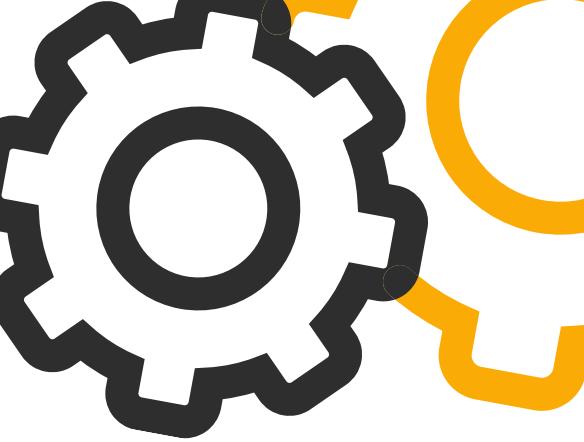




Software Development Practice Project



HOTEL BOOKING

Presenting By Lonewolf



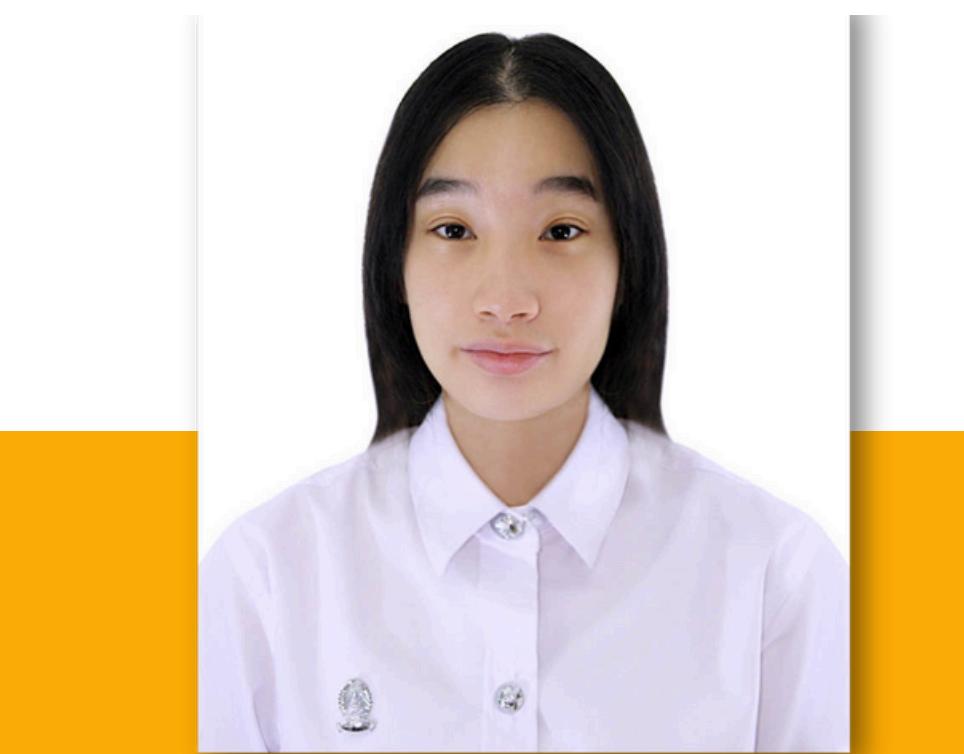


Our Team



Phanlop Bunkhan

6733174921



Tamakorn Khambanonda

6733104021



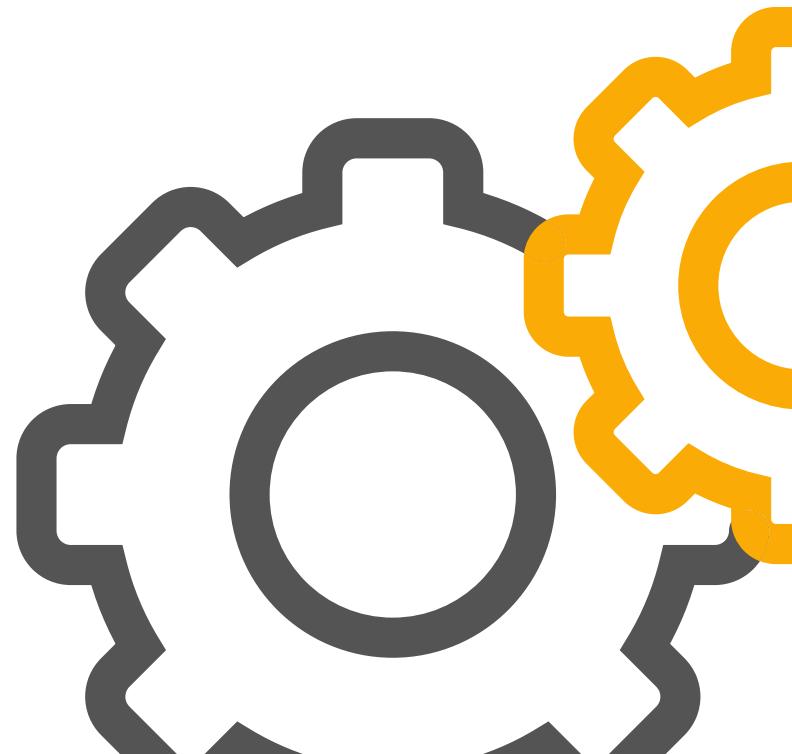
Waritsara Kongpom

6733235521



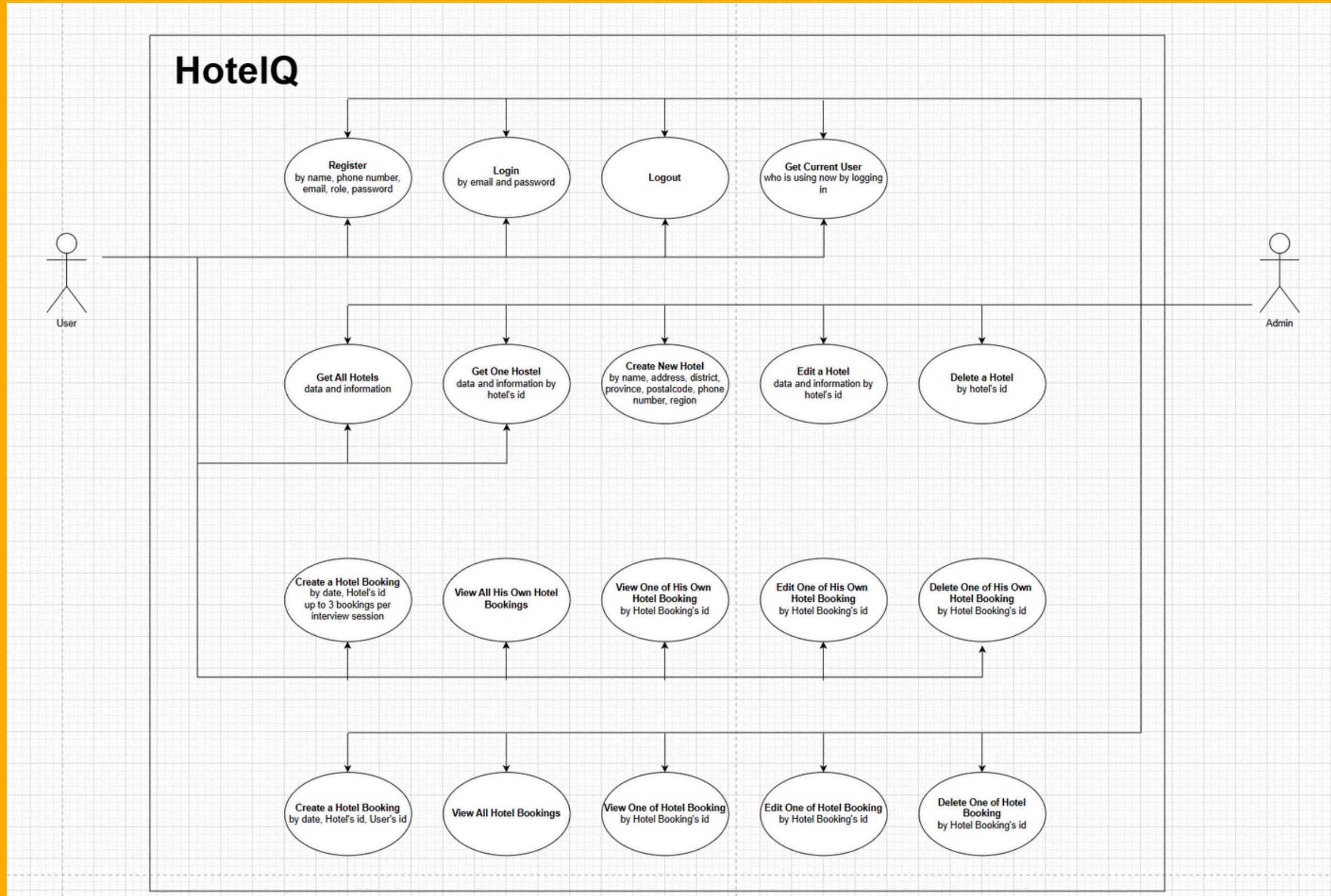
Project#1: Hotel Booking

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 log in to use the system by specifying the email and password. The system shall allow a registered user to log out.
3. After login, the system shall allow the registered user to book up to 3 nights by specifying the date and the preferred hotel. The hotel list is also provided to the user. A hotel information includes the hotel name, address, and telephone number.
4. The system shall allow the registered user to view his hotel bookings.
5. The system shall allow the registered user to edit his hotel bookings.
6. The system shall allow the registered user to delete his hotel bookings.
7. The system shall allow the admin to view any hotel bookings.
8. The system shall allow the admin to edit any hotel bookings.
9. The system shall allow the admin to delete any hotel bookings.



Use Case Diagram

Software Development Practice Project





การแก้ไขโค้ด

server.js

```
app.use('/api/appointments', appointments);
app.use('/api/hotels', hotels);
app.use('/api/auth', auth);
```

models/Appointment.js

```
hotel: {
  type: mongoose.Schema.ObjectId,
  ref: 'Hotel',
  required: true
},
```



การแก้ไขโค้ด

models/User.js

```
tel: {
  type: String,
  required: [true, 'Please add a phone number'],
  unique: true,
  match: [
    /^\\d{10}$/,
    'Please add a valid phone number (10 digits)'
  ],
},
```

controllers/auth.js

```
//@desc  Register user
//@route POST /api/auth/register
//@access Public
exports.register = async (req, res, next)=>{
  try{
    const {name, telNumber, email, password, role} = req.body;

    const user = await User.create({
      name,
      telNumber,
      email,
      role,
      password
    });
  }
}
```



การแก้ไขโค้ด

controllers/appointment.js

