**<객체지향 분석설계 팀 프로젝트>**

**<1팀>**

|  |  |  |
| --- | --- | --- |
| **학번** | **이름** |  |
| **2016156003** | **김동혁** |  |
| **2016156007** | **김영민** |  |
| **2016156009** | **김재현** |  |
| **2016156022** | **이영채** |  |

**목차**

[**1장 토의 보고서** 1](#_Toc57923999)

[**1-1 프로젝트 개요** 1](#_Toc57924000)

[**1.2 UML 상용 Tool에 대하여 사용설명서 작성** 4](#_Toc57924001)

[**2장 토의 보고서** 9](#_Toc57924002)

[**2.1 객체 지향 개념 정의** 9](#_Toc57924003)

[**3장 토의 보고서** 11](#_Toc57924004)

[**3.1 프로세스 정의** 11](#_Toc57924005)

[**3.2 유스케이스 다이어그램 모델링** 11](#_Toc57924006)

[4**장 토의 보고서** 17](#_Toc57924007)

[**4.1 프로세스 정의** 17](#_Toc57924008)

[4.1.1 문제 기술 17](#_Toc57924009)

[4.1.2 클래스 추출 18](#_Toc57924010)

[4.1.3 메서드 추출 18](#_Toc57924011)

[4.1.4 클래스간 관계 및 구조화 18](#_Toc57924012)

[**4.2 클래스 다이어그램 모델링** 19](#_Toc57924013)

[**5장 토의 보고서** 21](#_Toc57924014)

[**5.1 프로세스 정의** 21](#_Toc57924015)

[5.1.1 문제 기술 21](#_Toc57924016)

[5.1.2 액터,클래스 선정 22](#_Toc57924017)

[5.1.3 액터/객체간 메시지 정의 22](#_Toc57924018)

[5.1.4 객체 추가 24](#_Toc57924019)

[**5.2 모델링** 24](#_Toc57924020)

[**6장 토의 보고서** 27](#_Toc57924021)

[**6.1 프로세스 정의** 27](#_Toc57924022)

[6.1.1 문제 기술 27](#_Toc57924023)

[6.1.2 객체 선정 28](#_Toc57924024)

[6.1.3 객체를 다이어그램에 위치 및 링크정의 28](#_Toc57924025)

[6.1.4 객체간 메시지 정의하기 29](#_Toc57924026)

[6.2.1 순차다이어그램 변환 하여 통신 다이어그램 작성 30](#_Toc57924027)

[6.2.2 명세서기반 통신 다이어그램 작성 32](#_Toc57924028)

[**중간 발표 이후 토의 보고서** 35](#_Toc57924029)

[**1. 중간 발표 참관 후 팀의 회의 활동에 대한 피드백 및 진행 방향** 35](#_Toc57924030)

[**2. 개선 사항 및 결과** 35](#_Toc57924031)

[**7장 토의 보고서** 37](#_Toc57924032)

[**7.1 프로세스 정의** 37](#_Toc57924033)

[7.1.1 작성 대상 선정 37](#_Toc57924034)

[7.1.2 활동 상태 추출 38](#_Toc57924035)

[7.1.3 모델링 38](#_Toc57924036)

[1) 사전 조건 모델링 38](#_Toc57924037)

[2) 유스케이스 종료와 사후 조건 확인 40](#_Toc57924038)

[3) 선택 흐름 조건 작성 41](#_Toc57924039)

[4) 동기화 조건 작성 42](#_Toc57924040)

[**7.2 활동 다이어그램 작성** 43](#_Toc57924041)

[**8장 토의 보고서** 44](#_Toc57924042)

[**8.1 프로세스 정의** 44](#_Toc57924043)

[8.1.1 작성 대상 선정 44](#_Toc57924044)

[8.1.2 상태 정의 45](#_Toc57924045)

[8.1.3 상태 나열/전이 정의 45](#_Toc57924046)

[8.1.4 상세 부분 정의 47](#_Toc57924047)

[**8.2 상태 다이어그램 작성** 49](#_Toc57924048)

[**9장 토의 보고서** 50](#_Toc57924049)

[**9.1 프로세스 정의** 50](#_Toc57924050)

[9.1.1 컴포넌트 대상 정의 50](#_Toc57924051)

[9.1.2 컴포넌트 식별 51](#_Toc57924052)

[9.1.3 인터페이스 정의 및 컴포넌트 배치 51](#_Toc57924053)

[9.1.4 의존 관계와 실체화 관계 정의 52](#_Toc57924054)

[**9.2 컴포넌트 다이어그램 작성** 53](#_Toc57924055)

[**10장 토의 보고서** 54](#_Toc57924056)

[**10.1 프로세스 정의** 54](#_Toc57924057)

[10.1.1 노드를 식별하여 정의 54](#_Toc57924058)

[10.1.2 컴포넌트를 식별 55](#_Toc57924059)

[10.1.3 노드간 구성관계를 정의 55](#_Toc57924060)

[10.1.4 노드에 컴포넌트를 배치 58](#_Toc57924061)

[**10.2**  **배치 다이어그램 작성** 59](#_Toc57924062)

[**11장 토의 보고서** 60](#_Toc57924063)

[**11.1 프로세스 정의** 60](#_Toc57924064)

[11.1.1 패키지 요소 파악 60](#_Toc57924065)

[10.1.2 패키지의 외부 접근 허용 요소 정의 61](#_Toc57924066)

[10.1.3 외부 패키지에 의존하는 패키지 import 표현 62](#_Toc57924067)

[10.1.4 병합된 패키지의 경우 관계 표현 63](#_Toc57924068)

[**10.2**  **배치 다이어그램 작성** 63](#_Toc57924069)

|  |  |
| --- | --- |
| **1장 토의 보고서** | |
| **토의 사항** | **1장. 프로젝트 개요 및 Tool Manual**   * 1. **프로젝트 개요**  1. **프로젝트 팀 구성 (팀 운영 지침 및 보고서 양식 정의)** 2. **프로젝트 주제(정보통신 서비스 시스템)** 3. **프로젝트 개요 및 기능 정의**   **1.2 UML 상용 Tool에 대하여 사용설명서 작성** |
| **진행일자** | **2020년 9월 07일 17:20~19:00** |
| **참석자** | **김동혁(팀장), 김영민(서기), 김재현, 이영채** |
| **1-1 프로젝트 개요** | |
| **토의 내용** | **1) 프로젝트 팀 구성(팀 운영 지침 및 보고서 양식 정의)**  **[팀 운영 지침]**   1. **토의 결과 주 2회 디스 코드 및 카카오 톡으로 팀 프로젝트를 진행** 2. **차질이 없다면 월요일 5시부터 7시 사이, 필요하다면 추가적으로 목요일 10시부터 11시에 토의를 진행하기로 함** 3. **매 주마다 팀장과 서기를 돌아가면서 분담하기로 함** 4. **이번주 팀장을 한 인원이 다음주 서기를 한다.** 5. **팀장 순서는 이름 순서대로 김동혁, 김영민, 김재현, 이영채 순서로 결정** 6. **만약 개인의 사정으로 순서의 변경이 있다면 유동적으로 그 다음 사람이 팀장을 맡는다.** 7. **추가할 팀 운영 지침 및 보고서 양식이 있다면 해당 주 서기가 수정하도록 한다.**   **[보고서 양식]**   1. **보고서의 양식은 현장 실습에서 사용하던 보고서의 양식을 참고하여 활용할 것** 2. **개인 의견 -> 종합 의견 -> 팀 결론 순서로 보고서를 작성하기** 3. **매주 토의 후 서기가 보고서를 작성한 뒤 조원들과 수시로 공유한다.** 4. **조원들은 보고서를 피드백하고 수정이 필요하다면 서기에게 제시한다.** 5. **기존 보고서 양식을 유지하며 작성하되 양식을 변경 해야 하는 주제가 있다면 조원과 토의를 통해서 결정 후 수정**   **2) 프로젝트 주제 (정보통신 서비스)**  **[프로젝트 주제 선정]**   1. **토의를 통해서 IOT, 딥 러닝을 활용한 주제를 선정하기로 함** 2. **고장 예측 시스템 -> 내부 구조를 생각하기 어려움**   **음성인식을 통한 사람의 감정 추측 -> 사람의 감정을 아직 정확하게 추정이 불가능하다고 판단**  **IOT를 접합한 스마트 조명 -> 딥 러닝을 접목하기 어려움**     1. **어린이 보호구역 사고 방지를 위한 스마트 Area 채택** 2. **위의 주제들에 대해 토의를 통해서 효과적이고 적합한 주제를 선정하기로 함**   **3) 프로젝트 개요 및 기능 정의**   1. **어린이 보호구역에서 상호 예상치 못한 교통 사고를 미연에 방지하기 위한 스마트 Area** 2. **어린이 보호구역에서 갑작스러운 횡단으로 인한 사고 예방을 사전에 알림** 3. **네비게이션 혹은 어플리케이션과 연동하여 알람** 4. **실시간으로 해당 구역안에 카메라 뷰를 제공함으로써 사전에 운전자가 도로 근처에 어린이의 존재를 파악할 수 있음** 5. **운전자 근처에 어린이가 있을 때 알람을 해줌** 6. **3D 가상 뷰를 활용하여 위치 정보를 제공**   **[추가]**  **[보고서 작성 양식]**   1. **제목은 14pt, 소제목은 12pt, 내용은 10pt bold체로 작성한다.** 2. **제목 부류의 글자는 맑은 고딕(제목), 내용의 글자는 맑은 고딕(본문)으로 지정한다.** 3. **도출된 의견 중에 토론 결과의 핵심이 되는 의견은 노란색 형광펜을 사용한다.**   **[수정]**  **[팀 운영 지침]**   1. **화요일 수업 토의 시간에 해당 팀 프로젝트의 미리 수행해야할 주제들을 정의한다.** 2. **이후 개인마다 해당 주제에 대해 자신의 의견을 작성하고 디스코드 회의 전에 카카오톡으로 제출한다.** 3. **서기는 팀원들의 의견을 취합하여 회의전에 개인 의견 보고서를 작성한다.** 4. **차질이 없다면 매주 수요일 10시 혹은 목요일 12시에 팀원이 모두 모여 디스코드 회의를 통해 개인 의견 보고서에서 더 적합하다고 생각되는 내용들을 취합하여 토의를 통해 최종 결론을 도출한다.**   **<프로젝트 주제 변경 이유>**  선정 주제가 적절치 않아 팀원과의 토의 끝에 해당 주제로 변경하게 되었음.  **<프로젝트 주제 재선정>**   1. **기존 프로젝트주제를 변경하여 딥러닝을 활용한 모터고장주기예측 시스템을 구현하기로 함.**   **<프로젝트 개요>**   1. **전기모터의 고장주기를 예측하는 시스템을 구현한다.** 2. **수집한 다양한 데이터를 활용하여, 딥러닝을 수행, 고장주기를 예측한다.** 3. **중요설비가 고장 나기 전 미리 대비하여 예기치 못한 상황으로 인한 손해를 미연에 방지할 수 있다.**   **<프로젝트 기능정의>**  **(팀원의견)**   |  |  | | --- | --- | | **김영민** | 1. **ATmega를 이용하여 센서데이터를 수집해서 저장하자.** 2. **딥러닝 데이터를 알림을 통해 사용자에게 전달하는 것이 좋아보인다.** 3. **특정 모터의 고장 주기 예측을 위해 센서 데이터수집이 중요할 것 같다.** | | **이영채** | 1. **딥러닝을 진행해서 고장주기를 예측해야 할 것 같다.** 2. **딥러닝과 데이터 시각화는 파이썬으로 진행하면 좋을 것 같아** 3. **앱과 웹을 동시에 구현하는 것도 어렵지 않을 것 같다.** | | **김재현** | 1. **데이터 베이스에서 자료를 받아 딥러닝을 수행하자.** 2. **딥러닝한 데이터도 데이터베이스에 저장하자.** 3. **웹보다는 앱이 구현면에서 더 편리할 것 같다.** | | **김동혁** | 1. **데이터를 저장할 때 클라우드를 사용하여 저장하는게 좋을 것 같다.** 2. **클라우드 데이터를 이용하여 시각화로 사용한테 보여주는게 좋을 것 같아** 3. **시각화 데이터는 웹으로 전송해주면 확장성이 높을 것 같다** |   **[최종결론]**   1. **ATmega를 이용하여 센서데이터를 수집한 뒤 클라우드에 저장함.** 2. **저장한데이터를 기반으로 딥러닝을 진행하여, 고장주기를 예측함.** 3. **클라우드 내 데이터를 시각화하여 사용자에게 제공함.** 4. **웹 또는 앱을 통해 사용자에게 알림을 제공한다.** |
| **1.2 UML 상용 Tool에 대하여 사용설명서 작성** | |
| **토의 내용** | **팀 프로젝트 협업을 통해 사용 설명서를 작성하였음**   1. **사용할 UML은 토의를 통해 StarUML를 활용하기로 하였음**   **1. StarUML**  **[Tool 메뉴]**    **[단축키 설명]**    **[메뉴 설명]**        **2. Draw.io**  **[전체 메뉴]**    **[도형 선택] [스타일 변경]**    **[회의 툴 추가]**  **Discord는 원래 게이밍용 오디오 통화 어플리케이션이지만, 화면공유기능이나, 타 프로그램에 비해서 통화음질이 좋고 무설치로도 운용할 수 있어 활용하기 좋았기 때문에 해당 툴을 회의 툴로 선정하였다.**  **Discord (음성통화기능, 화면 공유기능.)**  **[초기 화면] : 팀원들이 참여하고 있는 객지분 채널이다.**  **우측에서는 참여인원을 확인할 수 있으며, 좌측에서는 음성채팅 채널과 화면공유 상황을 확인 할 수 있다.**    **해당 화면은 팀원이 모여 있는 채널이다.**  **[화면 공유 송출자 시점] : 화면공유 기능 사용시 설정하는 창이다.**  **각 프로그램별로, 디스플레이 드라이버로 화면공유대상을 지정할 수 있다.**    **[화면 공유 참여자 시점] : 화면공유 참여자의 시점이다. 화면 송출자의 화면을 그대로 확인 할 수 있으며, 레이턴시도 크게 느껴지지 않아 수정화면을 보면서 토의 하는 데에 어려움이 없다.** |

