

캡스톤디자인 Ⅱ 중간보고서(표지)

프로젝트명 : 신생아 호흡 곤란 증후군의 X-ray 진단 연구
캡스톤 디자인Ⅱ, 중간보고서

Version 1.0

개발 팀원 명(팀리더): 장어진
조한용

대표 연락처:010-3819-2963
e-mail: ej2b514@gmail.com

캡스톤 디자인 II 중간보고서 내용

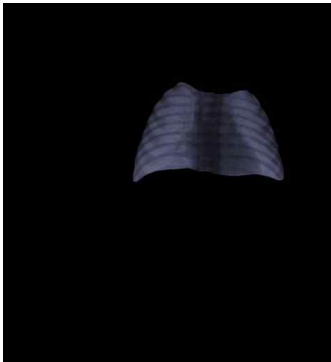
1. 요구사항 정의서에 명시된 기능에 대하여 현재까지 진척된 결과 및 그 내용을 기술하시오.

a) UNet으로부터 모델을 기반으로 폐 영역만 분할하여 이미지를 저장하는 코드 일부

```
for p in range(image_vis_t.shape[0]):
    for q in range(image_vis_t.shape[1]):
        if (pr_mask_t[p][q] == 0):
            for r in range(3):
                pr_image[p][q][r] = 0
        else:
            for r in range(3):
                pr_image[p][q][r] = image_vis_t[p][q][r].copy()

pr_image_convert = cv2.cvtColor(pr_image, cv2.COLOR_BGR2RGB)
```

결과 예시)



b) 분할된 결과를 바탕으로 폐 영역만 남기기 위해 crop 하는 코드 일부

```
img = cv2.imread(k, cv2.THREAD_COLOR)
if type(img) != type(None):
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

    blur = cv2.GaussianBlur(img, (3,3), sigmaX=0)
    ret, thresh1 = cv2.threshold(blur, 127, 255, cv2.THRESH_BINARY)

    edge = cv2.Canny(blur, 10, 250)

    kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (7,7))
    closed = cv2.morphologyEx(edge, cv2.MORPH_CLOSE, kernel)

    contours, _ = cv2.findContours(closed.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    total = 0

    contours_image = cv2.drawContours(img, contours, -1, (0,0,0), 3)

    contours_xy = np.array(contours)

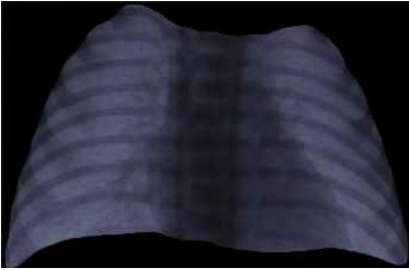
    x_min, x_max = 0, 0
    value = list()
    for i in range(len(contours_xy)):
        for j in range(len(contours_xy[i])):
            value.append(contours_xy[i][j][0])
            x_min = min(value)
            x_max = max(value)

    y_min, y_max = 0, 0
    value = list()
    for i in range(len(contours_xy)):
        for j in range(len(contours_xy[i])):
            value.append(contours_xy[i][j][1])
            y_min = min(value)
            y_max = max(value)

    x = x_min
    y = y_min
    w = x_max-x_min
    h = y_max-y_min

    print(k)
    img_trie = img[y:y+h, x:x+w]
    img_trie = cv2.cvtColor(img_trie, cv2.COLOR_BGR2RGB)
```

