

Business Requirements Document

<구성원과 효율적인 통신을 할 수 있는 이클래스 시스템>

AUTHORS

Name	Role	Department
정진환	조장	컴퓨터 전자 시스템 공학부
이다혜	팀원	전자 물리학과
배문혁	팀원	컴퓨터 전자 시스템 공학부
박정민	팀원	컴퓨터 전자 시스템 공학부

Contents

1. Introduction

1.1. Purpose of the Document

1.2. Scope

2. Overall Description

2.1. Product Features

2.2. User Requirements 도식도

2.3. User Classes and Characteristics

2.4. Design and Implementation Constraints

2.5. Assumptions, Dependencies, and Risks

3. System Features

3.1. <사용자 등록 (Register)>

3.2. <사용자 로그인>

3.3. <사용자 로그아웃>

3.4. <메인 메뉴 화면>

3.5. <클래스 메뉴 화면>

3.6. <게시판 작성 (Write)>

3.7. <게시판 읽기 (Read)>

3.8. <그룹 채팅 시스템>

3.9. <채팅 비속어 Filtering>

3.10. <채팅 이미지 첨부>

Introduction

Purpose of the Document

기존의 E-Class 시스템에서 클래스 내의 구성원들에 대한 소통 수단이 미비하다는 단점을 보완할 수단으로 채팅 시스템을 도입하기 위한 요구사항을 정리한 문서이다.

Scope

기존 E-Class 시스템의 사용자 로그인 관리 정보 관리, 클래스 별 구성원들의 문서, 메일 등의 통신과 함께 클래스 내 구성원들이 참여할 수 있는 실시간 채팅 시스템을 접목시킴으로써 구성원 간의 통신이 메일, 게시판 등으로 한정된 기존 시스템의 단점을 해소할 수 있을 것으로 기대해볼 수 있다.

Overall Description

1. Product Features

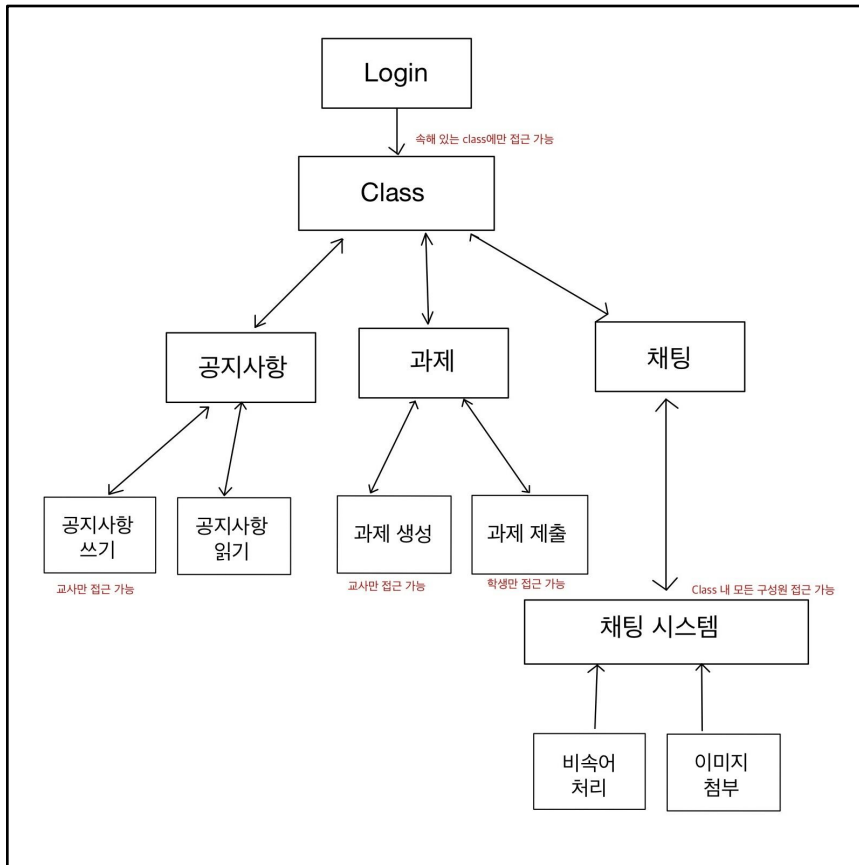
Class 관리

1. 사용자 접근 권한 관리
(ex. 사용자가 속해있는 클래스에만 접근 가능)
2. 사용자 로그인 정보 관리
3. 클래스 내 문서, 메일, 사용자 등을 구조적으로 관리
(ex. 클래스 내 사용자 정보, 문서 등을 Tree형태로 구성해 관리)

