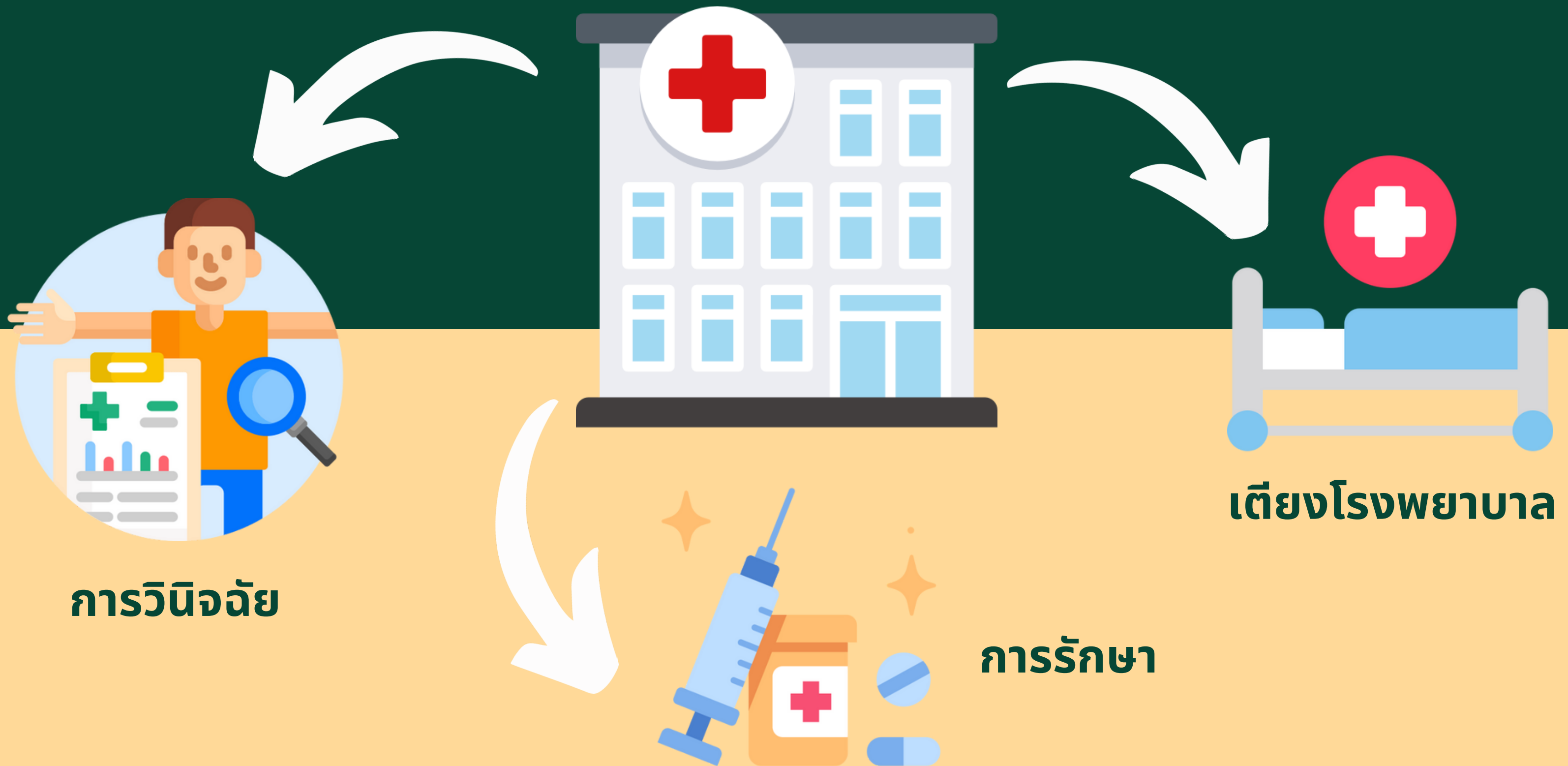


QHOSPITAL

ระบบจัดการคิวสำหรับโรงพยาบาล

Group 4

ที่มาและความสำคัญ



ปัญหาที่พบ

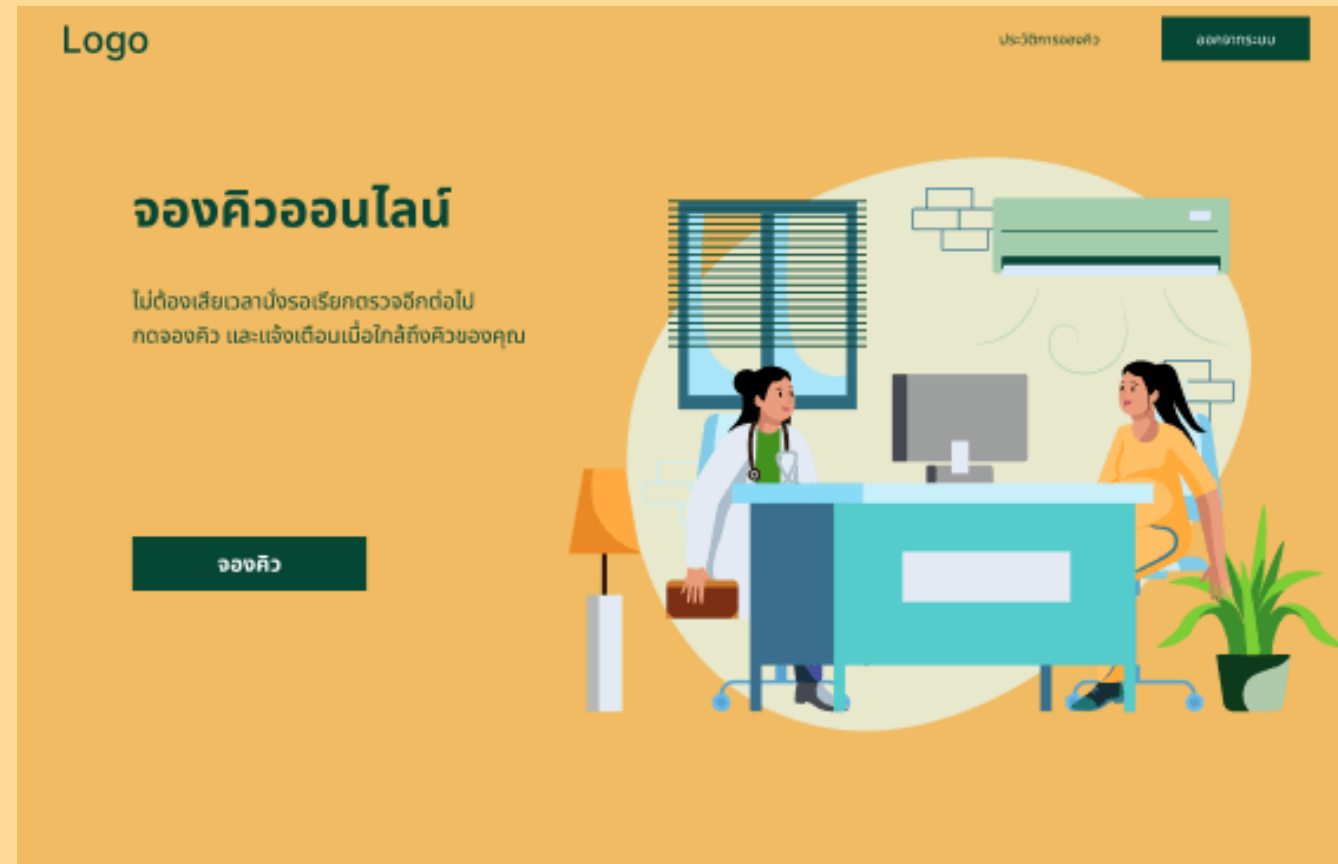


