

스터디 그룹 핵심 기능

현 상황 : 스터디를 구하려면 현재는 에타나 다른 커뮤니티에서 구해야하고, 주변에 있는 사람들과 함께 하기 힘들다.

학교단위로 스터디를 구할 수 있음.

기본기능은 주변사람들과 자격증, 학습 목적의 스터디를 구하는 기능

슬랙 같은 느낌의 단톡방을 제공

추가적으로 출석체크, 목표설정, 파일 정리 및 공유 폴더, 화상회의 등의 기능들을 추가 제공

인증샷 시스템을 통해 강제성 유도

메인이 될 만한 기능은 딱히 없는듯..

명세서에 적을 만 한 핵심 기능

튜터링

1. 형식적인 튜터링 감독 및 성실도 측정 + 보상

튜터링 때마다 어떤 것을 공부하였는지 인증샷, 보고서를 얼마나 성의있게 제출했는지 등으로 판단. But 어떻게 보면 형식적인 튜터링을 단속하기는 쉽지 않을 듯. 튜터 입장에서는 귀찮은 일만 많아질 뿐일 수도 있음. 그렇기에 튜티들의 평가참여 및 진실된 설문조사가 중요할듯.

보상 측면에서는 우수 팀 선정 및 성실도 등으로 시상하기

실시간 협업 문제 풀이는 온라인인 경우가 대다수. 줌의 화면 공유 + 메모기능 처럼 다같이 한화면을 보고 튜터링을 할 수 있도록. 또한 문제를 풀어보는 시간에는 각자 어떻게 푸는지 튜터가 확인 가능.

교과목의 교과과정 및 진도 계획서 등을 제공하고, 어떻게 튜터링을 하고 어떤 것을 중점적으로 할지 설정 가능

인터랙티브한 지원 도구는 줌 화상회의, 공유드라이브, 시간표 등록을 통해 튜터링 시간 설정 가

능

편의성 측면에서는 활동시간을 자동으로 계산하고 얼마나 남았는지 확인가능, 장소 추천 및 장소 예약 가능, 튜터들의 커리큘럼이나 활동계획을 좀 더 체계적으로 잡을 수 있게 형식 제공 및 자동 추천

튜티들을 위한 항목들은, 튜터링 Roadmap을 제공하고, 잘 모르는 학생들을 위해 선배들은 어떤 수업을 듣고, 어떤 로드맵을 따라가서 어떤 분야로 취직하는지도 소개해줌. 또한 평점 부여, 한줄 평 남기기로 서로에게 선순환 및 이상한 튜터들 정리가능

IC-PBL

이미 개발되고 있는 내용은 제외.

오프라인 지원 가능한 프로그램은 잘 모르겠음. 오프라인 PBL을 해본적이 없음. + 인터랙티브도 홀로그램 이나 상호작용이 따로 없는듯.

마인드 맵 작성을 통해 학생들이 말한 키워드나 회의, 공부한 것들을 한눈에 볼 수 있게 정리해 두고, 회의록 자동작성, 녹음 녹화등을 제공

포트폴리오

포트폴리오 내역에 따라 어떤 활동이 부족하고, 보충하면 좋을 맞춤형 정보 제공

취업활동에 유리하게 사용할 수 있게 기본 템플릿 및 작성할 수 있도록 가이드라인, 추가정보를 제공

요구사항 명세서 템플릿

1. 전반적인 개요

목적, 범위, 용어정의 (정의 등), 참조

2. 목차

주 기능

소프트웨어의 동작, 디자인

사용자의 특성

소프트웨어의 제약사항 및 의존성

3. 상세 정보 – 사진 참조

추가정보 :

외부 인터페이스 요구사항 – 사용자 구분

기능 요구사항 – 중요도, 어느화면에 띄울 것인지?