```
//@desc Add appointment
//@route POST /api/appointments
//@access Public
exports.addAppointment = async (req, res, next) =>{
  try{
    req.body.hotel=req.params.hotelId;
    req.body.user = req.user.id;
  }
}
```

```
//@desc Get single appointment
//@route Get /api/appointments/:id
//@access Public
exports.getAppointment = async (req, res, next) => {
  try{
    const appointment = await Appointment.findById(req.params.id).populate({
      path: 'hotel',
      select: 'name province tel'
    })
  }
}
```

```
//@desc Get all appointment
//@route Get /api/appointments
//@access Public
exports.getAppointments = async (req, res, next) =>{
  let query;
  //General User can only see their appointment
  if(req.user.role !== 'admin'){
    //console.log(req.user._id);
    query = Appointment.find({user:req.user.id}).populate({
      path: 'hotel',
      select: 'name province tel'
    });
  }else {//Admin can see all
    if(req.params.hotelId){
      console.log(req.params.hotelId);

      query = Appointment.find({hotel:req.params.hotelId}).populate({
        path: 'hotel',
        select: 'name province tel'
      });
    }else {
      query = Appointment.find().populate({
        path: 'hotel',
        select: 'name province tel'
      });
    }
  }
}
```

```
const Hotel = require('../models/Hotel');
```



การแก้ไขโค้ด

controllers/hotels.js

```
const Hotel = require('../models/Hotel.js');

//@desc Get all Hotels
//@route GET /api/hotels
//@access Public
exports.getHotels= asyncHandler(async (req, res, next)=>{

    const hotels = await query;

    res.status(200).json({succes:true, count:hotels.length, pagination, data:hotels});
```



การแก้ไขโค้ด

controllers/hotels.js

```
//@desc Get single Hotels
//@route GET /api/hotels/:id
//@access Public
exports.getHotel= async (req, res, next)=>{
  try{
    const hotel = await Hotel.findById(req.params.id);

    if(!hotel){
      return res.status(400).json({succes:false, message:"cannot get the Hotel"});
    }

    res.status(200).json({success:true, data:hotel});
  } catch(err){
    res.status(400).json({succes:false});
  }
};
```