เวลาเป็นสิ่งมีค่า



ความต้องการรับบริการอย่าง
รวดเร็วโดยเฉพาะช่วงเร่งรีบ

แนวทางแก้ไข

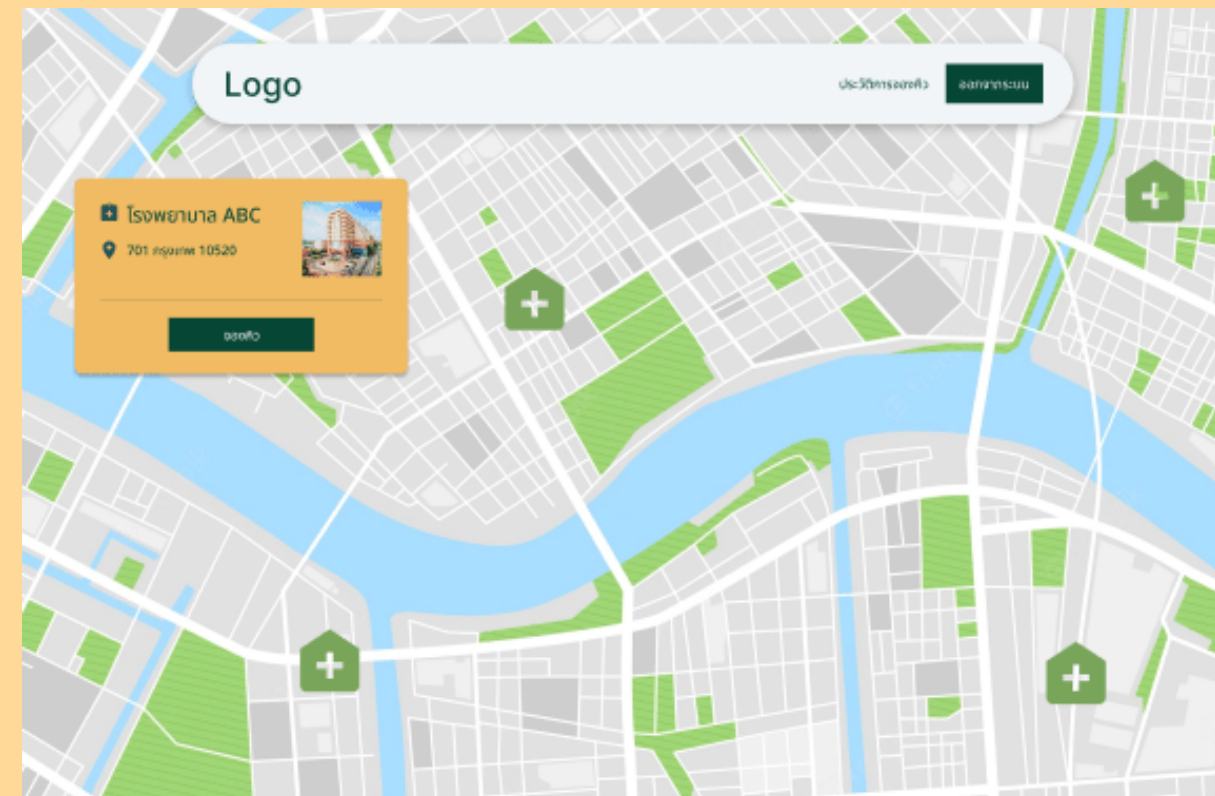


การสร้างระบบคิว :