|  |  |
| --- | --- |
| **2장 토의 보고서** | |
| **토의 사항** | **2장. 객체 지향 개념 정의**  **2-1. 기능으로부터 객체 지향 개념 정의** |
| **진행일자** | **2020년 9월 15일 14:50~15:20** |
| **참석자** | **김영민(팀장), 김재현(서기), 김동혁, 이영채** |
| **2.1 객체 지향 개념 정의** | |
| **토의 내용** | **<모터(팬) 고장 예측 시스템>**   |  |  | | --- | --- | | **각각의 객체가 사용하는 센서가 달려 있는 모터의 고장 주기를 예측한다.**  **에어컨은 온도를 측정하여 냉/난방 시스템을 상황에 따라 사용하여 내부 온도를 적정선에서 유지시키며 습도를 확인하여 제습기능을 이용한다. 환풍기는 이산화탄소 농도, 미세먼지 농도를 측정하여 내부 공기를 환기 시킨다. 블라인드는 일조량을 측정하여 상황에 맞게 창문의 햇빛을 가린다. 공통적으로 모든 객체에는 모터의 고장주기를 예측하는 시스템이 있어 모터의 정보를 받아 고장 주기를 알 수 있다.**  **모두 모터를 사용하더라도 각각의 객체는 공조 방식이 다르고 기능도 다르며, 역할도 다르다.**  **에어컨은 응축기, 증발기, 측정 센서를 가지고 있고 시스템 적으로 냉/난방, 제습 기능이 있다.**  **에어컨은 상황에 따라 냉방이나 난방을 하며 습도가 높을 경우 제습기능을 이용한다. 또, 내부의 모터가 고장이 생기기전 고장 주기를 예측해 사용자에게 알려준다.**  **모든 객체는 모터를 사용한다 하더라도 불량률, 내구도가 달라 고장 주기가 다르고 모터의 사용 방식이 다르다.**  **에어컨은 시스템의 상태를 모니터링할 수 있는 장치 객체와 연관을 가지고 있다.** | **추상화**  **상속**  **객체를 구분 짓는 정체성**  **에어컨 속성**  **캡슐화 과정**  **다형성**  **정보 제공 장치와 에어컨 객체의 연관성** | |

|  |  |  |
| --- | --- | --- |
| **3장 토의 보고서** | | |
| **토의 사항** | **3장. 유스케이스 다이어그램 작성**  **3-1. 프로세스 정의**  **3-2. 유스케이스 다이어그램 모델링**   1. **문제 기술서** 2. **유스케이스 다이어그램 모델링** 3. **유스케이스 명세서** | |
| **진행일자** | **2020년 9월 22일 14:50~15:20, 2020년 9월 23일 14:00~15:40** | |
| **참석자** | **이영채(팀장), 김동혁(서기), 김영민, 김재현,** | |
| **3.1 프로세스 정의** | | |
|  | | |
| **3.2 유스케이스 다이어그램 모델링** | | |
| **개인의견** | **김영민** | * **센서데이터를 기반으로 한 고장 주기시스템을 구현하자.** * **센서도 액터로 생각할 수 있다고 생각함.** |
| **이영채** | * **앱이나 웹 등을 관제시스템의 수신객체로 두자.** * **여러 개의 모터를 사용한 시스템을 구현하여 어디에나 사용될 수 있다는 점을 살리자.** |
| **김재현** | * **고장주기 예측시스템 포함한 중앙제어시스템을 구현하자** * **공장 설비등을 일반화하여 액터로 설정하자.** |
| **김동혁** | * **그럼 해당시스템을 활용할 수 있는 생산라인이나 공조시스템을 유스케이스로 작성하자.** * **공장에서 모터를 기반으로 사용되는 설비들을 액터로 설정하자.** |
| **토의 내용** | **주제 재설정 : 공장 모터기반 설비 고장예측 시스템.**  **1단계 요구사항 정의.**   * **사용자는 공장의 설비에 이상유무를 확인할 수 있다.** * **컨베이어 밸트는 다수의 모터로 물건을 일정한 속도로 운반시킨다.** * **환풍기는 공기질에 따라 실시간으로 환기를 시킨다.** * **로봇팔은 컨베이어 밸트와 연계하여 부품을 옮긴다** * **크레인은 모터를 이용하여 무거운 공장내 물건을 운반한다** * **사용자는 고장 예측 시스템이 측정한 데이터를 확인할 수 있다** * **고장 예측 시스템의 딥러닝을 사용하여 모터의 고장주기를 예측하는 시스템이다.** * **고장주기 예측시스템은 측정된 데이터를 활용하여 학습 데이터를 수집한다.** * **항상 모터를 실시간으로 확인한다.** * **사용자는 예측 정비 일정 알림을 받는다.** * **시스템을 사용하는 사용자와 유지보수 하는 관리자** * **관리자는 시스템의 유지 보수를 맡는다.**     **2단계 액터정의.**   * **사용자** * **관리자** * **공장설비(컨베이어 벨트, 환풍기, 로봇 팔, 크레인)** * **기계학습 시스템** * **센서**   **3단계 유스케이스 정의**   |  |  | | --- | --- | | **사용자** | **기능(유스케이스)** | | **공장설비** | **설비기능, 물건운반, 상품조립, 공장환기** | | **사용자** | **예측 정비 일정 알림, 결함보고, 정비지시, 데이터 시각화** | | **관리자** | **시스템 유지보수, 모터관리** | | **기계학습시스템** | **딥러닝, 고장예측** | | **센서** | **학습데이터 수집** |   **4단계 유스케이스 다이어그램**    **[변경된 유스케이스 다이어그램]**   |  |  | | --- | --- | | **김영민** | * **수집데이터는 데이터 베이스에 저장되어야 할 것 같다..** | | **이영채** | * **센서와 데이터 베이스 사이의 관계가 있어야 할 것 같다.** * **데이터 베이스와 딥러닝 관계가 있어야 할 것 같다.** | | **김재현** | * **데이터 시각화는 확장관계가 아닌 포함 관계가 되어야 할 것 같다.** | | **김동혁** | * **공장설비 액터와 유스케이스는 개발범위를 벗어난 것 같아 삭제하는게 좋을 것 같다.** * **데이터베이스도 액터로 지정되어야 한다.** |     **5단계 유스케이스 명세서**  **김영민 :**   |  |  | | --- | --- | | **유스케이스명** | **고장 예측 주기** | | **액터명** | **사용자, 공장 설비, 기계학습 시스템** | | **유스케이스 개요** | **공장 설비에 대한 고장 데이터를 딥러닝 한 결과를 활용한 후, 고장 주기를 예측하여 예측 정비 일정을 사용자에게 알린다.** | | **사전조건** | **딥러닝이 선행되어야 한다.** | | **이벤트 흐름** | **<정상 흐름>**   1. **공장 설비의 고장 주기 데이터를 수집한다.** 2. **수집한 데이터를 딥러닝 하여 예측되는 고장 주기를 구한다.** 3. **딥러닝에서 구한 결과를 활용하여 예측 정비 일정을 구한다.** 4. **사용자에게 예측 정비 일정 알람을 보낸다.** |   **이영채 :**   |  |  | | --- | --- | | **유스케이스명** | **학습데이터 수집** | | **액터명** | **센서** | | **유스케이스 개요** | **센서로부터 데이터를 받아 수집하는 유스케이스이다.** | | **사전조건** | **센서가 존재해야 한다.** | | **이벤트 흐름** | **<정상 흐름>**   1. **센서의 전원이 켜져 있는지 확인한다. (센서)** 2. **센서에게 데이터를 요청한다. (센서)** 3. **센서에게 받은 데이터를 해당 모터의 데이터베이스에 입력한다. (시스템)**   **<선택 흐름>**  **▶ 센서의 전원이 켜져 있지 않으면 오류메시지를 보인다.**  **▶ 센서의 데이터를 받지 못할 경우 오류메시지를 보인다.**  **▶ 데이터베이스에 입력이 되지않으면 오류메시지를 보인다** |   **김동혁 :**   |  |  | | --- | --- | | **유스케이스명** | **예측정비 일정 알림** | | **액터명** | **사용자** | | **유스케이스 개요** | **고장예측을 기반으로 사용자에게 예측정비시기를 알려주는 유스케이스이다.** | | **사전조건** | **학습데이터를 기반으로 한 고장예측이 선행되어야 한다.** | | **이벤트 흐름** | **<정상흐름>**   1. **사용자가 예측정비시스템을 확인한다.** 2. **장비의 센서데이터를 기반으로 딥러닝이 진행된다.** 3. **딥러닝이 진행됨에따라 장비의 고장예측주기를 나타낸다.** 4. **고장예측주기를 기반으로 예측정비주기를 사용자에게 알려줌**   **<선택흐름>**   * **고장이 발생했을경우 결함보고를 통해 결함부위를 구체적으로 알려준다.** * **예측정비일정을 통해 캘린더등을 통해 정비지시를 사용자에게 요청한다** * **데이터 시각화가 필요한 경우 그래프, 차트등으로 데이터를 시각화하여 사용자에게 알려준다.** |   **김재현 :**   |  |  | | --- | --- | | **유스케이스명** | **설비 기능** | | **액터명** | **공장 설비 시스템** | | **유스케이스 개요** | **공장 설비 시스템이 고장 발생 전 정상의 상태를 보여주는 유스케이스이다.** | | **사전조건** | **설비의 기능들은 고장 발생 전 정상의 상태이다.** | | **이벤트 흐름** | **<정상 흐름>**   1. **공장 설비들이 정상적으로 작동하고 있는지 검사한다(시스템).** 2. **내장된 고장 예측 프로그램이 실시간으로 설비 가동 데이터를 센서로 수집한다(액터).** 3. **수집한 데이터를 데이터 베이스에 전송한다(액터).**   **<선택 흐름>**   * **이상이 발생하면 데이터를 수집하고 정비를 알린다.** * **각각의 모터마다 데이터를 다르게 저장한다.** | | |

