

# Project Document

## #8 Message Reservation

Group อะไรก็ได้แล้วแต่เลย

6230238421 Thanin Sawetkititham

6231364021 Suppawich Tawornpichayachai

2110503 Software Development Practice, Semester 2, 2021

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

**GitHub link:** [github.com/stangthanin/message-reservation](https://github.com/stangthanin/message-reservation)

### **Non-Functional Requirements**

1. Security:

The system shall authenticate users using username-password.

\* เพิ่มข้อกำหนดว่า ใช้ email เป็น username \*

The system shall be able to keep user's transactions confidential.

2. Performance:

The system shall response to a request in 3 seconds.

3. Usability:

The system shall be used and test via Postman.

### **Functional Requirements**

1. The system shall allow a user to register by specifying the name, telephone number, email, and password.

2. After registration, the user becomes a registered user, and the system shall allow the user to login to use the system by specifying the email and password. The system shall allow a registered user to logout.

3. After login, the system shall allow the registered user to reserve up to 3 queues by specifying the date and the preferred message shop. The message shop list is also provided to the user. A message shop information includes the name, address, telephone number, and open-close time.

\* เพิ่มข้อกำหนดว่า message shop เปิดทุกวัน โดยเวลาเปิด-ปิด ในวันเดียวกันระหว่าง 00:00 - 23:59 \*

4. The system shall allow the registered user to view his/her message reservation.

5. The system shall allow the registered user to edit his/her message reservation.

6. The system shall allow the registered user to delete his/her message reservation.

7. The system shall allow the admin to view any message reservation.

8. The system shall allow the admin to edit any message reservation.

9. The system shall allow the admin to delete any message reservation.

## โค้ดโปรแกรมที่ได้แก้ไขไปจากของเดิม

### models

#### User

Field: name, tel, email, role, password

tel จะต้อง match กับ Regex เช่น 0123456789, 012-345-6789, +66123456789, (012) 345 6789

```
let telRegex = new RegExp(
  /^[+]?([0-9]{2,3})?[-\s\.]?[0-9]{3}[-\s\.]?[0-9]{4,6}$/
);

tel: {
  type: String,
  required: [true, "Please add a telephone number"],
  match: [telRegex, "Please add a valid telephone number"],
},
```

#### MessageShop

Field: name, address, tel, openTime (เวลาเปิดร้าน), closeTime (เวลาปิดร้าน)

openTime, closeTime จะต้อง match กับ Regex ในรูปแบบ 00:00 เช่น 09:30, 21:45

```
let timeRegex = new RegExp(/^([01]?[0-9]|2[0-3]):[0-5][0-9]$/);

openTime: {
  type: String,
  required: [true, "Please add an open time"],
  match: [timeRegex, "Please add a valid time in format 00:00"],
},
```

closeTime จะต้องเป็นเวลาที่ถัดจาก openTime

```
closeTime: {
  type: String,
  required: [true, "Please add a close time"],
  match: [timeRegex, "Please add a valid time in format 00:00"],
  validate: {
    validator: closeTimeValidator,
    message: "Close time must be after the open time",
  },
},

function closeTimeValidator(enterCloseTime) {
  return (
    (this.openTime.split(":")[0] < enterCloseTime.split(":")[0]) ||
    (this.openTime.split(":")[0] == enterCloseTime.split(":")[0] &&
      this.openTime.split(":")[1] < enterCloseTime.split(":")[1])
  );
}
```

## Appointment

Field: apptDateTime (วันเวลาที่จอง), durationMinute (ระยะเวลาที่จอง), user, messageShop, createAt

apptDateTime จะต้องเป็นเวลาหลังจาก openTime ของ messageShop

```
async function apptDateTimeValidator(enterApptDateTime) {
  const messageShop = await MessageShop.findById(this.messageShop);

  hour = new Date(enterApptDateTime).getUTCHours();
  minute = new Date(enterApptDateTime).getUTCMinutes();

  return !(
    hour < messageShop.openTime.split(":")[0] ||
    (hour == messageShop.openTime.split(":")[0] &&
      minute < messageShop.openTime.split(":")[1])
  );
}
```

