

# Capstone Design in CSE

## 컴퓨터공학 종합설계

### 1. Overview

# 강의 목표

1. 문제 정의, 요구조건 분석, 개념 설계, 상세 설계, 구현, 평가, 유지 등을 포함하는 소프트웨어 개발 방식의 이해
2. 각종 도구를 활용한 프로젝트 관리
3. 의사소통과 팀별 활동
  1. 팀원간 의사소통
  2. 체계적 문서화
  3. 결과에 대한 효과적인 발표

# 강의 개요

1. 컴퓨터공학 분야의 실용적 문제 선정
  - 브레인스토밍, 서베이, 토론 등을 통한 창의적 아이디어 도출
2. 해당 문제를 해결하기 위한 소프트웨어 시스템 설계
3. 해당 소프트웨어의 구현, 검증, 평가

# 강의 진행 방식

- 팀별 발표 및 토의
  - 팀 구성: 3인 1조
- 각 팀은 매주 또는 격주 발표 및 총 4회의 보고서 제출
  - 과제기획서, 제안서(상세요구사항명세 및 상세설계 포함), 중간 보고서, 최종보고서
- 최종 발표는 여러 분반을 통합하여 운영할 예정  
(presentation + live demonstration)
- 성적 평가: "절대평가"
  - 보고서: 60%
  - 발표 35%
  - 출석 10% (지각 2회=결석 1회)
  - 개별 학생별로 성적이 주어지므로, 팀원의 학점이 동일하지 않을 수 있음. 팀원들의 역할분담 명확히 할 것
  - 타 분반과 통합 발표 시행하며, 타 분반 교수로부터의 평가도 성적에 반영함

# 지난 수업에서의 학생들 의견 (최종보고서 발췌)

- 어렵다고 생각했던 프로젝트를 팀원들과 파트를 나누어 구현하다 보니 완성할 수 있었다. 팀원들 개인의 능력도 중요하지만 **가장 중요한 것은 협동**이라는 생각이 든다. 또한 학기 초에 세우는 계획서가 정말 중요함을 느꼈다. **계획을 2주 단위로 세세하게 짜 두니 프로젝트가 차질 없이 잘 진행됐다. 생각보다 시간이 빨리 지나가기 때문에 여유있게 계획을 세우는 것이 좋다.** 무리한 계획을 세우지 않으려면 이전의 프로젝트 계획서를 참고하는 것이 도움이 될 것 같다. 모르는 분야를 한 학기동안 배워가며 프로젝트를 하는 것은 많은 노력이 필요하다.
- 협업과 커뮤니케이션이 중요하다는 것을 알았다. 처음엔 팀프로젝트라 부담스러웠지만 각자 자신에게 주어진 목표를 수행하고 확인하니 하나의 어플이 개발이 되어있었다. 중간에 초기 간트차트에 문제가 생겨서 대거 변경을 하였는데, 그로 인해 계획의 소중함을 알게 되었다. 캡스톤만 하는 것이 아니라 다른 전공 수업도 들으면서 하니 생각한 것 만큼 시간을 많이 못 써서 아쉽지만 만족스러운 결과물이 나와서 보람있었다. **캡스톤 한 과목만 듣는 것이 아니기 때문에 중간고사 전까지 최대한 완성도를 높여놔야 한다.** 어떤 것을 할지는 모르지만 웹, 앱 등 기본적인 것이라도 배워두는 것이 본인이나 팀원들에게도 도움이 될 거라 생각한다.
- **팀원과의 의사소통이 가장 중요하다는 것을 알게 되었다. 자세하게 기능과 디자인에 대해 상의하고 만들지 않으면, 서로 미묘하게 다른 결과물을 만든다는 것을** 알게 되었다. 충분한 상의와 의견 교환 후에 결과물을 만드는 것이 오히려 더 개발 시간을 단축함을 알게 되었다. 주제를 잡게 되면, 빠르게 주제가 적합한지에 대해 확인해 보는 것이 좋다. **저작권이나 법적 문제, 이미 기존 어플에 있는 문제 등 주제 자체에 문제가 있을 수 있기에 구현 전에 최대한 알아보는 것을 권장한다.**

- **프로젝트를 진행하면서 팀 내에서 한 명이 제 몫을 해내지 못하면 팀 전체가 어려워진다**는 것을 피부로 느낄 수 있었습니다. 맡았던 많은 부분에서 부족했고 특히 연락이 끊기거나, 일정을 맞추지 못해서 개발 일정이 밀렸습니다. 개인적인 사정이나 다른 어떤 사유가 있었더라도 '팀'으로써 맡은 바 책임을 다하는 것이 얼마나 중요한지 알 수 있었던 프로젝트였다고 생각합니다.
- 개발에 있어서 거의 처음으로 해보는 안드로이드 개발이었는데 프로젝트가 끝날 즈음에 적어도 기초 중의 기초 정도를 할 수 있게 되었다고 느낄 수 있었습니다. 더불어 **알고리즘을 구상하는 것과 구상한 알고리즘을 실제 코드로 구현하는 것은 정말 큰 차이가 있다는** 것을 느꼈습니다. 학과 수업 중에 받았던 과제를 수행하거나 한 가지 기능을 집중적으로 구현하는 것이 아닌 하나의 어플리케이션을 구현하는 것은 정말 다른 문제이구나 하는 생각도 들었습니다.
- 개발 과정에 있어서 가장 커다란 힘이 되었던 것은 팀원들이었다고 자신 있게 말할 수 있습니다. 또한 팀원들과 소통하면서 기획하고 피드백을 진행하는 모든 부분이 포기하고 싶었던 마음을 다잡을 수 있었습니다. 프로젝트의 마무리에서 느낀 점을 빌어 팀원들에게 다시 한번 감사하다는 말을 전하고 싶습니다.
- 다음 수강생들을 위해 꼭 필요하다고 느꼈고 제안하고 싶은 부분은 **팀 내의 피드백 과정을 약식 보고서 양식으로 제출하는 것**입니다. 물론 귀찮을 수 있고, 힘든 과정이겠지만 과제의 형식을 빌어서라도 팀원들 간에 피드백과 대화가 중요하다는 것을 전해주고 싶습니다.