|  |  |
| --- | --- |
| 4**장 토의 보고서** | |
| **토의 사항** | **4장. 클래스 다이어그램 작성**   * 1. **프로세스 정의**      1. **문제기술**      2. **클래스 추출**      3. **메서드 추출**      4. **클래스간 관계 및 구조화**   2. **클래스 다이어그램 모델링** |
| **진행일자** | **2020년 9월 30일 21:00~23:00** |
| **참석자** | **김영민(서기), 김재현, 김동혁, 이영채(팀장)** |
| **4.1 프로세스 정의** | |
|  | |
| 4.1.1 문제 기술 | |
| **토의 내용** | **토의를 통해 예측 정비 일정 알림 클래스 다이어그램을 작성하기로 결정**  **예측 정비 일정 알림 명세서**   |  |  | | --- | --- | | **유스케이스명** | **예측정비 일정 알림** | | **액터명** | **사용자** | | **유스케이스 개요** | **고장예측을 기반으로 사용자에게 예측정비시기를 알려주는 유스케이스이다.** | | **사전조건** | **학습데이터를 기반으로 한 고장예측이 선행되어야 한다.** | | **이벤트 흐름** | **<정상흐름>**   1. **사용자가 예측정비시스템을 확인한다.** 2. **고장 예측 값을 받아 온다.** 3. **고장예측주기를 기반으로 예측정비주기를 사용자에게 알려준다.**   **<선택흐름>**   * **고장이 발생했을 경우 결함보고를 통해 결함부위를 구체적으로 알려준다.** * **예측정비일정을 통해 캘린더 등을 통해 정비지시를 사용자에게 요청한다** * **데이터 시각화가 필요한 경우 그래프, 차트 등으로 데이터를 시각화 하여 사용자에게 알려준다.** | |
| 4.1.2 클래스 추출 | |
| **토의 내용** | **▶ 출력 시스템(추상 클래스), 사용자, 데이터 시각화 시스템, 알림 시스템**  **(수정)**   * **예측정비 모듈, 사용자, 데이터 시각화모듈 ,알림모듈** |
| 4.1.3 메서드 추출 | |
| **토의 내용** | **▶ 예측정비 모듈 : 데이터 요청**  **▶ 사용자: 결과 요청**  **▶ 데이터 시각화 모듈: 결과 값 출력, 그래프 구현, 캘린더 구현**  **▶ 알림 모듈: 결과 값 출력, 정비 알림** |
| 4.1.4 클래스간 관계 및 구조화 | |
| **토의 내용** | **[출력 시스템]**   |  | | --- | | **예측정비 모듈** | | **예측 고장 주기 데이터: String** | | **예측 데이터 요청 : void**  **예측 데이터 전송 : void** |   **[사용자]**   |  | | --- | | **사용자** | | ID : String | |  |   **[데이터 시각화 모듈]**   |  | | --- | | **데이터 시각화 모듈** | | **그래프: String**  **캘린더: String** | | **결과 값 출력: void**  **그래프 구현: void**  **캘린더 구현: void** |   **[관계 및 구조화]**    **[수정]** |
| **4.2 클래스 다이어그램 모델링** | |
| **토의 내용** | **[수정]** |

|  |  |  |
| --- | --- | --- |
| **5장 토의 보고서** | | |
| **토의 사항** | **5장. 순차 다이어그램**   * 1. **프로세스 정의**   **5.1.1 문제 기술**  **5.1.2 액터, 클래스 선정**  **5.1.3 액터/객체간 메시지 정의**  **5.1.4 객체 추가**  **5.2. 순차 다이어그램 모델링** | |
| **진행일자** | **2020년 10월 7일 21:20~23:20** | |
| **참석자** | **김영민, 김재현, 김동혁(팀장), 이영채(서기)** | |
| **5.1 프로세스 정의** | | |
|  | | |
| 5.1.1 문제 기술 | | |
| [유스케이스 명세서]   |  |  | | --- | --- | | **유스케이스명** | **예측정비 일정 알림** | | **액터명** | **사용자** | | **유스케이스 개요** | **고장예측을 기반으로 사용자에게 예측정비시기를 알려주는 유스케이스이다.** | | **사전조건** | **학습데이터를 기반으로 한 고장예측이 선행되어야 한다.** | | **이벤트 흐름** | **<정상흐름>**   1. **사용자가 예측정비시스템을 확인한다.** 2. **고장 예측 값을 받아 온다.** 3. **고장예측주기를 기반으로 예측정비주기를 사용자에게 알려준다.**   **<선택흐름>**   * **고장이 발생했을 경우 결함보고를 통해 결함부위를 구체적으로 알려준다.** * **예측정비일정을 통해 캘린더 등을 통해 정비지시를 사용자에게 요청한다** * **데이터 시각화가 필요한 경우 그래프, 차트 등으로 데이터를 시각화 하여 사용자에게 알려준다.** | | | |
| 5.1.2 액터,클래스 선정 | | |
| **개인 의견** | **김동혁** | **액터 - 사용자.**  **클래스및 객체 - 예측주기출력 모듈(출력시스템), 시각화 모듈, 알림 모듈, 고장예측 모듈** |
| **김영민** | **사용자, 예측 정비 모듈, 알림 모듈, 데이터 시각화 모듈, 고장 예측 시스템** |
| **김재현** | **액터: 사용자**  **객체: 데이터 처리 시스템, 고장 주기 예측 시스템, 알림 시스템** |
| **이영채** | **액터 - 사용자,센서**  **클래스 - 예측정비일정알림 고장예측, 딥러닝** |
| **토의 내용** | **김동혁** | **액터는 사용자가 맞다. 객체로 모듈을 사용하는 것도 괜찮은 것 같다.** |
| **김영민** | **하나의 유스케이스로 분리하여 순차 다이어그램을 작성해야 하므로 시스템으로 명명한 몇 가지 객체들을 모듈로 바꾸는 것이 나아 보인다.** |
| **김재현** | **액터와 객체에 관한 큰 의견 차이는 없고 이벤트 흐름만 잘 정하면 될 것같다.** |
| **이영채** | **액터는 사용자로 통일하고, 객체는 예측정비, 데이터 시각화, 고장예측, 알림으로 통일하는 것이 좋을 것 같다.** |
| **결론** | **액터: 사용자**  **객체: 고장 예측 모듈, 알림 모듈, 시각화 모듈, 예측 주기 출력 모듈** | |
| 5.1.3 액터/객체간 메시지 정의 | | |
| **개인 의견** | **김동혁** | **1. 사용자가 예측주기 확인 실행**  **2. 고장예측주기 데이터 요청**  **3. 예측주기 데이터 반환**  **4. 데이터 시각화 요청**  **5. 시각화 데이터 반환**  **6. 알림 전송요청**  **7. 알림 전송** |
| **김영민** | **1) 사용자는 예측 정비 모듈에 확인하고 싶은 정보를 요구한다.**  **2) 예측 정비 모듈은 고장 예측 시스템에서 해당 데이터를 요구한다.**  **3) 고장 예측 시스템은 해당 데이터를 계산하여 산출한다.**  **4) 고장 예측 시스템은 데이터를 예측 정비 시스템에 넘긴다.**  **5) 예측 정비 시스템은 데이터 시각화 모듈에 데이터를 전달한다.**  **6) 데이터 시각화 모듈은 시각화 된 데이터를 산출한다.**  **7) 데이터 시각화 모듈은 산출된 데이터를 예측 정비 시스템에 넘긴다.**  **8) 예측 정비 시스템은 요구를 받은 데이터 수치와 시각화 된 데이터를 사용자에게 전달한다.** |
| **김재현** | **1. 사용자가 예측 시스템 실행**  **2. 데이터 처리 시스템이 고장 주기 예측 시스템에 데이터 요청**  **3. 예측 주기 날짜 알림 시스템에 전달**  **4. 예측 데이터 데이터 처리 시스템 전달**  **5. 데이터를 그래프로 시각화 하여 전달**  **6. 날짜에 대해 사용자에게 알림 발송** |
| **이영채** | **1. 사용자가 예측 정비 데이터 요구**  **2. 예측 정비 일정 알림 서비스에서 고장 예측 서비스에 예측 데이터 요구**  **3. 고장 예측 서비스는 딥러닝 서비스에 고장 예측 데이터 계산 요구**  **4. 딥러닝 서비스는 센서에게 현재 상태 데이터 요구**  **5. 센서는 딥러닝 서비스에 상태 데이터 전송**  **6. 딥러닝 서비스는 센서로 부터 받은 데이터를 딥러닝한 값에 대입 후 고장 예측 서비스에 데이터 반환**  **7. 고장 예측 서비스는 예측 데이터를 예츨 정비 시스템에 전송**  **8. 데이터를 시각화하여 사용자에게 표시** |
| **토의 내용** | **김동혁** | **김영민의 모듈 내부에서 메시지 흐름을 표현하는 것이 전달력이 더 좋은 것 같다.** |
| **김영민** | **알림 모듈을 추가하면서 흐름에 알림 전송 요청과 알림 전송이 김동혁의 메시지에 추가되는 것이 좋겠다는 의견이다.** |
| **김재현** | **김동혁의 것의 메시지에 데이터시각화 결과를 사용자에게 전달해주는 구간이 없다.** |
| **이영채** | **팀원들의 전체적인 메시지의 흐름이 본인의 흐름과 다른 부분이 있는데, 불필요한 메시지가 있는 것 같아 빼는 것이 좋을 것 같다.** |
| **결론** | 1. **사용자가 예측 주기 확인 실행** 2. **고장 주기 예측 데이터 요청** 3. **예측 주기 데이터 반환** 4. **데이터 시각화 요청** 5. **시각화 데이터 반환** 6. **시각화 데이터 출력** 7. **알림 전송 요청** 8. **알림 전송** | |
| 5.1.4 객체 추가 | | |
| **개인 의견** | **김동혁** | **객체 추가보다 선택흐름에 대한 주해사물이 들어가야할 것 같다.** |
| **김영민** | **이전에 생각한 모델에서 데이터 시각화 모듈을 추가하였다.** |
| **김재현** | **데이터의 흐름에 대해 정확히 제시하고 예외 흐름에 대한 메모를 했다.** |
| **이영채** | **딥러닝으로부터 데이터를 요구하는 것을 추가했다.** |
| **결론** | 1. **선택 흐름에 대한 주해사물 메모로 추가하였다.** 2. **몇몇의 팀원의 다이어그램에 없던 데이터 시각화 모듈을 추가했다.** | |
| **5.2 모델링** | | |
| **개인 의견** | **김동혁** |  |
| **김영민** |  |
| **김재현** |  |
| **이영채** |  |
| **토의 내용** | **김동혁** | **김재현의 다이어그램에 있는 것처럼 시각화 데이터를 전달하는 것을 사용하는 것이 좋을 것 같다.** |
| **김영민** | **김동혁의 다이어그램처럼 주해사물로 나타내는 것이 바람직한 것 같다.** |
| **김재현** | **팀원들처럼 시각화하는 모듈과 예측한 데이터를 출력하는 모듈로 분할하는 것도 좋을 것 같다.** |
| **이영채** | **김동혁의 다이어그램에 시각화 데이터를 추가하여 수정하는 것이 좋을 것 같다.** |
| **결론** | **김동혁 팀원의 다이어그램에 데이터 시각화 모듈을 추가하고 시각화 데이터를 전달하는 과정을 삽입한다.** | |