Class내 채팅 시스템

1. 클래스 내 구성원이 참여할 수 있는 실시간 그룹 채팅 시스템
2. 채팅 비속어 처리 기능
3. 채팅 내 사진 첨부 기능

2. User Requirements 도식도



3. User Classes and Characteristics

Potential Stakeholders

1. 선생님, 교사 등의 교육자들
2. 클래스에 참여하는 학생들
3. 조직 내 구성원들을 관리하는 Managers
4. 조직 내 구성원들

4. Design and Implementation Constraints

사용자 정보가 유실되거나 아무 이유 없이 바뀌어서는 안된다. 오직 교사 신분의 사용자만이 공지사항이나 과제란에 글을 올리거나 수정할 수 있다. 학생 신분의 사용자는 해당 글에 코멘트만 작성할 수 있다.

5. Assumptions, Dependencies, and Risks

사용자 정보 (학적, ID, PASSWORD등), 채팅 내역 정보 저장을 위한 SERVER사용 필요

이때, 여러 사용자가 채팅 내역을 읽고 쓰기 때문에 MUTEX기법이 필요함. (DBMS의 트랜잭션으로 해결 가능), Server에 미리 Class에 대한 정보 (Data Structure)가 저장되어 있어야 한다.

System Features

1. <사용자 등록 (Register)>

Use Case Name	사용자 등록 (Register)
Description	server에 사용자를 추가시킨다.
inputs	id, password, 이름, 소속, 학번(교번), 신분(학생/교사), 현재 소속된 Class들
Preconditions	inputs에 똑같은 정보가 server에 이미 있으면 안된다. (reject해야함)
Postconditions	입력된 사용자 data structure가 server에 저장된다.
Normal Flow	각 inputs에 대한 입력 칸이 주어지고 이에 대해 입력이 완료되어 확인되면 사용자 data structure가 형성되고 이것이 server에 저장되어진다.
Alternative Flow	inputs가 server에 이미 있는 내용이거나 공백이면 불가하다고 사용자에게 알리고 다시 입력받는다.
Exceptions	server의 내용과 inputs가 중복, inputs가 공백
Outputs	사용자 Data Structure (Server에 저장될 정보)

2. <사용자 로그인>

Use Case Name	사용자 로그인(Login)
Description	사용자를 system에 로그인 시킨다.
inputs	id, password
Preconditions	server와 연결되어있어야 한다.
Postconditions	사용자가 소속된 class에 대한 메뉴를 보여준다.
Normal Flow	id, password를 입력받으면 server에 같은 id, password가 있는지 확인하고 있다면 login 시킨다. (사용자 Data에 있는 사용자가 속해있는 class정보에 따른 Main Menu를 실행)
Alternative Flow	만약 id, password 하나라도 잘못되면(혹은 둘

	어온 입력 정보없이 확인한다면) 입력 정보가 틀렸다고 사용자에게 알린다.
Exceptions	id, password 둘 중 하나라도 server에 없는 (혹은 맞지 않는) 경우
Outputs	없음

3. <사용자 로그 아웃>

Use Case Name	사용자 로그 아웃 (Logout)
Description	사용자와 system간의 연결을 끊고 종료시킨다.
inputs	없음
Preconditions	사용자가 system에 로그 인 되어 있어야 한다.
Postconditions	사용자와 system간의 연결을 끊는다.
Normal Flow	사용자가 logout을 선택하면 system과의 연결을 끊고 프로그램을 종료시킨다.
Alternative Flow	
Exceptions	
Outputs	없음

4. <메인 메뉴 화면>

Use Case Name	메인 메뉴 화면 (Main Menu)
Description	사용자에게 메뉴 화면을 띄운다.
inputs	server에 저장된 사용자 data structure, 사용자가 속해있는 class의 data structure, 사용자가 선택한 메뉴
Preconditions	사용자는 로그인되어 system에 들어와 있어야 한다.
Postconditions	사용자가 선택한 메뉴에 대한 기능을 수행한다.
Normal Flow	사용자가 로그인되어 들어오면 사용자가 속해있는 class들에 대한 메뉴, 로그 아웃 메뉴를 보여주고 사용자가 해당 메뉴를 선택하면 그에 맞는 기능(클래스 메뉴 화면, 로그 아웃)을 수행한다.
Alternative Flow	
Exceptions	

