

실무중심산학협력프로젝트(캡스톤디자인) 지침서

과목 명 : 실무중심산학협력프로젝트 (캡스톤디자인-SW)

담당교수: 조성제

개설학기: 2018년 2학기

수강대상: 4학년

1. 설계 주제

실무중심산학협력프로젝트(캡스톤디자인)의 설계주제는 소프트웨어학과 1,2,3,4 학년에서 배운 전공지식을 활용하여 팀 단위로 진행할 수 있는 내용이면 된다. 소프트웨어 설계는 소프트웨어 개발방법론에 따라 분석 및 개발을 할 수 있어야 한다. 프로젝트는 4명 내외의 팀을 구성하여 진행한다. 주요 설계주제로는 인공지능, 빅데이터, 클라우드, IoT, 사이버보안 등과 관련 최신기술 관련 주제를 권장한다. 만약 학생이 적절한 설계주제를 제시하지 못할 경우 담당교수님이 제시한 분야에서 구체적인 하나를 선택하는 방식으로 한다. 선정된 소프트웨어를 소프트웨어공학(Software Engineering)을 기반으로 설계하고 구현한다. 개발하고자 하는 소프트웨어의 목표를 명확히 설정하고, 이 목표에 도달했음을 입증하는 논리적 또는 수리적 유도과정을 수립한다. 팀원 간의 토론을 통하여 논리적 수리적 유도를 진행하며, 소프트웨어의 구현을 진행하면서 팀원 간의 협동 및 조직운동 능력을 배양한다. 이러한 개발에서 평가에 이르는 전 과정을 소프트웨어 설계절차에 준하여 실시한다.

2. 설계 과제

소프트웨어학과 1,2,3학년에서 배운 지식을 활용하여, 인공지능, 빅데이터, 클라우드, IoT, 사이버보안 등의 최신 기술 관련분야의 소프트웨어 시스템 설계에 관한 주제를 정하도록 한다. 다음 주제들이 그 예이다.

- (1) PC 또는 서버에서 동작하는 윈도우/리눅스 애플리케이션 프로그램
- (2) 스마트 폰에서 동작하는 모바일 애플리케이션
- (3) 클라우드 컴퓨팅 환경과 연동되는 또는 클라우드에서 동작하는 애플리케이션
- (4) 모바일-클라우드 컴퓨팅 애플리케이션
- (5) 하드웨어 시스템(임베디드 시스템, 각종 전자기계장치)과 연동하거나 혹은 하드웨어 시스템내의 CPU에서 동작하는 소프트웨어
- (6) 게임 프로그램

3. 설계 내용

설계내용은 기본적으로 아래의 사항을 고려한 소프트웨어 시스템을 설계하는 것이며, 설계 주제에 따라 특정 방법론에 치중하거나 명확한 참조자료를 제시하고 그에 해당하는 방법론

으로 설계내용을 정하여도 된다. 가능한 한 최대한 자세히 설계방법과 내용에 대하여 기술하여야 한다.

- (1) 소프트웨어 시스템이 동작할 환경을 명확히 명시한다.
 - 소프트웨어 시스템이 동작할 컴퓨팅 시스템
 - 소프트웨어 시스템이 동작할 OS
- (2) IDE 등과 같은 소프트웨어 개발 환경에 대해서 명확히 설명한다.
- (3) 동작할 프로세스의 종류와 기능에 대해서 명확히 기술한다.
 - 블록다이어그램 혹은 좀 더 자세한 설계 톨로 프로그램의 구조를 설명한다.
 - 프로세스 간의 연동과정을 명시한다.
- (4) 프로세스 내부적으로 본 과목에서 배우게 되는 데이터 구조와 알고리즘에 대한 내용을 명확히 기술한다.
- (5) 데이터 구조에 대한 설계를 UML이나 일반적인 톨을 이용하던지 Java 구조체를 간략하게 표현해서 pseudo코드로 표현하든지 설계한다.
- (6) 프로세스 간의 통신 메시지들을 명확히 설계한다.
- (7) 성능이나 기능을 테스트할 방법을 명확히 제시한다.
- (8) 처음 개념설계시의 정의된 기능을 설계된 시스템에서 어떻게 동작하는지 확인한 후 설계의 수정이 필요하면 수정한다.