|  |  |  |
| --- | --- | --- |
| **6장 토의 보고서** | | |
| **토의 사항** | **6장. 통신 다이어그램**   * 1. **프로세스 정의**   **6.1.1 문제 기술**  **6.1.2 객체 선정**  **6.1.3 객체 다이어그램 위치 및 링크 정의**  **6.1.4 객체간 메시지 정의**  **6.2. 순차 다이어그램 모델링** | |
| **진행일자** | **2020년 10월 13일 14:20~15:20** | |
| **참석자** | **김영민(팀장), 김재현, 김동혁(서기), 이영채** | |
| **6.1 프로세스 정의** | | |
|  | | |
| 6.1.1 문제 기술 | | |
| **[유스케이스 명세서]**   |  |  | | --- | --- | | **유스케이스명** | **예측정비 일정 알림** | | **액터명** | **사용자** | | **유스케이스 개요** | **고장예측을 기반으로 사용자에게 예측정비시기를 알려주는 유스케이스이다.** | | **사전조건** | **학습데이터를 기반으로 한 고장예측이 선행되어야 한다.** | | **이벤트 흐름** | **<정상흐름>**   1. **사용자가 예측정비시스템을 확인한다.** 2. **고장 예측 값을 받아 온다.** 3. **고장예측주기를 기반으로 예측정비주기를 사용자에게 알려준다.**   **<선택흐름>**   * **고장이 발생했을 경우 결함보고를 통해 결함부위를 구체적으로 알려준다.** * **예측정비일정을 통해 캘린더 등을 통해 정비지시를 사용자에게 요청한다** * **데이터 시각화가 필요한 경우 그래프, 차트 등으로 데이터를 시각화 하여 사용자에게 알려준다.** | | | |
| 6.1.2 객체 선정 | | |
| **개인 의견** | **김동혁** | **액터는 사용자, 객체는 고장예측 모듈, 시각화 모듈, 알림 모듈, 예측주기출력 모듈** |
| **김영민** | **사용자, 예측 정비 알림 모듈, 고장 예측 모듈, 데이터 시각화 모듈** |
| **김재현** | **액터- 사용자, 예측 정비 모듈, 알림 모듈, 고장 주기 예측 모듈** |
| **이영채** | **사용자, 예측 정비 서비스, 데이터 시각화, 고장 예측 서비스** |
| **토의 내용** | **김동혁** | **예측주기 출력모델을 중심으로 다른객체들을 선정함.** |
| **김영민** | **예측정비알림모듈을 중심으로 객체들을 연결할 필요가 있다고 생각함** |
| **김재현** | **예측정비 모듈안에 데이터 시각화기능을 넣어 굳이 따로 정의하지 않음** |
| **이영채** | **명세서에 나와있는 객체를 선정함.** |
| **결론** | **객체: 사용자, 예측 주기 출력 모듈, 고장 예측 모듈, 시각화 모듈, 알림 모듈** | |
| 6.1.3 객체를 다이어그램에 위치 및 링크정의 | | |
| **개인 의견** | **김동혁** |  |
| **김영민** |  |
| **김재현** |  |
| **이영채** |  |
| **토의 내용** | **김동혁** | **예측주기 출력모델이 흐름을 정하는 중요객체로 보고 활성객체로 선정** |
| **김영민** | **가장 상호작용이 많은 예측정비 모듈을 활성객체로 두었음** |
| **김재현** | **김영민 팀원과 다르게 하나가 아닌 세 개의 객체를 활성객체로 둠** |
| **이영채** | **예측정비 서비스를 활성객체로 놓아야 겠다는 생각을 함.** |
| **결론** | **김동혁 팀원의 다이어그램에서 사용자를 액터로 표기한 것을 객체로 고치고 김영민 팀원의 다이어그램에서 알림 모듈을 추가해 두 명의 다이어그램을 섞어 최종 결론을 도출했습니다.** | |
| 6.1.4 객체간 메시지 정의하기 | | |
| **개인 의견** | **김동혁** | 1. **고장주기 예측 요청** 2. **고장 주기 예측데이터 요청** 3. **예측데이터 반환** 4. **데이터 시각화 요청** 5. **시각화 데이터 반환** 6. **사용자에게 시각화 데이터 출력** 7. **알림일정 예약** 8. **사용자에게 알림** |
| **김영민** | **1. 사용자가 예측 주기 확인 요청**  **2. 고장예측 주기 데이터 요청**  **3. 예측 주기 데이터 출력**  **4. 데이터 시각화 요청**  **5. 시각화 데이터 변환 출력**  **6. 시각화 데이터 출력 후 제공** |
| **김재현** | 1. **사용자가 고장주기를 요청한다.** 2. **고장 주기 예측 모듈에 데이터를 요청한다.** 3. **데이터를 예측 정비 모듈에 반환해준다.** 4. **알림 모듈은 예측 정비 모듈 고장 예측 날짜를 전달해준다.** 5. **사용자에게 고장 예측 날짜를 알림으로 전한다.** 6. **예측 정비 모듈에서 사용자에게 데이터를 시각화해 보여준다.** |
| **이영채** | **1. 예측 요구**  **2. 고장 예측 데이터 요구**  **3. 고장 예측 데이터 반환**  **4. 예측 데이터 전달**  **5. 예측 데이터 시각화** |
| **토의 내용** | **김동혁** | **최대한 활성객체를 중심으로 메시지 흐름을 잘 보일 수 있도록 선정해야한다고 생각 함.** |
| **김영민** | **이전에 작성한 명세서와 순차다이어그램을 참조하여 메시지를 정의함**  **또한 대체적으로 유사한 메시지들을 공통적으로 선정하는 것이 좋다고 생각함.** |
| **김재현** | **전체적으로 팀원들의 의견이 유사하다고 생각함.** |
| **이영채** | **객체와 링크간에 적절한 메시지를 정의함** |
| **결론** | 1. **사용자가 예측 주기 확인 실행** 2. **고장 예측 주기 데이터 요청** 3. **예측 주기 데이터 변환** 4. **데이터 시각화 요청** 5. **시각화 데이터 변환** 6. **시각화 데이터 출력** 7. **알림 전송 요청** 8. **알림 전송** | |
| 6.2.1 순차다이어그램 변환 하여 통신 다이어그램 작성 | | |
| **개인 의견** | **김동혁** |  |
| **김영민** |  |
| **김재현** |  |
| **이영채** |  |
| **결론** | **김영민 팀원의 다이어그램이 수정사항 없이 적합하여 김영민 팀원의 다이어그램을 결과물로 선정했습니다.** | |
| 6.2.2 명세서기반 통신 다이어그램 작성 | | |
| **김동혁** |  | |
| **김영민** |  | |
| **김재현** |  | |
| **이영채** |  | |
| **김동혁** | **김재현 팀원의 말처럼 굳이 시각화기능을 따로 시각화 모듈로 빼지 않았어도 좋았을 것 같다.** | |
| **김영민** | **이영채 김재현의 다이어그램에 있는 알림모듈이 적합한 것 같아 최종결과에 추가하는 것이 좋다고 생각한다.** | |
| **김재현** | **김동혁 팀원의 예측주기 출력모델로 한번에 연관시켜 처리하는게 괜찮아 보이지만 시각화 모듈을 굳이 지정했어야하는 생각이든다.** | |
| **이영채** | **데이터 시각화에서 사용자에게 데이터를 보내주는 것이 아닌, 예측정비 서비스를 활성객체로 하여 시각화 한 데이터를 예측정비서비스를 거쳐 사용자에게 보내주는 것이 좋다고 생각함.** | |
| **결론** | **김동혁 팀원과 김영민 팀원의 다이어그램을 섞어 사용자를 액터가 아닌 객체로 만들고 알림 모듈을 추가하고 그에 대한 메시지와 링크를 추가 했습니다.** | |

|  |  |
| --- | --- |
| **중간 발표 이후 토의 보고서** | |
| **토의 사항** | 1. **중간 발표 참관 후 팀의 회의 활동에 대한 피드백 및 진행 방향** 2. **개선 사항 및 결과** |
| **진행일자** | **2020년 10월 25일 22:40~24:30** |
| **참석자** | **김영민(팀장), 김재현(서기), 김동혁, 이영채** |
| **1. 중간 발표 참관 후 팀의 회의 활동에 대한 피드백 및 진행 방향** | |
| **토의 내용** | **[2조]**  **▶ 아이디어가 좋음**  **▶ 각자의 아이디어가 잘 작성**  **▶ 내용이 잘 정리**  **▶ 각 챕터마다 토론에 대한 최종 결론 명확**  **▶ 주제에 대한 아이디어 좋음**  **▶ 객체 선정에 의견이 다양**  **▶ 각각의 프로세스의 내용이 자세히 나옴**  **▶ 형식이 깔끔**  **▶ 주제 선정 시에 여러 번 고민 흔적**  **▶ 프로세스 잘 정리**  **[3조]**  **▶ 보고서 양식 구체적으로 명시**  **▶ 회의 내용 상세히 작성**  **▶ 개인 의견 표시 잘됨**  **▶ FeedBack이 있음 좋겠음**  **▶ 구현 가능성 높음**  **▶ 토의 과정 자세히 작성**  **▶ 결론 과정에서 각자의 의견에 점수 부여 참신**  **▶ 개인의 의견 상세하게 정리**  **▶ 각각의 선택, 최종 결론에 대한 이유가 명료**  **▶ 대체적으로 프로세스 정리**  **[모든 조 공통 문제]**  **▶ 최종 결론이 없음**  **▶ 프로세스 정의가 없음**  **▶ 메신저 툴에 대한 설명이 없음** |
| **2. 개선 사항 및 결과** | |
| **토의 내용** | **▶ 프로세스 정의 명확하게 추가**   * **프로세스를 정의하는 챕터부터 각 챕터의 결과물을 도출하기위한 프로세스를 정의 하였음**     **▶ 최종 결론 명확하게 추가**   * **각 챕터의 토의 내용에 대한 결론을 추가하고 그 결론을 사용하여 다음 프로세스를 진행했습니다.**   **▶ 토의에 사용하는 메신저 툴 정보 추가**   * **챕터 1 보고서의 1.2 부분에 메신저 툴의 정보를 추가함**   **▶ 보고서 폰트 양식 정형화**   * **챕터 1 보고서의 1.1 보고서 양식 부분에 [추가] 탭을 이용하여 명시**   **▶ 토의 결과 내용에 대한 개인 의견 중요도 표시**   * **이후에 시행하는 토의 및 활동에서 결과를 도출하는데 비중을 많이 차지한 의견을 노란색 형광펜 효과를 추가하여 표시할 것** * **챕터 1 보고서의 1.1 보고서 폰트 양식 정형화시 포함하여 작성함** |

