee" && stateEMP == 0) {
                client = "employee";
            }
            if (txt == "customer" && stateCTM == 0) {
                client = "customer";
            }
            sock.emit(client);
        });
    });
```

- ถ้า Client เป็น Employee

ถ้าเข้ามาครั้งแรกก็จะส่ง “ Ready for Employee client ”

ครั้งถัดไปทำการรับข้อมูลมาจาก Client Employee เพื่อเก็บข้อมูลของ Product และส่งจำนวนของ Product ให้ Client Employee

```
sock.on("employee", function () {
    switch (stateEMP) {
        case 0:
            sock.write("Ready for Employee client");
            stateEMP = 1;
            break;
        case 1:
            productList.push(txt);
            console.log("Employee add product success");
            sock.write("Num of Product is " + productList.length);
            break;
    }
});
```

- ถ้า Client เป็น Customer

ครั้งแรกจะทำการส่ง List Product ไปยัง Client Customer

ในครั้งถัดไปจะรับข้อมูลมาจากชื่อ Product มาจาก Customer แล้วทำการ Check ว่าใน Server ว่ามี Product ซีนี่ไหม ถ้าหากมี Product ซีนี่ก็จะทำการส่ง Success to select แต่ถ้าหากไม่มีก็จะทำการส่ง Failed to select ไปยัง Client Customer

```
sock.on("customer", function () {
  switch (stateCTM) {
    case 0:
      sock.write("Product List :\n" + productList);
      stateCTM = 1;
      break;
    case 1:
      if (txt == "order") {
        sock.emit("success");
        stateCTM = 2;
      } else {
        for (var i = 0; i < productList.length; i++) {
          if (txt == productList[i]) {
            isHasProduct = true;
          }
        }
        if (isHasProduct) {
          selectProductList.push(txt);
          console.log("Customer select product success");
          sock.write("Success to select " + txt);
          isHasProduct = false;
        } else {
          console.log("Customer failed to select " + txt);
          sock.write("Failed to select " + txt);
        }
      }
      break;
  }
});
```

```
sock.on("success", function () {
  sock.write("Order success");
});
});
```

อธิบาย Code Employee.js

- กำหนด Host และ Port

```
var net = require("net");  
var HOST = "127.0.0.1";  
var PORT = 6969;
```

- ทำการสร้าง Socket จากนั้นทำการเชื่อมต่อกับ Server และส่ง Data “employee” ไปยัง server

```
var employee = new net.Socket();  
employee.connect(PORT, HOST, function () {  
  employee.write("employee");  
});
```

- สร้างตัวรับข้อความ หลังจากนั้นทำการรับข้อความจาก Employee ส่ง Data ให้กับ Server

```
const readline = require("readline").createInterface({  
  input: process.stdin,  
  output: process.stdout,  
});  
employee.on("data", function (data) {  
  console.log(data.toString());  
  readline.question("Add Product: ", (input) => {  
    employee.write(input);  
  });  
});
```

อธิบาย Code Customer.js

- กำหนด Host และ Port

```
6210110334, 23 minutes ago | 1 author (
var net = require("net");
var HOST = "127.0.0.1";
var PORT = 6969;
```

- ทำการสร้าง Socket จากนั้นทำการเชื่อมต่อกับ Server และส่ง Data “Customer” ไปยัง server

```
var customer = new net.Socket();
var state = 0;
var txt;

customer.connect(PORT, HOST, function () {
  customer.write("customer");
});
```

- สร้างตัวรับข้อความ หลังจากนั้นทำการรับข้อความจาก Customer ส่ง Data ให้กับ Server

```
customer.on("data", function (data) {
  txt = data.toString();
  console.log(txt);
  if (txt == "Order success.") {
    state = 1;
  }
  switch (state) {
    case 0:
      readline.question("Select Product: ", (input) => {
        customer.write(input);
      });
      break;
    case 1:
  }
});
```

สมาชิกกลุ่ม

- |                |                |                 |
|----------------|----------------|-----------------|
| 1.นายชอฟรอน    | ภาวะ           | รหัส 6210110081 |
| 2.นายฉัตร      | ปรัรักษกิจดำรง | รหัส 6210110097 |
| 3.นายภูมิทัศน์ | ใจแก้ว         | รหัส 6210110271 |
| 4.นายวิรุฬ     | หวังยุ้ย       | รหัส 6210110334 |