การแก้ไขโค้ด

controllers/hotels.js

```
//@desc Create single Hotels
//@route POST /api/hotels
//@access Public
exports.createHotel= async (req, res, next)=>{
  const hotel = await Hotel.create(req.body);
  res.status(201).json({success:true, data:hotel});
};
```



การแก้ไขโค้ด

controllers/hotels.js

```
//@desc Update single Hotels
//@route PUT /api/hotels
//@access Public
exports.updateHotel= async (req, res, next)=>{
  try{
    const hotel = await hotel.findByIdAndUpdate(req.params.id, req.body, {
      new: true,
      runValidators: true
    });

    if(!hotel){
      return res.status(400).json({success:false});
    }

    res.status(200).json({success:true, data:hotel});
  } catch (err) {
    next(err);
  }
}
```



การแก้ไขโค้ด

controllers/hotels.js

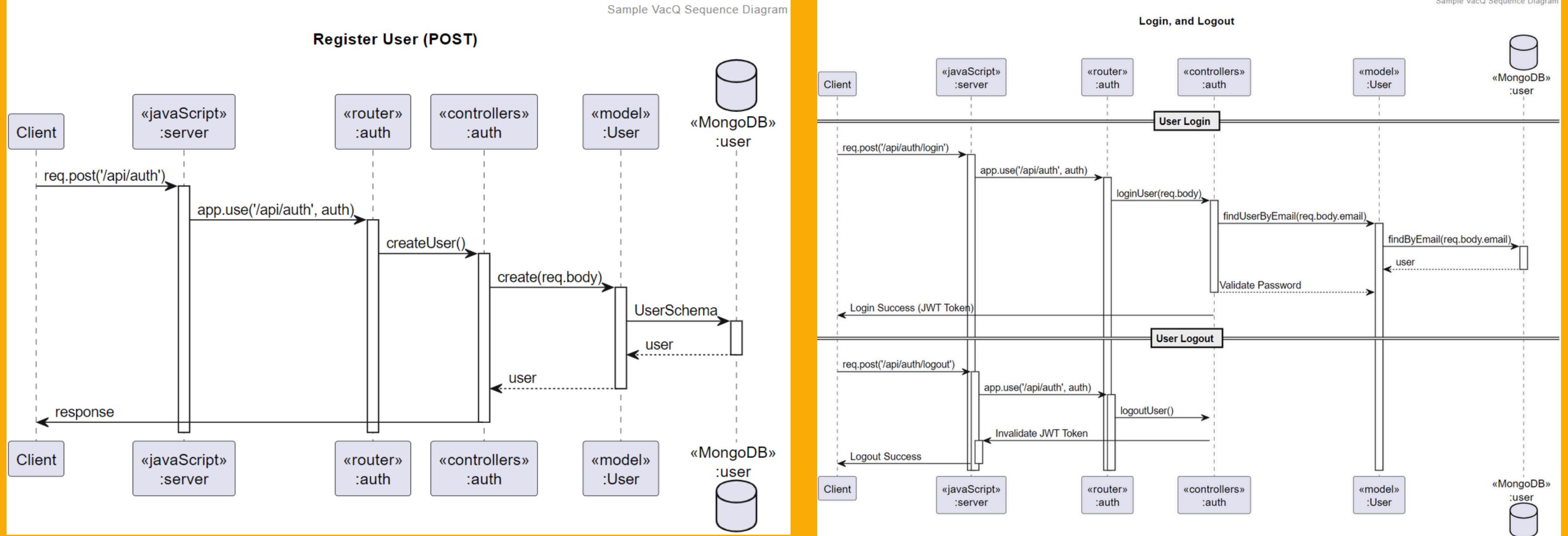
```
//@desc Delete single Hotels
//@route POST /api/hotels
//@access Public
exports.deleteHotel= async (req, res, next)=>{
  try{
    const hotel = await Hotel.findByIdAndDelete(req.params.id);

    if(!hotel){
      return res.status(404).json({success:false,
        message:`Hotel not found with id of ${req.params.id}`});
    }

    await Appointment.deleteMany({hotel: req.params.id});
    await Hotel.deleteOne({_id: req.params.id});
  }
}
```