apptDateTime เมื่อบวกด้วย durationMinute แล้ว จะต้องไม่เกิน closeTime ของ messageShop

```
durationMinute: {
  type: Number,
  required: [true, "Please add an appointment duration (minute)"],
  validate: {
    validator: durationMinuteValidator,
    message: "Appoint duration is exceed the shop close time",
  },
},

async function durationMinuteValidator(enterDurationMinute) {
  const messageShop = await MessageShop.findById(this.messageShop);

  hour = new Date(this.apptDateTime).getUTCHours();
  minute = new Date(this.apptDateTime).getUTCMinutes();

  minute += enterDurationMinute;
  hour += Math.floor(minute / 60);
  minute %= 60;

  return !(
    hour > messageShop.closeTime.split(":")[0] ||
    (hour == messageShop.closeTime.split(":")[0] &&
      minute > messageShop.closeTime.split(":")[1])
  );
}
```

## controllers

### appointments

ใน method addAppointment เพิ่มการเช็ค user.role ในกรณีที่ admin เป็นผู้สร้าง appointment จะต้องระบุ user.id ใน body ด้วย เพื่อไม่ให้ user.id ใน appointment เป็น id ของ admin

```
if (req.user.role !== "admin") { req.body.user = req.user.id }
```

เพิ่ม totalAppointment ใน response เพื่อแสดงจำนวน appointment ทั้งหมดที่ได้จออกไป

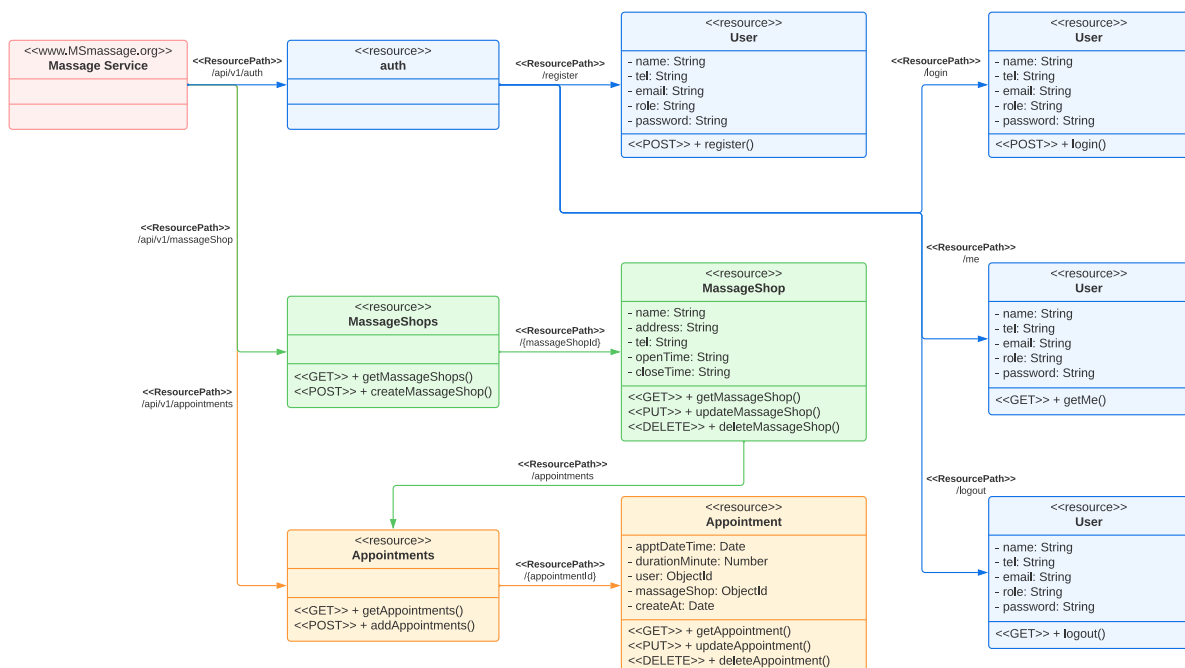
```
res.status(200).json({
  success: true,
  data: appointment,
  totalAppointment: existedAppointment.length + 1,
});
```