|  |  |  |
| --- | --- | --- |
| **7장 토의 보고서** | | |
| **토의 사항** | **7장. 활동 다이어그램**   * 1. **프로세스 정의**   **7.1.1 작성 대상 선정**  **7.1.2 활동 상태 추출**  **7.1.3 모델링**  **7.1.4 전체 모델링 조합**   * 1. **활동 다이어그램 작성** | |
| **진행일자** | **2020년 11월 4일 14:20~15:20** | |
| **참석자** | **김영민(팀장), 김재현, 김동혁(서기), 이영채** | |
| **7.1 프로세스 정의** | | |
|  | | |
| 7.1.1 작성 대상 선정 | | |
| **결론** | **[유스케이스 명세서]**   |  |  | | --- | --- | | **유스케이스명** | **예측정비 일정 알림** | | **액터명** | **사용자** | | **유스케이스 개요** | **고장예측을 기반으로 사용자에게 예측정비시기를 알려주는 유스케이스 이다.** | | **사전조건** | **학습데이터를 기반으로 한 고장예측이 선행되어야 한다.** | | **이벤트 흐름** | **<정상흐름>**   1. **사용자가 예측정비시스템을 확인한다.** 2. **고장 예측 값을 받아 온다.** 3. **고장예측주기를 기반으로 예측정비주기를 사용자에게 알려준다.**   **<선택흐름>**   * **고장이 발생했을 경우 결함보고를 통해 결함부위를 구체적으로 알려준다.** * **예측정비일정을 통해 캘린더 등을 통해 정비지시를 사용자에게 요청한다** * **데이터 시각화가 필요한 경우 그래프, 차트 등으로 데이터를 시각화 하여 사용자에게 알려준다.** | | |
| 7.1.2 활동 상태 추출 | | |
| **결론** | **명세서에서 활동 상태를 추출하였다.**   1. **사용자가 예측정비시스템을 확인한다.** 2. **고장 예측 값을 받아 온다.** 3. **고장예측주기를 기반으로 예측정비주기를 사용자에게 알려준다.** 4. **고장이 발생했을 경우 결함보고를 통해 결함부위를 구체적으로 알려준다.** 5. **예측정비일정을 통해 캘린더 등을 통해 정비지시를 사용자에게 요청한다** 6. **데이터 시각화가 필요한 경우 그래프, 차트 등으로 데이터를 시각화 하여 사 용자에게 알려준다.** | |
| 7.1.3 모델링 | | |
| 사전 조건 모델링 | | |
| **개인 의견** | **김동혁** |  |
| **김영민** |  |
| **김재현** |  |
| **이영채** |  |
| **토의 내용** | **김동혁** | **전이 지점을 사용하여 데이터 유무를 파악하는 부분을 추가해야 될 것 같다.**  **시전 조건에 대한 이해가 더 필요할 것 같다.** |
| **김영민** | **데이터가 없을 때 요청 이후 프로그램이 종료 될 것인지 여부를 생각해야 할 것 같다.** |
| **김재현** | **전이 부분에 데이터 보유 여부를 확인할 것인지, 응답 확인 할 것인지 선택 해야 할 것 같다.** |
| **이영채** | **전이와 대한 분기에 대한 차이점에 대해서 좀더 확실히 알게 되었다. 다시 다이어그램을 작성 할 때 참고하겠다.** |
| **결론** | **토의 결과 이영채 팀원의 다이어그램을 중심으로 좀 더 수정 및 추가를 하였고, 전이 부분에 대한 정의를 확실히 하였다.** | |
| 유스케이스 종료와 사후 조건 확인 | | |
| **개인 의견** | **김동혁** |  |
| **김영민** |  |
| **김재현** |  |
| **이영채** |  |
| **토의 내용** | **김동혁** | **분기에 따라서 나뉜 액티비티들이 동기화를 통해 일률적으로 종료 되야 한다고 생각한다.** |
| **김영민** | **다른 팀원들의 의견처럼 분기 이후 동기화를 통해 모듈의 기능에 대한 종료 시점 통일을 추가해야 할 것 같다.** |
| **김재현** | **데이터 시각화 부분과 사용자 알림 부분을 병렬 처리하여 동시에 출력하는 것이 좋다고 생각한다.** |
| **이영채** | **정비 일정과 고장 발생 여부를 동기화로 한번에 데이터 시각화 하는 것이 좋을 것 같다.** |
| **결론** | **동기화를 활용하여 분기로 나뉜 액티비티들을 동시에 처리하기로 한다.** | |
| 선택 흐름 조건 작성 | | |
| **개인 의견** | **김동혁** |  |
| **김영민** |  |
| **김재현** |  |
| **이영채** |  |
| **토의 내용** | **김동혁** | **데이터 시각화도 선택 흐름에 들어가 있는 부분이기 때문에 김영민 팀원의 의견도 추가하는 것이 좋다고 생각한다.** |
| **김영민** | **예측 정비 일정 알림 유무와 정비 지시에 대한 선택 흐름을 간과한 것 같다. 추가하는 것이 좋다고 생각한다.** |
| **김재현** | **결함이 생겼을 때 결함 지시 분기를 추가하는 것이 좋다고 생각한다.** |
| **이영채** | **팀원의 의견을 종합하여 선택흐름을 추가로 작성하는 것이 좋다고 생각한다.** |
| **결론** | **토의 결과 예측 정비 일정 알림, 결함 보고, 데이터 시각화 여부를 최종적으로 추가하기로 결정 하였다.** | |
| 동기화 조건 작성 | | |
| **개인 의견**  **및 토의 내용** | **김동혁** | **이 다이어그램에서는 사후조건과 동기화 조건이 동일하다고 생각한다.** |
| **김영민** | **해당 팀 프로젝트 모듈에는 동기화 조건에 해당사항이 없다고 생각하였으나, 토의 결과 팀원들의 의견이 이해가 되었고 동의한다.** |
| **김재현** | **사후 조건과 동기화 조건이 동일하다고 생각한다.** |
| **이영채** | **팀원의 내용에 동의한다.** |
| **결론** | **동기화 조건은 이전에 작성하였던 사후 조건과 일치시키도록 한다.** | |
| **7.2 활동 다이어그램 작성** | | |
| **결론** | * **동기화를 활용하여 분기로 나뉜 액티비티들을 동시에 처리** * **토의 결과 예측 정비 일정 알림, 결함 보고, 데이터 시각화 여부를 최종적으로 추가** | |

|  |  |  |
| --- | --- | --- |
| **8장 토의 보고서** | | |
| **토의 사항** | **8장. 상태 다이어그램**   * 1. **프로세스 정의**   **8.1.1 작성 대상 선정**  **8.1.2 상태 정의**  **8.1.3 상태 나열/전의 정의**  **8.1.4 상세 부분 정의**  **8.1.5 전체 모델링 조합**  **8.2 활동 다이어그램 작성** | |
| **진행일자** | **2020년 11월 12일 13:20~14:50** | |
| **참석자** | **김영민, 김재현, 김동혁(팀장), 이영채(서기)** | |
| **8.1 프로세스 정의** | | |
|  | | |
| 8.1.1 작성 대상 선정 | | |
| **결론** | **[유스케이스 명세서]**   |  |  | | --- | --- | | **유스케이스명** | **예측정비 일정 알림** | | **액터명** | **사용자** | | **유스케이스 개요** | **고장예측을 기반으로 사용자에게 예측정비시기를 알려주는 유스케이스 이다.** | | **사전조건** | **학습데이터를 기반으로 한 고장예측이 선행되어야 한다.** | | **이벤트 흐름** | **<정상흐름>**   1. **사용자가 예측정비시스템을 확인한다.** 2. **고장 예측 값을 받아 온다.** 3. **고장예측주기를 기반으로 예측정비주기를 사용자에게 알려준다.**   **<선택흐름>**   * **고장이 발생했을 경우 결함보고를 통해 결함부위를 구체적으로 알려준다.** * **예측정비일정을 통해 캘린더 등을 통해 정비지시를 사용자에게 요청한다** * **데이터 시각화가 필요한 경우 그래프, 차트 등으로 데이터를 시각화 하여 사용자에게 알려준다.** | | |
| 8.1.2 상태 정의 | | |
| **개인 의견** | **김동혁** | **데이터 요청, 데이터 보유, 데이터시각화, 고장 주기 알림, 대기, 종료** |
| **김영민** | **데이터 요청, 데이터 확인, 오류 확인 ,데이터 시각화, 고장 주기 알림, 종료** |
| **김재현** | **데이터 요청, 장비고장, 사용자 요청 여부, 데이터 시각화 알림** |
| **이영채** | **데이터 요청, 데이터 보유 확인, 고장 주기 알림, 데이터 시각화** |
| **토의 내용** | **김동혁** | **모두가 가지고 있는 데이터 요청이 가장 처음에 나와야 할 것 같다.** |
| **김영민** | **오류 확인이 추가하는 것이 좋을 것 같다.** |
| **김재현** | **선택 흐름에 장비가 고장 났을 때 결함보고도 해주어야 한다.** |
| **이영채** | **대기가 있어야하는 이유가 무엇인지 다시 생각해보아야겠다.** |
| **결론** | **데이터 요청, 사용자 요청 여부, 출력여부, 데이터 시각화 알림, 장비 결함을 상태로 두기로 했다.** | |
| 8.1.3 상태 나열/전이 정의 | | |
| **개인 의견** | **김동혁** |  |
| **김영민** |  |
| **김재현** |  |
| **이영채** |  |
| **토의 내용** | **김동혁** | **장비 결함을 데이터 요청에서 동기바로 연결하는 것이 아니라, 데이터 요청을 통해서 알아야 한다고 생각한다.** |
| **김영민** | **오류 확인을 추가하는 것이 맞다고 생각한다. 김재현 팀원의 의견처럼 동기바 대신에 조건문을 활용하는 것이 맞는 것 같다.** |
| **김재현** | **데이터 요청 후에 데이터의 존재 유무 분기문이 있어야한다고 생각한다.** |
| **이영채** | **김재현 팀원의 다이어그램대로 출력이라는 슈퍼 상태에 알림과 데이터 시각화를 서브 상태로 하는 것이 좋겠다고 생각한다.** |
| **결론** | **김재현 팀원의 동기바 사용과 슈퍼, 서브 상태의 사용이 올바르다고 생각하여, 김재현 팀원의 다이어그램을 따르기로 했다.** | |
| 8.1.4 상세 부분 정의 | | |
| **토의 내용** | **김동혁** | * **사용자의 요청으로 상태 시작** * **장비 결함이 있을 경우 알림 상태가 되어야 하므로, 이벤트는 결함 상태 전송이다.** * **알림에 이력상태를 추가하는 것이 좋을 것 같다.** |
| **김영민** | * **장비 결함의 경우에는 사용자의 요청 없이도 발생하기 때문에, 상태 시작은 주기적으로 시작되는 것이 맞는 것 같다.** * **알림을 모듈화하여 하나로 통일하는 것이 좋을 것 같다.** |
| **김재현** | * **데이터를 요청했을 때, 데이터가 없으면 종료하고 존재하면 사용자 요청 여부 단계로 진행한다.** * **데이터 시각화 요청이 있을 경우, 데이터 시각화와 알림을 진행하고 없으면 알림만 진행한다.** |
| **이영채** | * **시스템은 항상 장비 결함과 데이터 요청에 대해 주기적으로 확인해야한다.** * **장비 결함의 대한 알림은 사용자의 요청 여부와 관계없이 작동하게 한다.** |
| **결론** |  | |
| **8.2 상태 다이어그램 작성** | | |
| **결론** |  | |