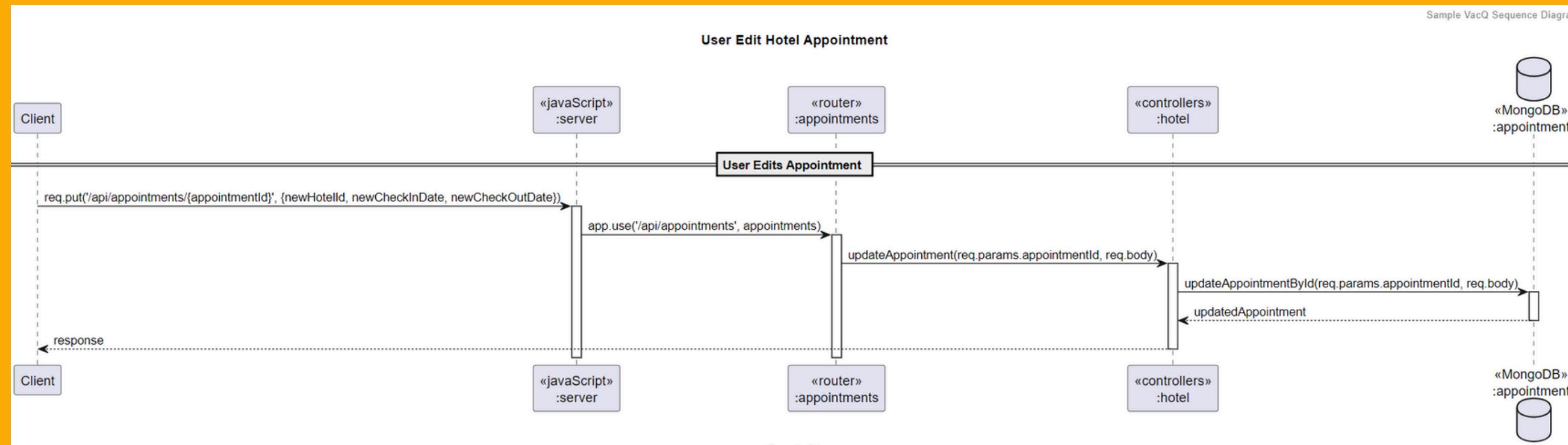
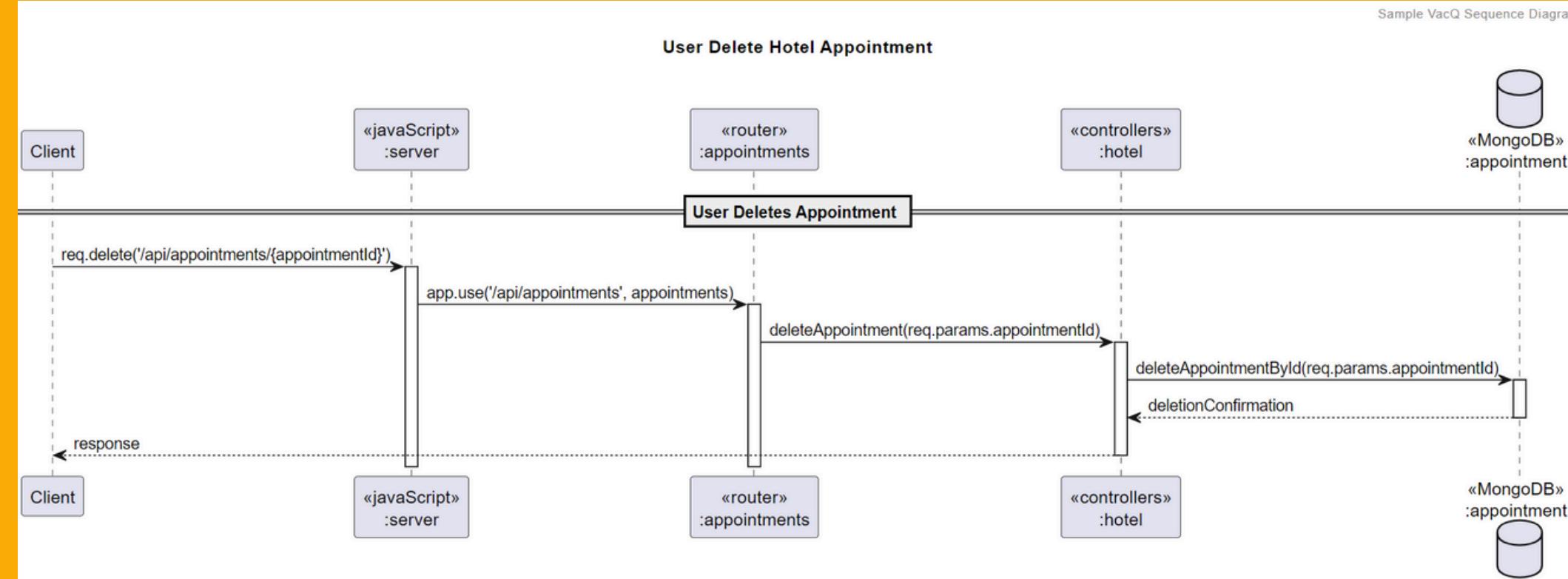
Sequence Diagram