결과 예시)



c) 위 결과를 바탕으로 RDS 진단하기 위한 모델 학습 코드 (pretrained model load해 사용)

```
image_datasets = {x: datasets.ImageFolder(os.path.join(data_dir, x),
                                              data_transforms[x])
                  for x in ['train', 'val']}

dataloaders = {x: torch.utils.data.DataLoader(image_datasets[x], batch_size=batch_size,
                                              shuffle=True, num_workers=8)
              for x in ['train', 'val']}

dataset_sizes = {x: len(image_datasets[x]) for x in ['train', 'val']}
class_names = image_datasets['train'].classes

device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")

model = EfficientNet.from_pretrained('efficientnet-b5', num_classes=2)
model.to(device)

optimizer = optim.Adam(model.parameters(), lr=0.001)
criterion = nn.CrossEntropyLoss()
scheduler = lr_scheduler.StepLR(optimizer, step_size=7, gamma=0.1)

since = time.time()

best_model_weights = copy.deepcopy(model.state_dict())
best_acc = 0.0
```

d) 진단 모델의 가중치 시각화 코드 (grad-cam 일부)

```
#####model2#####
model2._modules.get('blocks')._modules.get('38')._swish.register_forward_hook(hook_feature)
model2._modules.get('blocks')._modules.get('38')._swish.register_backward_hook(backward_hook)

params = list(model2.parameters())
weight_softmax = np.squeeze(params[-2].cpu().detach().numpy()) # [1000, 512]

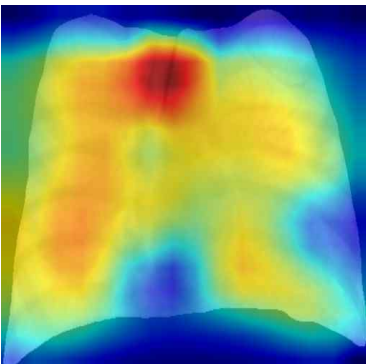
logit = model2(normed_torch_img)
h_x = F.softmax(logit, dim=1).data.squeeze() # softmax 결과

_, preds = torch.max(logit, 1)
pred = preds.cpu().numpy()
score = logit[:, 1].squeeze()
score.backward(retain_graph=True)

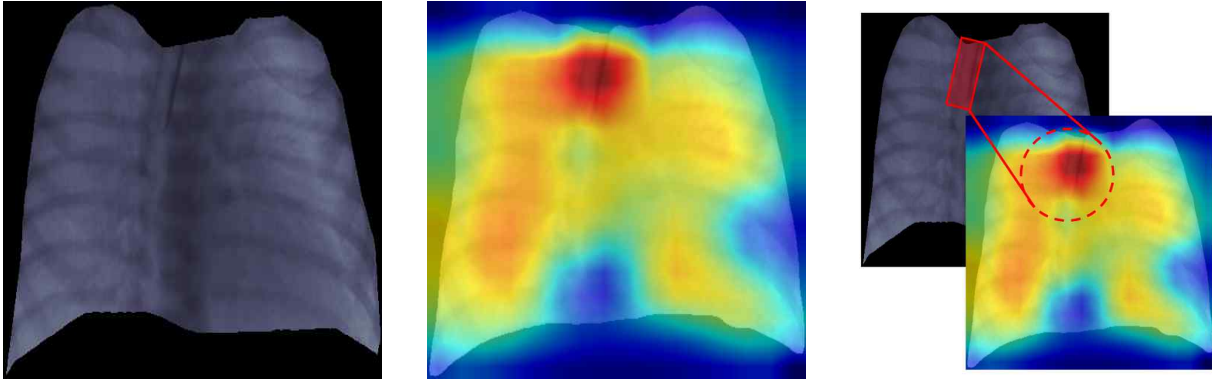
activations = feature_blobs[0].to(device)
gradients = backward_feature[-1][0]
b, k, u, v = gradients.size()

alpha = gradients.view(b, k, -1).mean(2)
weights = alpha.view(b, k, 1, 1)
grad_cam_map = (weights * activations).sum(1, keepdim=True)
grad_cam_map = F.relu(grad_cam_map)
grad_cam_map = F.interpolate(grad_cam_map, size=(512, 512), mode='bilinear', align_corners=False)
grad_cam_map2 = grad_cam_map
```

결과 예시)



2. 프로젝트 수행을 위해 적용된 추진전략, 수행 방법의 결과를 작성하고, 만일 적용과정에서 문제점이 도출되었다면 그 문제를 분석하고 해결방안을 기술하시오.