เพิ่ม error.message จาก model มาแสดงใน response

```
return res.status(500).json({
  success: false,
  message: "Cannot create Appointment",
  error: error.message,
});
```

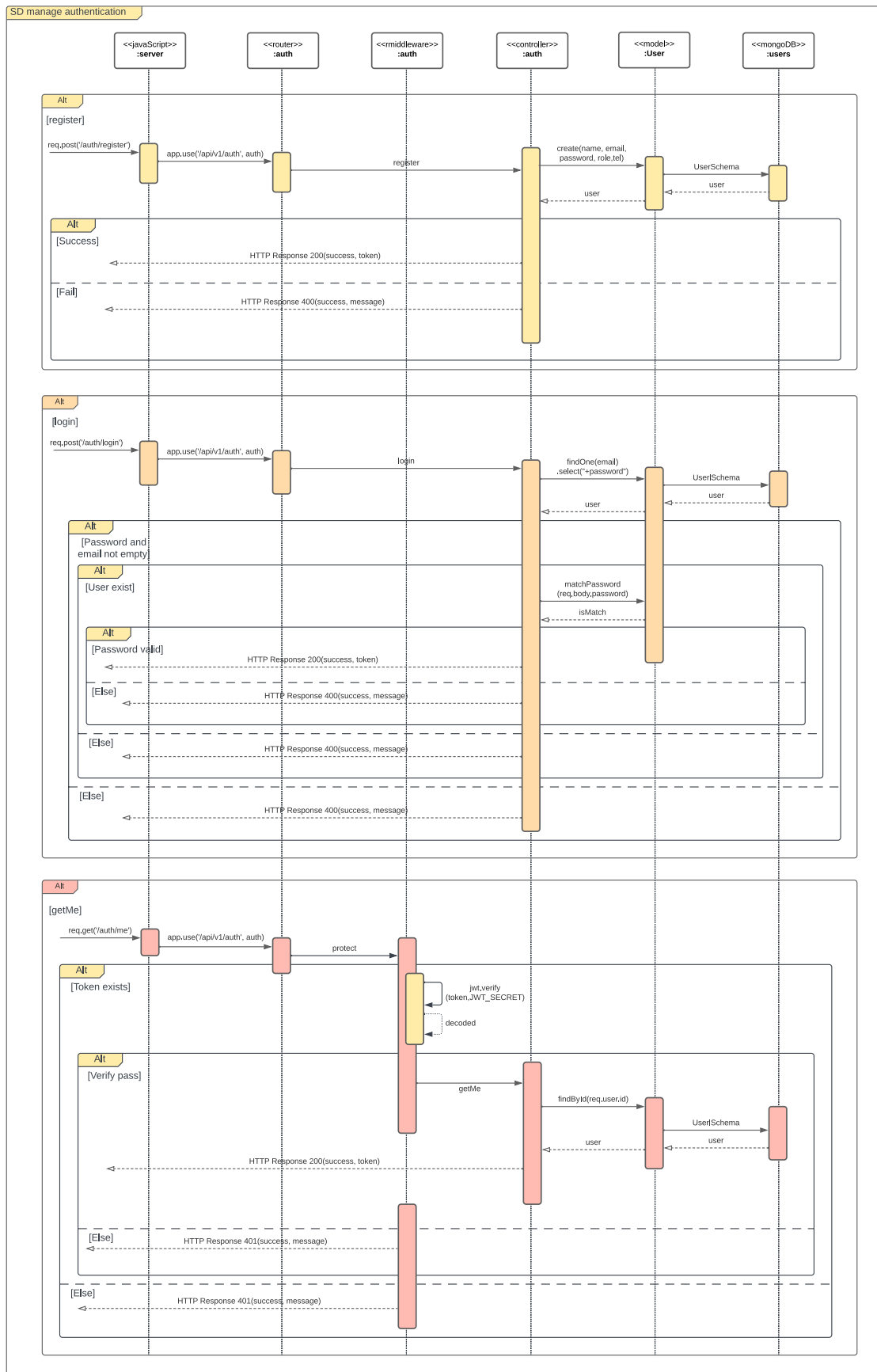
สำหรับ controller อื่นๆ, routes, middleware, config และ server ใช้ตัวอย่างจากโปรเจกในวิชาเรียน