Software Development Practice Project



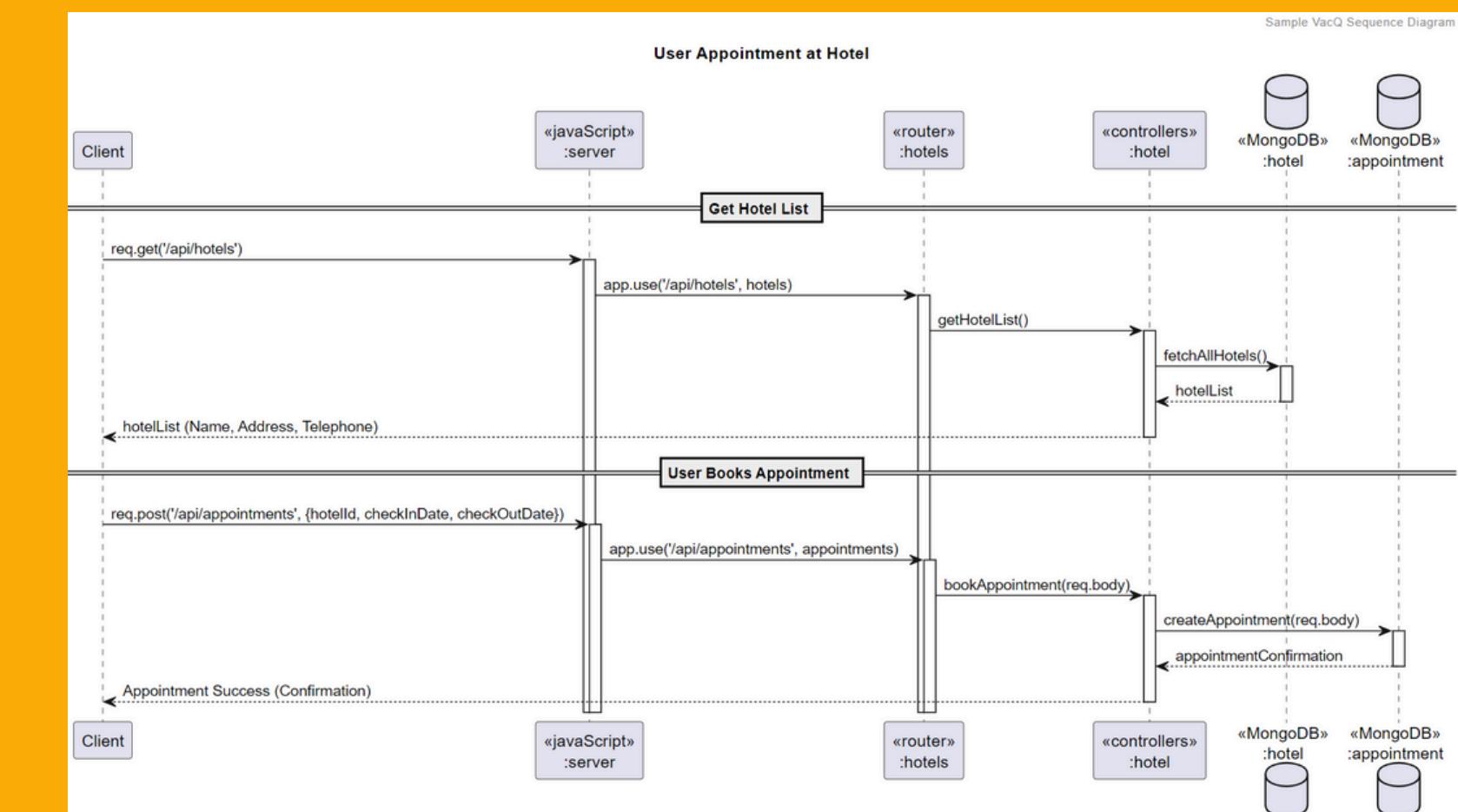
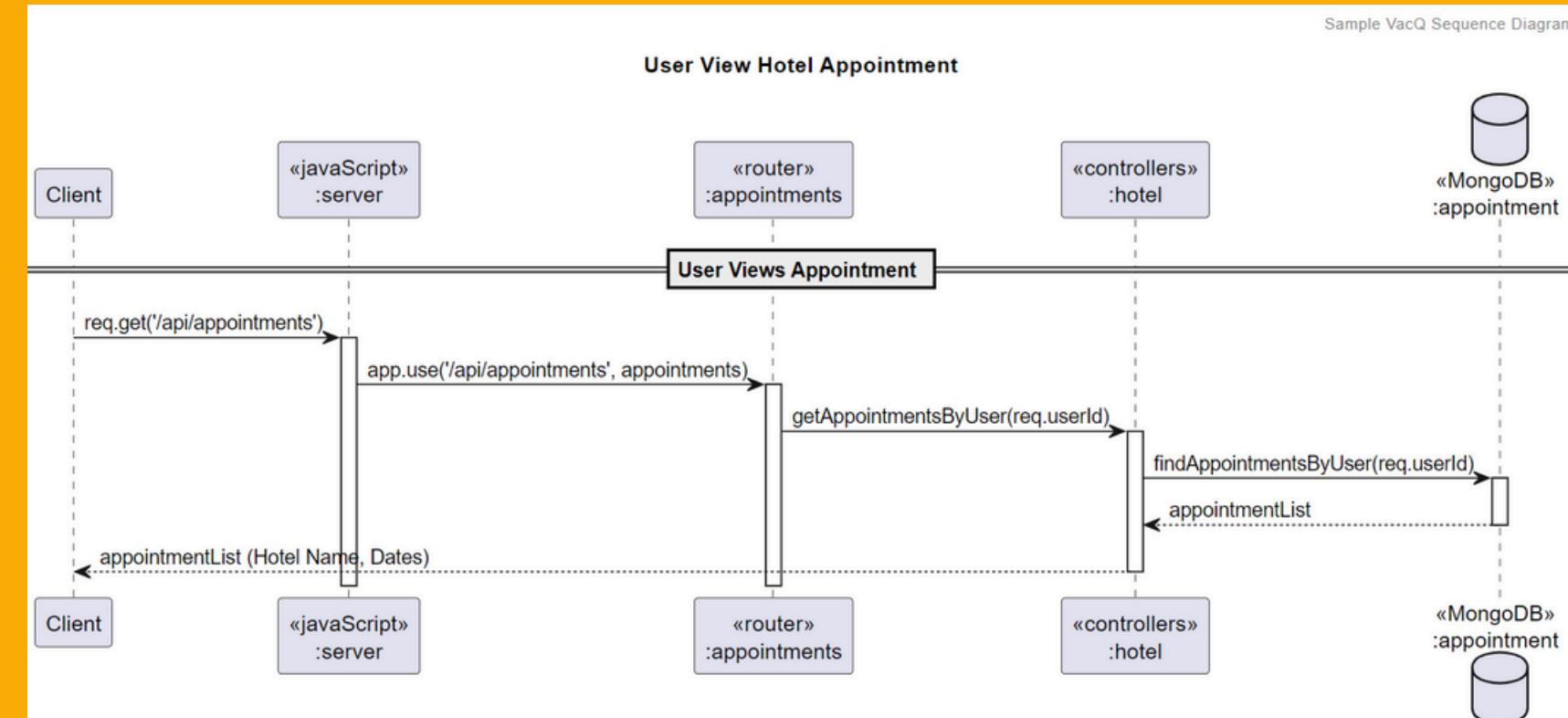
Sequence Diagram