항목	설명
외부 인터페이스 요구사항 (External Interface Requirement)	<p>모든 소프트웨어 시스템으로의 입력과 출력에 대한 요구사항을 상세히 기술한다. 사용자 인터페이스, 하드웨어 인터페이스, 소프트웨어 인터페이스, 통신 인터페이스 등으로 분류될 수 있으며, 다음과 같은 형식의 내용이 포함되어야 한다.</p> <ul style="list-style-type: none"> - 항목의 이름 (Name of item) - 목적 상세 설명 (Description of purpose) - 입력의 시작점 및 출력의 도착점 (Source of input or destination of output) - 유효 범위, 정확도, 오차 (Valid range, accuracy, and/or tolerance) - 측정 단위 (Units of measure) - 시간 (Timing) - 다른 입력 및 출력과의 관계 (Relationships to other inputs/outputs) - Screen formats/organization; - Window formats/organization; - 데이터 형식 (Data formats) - 명령 형식 (Command formats) - 종료 메시지 (End message)
기능 요구사항 (Functional Requirement)	<p>소프트웨어의 입력 처리와 출력을 생성하는 처리 과정에서 발생할 수 있는 기본적인 동작에 대하여 기술하며, '시스템은 ~해야 한다' 형식으로 작성되어야 하며 아래와 같은 항목들이 포함된다.</p> <ul style="list-style-type: none"> - 입력의 유효성 확인 (Validity checks on the inputs) - 동작의 정확한 흐름 (Exact sequence of operations) - 비정상 상황에 대한 동작 (Responses to abnormal situations, including, Overflow, Communication facilities, Error handling and recovery) - 파라미터의 영향 (Effect of parameters) - 출력에서 입력까지의 관계 (Relationship of outputs to inputs, including Input/output sequences, Formulas for input to output conversion)
성능 요구사항 (Performance Requirement)	<p>소프트웨어 전체적으로 사람과의 상호작용 혹은 소프트웨어에서 확인할 수 있는 정적이고 동적인 수치적 요구사항을 기술한다.</p> <ul style="list-style-type: none"> - 정적인 수치적 요구사항(static numerical requirements): 소프트웨어에서 다루어지고 있는 정보의 양이나 타임과 같은 고정적인 수치적 요구사항 - 동적인 수치적 요구사항(dynamic numerical requirements): 정해진 시간이 아니라 일정 시간 안에 수행해야 하는 것처럼 정해진 수치가 없는 요구사항
논리적 데이터베이스 요구사항 (Logical Database Requirement)	<p>데이터 베이스에서 사용될 정보를 위한 논리적 요구사항에 대하여 기술하며, 다음과 같은 항목을 포함한다.</p> <ul style="list-style-type: none"> - 여러 기능에서 사용될 정보의 타입 (Types of information used by various functions) - 사용 빈도 (Frequency of use) - 접근 가능성 (Accessing capabilities) - 데이터 엔티티 및 관계 (Data entities and their relationships) - 무결성 제약조건 (Integrity constraints) - 데이터 유지 요구사항 (Data retention requirements.)
설계 제약사항 (Design Constraints)	<p>다른 표준이나 하드웨어적 제한으로 인해 적용되는 설계적 제한사항에 대하여 기술한다.</p>
소프트웨어 시스템 속성 (Software System Attribute)	<p>신뢰도(Reliability), 사용가능성(Availability), 보안(Security), 유지보수(Maintainability), 이식성(Portability) 등의 소프트웨어 속성을 요구사항으로서 기술한다.</p>

Aa 항목	≡ 설명
ID	내부 규칙에 따라 식별자를 부여한다. 하나의 요구사항에 하나의 식별자이다.
화면명	어느 화면에서 구현할 기능인지 기재한다. 화면에 속하지 않는 요구사항도 있을 수 있다.
요구사항명	요구사항의 설명을 요약하여 기재한다.
내용	요구사항의 상세한 내용을 기재한다.
중요도	상중하, 1~5 등 내부 규칙에 따라 부여한다.
부서 / 작성자	요구사항을 기재한 담당자를 기재한다. 부서가 없는 경우 생략해도 무방하다.
날짜	요구사항을 기재한 날짜를 명시한다.
진행사항 (구현 여부)	검토 예정, 진행 확정, 진행 불가, 추후 진행 등 내부에서 결정된 사항을 기재한다. 진행 불가이거나 추후 진행일 경우 이유를 같이 적는 것이 좋다.
버전명	요구사항이 변경될 수 있으므로 버전(혹은 다른 방식)으로 표기하여 타인이 알 수 있게 해야 한다.
그 외	유형(기능, 비기능, 제약사항, 인터페이스, 기타 등) ,출처(관려자 이름, 11/15 회의, 사업계획서 등)

소프트웨어 요구사항 명세서의 양식을 제시하고 있는 표준인 IEEE Std. 830(IEEE Std. 830-1998: IEEE Recommended Practice for Software Requirements Specifications)에 의하면, 요구사항 명세서는 다음 그림과 같은 기본적인 양식을 갖는다.

Table of Contents
1. Introduction
1.1 Purpose
1.2 Scope
1.3 Definitions, acronyms, and abbreviations
1.4 References
1.5 Overview
2. Overall description
2.1 Product perspective
2.2 Product functions
2.3 User characteristics
2.4 Constraints
2.5 Assumptions and dependencies
3. Specific requirements (See 5.3.1 through 5.3.8 for explanations of possible specific requirements. See also Annex A for several different ways of organizing this section of the SRS.)
Appendixes
Index

Figure 1—Prototype SRS outline

<그림1.IEEE Std.830의 소프트웨어 요구사항 명세서 양식>

1. Introduction'에는 문서의 목적과 범위, 용어 정의(정의, 두문자어, 약어), 참조 등 문서의 전반적인 개요에 대한 내용이 작성된다.

2. Overall Description'에는 제품 관점에서의 소프트웨어 동작이나 소프트웨어의 주 기능, 경험이나 전문 지식을 포함한 소프트웨어 사용자의 일반적인 특성, 소프트웨어의 제약사항과 가정 및 의존성 등에 대한 내용이 기술된다. 해당 절의 경우, 명확한 요구사항을 작성하기 보다는 요구사항에 대한 전반적인 내용이 기술되는데, 요구사항을 상세히 작성하는 '3. Specific requirement'의 이해를 돕기 위한 배경과 같은 내용이 포함된다.

3. Specific Requirement'에는 소프트웨어의 모든 세부적인 요구사항을 명확하게 기술하며, 명시되어야 하는 요구사항은 아래 표와 같다.