4. 설계 제한 요소

강의계획서에 제시된 다음의 기본 설계 요소를 따른다.

- 경제, 환경, 사회, 미학 : 가능한 한 경제적인 효과, 오락적 요소, 사회 환경적으로 유용성이 있는 소프트웨어 설계
- 윤리 : 소프트웨어의 기능은 윤리, 보건 및 안전적으로 문제가 되지 않는 범위의 소프트웨어시스템 설계
- 생산성과 내구성 : 필요시 입출력 데이터를 효과적으로 처리하고, 이상종료가 발생하지 않는 소프트웨어 시스템 설계
- 산업표준 : 표준화된 요소의 사용/기능요건 선정 시 표준 검토

- (1) 기본적인 설계 방법은 설계수행자의 몫이나 정립된 이론이나 방법론에 따른 설계 방법에 따를 것.
- (2) 4명 내외의 팀으로 프로젝트 팀을 구성하고 회의를 통해 각자의 업무를 결정하고 협의를 통해 설계 개발 일정을 정할 것.
- (3) 최종보고서를 반드시 주어진 포맷을 바탕으로 갱신할 것.
- (4) 시연이 가능한 설계과제 혹은 최종보고서로 발표할 수 있는 설계과제를 선택한다.
- (5) 설계과제 진행 전에 제안서를 작성하여 설계과제 진행계획의 타당성을 지도교수에게 반드시 승인을 받을 것.
- (6) 가능한 한 창의적인 새로운 아이디어를 시스템에 적용할 것.
- (7) 종합설계2와 연계가능하거나 혹은 종합설계2에서의 설계과제에 필요한 과제를 선택해도

무방하다.

5. 과제 수행 방법

기본적인 수행 순서는 아래와 같이 하도록 할 것이며 필요에 따라 수정 가능하다. 반드시 아이디어와 기능 명세서는 팀원이 모두 참석한 회의에서 확정할 것.

- (1) 팀 구성
- (2) 아이디어 도출 및 제안서 초안 작성
- (3) 기능명세서 작성
- (4) 개발 일정 및 역할 분담
- (5) 개념 설계
- (6) 세부 기능 설계
- (7) 데이터 구조 설계
- (8) 설계 내용 구현 (중간보고서 작성)
- (9) 문제점 보완 및 설계수정
- (10) 보완된 내용 구현
- (11) 테스트 방안 도출
- (12) 테스트
- (13) 최종보고서 작성

6. 과제평가방법

종합설계의 점수는 다음과 같이 리포트작성, 프로그램 동작시연 및 발표, 창의성, 역할분배 및 협동성에 따라 배분된 점수를 더하는 것이며, 100점을 만점으로 한다.

실무중심산학협력프로젝트 (캡스톤디자인) 채점표 (100점 만점)

(1) 리포트 작성	35점
제공된 템플릿에서 필요한 부분을 잘 취사선택해서 체계적으로 작성하였나?	5점
추론으로 전체적인 내용을 한 번에 알 수 있게 작성하였나?	3점
Introduction에 필요한 내용들을 잘 작성하였나?	5점
본론내용을 구체적으로 잘 구성하였나?	5점
프로그램의 설계 및 구성이 잘되었나?	10점
결과, 토론 및 결론 부분을 어떻게 구성하였나?	5점
전체적인 보고서의 스타일과 가독성이 뛰어나게 작성되었나?	2점
(2) 프로그램 동작 및 시연 혹은 최종보고서 발표	40점
초기에 계획했던 기능들이 잘 구현되었나, 혹은 설계보고서에 설계가 명확이 되었나?	15점
프로젝트에 대해서 간략 명료하게 잘 설명하였는가?	5점
데이터 구조 및 알고리즘이 어떻게 잘 적용이 되었나?	10점
제안한 기능들을 최종설계보고서에 상세하게 설계되거나 실제 구현되었나?	5점
(3) 프로젝트 창의성	10점
비슷한 프로젝트가 없고 새로운 기능이나 서비스 혹은 알고리즘을 제시하였는가?	10점
(4) 개인별 역할 점수	10점
개별인원에게 질문을 통해서 각자의 역할과 과제 프로젝트에서의 역할의 중요성 및 기여도 평가	10점
(5) 협동성 점수	5점
프로젝트 전반과 시연 시에 설명하는 상황에 따라 채점. 모든 멤버가 전체적인 개발상황에 대한 이해가 필수	5점