위 grad cam 결과에서 볼 수 있듯 가중치 시각화 결과 RDS 진단에 있어 삼관 영역을 중점적으로 보고 있음을 확인함. 따라서 해당 영역을 제거하고 모델을 다시 학습시켜 결과를 확인하고자 함.

[별첨1]

프로젝트명 : 신생아 호흡 곤란 증후군의 X-ray 진단 연구

소프트웨어 요구사항 정의서

Version 1.0

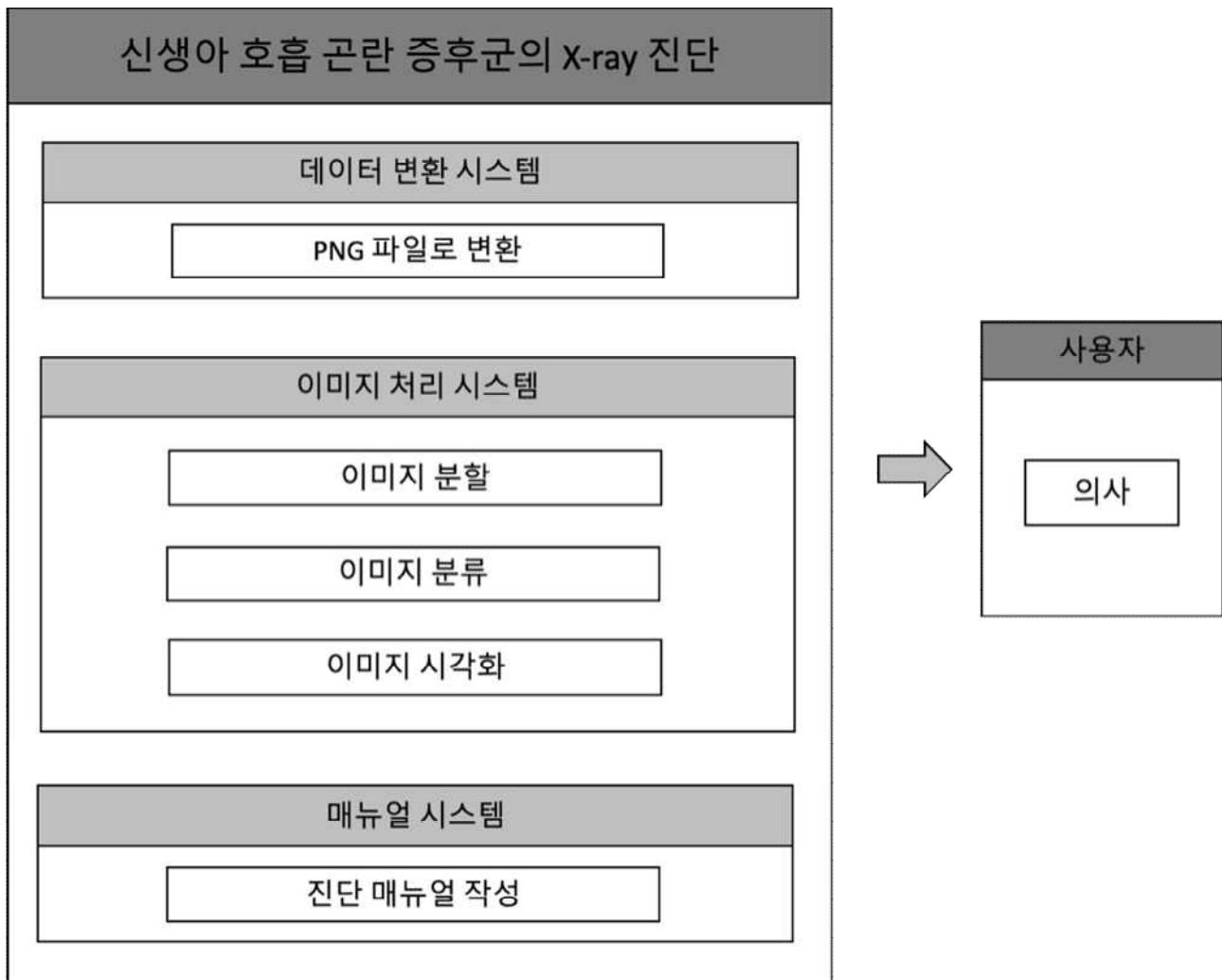
개발 팀원 명(팀리더): 장어진
조한용

대표 연락처:010-3819-2963
e-mail: ej2b514@gmail.com

목차

1. 개요
2. 시스템 장비 구성요구사항
3. 기능 요구사항
4. 성능 요구사항
5. 인터페이스 요구사항
6. 데이터 요구사항
7. 테스트 요구사항
8. 보안 요구사항
9. 품질 요구사항
10. 제약 사항
11. 프로젝트 관리 요구사항