เป็นเว็บไซต์ที่ผู้ป่วยสามารถจองคิวการเข้ารักษากับโรงพยาบาลที่ต้องการได้

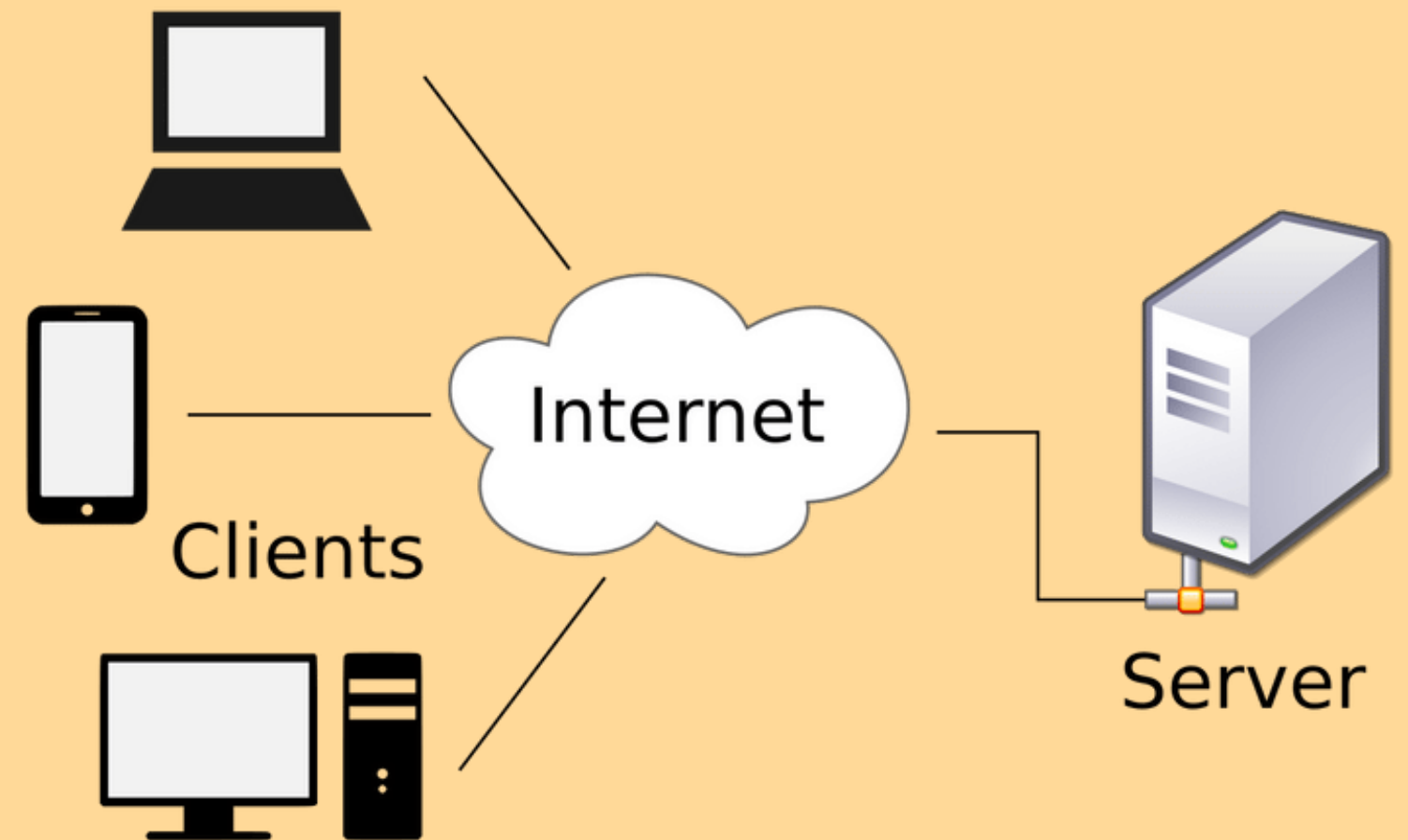
- หน้าแสดงโรงพยาบาลที่อยู่ใกล้กับผู้ป่วย
- ข้อมูลต่างๆของโรงพยาบาล
- ปริมาณความมาก-น้อยของจำนวนคิวที่มีผู้คนจองในปัจจุบัน

ใบจองคิว				
โรงพยาบาลสมเด็จฯ				
● ผู้ป่วยใหม่ ● ผู้ป่วยเก่า				
วันที่ 1/11/2022				
เวลา 10:30 - 11:00น.				
ยืนยัน				
วันที่	เวลา	สถานะ		
1/11/2022	9:00 - 9:30น.	ปานกลาง	จอง	
1/11/2022	9:30 - 10:00น.	ปานกลาง	จอง	
1/11/2022	9:00 - 9:30น.	เยอะมาก	จอง	
1/11/2022	10:30 - 11:00น.	ปานกลาง	จอง	
1/11/2022	11:00 - 11:30น.	ปานกลาง	จอง	
1/11/2022	11:30 - 12:00น.	ปานกลาง	จอง	
1/11/2022	12:00 - 12:30น.	ปานกลาง	จอง	
1/11/2022	13:00 - 13:30น.	ปานกลาง	จอง	