|  |  |  |
| --- | --- | --- |
| **9장 토의 보고서** | | |
| **토의 사항** | **9장. 컴포넌트 다이어그램**  **9.1 프로세스 정의**  **9.1.1. 컴포넌트 대상 정의**  **9.1.2. 컴포넌트 식별**  **9.1.3. 인터페이스 정의 및 컴포넌트 배치**  **9.1.4. 의존 관계와 실체화 관계 정의.**  **9.2 컴포넌트 다이어그램 작성** | |
| **진행일자** | **2020년 11월 17일 13:20~14:50** | |
| **참석자** | **김영민(팀장), 김재현, 이영채, 김동혁(서기)** | |
| **9.1 프로세스 정의** | | |
|  | | |
| 9.1.1 컴포넌트 대상 정의 | | |
| **기존**  **클래스** |  | |
| 9.1.2 컴포넌트 식별 | | |
| **개인 의견** | **김동혁** | **데이터 시각화 모듈과 알림시스템을 묶어서 데이터 알림 컴포넌트가 있어야 할 것 같다.** |
| **김영민** | **고장상태관리를 상태출력이라는 컴포넌트로 설정하면 될 것 같다.** |
| **김재현** | **딥러닝과 데이터 저장, 예측데이터 제공을 하나로 묶어 컴포넌트를 정의하면 될 것 같다.** |
| **이영채** | **예측정비 모듈과 결과출력을 묶어서 컴포넌트로 정의하면 될 것 같다** |
| **토의 내용** | **각각의 클래스들 중에서 데이터 시각화 모듈, 알림시스템은 묶어서 데이터 알림 컴포넌트로 정의하고, 고장상태 관리는 상태출력클래스 자체를 컴포넌트로 설정한다.**  **그리고 딥러닝과 데이터 저장, 예측데이터 또한 하나로 묶어 컴포넌트를 설정하고 나머지 예측정비 모듈과 결과출력을 묶어서 컴포넌트로 정의 하였다.** | |
| **결론** |  | |
| 9.1.3 인터페이스 정의 및 컴포넌트 배치 | | |
| **개인 의견** | **김동혁** | **각각의 데이터를 시각화 해주는 데이터 알림 컴포넌트의 인터페이스를 통해 다른 컴포넌트와 연결되어야한다.** |
| **김영민** | **상태를 출력하여 정보를 제공하는 인터페이스가 정의 되어야 할 것 같다.** |
| **김재현** | **딥러닝된 결과를 예측정비컴포넌트에 보낼수 있는 인터페이스가 정의되어야 할 것 같다.** |
| **이영채** | **데이터 알림 컴포넌트와 예측결과출력 컴포넌트 사이에 인터페이스가 정의 되어야 할 것 같다.** |
| **토의 내용** | **김동혁** | **다른 팀원들의 의견처럼 데이터 알림 컴포넌트와 예측결과 컴포넌트 사이에 시각화 인터페이스가 존재해야 한다.** |
| **김영민** | **예측정비컴포넌트에 인터페이스를 추가로 정의하는 것은 어떨지 생각한다.** |
| **김재현** | **시각화 컴포넌트와 예측정비 컴포넌트를 굳이 따로 두어야 하는가 하는 생각이 든다.** |
| **이영채** | **데이터 클리닝 컴포넌트 관련하여 해당 컴포넌트가 더 나뉘어야 하는 지 의문이든다.** |
| **결론** |  | |
| 9.1.4 의존 관계와 실체화 관계 정의 | | |
| **토의 내용** | **김동혁** | * **Print status 가 예측데이터 컴포넌트랑 연결되는 것이 아닌 개별적으로 사용자 컴포넌트와 연결되어야 한다.** |
| **김영민** | * **출력 컴포넌트가 데이터 예측데이터 저장에 의존관계로 이어주는 것이 맞다고 생각하였으나 토의 결과 연관관계로 이어야 한다고 생각한다.** |
| **김재현** | * **딥러닝 인터페이스와 예측정비 컴포넌트가 의존관계가 아닌 연관관계로 나타내어야 한다** |
| **이영채** | * **각가의 인터페이스들은 클래스 다이어그램에서 사용한 인터페이스들을 사용하면 될 것 같다.** |
| **결론** |  | |
| **9.2 컴포넌트 다이어그램 작성** | | |
| **결론** |  | |