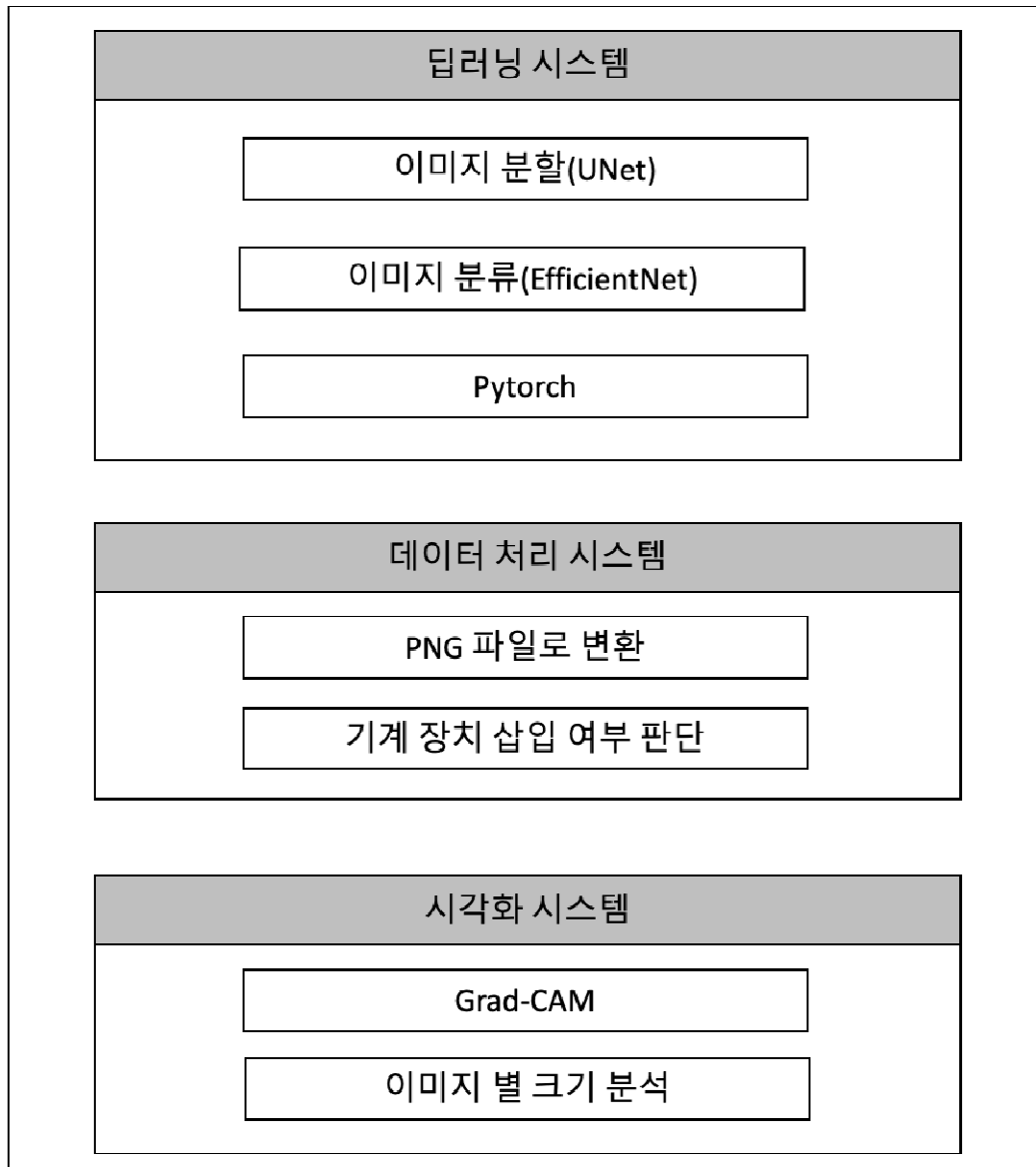
1. 시스템 개요



2. 시스템 장비 구성요구사항

요구사항 고유번호		ECR-001		
요구사항 명칭		딥러닝 모델 학습용 서버		
요구사항 분류		시스템 장비구성 요구사항	응락수준	필수
요구사항 상세설명	정의	- 딥러닝 모델 학습		
	세부 내용	<ul style="list-style-type: none"> - 장비 품목 : NVIDIA RTX A5000 - 장비 수량 : 1개 - 장비 기능 : 이미지 Segmentation 및 Classification 모델을 학습하기 위한 서버 - 장비 성능 및 특징 : 대용량 영상 데이터를 처리하기 위해 고성능 GPU를 이용하여 모델 학습을 수행함 		

3. 기능 요구사항



요구사항 고유번호		SFR-001		
요구사항 명칭		딥러닝		
요구사항 분류		기능요구사항	응락수준	필수
요구사항 상세설명	정의	- 딥러닝 모델을 이용하여 이미지 분할		
	세부 내용	<ul style="list-style-type: none"> - Unet++ - Image Segmentation을 사용하여 폐 영역 분할 		

요구사항 고유번호		SFR-002		
요구사항 명칭		딥러닝		
요구사항 분류		기능요구사항	응락수준	필수
요구사항 상세설명	정의	- 딥러닝 모델을 이용하여 이미지 분류		
	세부 내용	<ul style="list-style-type: none"> - EfficientNet - Image Classification을 사용하여 RDS 진단 		