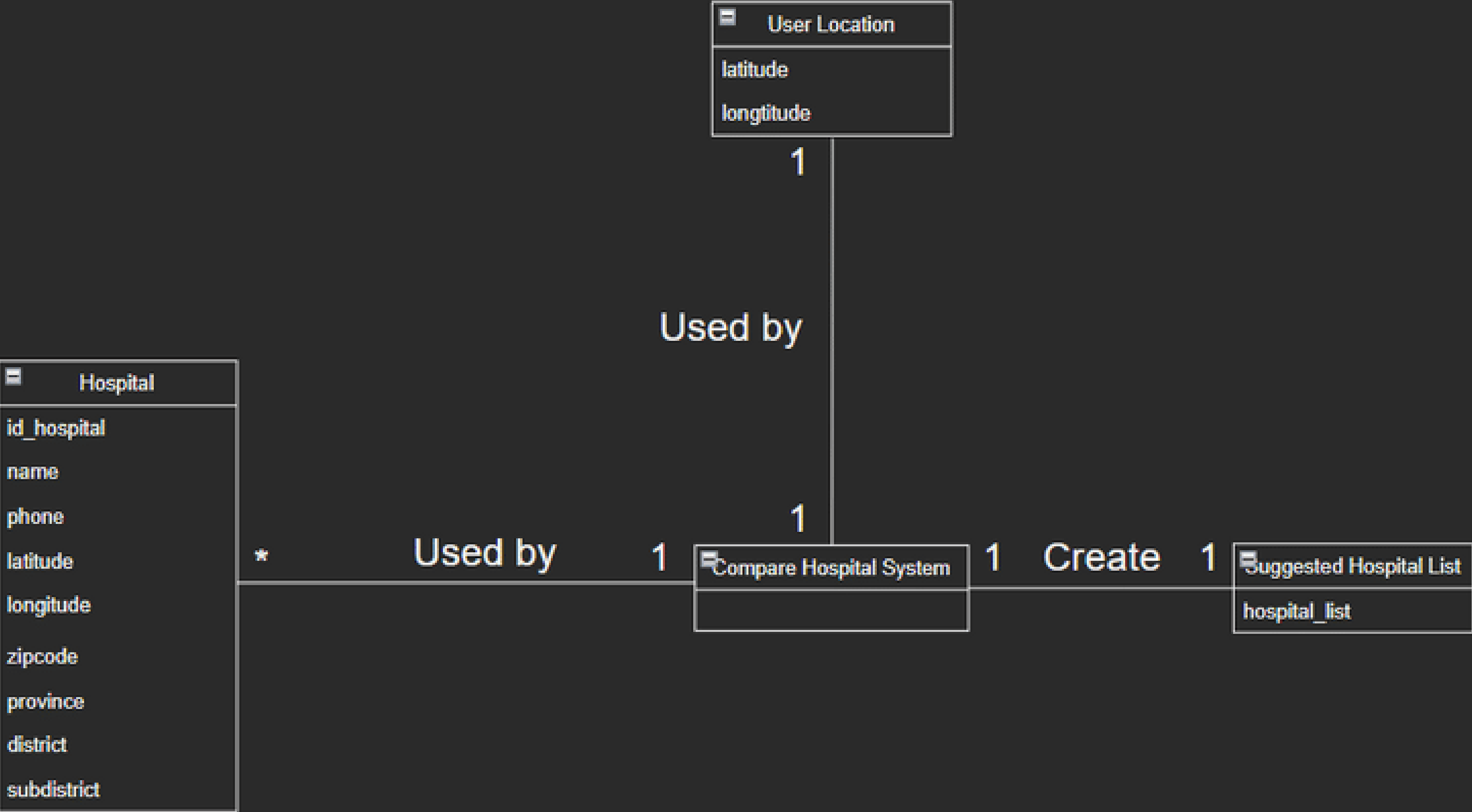
Software Architecture

- **Client-Server & N-tier**

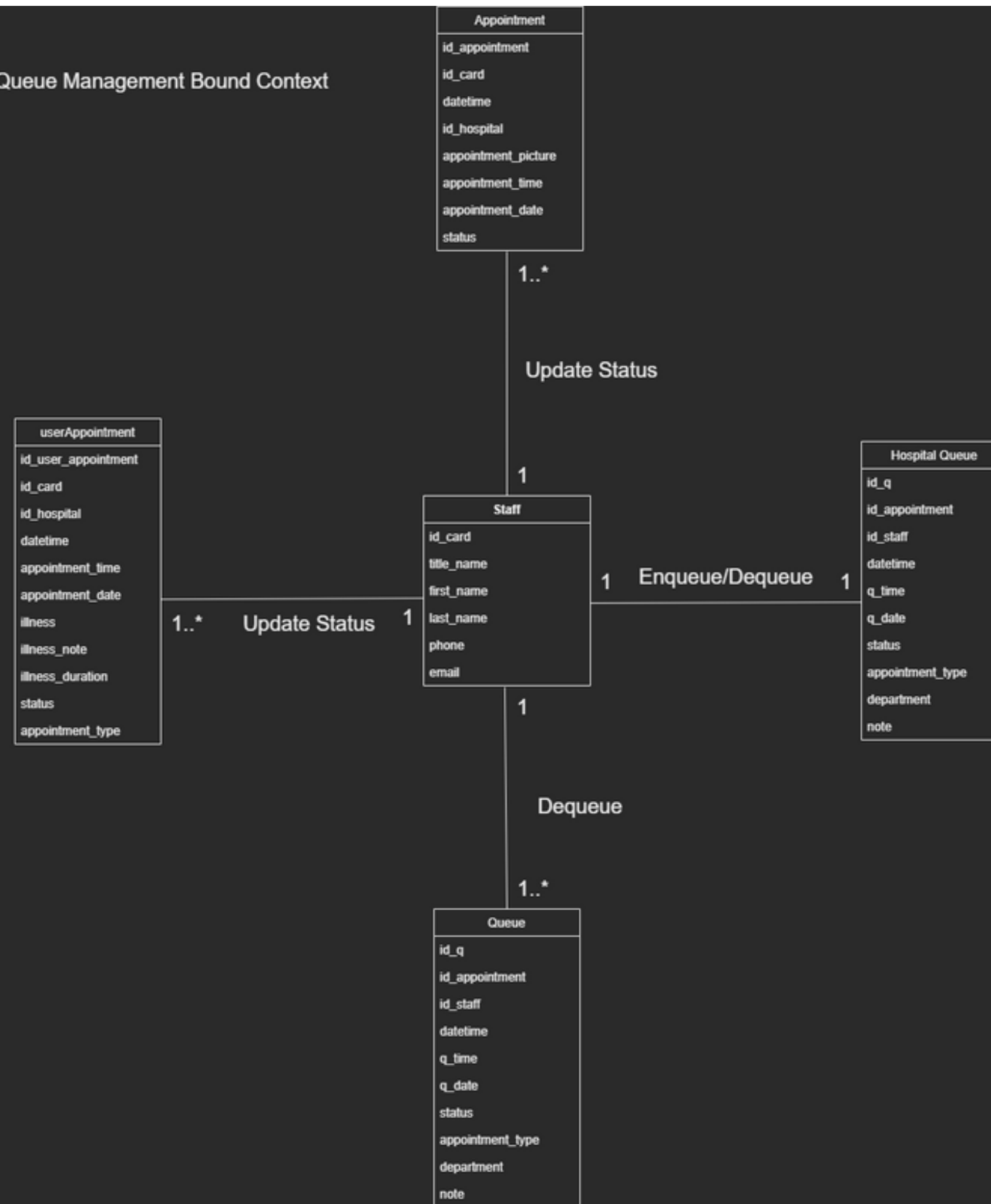


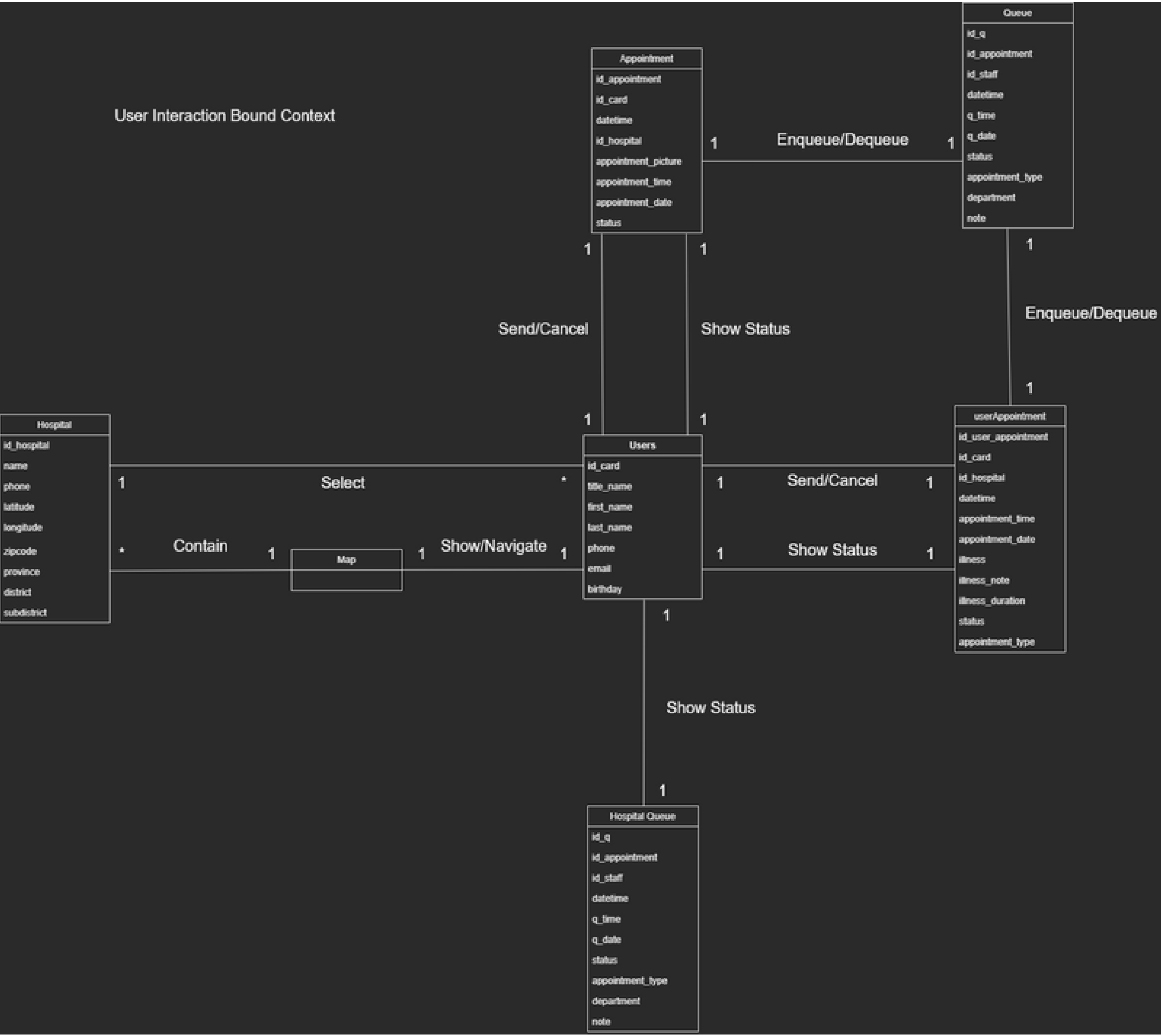
Software Design

Hospital Suggestion Bound Context



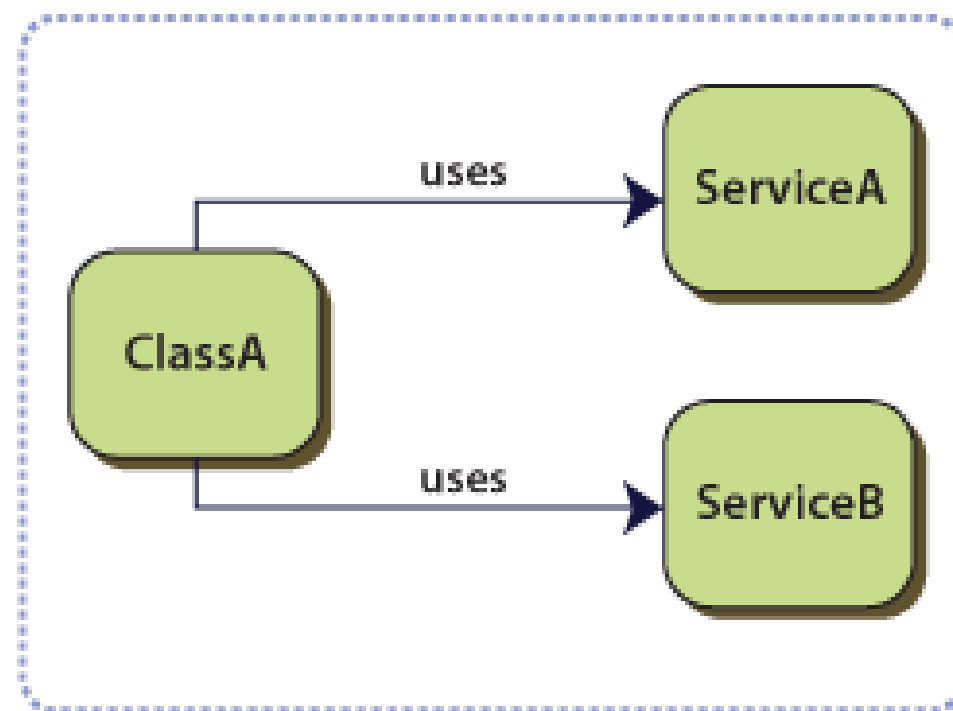
Hospital Queue Management Bound Context



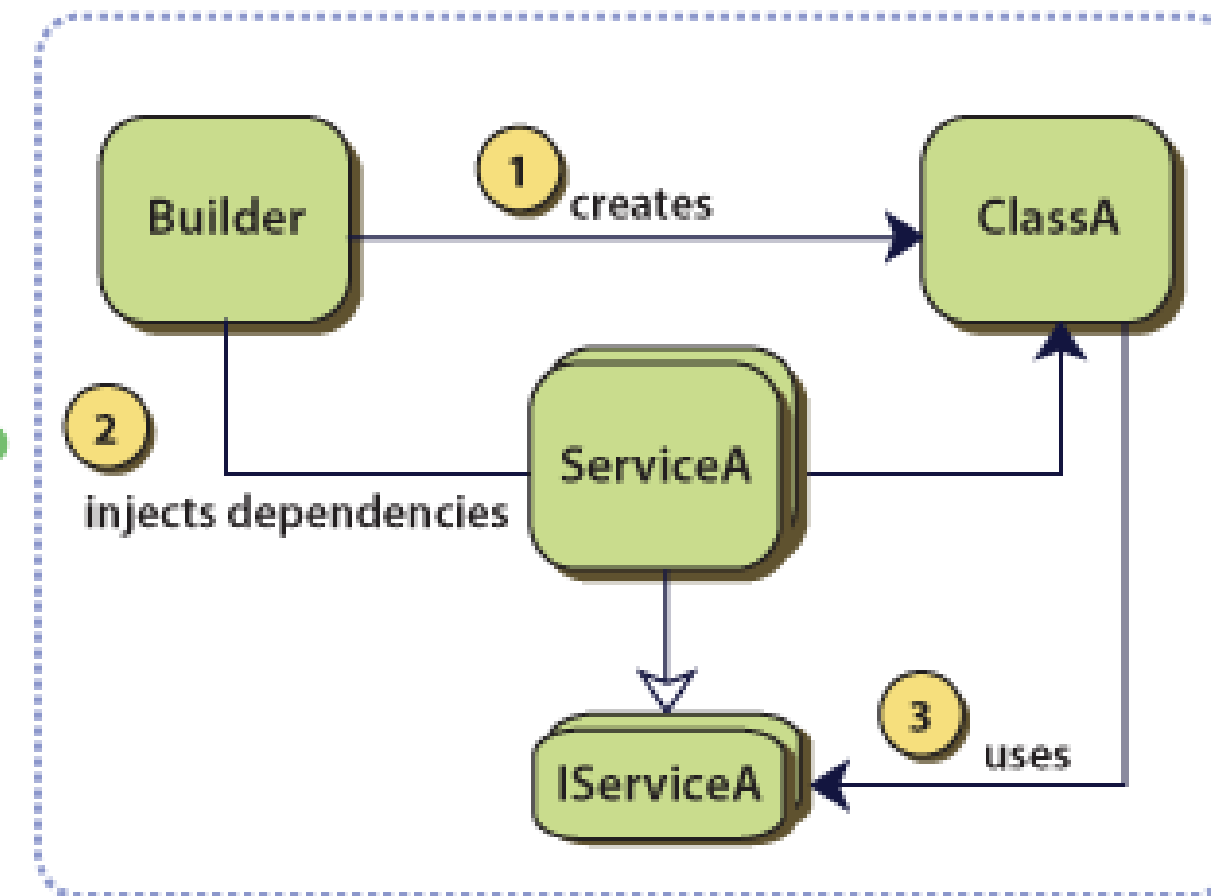


Design Pattern





Refactor
Code



Singleton Service

```
@Injectable()
export class UserAppointmentService extends ResponseHandler {
  constructor(
    @InjectRepository(UserAppointment)
    private readonly userAppointmentRepository: Repository<UserAppointment>,
    @InjectRepository(Staff)
    private readonly staffRepository: Repository<Staff>,
  ) {
    super();
  }
}
```

```
@Injectable()
export class RegisterService extends ResponseHandler {
  constructor(
    @InjectRepository(Registration)
    private readonly registrationRepository: Repository<Registration>,
    @InjectRepository(User)
    private readonly userRepository: Repository<User>,
  ) {
    super();
  }
}
```

```
@Injectable()
export class UsersService extends ResponseHandler {
  constructor(