### Class Diagram

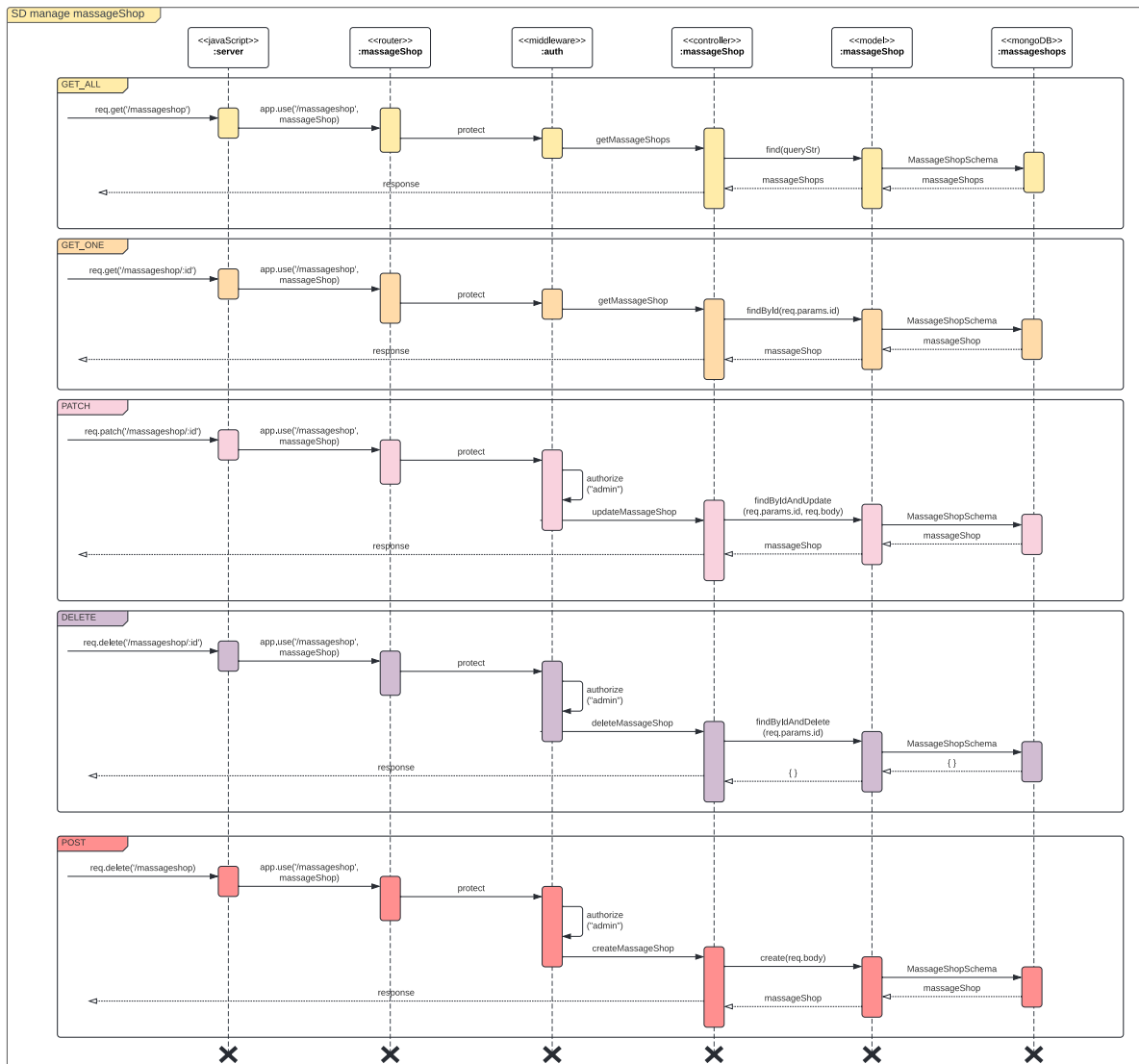


## Sequence Diagram

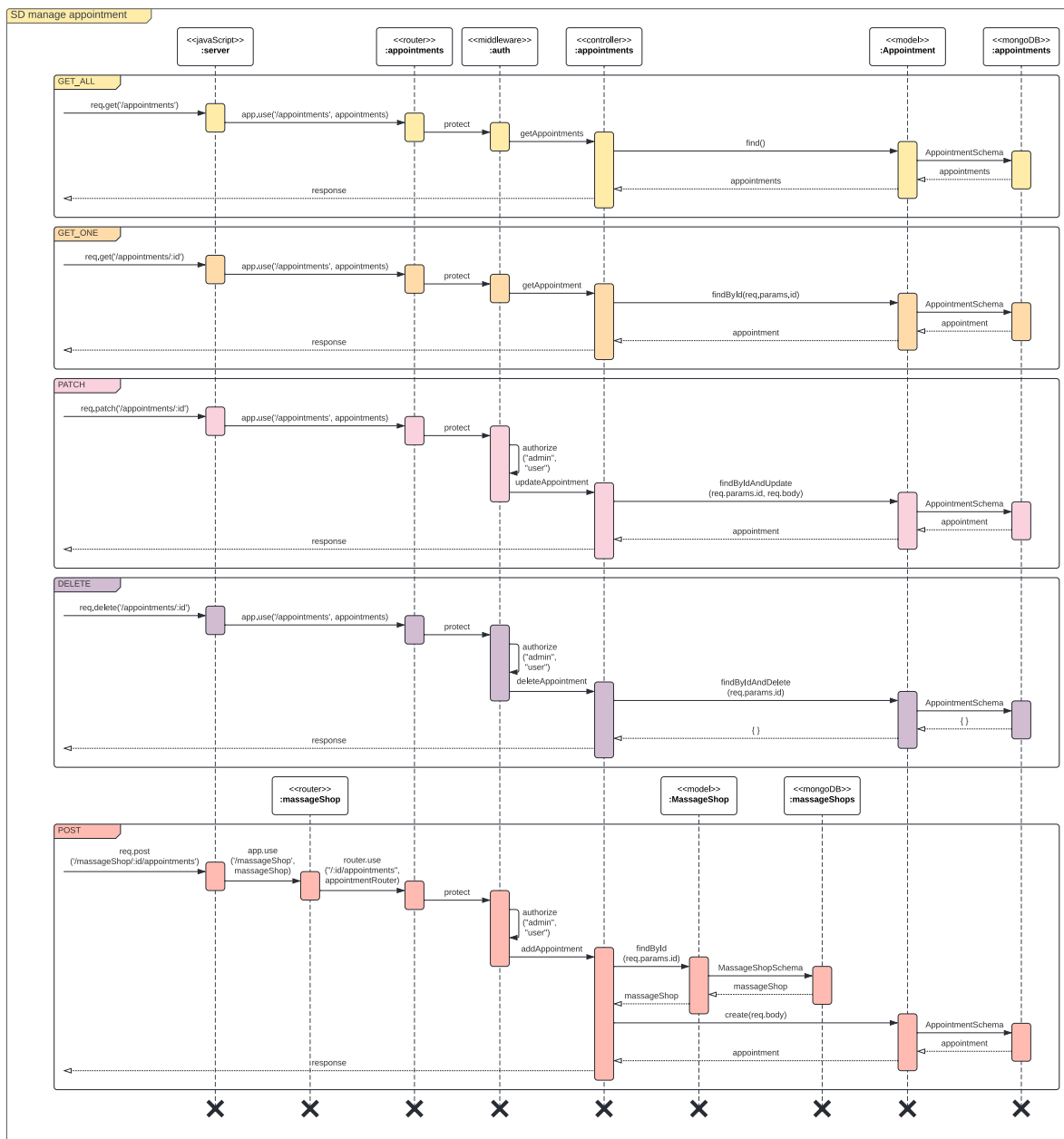
auth



## messageShop



## appointments



## การแบ่งงานในทีม

Thanin Sawetkititham	Suppawich Tawornpichayachai
models: User, Appointment	models: MessageShop
controllers: auth, appointments	controllers: messageShop
routes: auth, appointments	routes: messageShop
Sequence Diagram: appointments	Sequence Diagram: auth, messageShop Class Diagram
Project document	Project presentation