요구사항 고유번호		SFR-003		
요구사항 명칭		시각화		
요구사항 분류		기능요구사항	응락수준	필수
요구사항 상세설명	정의	- Grad-CAM을 사용하여 결과 시각화		
	세부 내용	<ul style="list-style-type: none"> - Grad-CAM - Grad-CAM을 사용해서 결과 시각화 		

요구사항 고유번호		SFR-004		
요구사항 명칭		시각화		
요구사항 분류		기능요구사항	응락수준	필수
요구사항 상세설명	정의	- 데이터 시각화		
	세부 내용	<ul style="list-style-type: none"> - Matplotlib, Seaborn - Image 별 크기 분석 		

요구사항 고유번호		SFR-005		
요구사항 명칭		딥러닝		
요구사항 분류		기능요구사항	응락수준	필수
요구사항 상세설명	정의	- 오픈소스 라이브러리		
	세부 내용	<ul style="list-style-type: none"> - <u>Pytorch</u> - 모델 학습, 데이터 증대 등 <u>딥러닝에 필요한 라이브러리</u> 		

요구사항 고유번호		SFR-006		
요구사항 명칭		데이터 처리		
요구사항 분류		기능요구사항	응락수준	필수
요구사항 상세설명	정의	- 데이터 처리		
	세부 내용	<ul style="list-style-type: none"> - OpenCV - 기계 장치 삽입 여부 파악 		