    @InjectRepository(User)
    private readonly userRepository: Repository<User>,
    @InjectRepository(Registration)
    private readonly registrationRepository: Repository<Registration>,
  ) {
    super();
  }
}
```

```
@Module({
  imports: [TypeOrmModule.forFeature([Registration, User])],
  controllers: [RegistrationController],
  providers: [RegisterService],
})
export class RegistrationModule {}
```

Adapter

```
import { MiddlewareConsumer } from '../middleware/middleware-consumer.interface';  
export interface NestModule {  
  configure(consumer: MiddlewareConsumer): any;  
}
```

```
@Module({  
  imports: [  
    TypeOrmModule.forRoot(configService.getTypeOrmConfig()),  
    AuthModule,  
    TestModule,  
    AppointmentModule,  
    HospitalModule,  
    MapModule,  
    QueueModule,  
    RegistrationModule,  
    StaffModule,  
    TimelistModule,  
    UserAppointmentModule,  
    UsersModule,  
  ],  
})  
export class AppModule implements NestModule {  
  configure(consumer: MiddlewareConsumer) {  
    consumer  
      .apply(LoggerMiddleware)  
      .forRoutes({  
        path: '*',  
        method: RequestMethod.ALL,  
      });  
  
    // apply auth  
    consumer  
      .apply(AuthMiddleware)  
      .exclude('api/(register|login)')  
      .forRoutes({  
        path: '*',  
        method: RequestMethod.ALL,  
      });  
  }  
}
```

```
export abstract class ResponseAbstract {  
  public abstract responseBadRequest();  
  public abstract responseUnauthorized();  
  public abstract responseForbidden();  
  public abstract responseMethodNotAllowed();  
  public abstract responseConflict();  
}
```

Template Method

```
// 400 Bad Request  
override responseBadRequest(message?: string, payload?: object): ResponseInterface {  
  return {  
    statusCode: 400,  
    message: `failed: ${message || 'bad request'}`,  
    timestamp: new Date().toISOString(),  
  };  
}  
  
// 401 Unauthorized  
override responseUnauthorized(message?: string): ResponseInterface {  
  return {  
    statusCode: 401,  
    message: `failed: ${message || 'unauthorized'}`,  
    timestamp: new Date().toISOString(),  
  };  
}  
  
// 403 Forbidden  
override responseForbidden(message?: string): ResponseInterface {  
  return {  
    statusCode: 403,  