Outputs	없음
---------	----

5. <클래스 메뉴 화면>

Use Case Name	클래스 메뉴 (Class Manu)
Description	사용자가 선택한 클래스에 대한 정보 (공지사항, 과제란, 클래스 그룹 채팅, 클래스 나가기)를 보여 주고, 각 메뉴에 대한 기능을 수행한다.
inputs	server에 저장된 사용자가 속한 (선택한) class 의 data structure
Preconditions	사용자가 메인 메뉴 화면에서 해당 class 메뉴를 선택한 상황이다.
Postconditions	사용자가 선택한 해당 메뉴 (공지사항, 과제란, 클래스 그룹 채팅, 클래스 나가기) 기능을 수행한다.
Normal Flow	사용자에게 해당 클래스에서 할 수 있는 기능들 (공지사항, 과제란, 클래스 그룹 채팅, 클래스 나가기)을 보여주고 사용자가 그 중 하나를 선택하면 해당 기능을 수행한다.
Alternative Flow	
Exceptions	
Outputs	없음

6. <게시판 작성 (Write)>

Use Case Name	게시판 작성 (Write)
Description	사용자가 공지사항, 또는 과제란에 글을 올릴 수 있게 한다.
inputs	사용자가 입력한 글
Preconditions	
Postconditions	사용자가 입력한 글이 Server에 저장된 Class 의 Data Structure에 업데이트된다.
Normal Flow	사용자가 글을 작성하면 이를 입력으로 받고 글을 다 쓰면 Class내의 모든 인원이 글을 볼 수 있도록 해당 Class DataStructure에 이 글을 업데이트한다.
Alternative Flow	아무 입력도 받지 않고 사용자가 등록하면 불가하

	다른 메시지를 띄운다.
Exceptions	input 공백
Outputs	없음

7. <게시판 읽기 (Read)>

Use Case Name	게시판 읽기 (Read)
Description	Class에 등록된 게시판 글 중 하나를 사용자가 선택해 읽게 해준다.
inputs	사용자가 선택한 등록된 게시판 글 중 하나의 index
Preconditions	
Postconditions	등록된 게시판의 글을 보여준다.
Normal Flow	사용자에게 등록된 게시판의 글(공지사항 혹은 과제란)들을 보여주고 (각 글은 index로 구분된다) 그 글 중 하나를 선택하면 이를 보여준다.
Alternative Flow	만약 등록된 글이 없다면 등록된 글이 없다는 알림을 사용자에게 보여준다.
Exceptions	
Outputs	없음

8. <그룹 채팅 시스템>

Use Case Name	클래스 그룹 채팅 시스템 (Class Group Chat)
Description	클래스에 속해있는 사용자들이 모두 참여 가능한 그룹 채팅방을 보여주고 자유롭게 글을 쓰고 읽게 한다. 언제 접속해도 이전의 채팅 내역을 다시 보여준다.
inputs	사용자 Data Structure, Class의 Data Structure, Server의 채팅 내역, 사용자의 글 (String)
Preconditions	메인 메뉴에서 클래스 그룹 채팅 메뉴를 사용자가 선택한 상황이다.
Postconditions	글을 올리면 Server에 저장된 채팅 내역이 업데이트된다

Normal Flow	메인 메뉴에서 클래스 그룹 채팅 메뉴를 선택하면 Server에 있는 채팅 내역을 보여준다. 사용자가 글을 올리면 채팅 내역이 업데이트되고 채팅에 참여하는 모든 사용자들에게 글이 보여진다.
Alternative Flow	Server에 연결이 끊어지면 그 상황을 사용자에게 알린다.
Exceptions	Server에 연결이 끊어짐
Outputs	업데이트된 채팅 내역

9. <채팅 비속어 Filtering>

Use Case Name	채팅 비속어 Filtering
Description	사용자가 채팅에 비속어를 섞어 쓰면 해당 단어를 Filtering한다.
inputs	사용자의 글 (String)
Preconditions	사용자가 채팅에 글을 쓴 상태이다.
Postconditions	비속어가 처리된 글이 채팅 내역으로 업데이트된다.
Normal Flow	사용자가 채팅에 글을 쓰면 그 글에 비속어가 있는지 확인하고 만약 있다면 비속어가 안보이게 처리한 뒤, 그 글을 채팅 내역으로 업데이트한다.
Alternative Flow	
Exceptions	없음
Outputs	비속어가 처리된 사용자의 글

10. <채팅 이미지 첨부>

Use Case Name	채팅 이미지 첨부
Description	사용자가 채팅에 이미지를 첨부하게 한다.
inputs	이미지 파일 (jpeg, png, gif)
Preconditions	사용자가 채팅 시스템에 참여한 상태이다.
Postconditions	채팅창에 이미지 파일이 업로드되고, 채팅 내역이 업데이트된다.
Normal Flow	사용자가 채팅에 이미지 파일을 업로드하면 Server에 이미지 파일이 저장되고 이 이미지를

	채팅 내역에 포함시켜 보여주게 한다.
Alternative Flow	Server에 이미지 저장이 실패하면 이에 대해 사용자에게 업로드 실패 메시지를 전달한다.
Exceptions	없음
Outputs	업데이트된 채팅 내역