7. 설계보고서에 포함되어야 할 사항

실무중심산학협력프로젝트(캡스톤디자인-SW)에서는 기본적으로 3가지의 보고서를 제출한다. 먼저 강의 1-3주차에 제출하게 되는 과제제안서, 중간시험기간에 제출하는 중간보고서, 그리고 기말에 최종 제출하게 되는 최종 설계보고서이다. 중간보고서는 최종설계보고서의 준비과정이며 생략 될 수 있다. 각 보고서에서의 배점은 제안서(10%), 중간보고서(5%), 최종보고서(85%)이며, 각 보고서에 대한 형식은 아래에 자세히 설명하였다.

(1) 과제제안서

과제 제안서는 ppt파일로 10장내외의 슬라이드로 작성한다.

(2) 중간보고서

중간보고서는 제안서의 내용에 따라 중간시점까지 진행한 내용을 자유형식으로 설명하는 것이다. 아래 제시되는 최종설계보고서의 양식을 따르면, 이를 보완하여 학기말에 최종설계보고서를 쉽게 완성할 수 있다.

(3) 최종설계보고서

기본적인 최종설계보고서의 구성은 아래와 같으며, 설계과제의 상황에 따라 어느 정도는 수정 가능함. 5가지 항목이상을 충족하지 못하는 경우에는 반드시 담당교수와 협의할 것.

혹은 컴퓨터 과학 분야의 일반적인 학위논문의 구성방식에 따라 설계보고서를 작성하여도 무방하다.

최종 설계보고서

Authors (작성자) :

1. Title (제목)

프로젝트의 내용을 함축적으로 나타낼 수 있는 한 문장. (ex. AVL 트리를 활용하여 Particle Filter의 성능을 향상한 로봇 미로 탐색기)

2. Abstract (초고)

전체적인 보고서의 내용을 한 두 문단으로 요약하여 기술할 것

3. Introduction & Background

이 프로그램을 개발하게 된 배경. 예를 들어 기존의 다른 프로그램이 존재했는데 특정기능을 개선 혹은 추가한 배경. 혹은 기존의 어떤 프로그램이 있었는데 그것을 분석해보니 이러한 방식으로 구현되었을 것으로 예상되는 비슷하지만 조금 다른 방식으로 개발했다. 마지막 문단에는 이런 프로그램을 개발하기 위해 누가 무엇을 개발하고 누가 무엇을 개발했더라고 할 것.

4. Design & Method

먼저 전체적인 프로그램의 목적 및 디자인에 대해서 간략하게 서술한다. 전체적인 구성을 블록다이어그램 같은 걸로 전체적인 시스템에서의 프로그램의 기능을 설명하는 방법을 추천함.

4.1 Requirement & Constraint	
4.1.1 Major Software Requirement	소프트웨어 설계 시 요구조건을 기술한다. 특정기능이라던가 특정상황에 맞는 것을 구체적으로 기술할 것
4.1.2 Design Constraint	
4.2 Data Design	
4.2.1 Data objects and Data Structures	
4.2.2 File and Database Structure	
4.3 Architectural Design : 프로그램의 구조를 설명한다. 그림이나 Flow chart 추천.	
4.3.1 Data and Control Flow	
4.3.2 Program Structure	
4.4 Interface Design	

4.5 Procedural Design : 각 모듈 혹은 클래스에 대하여 interface 혹은 연동되는 방식을 상세 기술할 것.

5. Discussion & Conclusion

다음의 기본적인 내용을 포함하도록 한다.

- 설계 배경 (Background) 혹은 설계 소개 (Introduction)
- 제안 아이디어 혹은 RFP (request for proposal)
- 기본적인 시스템 설계 및 아이디어 세부 설명
- 개발환경
- 설계 일정
- 팀원별 역할