표

|  |  |  |  |
| --- | --- | --- | --- |
| **10장 토의 보고서** | | | |
| **토의 사항** | **10장. 배치 다이어그램**  **10.1 프로세스 정의**  **10.1.1 노드를 식별하여 정의**  **10.1.2 컴포넌트를 식별**  **10.1.3 노드간 구성관계를 정의**  **10.1.4 노드에 컴포넌트를 배치**  **10.2 배치 다이어그램 작성** | | |
| **진행일자** | **2020년 11월 26일 9:00~11:00** | | |
| **참석자** | **김영민(서기), 김재현(팀장), 김동혁, 이영채** | | |
| **10.1 프로세스 정의** | | | |
|  | | | |
| 10.1.1 노드를 식별하여 정의 | | | |
| **개인 의견** | **김동혁** | - 데이터 베이스  - 딥러닝하고 데이터수집 합쳐서 딥러닝처리서버  - 데이터 출력서버  - 시각화 서버  - 알림서버 | |
| **김영민** | - 사용자  - DB  - 알림 서버  - 딥러닝/데이터 처리 서버 | |
| **김재현** | - 데이터 베이스  - 사용자 클라이언트  - 시각화, 알림 서버  - 데이터 정제 서버  - 출력 서버 | |
| **이영채** | - 유저 컴퓨터  - 딥러닝 서버  - 시각화 및 알람 서버  - 데이터 수집 서버  - 데이터베이스 | |
| **토의 내용** | **김동혁** | - 딥러닝 서버와 데이터 수집 서버를 나누지 않고 한번에 처리하는 것이 좋을 것 같다. | |
| **김영민** | - 데이터의 수집, 저장, 딥 러닝은 한번에 모아서 하는 것이 좋을 것 같다. | |
| **김재현** | - 딥러닝 서버에 인터페이스를 만들기 보다는 딥러닝 기능을 데이터 정제 서버에서 실행하는 방식으로 구현하는 것이 좋을 것 같다. | |
| **이영채** | - 시각화와 알람 출력이 같은 서버에서 실행되는 것이 좋을 것 같다. | |
| **결론** | - 데이터 베이스  - 사용자 클라이언트  - 시각화, 알림 서버  - 데이터 정제 서버 | | |
| 10.1.2 컴포넌트를 식별 | | | |
| **결론** | 이전 팀 프로젝트에서 식별한 내용을 차용 하였음 | | |
| 10.1.3 노드간 구성관계를 정의 | | | |
| **개인 의견** | **김동혁** |  | |
| **김영민** |  | |
| **김재현** |  | |
| **이영채** |  | |
| **토의 내용** | **김동혁** | - 다른 팀원의 의견처럼 데이터베이스와 시각화 알림 서버 사이에 연결이 있어야 할 것 같다. | |
| **김영민** | - 연결해주는 과정에서 프로토콜을 명시해주는 것이 바람직한 것 같다.  - 알림 서버와 사용자 클라이언트 서버의 관계는 1 대 n 관계가 맞는 것 같다. | |
| **김재현** | - 사용자 클라이언트 서버와 데이터 정제 서버를 연결하는 것이 아닌 데이터 정제 서버와 시각화 알림 서버를 연결하는 것이 맞는 것 같다. | |
| **이영채** | - 서버와 서버, 서버와 데이터베이스 사이에 관계는 1 대 1 관계가 맞다고 생각하고, 시각화 알림 서버와 사용자 클라이언트는 1 대 n 관계가 맞다고 생각한다. | |
| **결론** | 토의 결과 이영채 팀원의 노드 배치를 중심으로 수정하여 최종 결론을 작성하였음 | | |
| 10.1.4 노드에 컴포넌트를 배치 | | | |
| **토의 내용** | **김동혁** | - 컴포넌트 의존 관계의 경우 Data Cleaning Component와 Print Predict Data Component를 연결해주어서 컴포넌트 실행에 순차적 단계를 명시해 주어야 하며 이 컴포넌트들은 모두 데이터 정제 서버에 적재 되어야 한다. | |
| **김영민** | - Data Cleaning Component는 데이터 정제에 배치하고 Print Status, Data Alarm에서 User로 의존 관계를 만들어 주는 것이 좋을 것 같다.  - Data Cleaning 와 Print Predict Data을 의존 관계로 만들어 주는 것이 좋을 것 같다. | |
| **김재현** | - 데이터 정제 서버에는 Data Cleaning Component을 배치한다.  - 사용자 클라이언트 에 User를 배치시킨다.  - 알림, 시각화 서버에 Print Status, Data Alarm, Print Predict Data Component를 배치 시킨다.  - 의존 관계의 경우에는 Data Cleaning Component와 Print Status, Data Alarm, Print Predict Data Component 3개를 연결 시키고 위의 3개를 User Component에 연결 시켜주는 것이 좋을 것 같다. | |
| **이영채** | - Data Cleaning Component와 Print Predict Data를 연결  - Data Alarm과 Print Predict를 연결하고, Print Predict를 User와 연결해준다.  - 시각화 알림 서버에 Print Predict Data 적재, 데이터 정제 서버에 Data Cleaning Component 적재한다. | |
| **결론** |  | | |
| **10.2**  **배치 다이어그램 작성** | | |
| **결론** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **11장 토의 보고서** | | | |
| **토의 사항** | **11장. 패키지 다이어그램**  **11.1 프로세스 정의**  **11.1.1 추상적으로 비슷한 요소 집합을 패키지로 구성**  **11.1.2 패키지의 외부 접근 허용 요소 정의**  **11.1.3 외부 패키지에 의존하는 패키지 import 표현**  **11.1.4 병합된 패키지의 경우 관계 표현**  **11.2 패키지 다이어그램 작성** | | |
| **진행일자** | **2020년 12월 02일 9:00~10:30** | | |
| **참석자** | **김영민(팀장), 김재현(서기), 김동혁, 이영채** | | |
| **11.1 프로세스 정의** | | | |
|  | | | |
| 11.1.1 패키지 요소 파악 | | | |
| **개인 의견** | **김동혁** | - 컴포넌트 단위로 묶어서 데이터 정제 패키지와 알림 패키지, 사용자 패키지로 나누는 것이 좋다. | |
| **김영민** | - 클라이언트 패키지를 만들고 User컴포넌트를 관리자 패키지와 사용자 패키지로 나눈다. | |
| **김재현** | - 서버 패키지와 서버 내부에서 실행할 컴포넌트의 패키지를 따로 만들어 할 것 같다. | |
| **이영채** | - 시각화 알림 서버랑 데이터 정제 서버 묶어서 하나의 패키지로 만드는 것이 좋다고 생각한다. | |
| **토의 내용** | **김동혁** | - 김영민 팀원처럼 관리자 패키지가 필요하다고 생각한다. | |
| **김영민** | - 김동혁 팀원의 의견처럼 데이터 베이스, 알림, 정제, 사용자로 패키지를 나누는 것이 좋다고 생각한다. | |
| **김재현** | - 김영민 팀원 말대로 User컴포넌트를 패키지화 할 때 사용자와 관리자를 나누는 것이 좋다고 생각한다. | |
| **이영채** | - 김동혁 팀원의 의견에 동의하여 패키지를 나누는 것이 좋다고 생각한다. | |
| **결론** | - 데이터 서버 패키지  - 사용자, 관리자 패키지  - 데이터 정제 패키지  - 알림 패키지  - 서버 패키지 | | |
| 11.1.2 패키지의 외부 접근 허용 요소 정의 | | | |
| **개인 의견** | **김동혁** | | - 사용자 패키지 -> +UI  - 관리자 패키지 -> +UI  - 서버 패키지 -> +서버  - 알림 패키지 -> +예측 데이터, -요청 데이터  - 데이터 서버 패키지 -> -데이터 베이스  - 데이터 정재 패키지 -> -정제된 데이터 |
| **김영민** | | - 사용자 패키지 -> +사용자 모니터  - 관리자 패키지 -> + 관리자 모니터  - 서버 패키지 -> +데이터  - 알림 패키지 -> + 예측 값, + 요청 값  - 데이터 정제 패키지 -> -딥러닝, -데이터 정제, + 결과 값  - 데이터 서버 패키지 -> -데이터 베이스 |
| **김재현** | | - 사용자 패키지 -> +모니터  - 관리자 패키지 -> +모니터  - 서버 패키지 -> +시각화 서버, -데이터 정제 서버  - 알림 패키지 -> -예측 데이터, -요청 데이터  - 데이터 정재 패키지 -> -딥러닝, -데이터 정제  - 데이터 서버 패키지 -> -데이터 베이스 |
| **이영채** | | - 사용자 패키지 -> + 사용자 클라이언트  - 관리자 패키지 -> + 관리자 클라이언트  - 서버 패키지 -> - 데이터  - 데이터 서버 패키지 -> -데이터 베이스  - 알림 패키지 -> + 알람 데이터, +예측 결과  - 데이터 정제 패키지 -> - 딥러닝 데이터 |
| **토의 내용** | **김동혁** | | - 알림 패키지에서 예측 값을 -로 설정 안해도 된다고 생각한다. |
| **김영민** | | - 김동혁 팀원의 의견대로 사용자, 관리자 UI가 필요하다고 생각한다. |
| **김재현** | | - 예측 데이터를 -로 두었는데 굳이 외부 접속을 막을 필요가 없다고 생각이 변했다. |
| **이영채** | | - 김재현 팀원의 의견처럼 시각화 서버의 데이터만 +로 설정하는 것이 좋다고 생각한다. |
| **결론** | - 사용자 패키지 -> +사용자 UI  - 관리자 패키지 -> + 관리자 UI  - 서버 패키지 -> #시각화 서버, #데이터 정제 서버  - 알림 패키지 -> + 예측 값, - 요청 값  - 데이터 정제 패키지 -> -딥러닝, -데이터 정제, + 결과 값  - 데이터 서버 패키지 -> -데이터 베이스 | | |
| 11.1.3 외부 패키지에 의존하는 패키지 import 표현 | | | |
| **개인 의견** | **김동혁** | - 데이터 서버 패키지 –(import)> 서버 패키지  - 서버 패키지 -(import)> 데이터 정제패키지  - 데이터 정제패키지 -(import)> 서버 패키지  - 데이터 정제패키지 -(import)> 알림 패키지  - 알림 패키지 -(import)> 사용자 패키지  - 알림 패키지 -(import)> 관리자 패키지 | |
| **김영민** | - 사용자 패키지 -(import)> 알림 패키지  - 서버 패키지 –(import)> 데이터 서버 패키지  - 관리자 패키지 -(import)> 서버 패키지  - 알림 패키지 -(import)> 데이터 정제 패키지  - 데이저 정제 패키지 -(import)> 데이터 서버 패키지 | |
| **김재현** | - 서버 패키지가 데이터 서버 패키지 import  - 데이터 서버 패키지가 서버 패키지 export  **-** 데이터 정제 패키지가 서버 패키지 import  **-** 서버 패키지가 데이터 정제 패키지 export  - 알림 패키지가 데이터 정제 패키지 import  - 데이터 정제 패키지가 서버 패키지 export  **-** 사용자, 관리자 패키지가 알림 패키지 import  - 알림 패키지가 사용자,관리자 패키지 export | |
| **이영채** | - 알림 패키지 -import-> 서버 패키지  - 서버 패키지 -import-> 데이터 정제 패키지  - 알림 패키지 -Export-> 사용자 클라이언트  - 서버 패키지 -Export-> 관리자 클라이언트  - 데이터 서버 패키지 -Export> 서버 패키지 | |
| **토의 내용** | **김동혁** | - 김영민 팀원 표현에서 import가 아닌 export로 바뀌는 것이 맞는 것 같다 | |
| **김영민** | - import 뿐만 아니라 export도 활용하는 것이 좋을 것 같다. | |
| **김재현** | - 김영민 팀원의 내용이 잘못된 건 아니지만 표현 방식이 틀린 것 같다.  - 김동혁 팀원의 내용처럼 양방향으로 import를 해야할 것 같다. | |
| **이영채** | - 한쪽에서만 import를 하는 것이 맞는 것 같다. | |
| **결론** | - 데이터 서버 패키지 –(import)> 서버 패키지  - 서버 패키지 -(import)> 데이터 정제패키지  - 데이터 정제패키지 -(import)> 서버 패키지  - 데이터 정제패키지 -(import)> 알림 패키지  - 알림 패키지 -(import)> 사용자 패키지  - 알림 패키지 -(import)> 관리자 패키지 | | |
| 11.1.4 병합된 패키지의 경우 관계 표현 | | | |
| **토의 내용** | **김동혁** | - 관리자, 사용자 패키지를 병합하는 것이 좋을 것 같다. | |
| **김영민** | - 병합은 굳이 사용하지 않아도 될 것 같다. | |
| **김재현** | - 비슷한 기능이 있는 패키지가 없어서 병합은 안해도 될 것 같다. | |
| **이영채** | - 지금이 제일 적절한 것 같다. | |
| **결론** | - 관리자, 사용자 패키지가 생각보다 공통으로 묶을 이유가 없어서 현상 유지하기로 했다. | | |
| **11.2**  **패키지 다이어그램 작성** | | | |
| **결론** |  | | |