    message: `failed: ${message || 'forbidden'}`,  
    timestamp: new Date().toISOString(),  
  };  
}  
  
// 405 Method Not Allowed  
override responseMethodNotAllowed(message?: string): ResponseInterface {  
  return {  
    statusCode: 403,  
    message: `failed: ${message || 'method not allowed'}`,  
    timestamp: new Date().toISOString(),  
  };  
}  
  
// 409 Conflict  
override responseConflict(message?: string): ResponseInterface {  
  return {  
    statusCode: 409,  
    message: `failed: ${message || 'conflict'}`,  
    timestamp: new Date().toISOString(),  
  };  
}
```

Quality Attribute

Availability : Scenario 1

Source of Stimulus: Server ของเว็บไซต์

Stimulus: ไม่ตอบสนองต่อ request

Artifact: process

Environment: ช่วงเวลาทำงานปกติ

Response:

- แจ้งเตือนให้ผู้ดูแลทราบ
- หยุดการทำงานของ server เพื่อทำการแก้ไขข้อผิดพลาดที่เกิดขึ้น

Response Measure: ระยะเวลาที่ server ไม่สามารถใช้งานได้จะไม่เกิน 1 ชั่วโมง

Quality Attribute

Availability : Scenario 2

Source of Stimulus: คนดูแลพัฒนา backend ของเว็บไซต์

Stimulus: แก้ไขโค้ดผิดรูปแบบ

Artifact: process

Environment: ช่วงเวลาทำงานปกติ

Response:

- แจ้งเตือนให้ผู้พัฒนาทราบ
- ปิดระบบเว็บไซต์เพื่อทำการแก้ไขโค้ดที่เกิดข้อผิดพลาด

Response Measure: ระยะเวลาปิดเว็บไซต์และทำการแก้ไขจะใช้ไม่เกิน 30 นาที

Quality Attribute

Availability : Scenario 3

Source of Stimulus: ฮาร์ดแวร์ของ Server เว็บไซต์

Stimulus: เครื่อง Server ดับ

Artifact: communications channels

Environment: ช่วงเวลาทำงานปกติ

Response:

- บันทึกข้อผิดพลาด
- แจ้งเตือนให้ผู้ดูแลระบบ monitor
- ปิดตัว server ที่เกิดข้อผิดพลาดและทำการแก้ไข

Response Measure: ระยะเวลาในการแก้ไขจะใช้ไม่เกิน 1 ชั่วโมง

Quality Attribute

Integrability : Scenario 1

Source of Stimulus: ตลาดคอมพิวเตอร์

Stimulus: integrate เวอร์ชันใหม่ของคอมพิวเตอร์ที่มีอยู่ในระบบ

Artifact: ระบบทั้งหมด

Environment: หลังเปิดการทำงานระบบ

Response: ผ่านการทดสอบ integrate และ integrate สำเร็จ

Response Measure: ใช้เวลาในการทำไม่เกิน 14 วันทำงาน

Quality Attribute

Modifiability : Scenario 1

Source of Stimulus: ทีมผู้พัฒนา

Stimulus: ต้องการเพิ่มฟังก์ชันการทำงานใหม่

Artifact: โค้ด

Environment: ช่วงเวลาการเขียนโค้ด

Response: เพิ่มฟังก์ชันการทำงานใหม่ได้สำเร็จ

Response Measure: ไม่ส่งผลกับการทำงานส่วนอื่นๆ

Quality Attribute

Performance : Scenario 1

Source of Stimulus: ผู้ใช้งาน 2000 คน

Stimulus: ทำการจองคิวโรงพยาบาล 900 รายการภายในระยะเวลา 1 นาที

Artifact: ระบบจัดการคิวโรงพยาบาล

Environment: ช่วงเวลาทำงานปกติ

Response: ระบบจัดลำดับคิวโรงพยาบาลได้ทั้งหมด

Response Measure: ระยะเวลาในการจัดลำดับคิวเฉลี่ยไม่เกิน 5 วินาที

Quality Attribute

Performance : Scenario 2

Source of Stimulus: ผู้ใช้งาน 300 คน

Stimulus: ทำการยกเลิกคิวที่จอง 300 รายการภายในระยะเวลา 3 นาที

Artifact: ระบบจัดการคิวโรงพยาบาล

Environment: ช่วงเวลาทำงานปกติ

Response:

- ระบบนำคิวออกจากคิวโรงพยาบาลได้ถูกต้องทั้งหมด

Response Measure:

ระยะเวลาในการตรวจสอบและจัดลำดับคิวใหม่เฉลี่ยไม่เกิน 10 วินาที

Quality Attribute

Performance : Scenario 3

Source of Stimulus: ผู้ใช้งาน 200 คน

Stimulus: ทำการเข้าดูตำแหน่งโรงพยาบาลใน Google map 150 ครั้งใน 1 นาที

Artifact: ระบบเชื่อมต่อ Google map

Environment: ช่วงเวลาทำงานปกติ

Response: ระบบแสดงตำแหน่งโรงพยาบาลใน Google map ได้ถูกต้อง

Response Measure: ระยะเวลาในการค้นหาและแสดงตำแหน่งโรงพยาบาลเฉลี่ยไม่เกิน 5 วินาที

สมาชิกกลุ่ม

62010728 นายภูรินทร์

63010022 นายกฤต

63010052 นายก้องเกียรติ

63010062 นางสาวกนต์กนิษฐ์

63010086 นางสาวกิริติกร

63010226 นายชานน

63010249 นางสาวญาณิศา

63010271 นางสาวณกมล

บุญกระสินธ์

รุ่งโรจน์กิจกุล

ชุนงาม

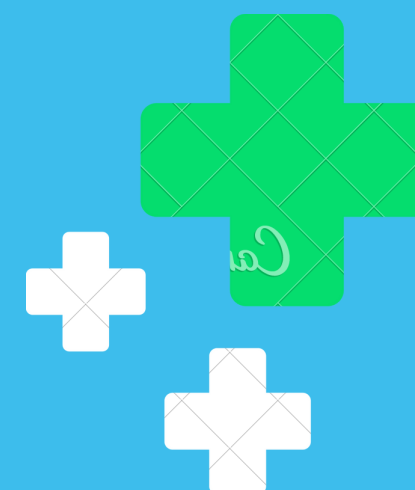
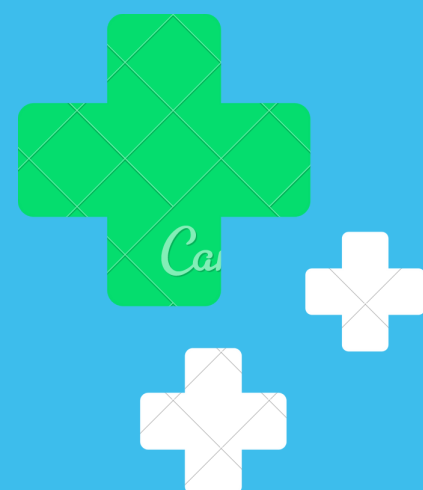
ทองแก่ง

พลับพลา

เต็มกมลศิลป์

พงษ์เมธา

แซ่เฮง



Thank You