Software Development Practice Project



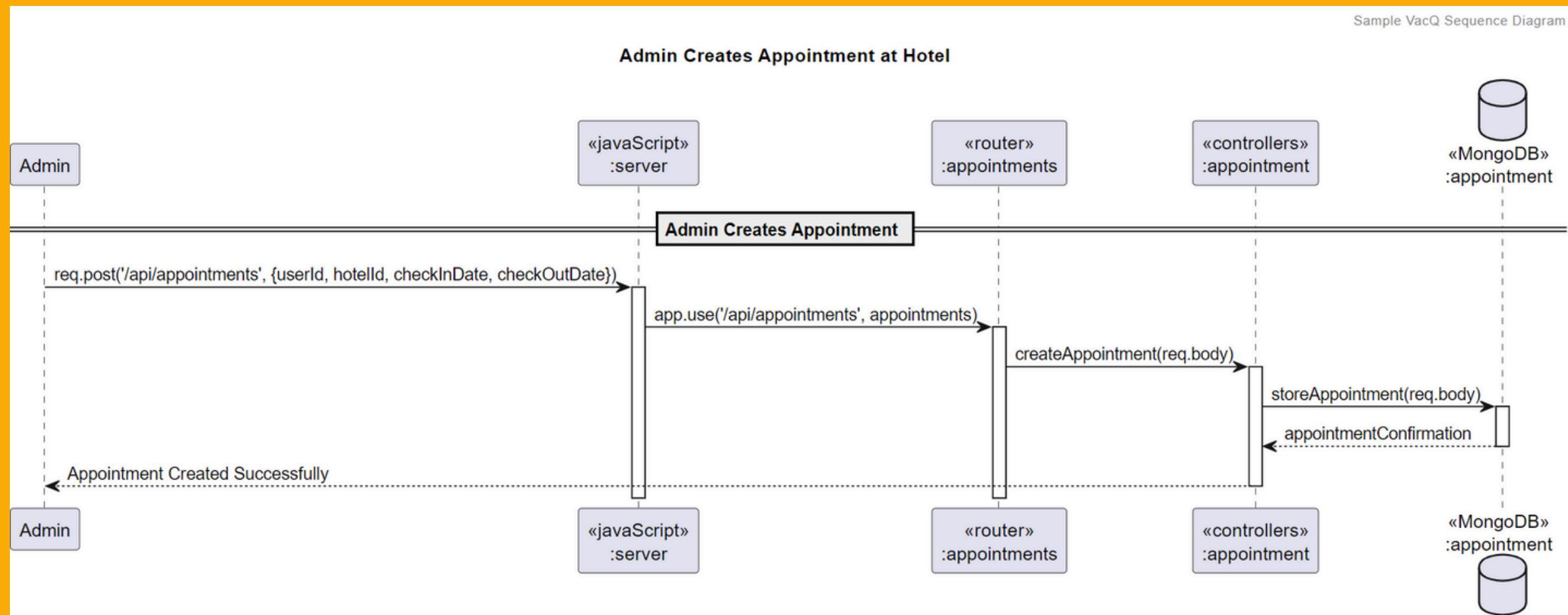
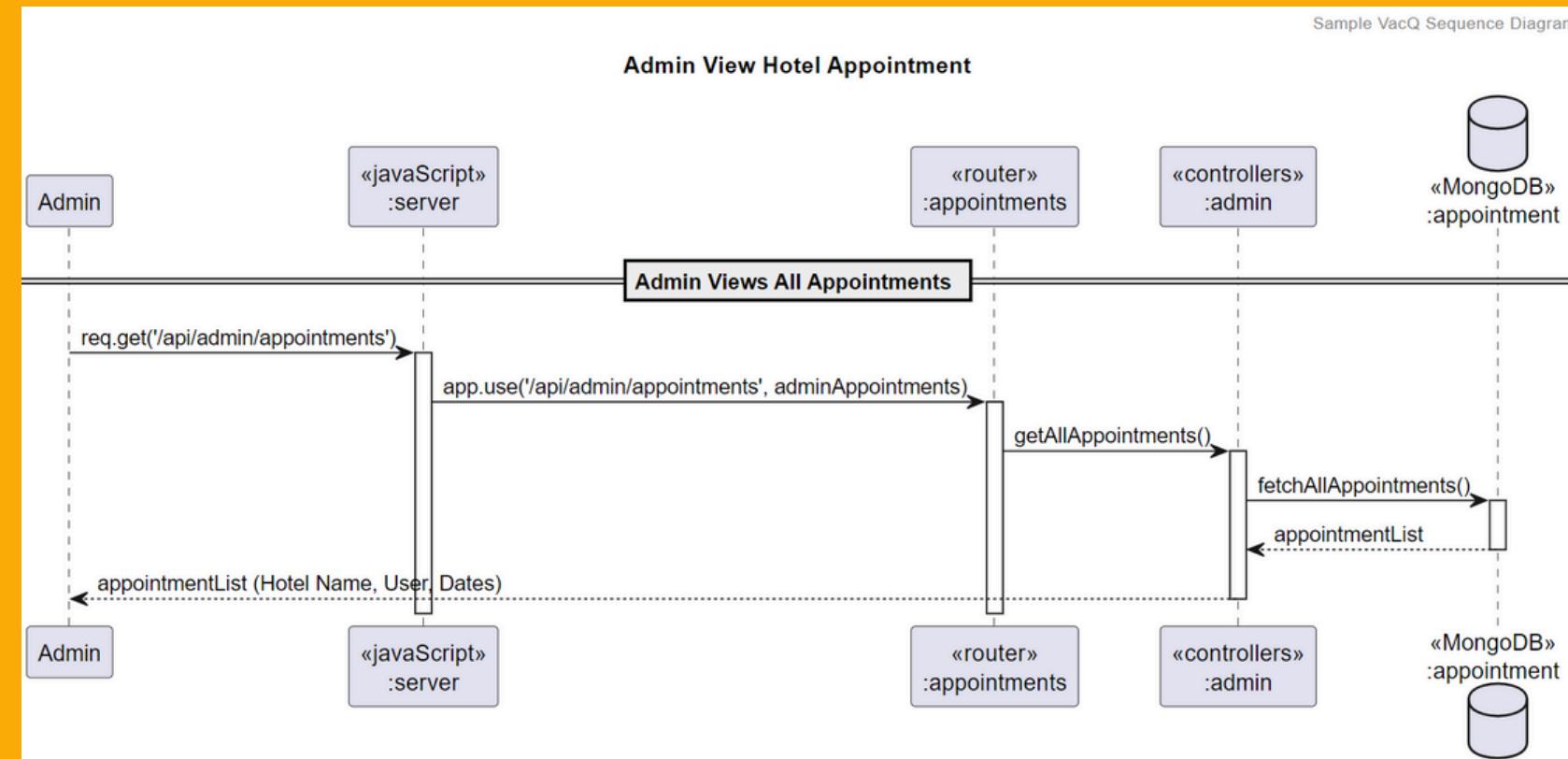
Sequence Diagram