|  |  |  |
| --- | --- | --- |
| **12장 수업 Review** | | |
| **김영민** | 프로젝트 과정 | 이번 객체 지향 분석 팀 프로젝트를 통해서 많은 것을 배우고 과정을 터득해 나간 것 같습니다. 우선 기본적인 UML 개념에 대해서 배우고 이후 여러가지 다양한 다이어그램들에 대한 개념과 특징, 장단점, 사용법, 여러가지를 자세하게 배울 수 있었습니다. 수업에서 배운 다이어그램을 직접 복습하고 과제를 통해 반복해서 학습하면서 개념을 익힌 이후 팀 프로젝트를 통해 저희 팀 만의 주제에 대해서 다시 다이어그램과 보고서를 작성하는 방법을 익힐 수 있었습니다. |
| 배운 점 | 이번 객체 지향 분석 팀 프로젝트를 통해서 배운 것이 많았지만 먼저 첫째로 팀 프로젝트 주제를 통한 다이어그램 개념 학습이 있었습니다. 기존 수업에서 배운 내용들을 실제로 팀원들과 토의해 나가면서 잘 모르는 개념을 정립하고 활용해 가면서 여러가지 다양한 다이어그램들을 작성해 나가면서 동시에 개념을 익힐 수 있는 좋은 기회였습니다. 둘째로, 팀 프로젝트를 통한 협업 능력을 기를 수 있었습니다. 한 학기를 통째로 할애하는 장기적인 팀 프로젝트였기 때문에 팀원과의 의사소통과 협업이 무엇보다 중요했습니다. 팀원들과 함께 완벽하진 않지만 만족스러운 팀 프로젝트를 진행할 수 있어서 더욱 성장할 수 있는 경험을 할 수 있어서 정말 좋았습니다. 마지막으로 팀프로젝트를 통해 여러가지 다이어그램을 활용하고 작성하면서 다이어그램 툴에 대한 이해도를 올릴 수 있었습니다. 여러가지 툴을 사용해보면서 좀 더 좋아 보이는 툴을 고를 수 있었고 툴에 대한 이해도가 오르면서 작업의 효율 또한 올릴 수 있었습니다. 추후 현업에 종사할 때도 이번 팀 프로젝트에 대한 경험을 살려 팀원들과 원활한 협업 관계를 구축할 수 있도록 하겠습니다. |
| 배운 점을 통한 다른 프로젝트에서의 적용점 | 이번 팀 프로젝트를 통해 배운 점들은 또 다시 그대로 다른 팀 프로젝트에서 적용하기 용이할 것입니다. 무엇보다 장기적인 팀 프로젝트를 통해서 팀원 간의 의사소통 능력과 협업 능력이 상승하였고 이렇게 배운 점들은 다른 팀 프로젝트에서도 충분히 활용할 수 있는 점들입니다. 특히 프로젝트를 진행하는 동안 더 나은 결과를 위해 팀원들과 함께 적극적인 토론을 하는 법을 배웠고, 이 또한 다른 팀 프로젝트의 협업 과정에서 원활한 의사소통과 토론을 통해서 더욱 발전된 결과물들을 도출해 낼 수 있다고 생각됩니다. 이번 팀 프로젝트 수업을 통해서 배운 점들을 잘 기억해서 추후 다른 수업 또는 다른 협업에서 활용할 수 있도록 더 노력 하겠습니다. |
| **김동혁** | 프로젝트 과정 | 플립드 러닝 수업을 진행하면서 처음에는 거꾸로 학습..? 이라는 말에 의아함을 가지고 임했던 것 같습니다. 대면 강의를 보통 진행하던 저의 습관들은 플립드 러닝이라는 방식을 쉽게 받아들이지 못했던 것 같습니다.  그래서 그런지 처음은 너무나 힘들었습니다. 요약정리는 끝이 안보이고, 질문거리를 찾기도 어렵고 그리고 모호한 개념으로 인해 제가 뭘 알고 뭘 모르고 있는 지조차 파악할 수 없었던 것 같습니다.  하지만 한주 한주가 지나고 동영상을 통한 자율학습, 그리고 그를 통한 질문을 바탕으로 대면수업때는 교수님 과의 질의 등을 통해 피드백을 진행하고 피드백의 과정속에서 전체적인 view의 관점으로 설명을 들으면서, UML 다이어그램, SW의 전체적인 관점에 대한 이해를 한 주 한 주 해 나갈 수 있는 시간이었습니다. |
| 배운 점 | 무엇보다 이 수업을 통해서 얻을 수 있었던 점은 자율학습 측면이 컸던 것 같습니다.  내가 뭘 모르는지 파악을 하고 그것들을 해결하려고 하는 문제 해결능력을 가장 먼저 키울 수 있었던 것 같습니다.  ‘ 또한 키울 수 있었던 점은 팀워크였습니다. 모든 강의마다 팀플레이가 요구되고 그 과정속에서 교수님께서 말씀하신 배려, 경청 등의 영역이 가장 많이 시험 받게 되고 훈련이 되는 영역이었습니다. 또한 더 나아가 객체지향에 대한 개념들을 다시금 정리하고 재정립할 수 있었으며, 4+1 뷰를 통해 SW를 바라보는 관점을 더 자세하게 알 수 있었고 설계에 있어 문서의 중요성, 또한 프로세스의 중요성에 대해 다시한번 상기시킬 수 있는 시간이었습니다.  마지막으로 무엇보다 문서작업에 대한 영역을 발전시킬 수 있었습니다. 정형화된 문서를 사용하여 팀원 간의 의견을 정리하고 양식을 통일하여 이전과 이후의 과정마저 모두 담을 수 있는 방식을 통해, 이 한 학기동안 어느정도 발전했는지를 알 수 있는 시간이었습니다. |
| 배운 점을 통한 다른 프로젝트에서의 적용점 | 앞으로의 어떤 실무의 영역에서도 이번학기의 수업은 계속 기억에 남을 것 같고, 이런 협업의 능력이나 tool의 사용, 그리고 자율학습을 통해 나의 위치를 파악하고 또한 다이어그램을 통한 시스템확인 등등의 영역에서 이번 경험을 토대로 잘 적응해 나가고 싶다.  좋았고 배울 것이 많았던 한 학기였습니다. 감사합니다. |
| **김재현** | 프로젝트 과정 | 본격적인 수업에 들어가기 전 수업의 소개를 하며 팀을 3~4명으로 구성하였습니다. 이 팀원들과 학기말까지 함께 공부하며 공부해본 것을 적용해보기 위한 팀프로젝트를 수행하였습니다. 먼저 UML을 나타낼 수 있는 주제를 선정하고 주제를 각 챕터에 맞게 생각하고 토의하며 결과물을 만들었습니다. 또한 플립드 러닝을 통해 교수님께서 강의만 하는 수업이 아닌 자습과 팀과의 스터디 그룹을 통해 학습하고 생기는 질문을 교수님께서 해결해주는 방식이었습니다. 매주 학습한 내용을 요약 및 정리하여 제출하고 책의 예제를 풀어보고 선정한 주제를 챕터에 맞는 결과물을 내기 위하여 토의를 진행하고 이런 과정에 대한 자기 자신, 동료 평가를 진행했습니다. 마지막으로 다른 팀들의 팀프로젝트를 중간 발표, 최종 발표를 통해 확인할 수 있었습니다. |
| 배운 점 | UML의 개념과 각각의 다이어그램의 특징, 모델링 프로세스, 모델링시 주의사항을 배웠습니다. 각각의 다이어그램이 언제 만들어지고, 만들어지는데 어떤 선행 조건이 필요하며 유스케이스 다이어그램이 거의 모든 다이어그램의 기본이 되는 것을 배웠습니다. 또한 다이어그램마다 개발하는 4+1아키텍처가 다른 것도 배웠습니다. 그리고 내용 요약정리, 평가, 팀프로젝트 보고서를 작성하며 문서 tool에 대해 많이 배우고 보고서를 작성하며 기본적인 작성법에 대해 배웠습니다. 팀프로젝트를 통해 팀원들 과의 협업을 해보고 이런 협업을 위해 필요한 업무 분담, 배려, 의사 소통에 대해 배웠습니다. 이런 것들을 배우며 사회에 나갈 수 있는 기본적인 틀을 확립할 수 있었으며 팀프로젝트를 통해 협업의 중요성, 장점에 대해 특히 많이 깨달았습니다. 또, 플립드 러닝을 통해 강의는 듣는 것만이 아닌 스스로의 학습이 매우 중요하다는 것을 배우고 아무리 강의를 듣고 질문을 한다 하여도 자신 스스로가 이해하지 못하고 러닝을 하지 않는다면 쓸모가 없다는 것을 직접 몸으로 느꼈습니다. 마지막으로 자신에 대한 평가를 쓰며 프로젝트 수행 시의 자신의 모습을 알 수 있었고 어떻게 행동해야 하는지 지표가 될 수 있었습니다. |
| 배운 점을 통한 다른 프로젝트에서의 적용점 | 이번 수업의 경험을 통해서 다른 곳에서 보고서를 자신 있게 작성할 수 있는 능력을 얻었습니다. 그리고 팀 프로젝트의 경험을 통해 다른 팀 프로젝트들도 남을 배려하며 의사소통을 하고 토의를 통해 결과를 도출할 수 있게 되었습니다. 중간 발표와 기말 발표를 하며 많은 내용을 어떻게 요약해서 시간 내에 발표할 수 있는지 경험을 해볼 수 있었고 이를 통해 발표준비에 대한 능력이 생겼습니다. 그리고 플립드 러닝을 수업 내용을 요약, 정리하여 자기주도 학습을 통해 공부하는 법을 배웠고 질문을 스스로 만들어 완벽히 이해하는 방법을 배웠습니다. 이런 내용들을 수행하며 자신에 대한 평가를 하고 보완점을 찾을 수 있게 되었습니다. 이 과목을 수강하며 인생에 도움이 되는 내용을 많이 배웠고 이런 수업을 해주신 교수님께 감사드립니다. |
| **이영채** | 프로젝트 과정 | 플립드 러닝을 기본으로 하는 객체 지향 분석 수업에서는 수업을 하기 전에 미리 교수님이 올리신 동영상을 통해 UML에 대한 기본 개념들을 파악합니다. 동영상을 통해 강의를 시청한 후 책과 교수님이 올리신 강의 자료를 통해 스스로 해당 주차의 Chapter를 요약하고, 모르는 것에 대해 교수님께 질문합니다. 교수님께서는 질문을 받아 대면으로 하는 수업시간에 답변을 해주시고, 이해가 부족했던 부분을 다시 알아갔습니다. 이러한 학습을 마친 후, 팀 활동에서 주제를 하나 정해 그 주제에 관한 해당 주차의 다이어그램을 그렸습니다. 팀 활동은 각각의 프로세스마다, 개인 의견을 낸 후, 토론을 하고 결론을 내는 과정을 통해 팀 활동을 했으며, 매주 자신의 학습을 되돌아보는 자가평가와 팀 활동을 하며 팀원들을 평가하는 팀원 평가를 했습니다. 또한, 교수님이 주신 Homework를 통해 직접 다이어그램을 모델링 해보며, Self-Study를 통해 다이어그램 모델링에 대한 능력을 키워 나갔습니다. 중간고사 전과 기말고사 전에 지금까지 배운 내용에 대해 팀당 3명씩 나와 시험 전까지의 활동을 발표하는 시간을 가졌습니다. |
| 배운 점 | 객체 지향 분석이라는 과목을 통해 기본적인 UML의 개념과 각각의 다이어그램의 특징, 모델링 프로세스, 모델링을 하며 주의해야할 점에 대해 배웠지만, 이번 활동을 통해 이런 지식 외에 다른 가치 있는 것들을 더 많이 배운 것 같습니다. 이번 학습을 통해 첫번째로, 팀원들과 의사 소통하며 프로젝트를 원활하게 진행하는 방법을 배웠습니다. 지금까지 팀 프로젝트를 여러 개 해봤지만, 이번 수업만큼은 교수님이 강조하신 팀원들을 배려하는 것에 집중하려고 노력했으며, 팀 프로젝트에서 업무 분담, 협업, 의사소통에 대해 배웠던 시간이었습니다. 두번째로는, 보고서 작성 등과 같은 문서 작성 능력을 배웠습니다. 항상 보고서를 투박하게 만들었는데, 보고서의 목차와 표를 사용하며 깔끔한 보고서를 작성했습니다. 또한 UML 다이어그램을 그리기 위한 Tool에 대해 배웠습니다. |
| 배운 점을 통한 다른 프로젝트에서의 적용점 | 이번 수업을 통해서 각각의 UML이 언제 쓰이는지 알게 되었고 어떻게 쓰이는지도 알게 되었습니다. 이러한 지식을 실무에서도 잘 적용하여 알맞은 때에 적당한 다이어그램을 적용하여 사용하면 좋겠다고 생각했습니다. 또한 교수님이 올려 주신 영상을 보며 미리 자기 주도적으로 학습하는 방법을 배웠는데, 이러한 방법을 습관처럼 다른 과목이나 공부에도 적용하여 사용하겠습니다.그리고 나중에 실무에 가면 더 많은 사람들과 하는 장기적인 프로젝트가 많을 예정인데, 이번에 팀 프로젝트를 하며 배운 점들인 협업, 배려, 의사소통, 토론 방법 등을 잘 활용하여 프로젝트에 적용하면 좋겠다고 생각했습니다. |

**▶ 프로세스 정의 명확하게 추가**

**▶ 최종 결론 명확하게 추가**

**▶ 토의에 사용하는 메신저 툴 정보 추가**

**▶ 보고서 폰트 양식 정형화**

**▶ 토의 결과 내용에 대한 개인 의견 중요도 표시**

이후에 시행하는 토의 및 활동에서 결과를 도출하는데 비중을 많이 차지한 의견을 노란색 형광펜 효과를 추가하여 표시할 것

**<7장 어려웠던 점>(활동 다이어그램)**

**전이 부분에 데이터 보유 여부를 확인할 것인지, 응답 확인 할 것인지 선택할 때 어려움이 있었다.**

**<8장 어려웠던 점>(상태 다이어그램)**

**데이터 시각화와 알림에 대한 상태를 서브상태로 놓아야 하는가에 대한 논의에 어려움이 있었다.**

(어떤 것을 슈퍼 상태와 서브 상태로 놓을 것인지 논의할 때 어려움이 있었다.)

**<9장 어려웠던 점>(컴포넌트 다이어그램)**

**컴포넌트 다이어그램 작성 이전 클래스에서 대상추출이 상당히 어려웠다**

**(컴포넌트 대상 정의할 때 이전 클래스에서 대상 추출이 상당히 어려웠다.)**

**<10장 어려웠던 점>(배치 다이어그램)**

**노드를 정의하고 식별하는데 에서 어려움을 겪었다.**

**<11장 어려웠던 점>(패키지 다이어그램)**

**패키지간 import와 export관계 지정에 어려움을 겪었다.**