요구사항 고유번호		SFR-007		
요구사항 명칭		데이터 처리		
요구사항 분류		기능요구사항	응락수준	필수
요구사항 상세설명	정의	- 데이터 변환		
	세부 내용	<ul style="list-style-type: none"> - <u>Pydicom</u> - 의료 영상에서 사용하는 DICOM 형식을 학습을 위해 PNG로 변환 		

4. 성능 요구사항

요구사항 고유번호		PER-001		
요구사항 명칭		이미지 분할 시간		
요구사항 분류		성능	응락수준	필수
요구사항 상세설명	정의	- Unet++을 통해 이미지를 분할 후 반환하는 시간		
	세부 내용	- 이미지 한 장당 1분 내에 분할이 이루어져야 하며 분할된 이미지를 반환한다.		

요구사항 고유번호		PER-002		
요구사항 명칭		이미지 분류 시간		
요구사항 분류		성능	응락수준	필수
요구사항 상세설명	정의	- EfficientNet을 통해 이미지를 분류 후 반환하는 시간		
	세부 내용	- 이미지 한 장당 1분 내에 분류가 이루어져야 하며 분류된 이미지를 반환한다.		

요구사항 고유번호		PER-003		
요구사항 명칭		이미지 시각화 시간		
요구사항 분류		성능	응락수준	필수
요구사항 상세설명	정의	- Grad-CAM을 통해 이미지를 분류 후 반환하는 시간		
	세부 내용	- 이미지 한 장당 1분 내에 시각화가 이루어져야 하며 <u>시각화된</u> 이미지를 반환한다.		

요구사항 고유번호		PER-004		
요구사항 명칭		데이터 처리 오류 응답시간		
요구사항 분류		성능	응락수준	필수
요구사항 상세설명	정의	- 사용자 입력 응답시간		
	세부 내용	- 모든 입력에 대해서 잘못된 형식에 대한 오류는 1초 이내로 응답을 해줘야 한다.		

5. 인터페이스 요구사항

요구사항 고유번호		SIR-001		
요구사항 명칭		사용자 인터페이스 가이드		
요구사항 분류		인터페이스 요구사항	응락수준	필수
요구사항 상세설명	정의	- 서비스를 이용하기 위한 방법		
	세부 내용	<ul style="list-style-type: none"> - RDS 판별을 위한 인터페이스는 Python 파일 실행이 가능한 환경에서 가능하도록 구성 - 이미지 분할, 분류, 시각화를 위한 Python 파일을 제공 - 사용 가능한 데이터를 이용하여 Python 파일을 실행 - 실행 환경은 터미널 혹은 Google Colab 환경에서 실행 가능하도록 가이드 라인을 제시 		

6. 데이터 요구사항

요구사항 고유번호		DAR-001		
요구사항 명칭		초기자료 구축		
요구사항 분류		데이터	응락수준	필수
요구사항 상세설명	정의	- 초기자료 구축		
	세부 내용	<ul style="list-style-type: none"> - 시스템 테스트를 위해서 초기 자료를 사전에 구축해야 한다. 		

요구사항 고유번호		DAR-002		
요구사항 명칭		데이터 관리		
요구사항 분류		데이터	응락수준	필수
요구사항 상세설명	정의	- 데이터를 관리하기 위한 방법		
	세부 내용	<ul style="list-style-type: none"> - 누락된 데이터가 없도록 한다. - 이미지는 경로를 저장한다. 		