- -----를 다루기에, 초반에 API를 구하며 **저작권 문제 때문에 시간을 2주정도 아무것도 못하고 보내 버린** 게 너무 아쉽다.
- 초반에 기획했던 기능들은 무조건 구현하자 란 생각으로 임했기에, 초반 기획을 정말 잘해야 한다고 생각한다.
- 매주 2회씩 2시간 회의를 의무적으로 했고, 이도 모자라 수없이 회의를 했다. 팀원들 전부 정말 고생했고 다들 열심히 했다.
- 단 한 명도 빠지면 안 될 정도로 필요한 기능들을 각자 구현해 냈다.
- 부분등록으로 실 수강 하는 수업이 이 수업 뿐 임에도 불구하고 거의 모든 것을 쏟아 부어도 모자랐던 한학기가 되었습니다. 밤을 새가며 개발을 하느날도 많았고 기초조차 모르던 안드로이드를 공부 하기 위해 무작정 책을 사서 그냥 읽고 따라 치고 주먹구구식으로 한권을 공부 했습니다. 지속적으로 발표와 발표 자료 또한 준비해야 하는 것이 정말 학교 4년 통틀어 제일 힘들지만 알찬 수업이었습니다. 우리 과 특성상 이렇게 본격적으로 팀플 형태로 수업을 진행하는 것이 거의 없기 때문에 무척이나 신선한 경험이었었고 회사에 들어 간 이후에 협업을 한다면 이런 느낌이 아닐까라는 생각이 들었습니다.
- 다음 학기 수강생들을 위하여 한 말씀 드린다면, **주제나 분야 정도는 미리 정해와서 미리 공부하는 걸 추천** 드리고 싶습니다. 우리 팀의 경우 주제 정하고 처음인 안드로이드를 공부 하면서 개발을 하느라 초반에 시간을 많이 소모 했습니다. 그 시간을 좀 더 유용하게 쓸 수 있었다면 조금 더 좋은 결과 가 나오지 않을까 싶습니다.

- 팀원들 모두 팀 프로젝트에 익숙하지 않아 **소통의 어려움**이 있었음, 다음 팀 프로젝트를 진행한다면 **작업을 시작하기 전 스케줄링, 코드 공유, 의견 공유를 할 수 있는 방법을 의논하고 시작**하는 것이 더 효율적일 것이라고 생각됨
- 어플 개발을 하면서 라이브러리 및 api 활용에서 버전에러가 많이 발생해서 어려웠었다.
- 구직활동을 병행하면서 진행했던 프로젝트여서 시간이 모자라서 아쉬웠다.
- 사용해보지 않았던 -----를 활용한 프로그래밍이 낯설어서 어려웠었다.
- 처음에 계획했던 감정분석 ----이 심전도 기능 활용 불가로 행동패턴 분석 ---이 되었다. **사전조사를 할 때 조금 더 확실하게 조사하는 것이 중요**하다
- 학사 과정 동안 진행한 프로젝트 중에서 가장 도전적이었다. 단순히 하나의 프레임워크를 이용하는 수준이 아니라 클라이언트, 서버, AI 파트 각각 다른 기술 스택을 이용해 상호 작용한다는 게 의미 있었다. 이러한 경험을 통해 더 나은 개발자로 성장할 수 있었다고 확신한다. 또한 다양한 도메인을 가진 팀원들과 **Github를 이용하여 프로젝트를 진행**하면서 체계적인 프로그래밍 능력을 기를 수 있었다.
- 프로젝트를 진행하는 동안 이슈도 많이 발생했는데, 수업시간을 통해 배워왔던 지식을 바탕으로 차근차근 해결해나갈 수 있었다. 쌓아온 전공 지식들을 활용하여 새로운 언어를 학습하고, 실용적인 문제를 해결하며, 사용자 친화적인 서비스를 제공하기 위해 노력했던 경험은 앞으로의 다른 프로젝트에도 큰 도움이 될 것 이라 생각한다.



# 강의 진행 주요 일정

주차	내용	과제
1	Overview - 강의 개요, 이전 프로젝트 소개	기업 제안 과제 확인 후 희망 과제 리스트 제출
2	Brainstorming, 주제 선정 - Team discussion on candidate problems - Brainstorming / requirement gathering - 팀 구성	2주차: 과제기획서
3-4	Proposal 발표 (팀별)	4주차: 팀 proposal (제안서)
5-6	설계 완료 및 구현 시작 - 진행사항 발표 (팀별)	
7-14	구현 단계 - 진행사항 발표 (팀별)	10주: 중간보고서
15	완성 및 데모 단계 - 분반 내 최종발표 및 분반통합 발표	
16		16주차: 최종보고서