Software Development Practice Project



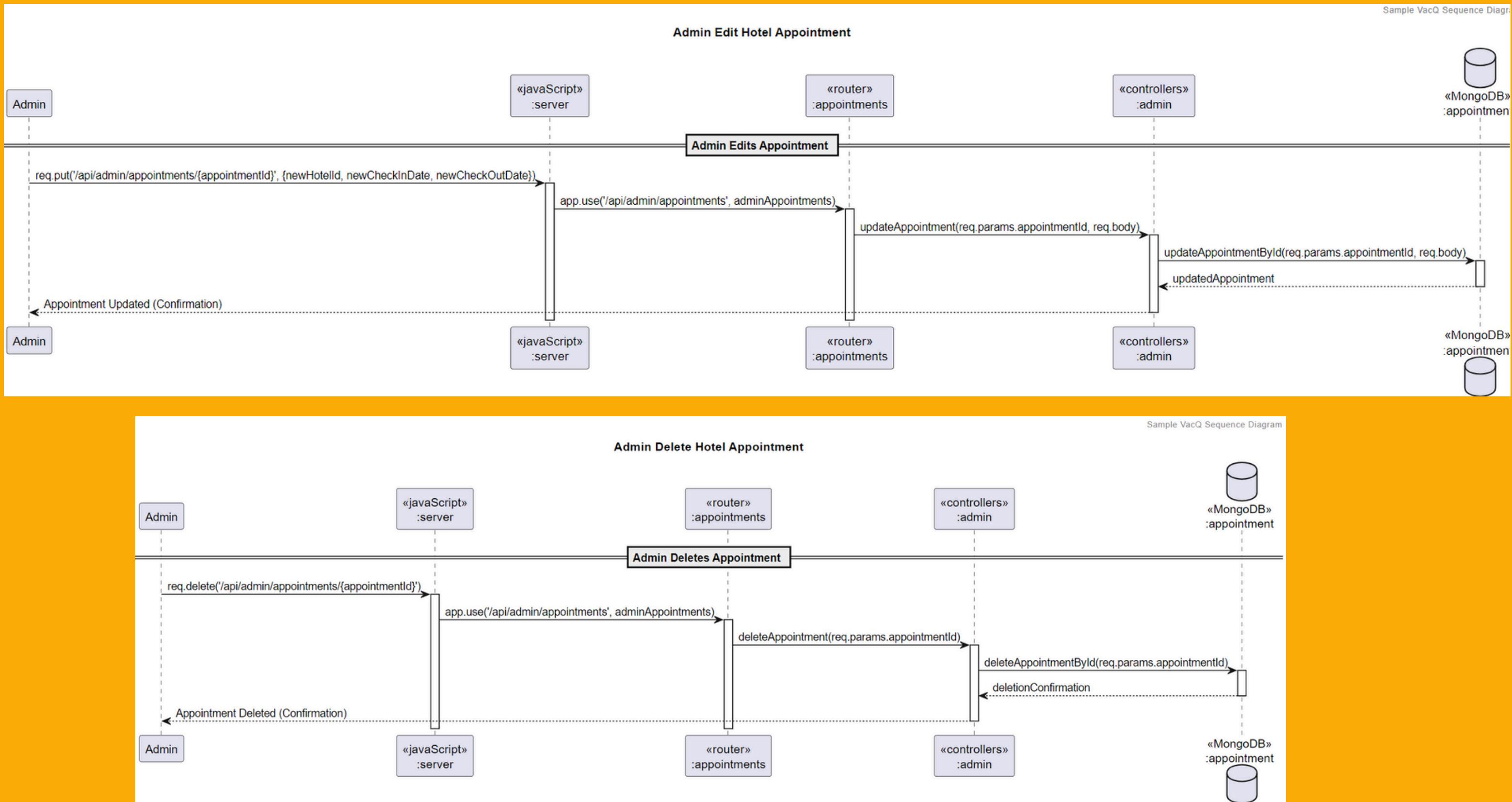
Sequence Diagram

Software Development Practice Project



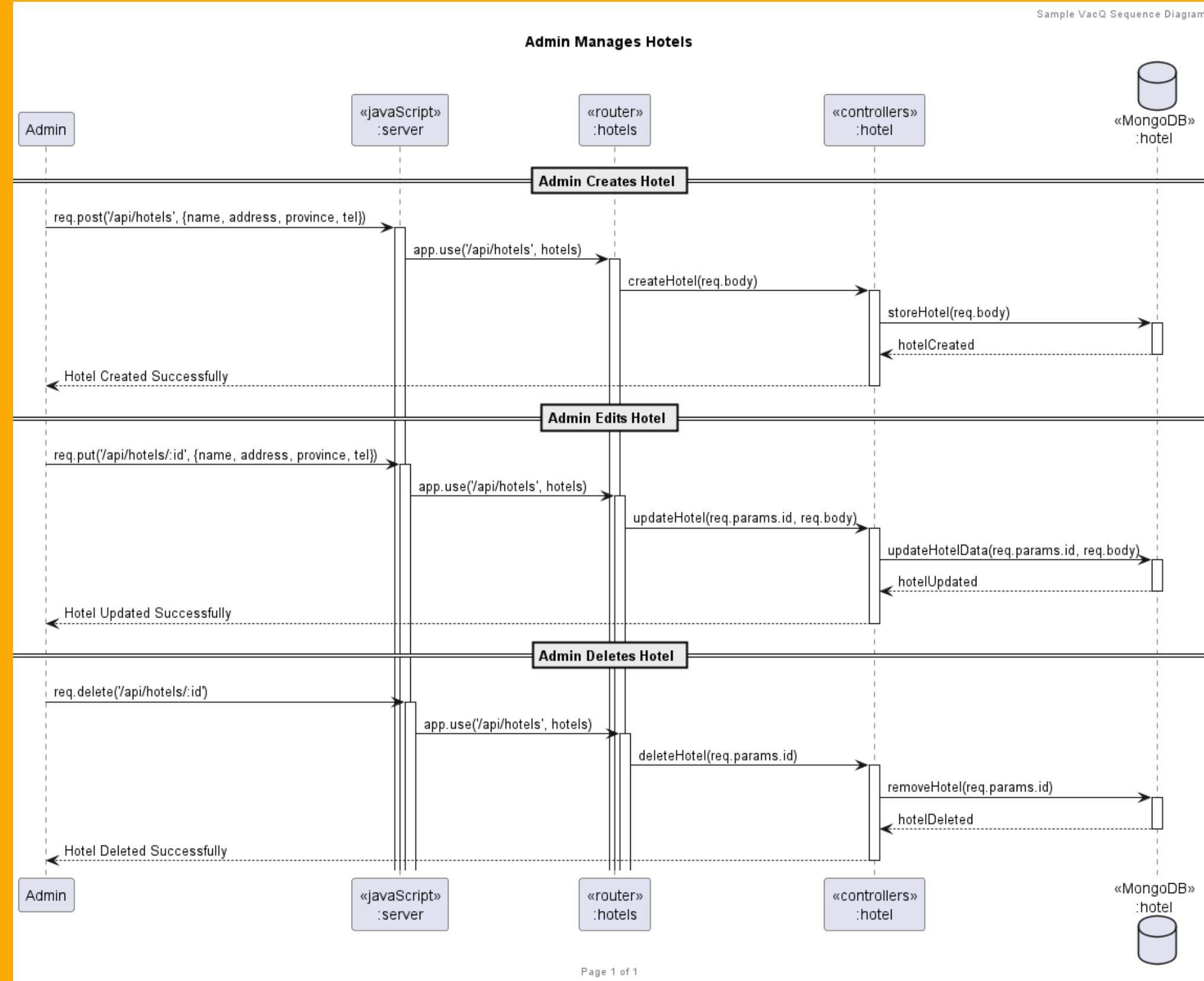
Sequence Diagram

Software Development Practice Project



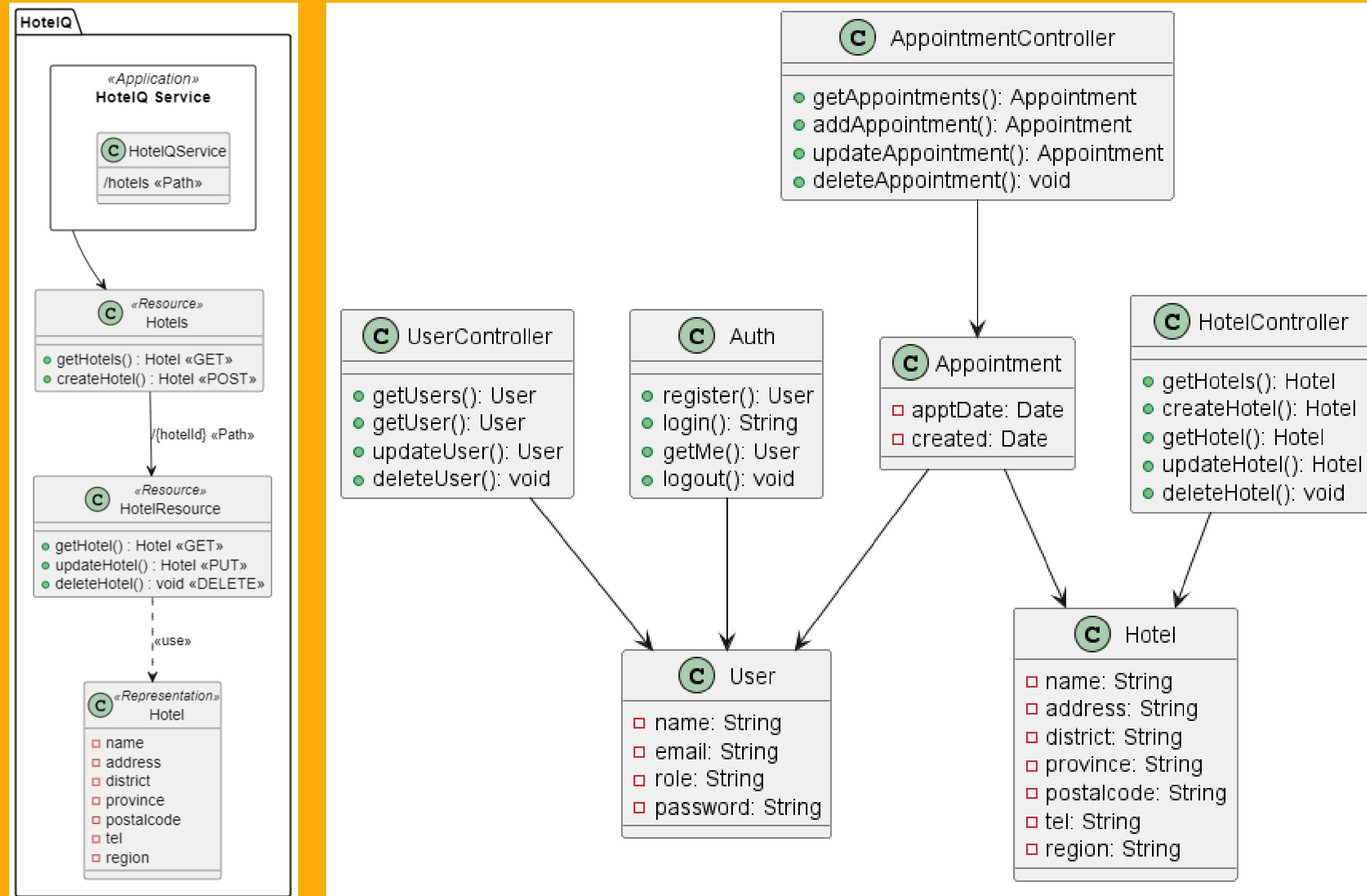
Sequence Diagram

Software Development Practice Project



UML Diagram

Software Development Practice Project



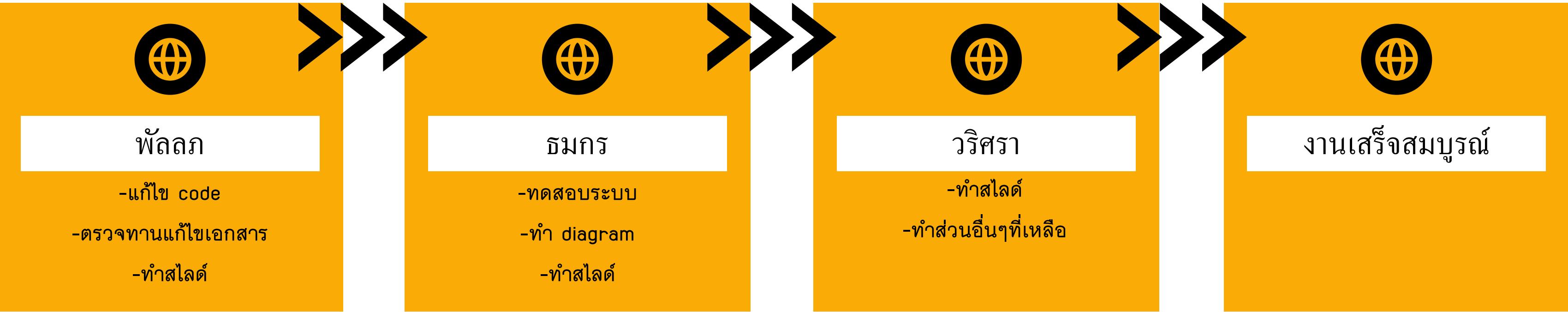


Extra credit

https://drive.google.com/file/d/1SpKfVLvTQpt_nl5SWi-744X_T1CJklyI/view?usp=sharing



ขั้นตอนการแบ่งงาน การดำเนินงาน



git: lonewolf

<https://github.com/2110503CEDT67/be-project-lone-wolf>

A screenshot of a GitHub repository showing a list of recent commits:

- Add blacklist user feature
- add plant-uml document
- Limit user access to others appointment in getAppointment
- add logout and User Model
- Fix addAppointment
- Change from hospital to hotel
- Initial commit
- add deadline

Each commit includes the author (Ayaiko), the branch (main), the commit hash, and the time of the commit.



link to project github repo

<https://github.com/2110503CEDT67/be-project-lone-wolf>