7. 테스트 요구사항

요구사항 고유번호	TER-001		
요구사항 명칭	테스트 방안		
요구사항 분류	테스트	응락수준	필수
요구사항 세부내용	<ul style="list-style-type: none"> - 시스템 테스트는 제공되기로 한 요구사항을 모두 제공하며, 베이스라인을 제공 - 제공되기로 한 요구사항을 제공하는 여부는 검증 활동을 통해 예상 결과가 도출되었을 경우를 기준으로 평가 - 단위 기능에 대한 테스트는 자동화하여 테스트 함 - 각 테스트를 위한 구체적인 틀과 적용 방안을 제안하여 기능의 오류를 빠르게 발견하여 더 많은 디버그 과정을 거쳐 시스템의 오류를 줄임 		

8. 보안 요구사항

요구사항 고유번호	SER-001		
요구사항 명칭	데이터 보안		
요구사항 분류	보안	응락수준	필수
요구사항 세부내용	<ul style="list-style-type: none"> - 민감한 정보가 들어있는 의료 데이터는 개발 후 바로 삭제한다. - 사용자 이외 다른 사용자가 데이터를 볼 수 없게 한다. 		

9. 품질 요구사항

요구사항 고유번호		QUR-001		
요구사항 명칭		결함 발생률		
요구사항 분류		품질 요구사항	응락수준	필수
요구사항 상세설명	정의	- 품질 관리(기술 관점)		
	세부 내용	<ul style="list-style-type: none"> - 시험 운영 기간 동안 발견된 결함 수를 측정하고, 결함 발생률이 5% 이상이면 시스템 오픈 기간을 연장해야 함 - 시험 운영 기간 동안 결함이 계속해서 발견되면 운영을 중단하여야 함 		

요구사항 고유번호		QUR-002		
요구사항 명칭		결함 발생률		
요구사항 분류		품질 요구사항	응락수준	필수
요구사항 상세설명	정의	- 품질 관리(프로젝트 관리 관점)		
	세부 내용	<ul style="list-style-type: none"> - 산출물 및 각종 안내서(사용자 측면, 시스템 운영자 측면 등)의 관리 방안을 제시해야 함 - 개발 보고서를 제출해야 함. 		

요구사항 고유번호		QUR-003		
요구사항 명칭		상호 운용		
요구사항 분류		품질 요구사항	응락수준	필수
요구사항 상세설명	정의	- 이식성		
	세부 내용	<ul style="list-style-type: none"> - 가상화 기술(docker 등)을 사용하여 다른 운영체제로 이식하더라도 기존의 환경을 그대로 유지하여 운영하기 쉽도록 함. 		

10. 제약 사항

요구사항 고유번호	COR-001		
요구사항 명칭	개발업무 모듈화 및 자원 활용 방안		
요구사항 분류	제약사항	응락수준	필수
요구사항 세부내용	<ul style="list-style-type: none"> - 유연성,확장성을 확보할 수 있도록 모듈화를 지향함 - 현재 보유한 H/W,S/W를 최대한 재활용함 		

요구사항 고유번호	COR-002		
요구사항 명칭	시스템 구조 설계		
요구사항 분류	제약사항	응락수준	필수
요구사항 세부내용	<ul style="list-style-type: none"> - 분산설계, 데이터 유형, 프로세스 환경유형,사용자 유형, 시스템 토폴로지가 고려되어 구조설계가 이루어져야한다. - 개발시 애자일 방법론에 맞춰서 개발함. - 디자인 패턴에 맞춰서 개발함 		

11. 프로젝트 관리 요구사항

요구사항 고유번호	PMR-001		
요구사항 명칭	품질 관리		
요구사항 분류	프로젝트 관리	응락수준	필수
요구사항 세부내용	<ul style="list-style-type: none"> - 품질 보증의 범위, 조직, 절차, 점검 방법 등을 제시하여야 함 - 제안사는 품질 보증을 보장하기 위한 품질 보증 방안을 제시하여야 함 - 대외적으로 인정받을 만한 품질 보증 관련 인증을 받은 경우 이를 입증해야 함 		

요구사항 고유번호	PMR-002		
요구사항 명칭	품질 관리		
요구사항 분류	프로젝트 관리	응락수준	필수
요구사항 세부내용	<ul style="list-style-type: none"> - 산출물 및 각종 안내서(시스템 운영자, 사용자 안내서 등)의 관리 방안 제시 - 개발 및 커스터마이징 보고서 제출 		