COM

(Club On MMU)

2023년 12월 13일

팀명: PHKJ 팀

팀원: 박성국(팀장), 한태민, 김정현, 정유성, 김태한

텍스트, 폰트, 로고, 상징이(가) 표시된 사진

자동 생성된 설명

**1. 개요**

1.1 개발 배경

사실, 처음 목표는 게임이였으나, 교수님이 게임은 이미 나와있어서 다른 거대한 프로젝트를 한번 해보라고 하였다. 그래도 다들 3학년이기도 하고 이제 다들 열심히 하겠지라는 생각으로 팀원들에게 여러가지 아이디를 제안했고 그 중에 하나가 동아리 신청하는 사이트를 간단하게 만들자는 것이였다.(~~하지만 이 결정이 매우 큰 실수였음을 지금 깨달았다.~~)

그리고 현재 동아리가 어떻게 신청이 되는지 조사를 시작하였다. 큰 대학부터 시작해서 각종 커뮤니티를 조사한 결과, 다음과 같이 결론 내릴 수 있었다.

1. 여러 대학교의 동아리 신청 방식 중 대다수는 SNS, 학생의 전화를 이용해서 신청을 받고 있다
2. 동아리 신청, 개설에 필요한 서류가 무엇인지, 해당 동아리에서 무슨 활동을 하는지 관련된 사람이 직접 설명하지 않는 한, 알 수 없다
3. 여러 대학교의 동아리 사이트를 둘러봤지만, 대학 자체적으로 동아리를 운영하고 같이 소통할 수 있는 장소가 되는 사이트는 별로 없다
4. 우리 대학교만 보아도 학과 내에 어떤 동아리가 있는지 학생들이 모른다
5. 학생들의 자유 시간이 각각 다르기 때문에 직접 만날 수 없는 상황이 많이 벌어진다

또한 이런 문제들을 해결할 맨 처음 언어가 HTML5가 생각났었다. 그리고 이걸로 다들 결정하고 이걸로 문제를 해결하기 위해 역할 분담을 하고 정기적으로 만나는 시간을 정하고 각자 결과물이 어떻게 되는지 git으로 알려주고 카톡이나 디스코드, 대면으로 대화하였다.

(이것을 더 알고 싶다면 맨 마지막에 있는 느낀 점으로 이동하시길)

1.2 개발 목표

동아리를 신청할 수 있는 프로그램(또는 시스템)을 만들고 그 시스템이 일단 구축이 되면, 각종 기능들을 추가해서 완성도를 높여가는 진화적 프로세스를 채택하였다. (하지만 과제를 위한 문서를 제출하는 형식이 우리들의 형식과 너무 달라서 상당한 문제를 직면했었다.) 그리고 사람들이 각종 활동 인터넷으로 올려서 사람들이 각종 활동을 보고 자극을 받아서 더 많은 사람이 가입하고 활동할 수 있도록 하는 것을 목표로 하였다.

또한 개발 목표를 잡을 때, 우리 학교에 있는 문제를 조사해서 해당 문제를 해결하는 방식으로 접근하면, 다른 학교에서도 대부분의 문제를 해결할 수 있을 것 같다는 생각이 있었다. 그래서 우리 학교에 문제들이 무엇인지 발견하는 시간을 가졌었다. 아래는 우리 학교의 학습 상황 또는 문제들을 나열한 것이다.

발견한 우리 학교의 동아리 현황

1. 교수님들은 학생들이 모여서 어떤 활동을 하고 같이 공부할 수 있는 활동이 적다는 것을 알고 있다. ( 어쩌면 활동이 있는데 모르는 것일 수도 있다, 친목을 다질 수 있는 활동은 많을 지 모르겠지만… )
2. 위의 상황은 학생들이 같이 활동을 통해 더 큰 프로젝트를 할 수 있도록 하는 것과 상반되는 것이다.
3. 동아리를 신청할 수 있는 사이트나 프로그램이 없으며, 대부분은 포스터나 지인을 통해서 동아리에 유입되며, 각 동아리가 어떤 활동을 주로 하는지 알려진 것이 거의 없다.
4. 위의 이유들 때문에 대학생들이 우리가 만든 사이트에서 동아리 활동을 더 효율적으로 할 수 있고 시간적, 공간적 제약으로 같이 모일 수 없는 사람들이 모여서 이야기할 수 있는 공간을 만들자는 목적을 시작으로 이 프로젝트를 기획하게 되었다

**2. 기존 시스템**

조사한 것들을 살펴봤을 때, 기존에 존재하는 시스템은 시스템이라고 부를 수 없는 것들 이 대부분 이였다.

(그나마 캠퍼스 픽이 시스템 형태를 갖추었다.)

1. **텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

   자동 생성된 설명**목포해양대

위의 이미지는 목포해양대학교에서 동아리를 신청하는 방식을 설명한 것이다.  
([**https://www.mmu.ac.kr/main/board/301/27/read/67586**](https://www.mmu.ac.kr/main/board/301/27/read/67586))

1. **신청 방식**
   1. 종이로 직접 작성해서 해당 건물로 찾아가서 제출하는 방식
2. **보면서 느낀 것들**
   1. 너무 번거롭다( 많은 과정, 걸어야 함 )
3. 텍스트, 스크린샷, 소프트웨어, 웹 페이지이(가) 표시된 사진

   자동 생성된 설명캠퍼스픽

**위 이미지는 캠퍼스픽의 홈페이지 레이아웃을 보여준다**

**(**<https://www.campuspick.com/>**)**

1. **신청 방식**
   1. 여러 대학생들이 가입을 하고 동아리 신청 게시물을 올리면 해당 대학교 학생들이 동아리를 올리는 방식
2. **보면서 느낀 것들**
   1. 이 사이트에서 존재하는 대학교 커뮤니티가 별로 없어서 다른 대학생들은 구경만 해야 한다 ( 설령 가입을 한다고 해도 커뮤니티가 없다 )
   2. 유명한 대학교(고려대, 연세대 등)들의 동아리가 별로 없다 ( 이 소리는 사이트를 정말로 목적대로 이용하는 사람이 적다는 이야기이다 )
3. **텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

   자동 생성된 설명**세종대

위의 이미지는 세종대가 동아리를 신청을 인스타로 대신하는 방식이다.

(<https://find.sejongclubunion.com/clubs/show>)

1. **신청 방식**
   1. 동아리는 많지만 해당 동아리를 신청 하려면 인스타로 DM을 보내는 방식
2. **보면서 느낀 것들**
   1. 학교 학생들이 아닌 학생들이 게시물을 볼 수 있다 ( 관련 서류, 자세한 위치 등, 민감한 정보를 넣을 수 없다 )
   2. 어려운 관리 ( 인스타 게시물을 학생들이 한 명 한 명 씩 돌아가면서 수작업으로 관리해야 한다, 얼마나 많은 학생이 신청했는지 수작업으로 작업해야 한다 )
   3. SNS이므로 해당 동아리 DM 이외의 대화가 있어서 관리에 혼동이 될 수 있다
3. 텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

   자동 생성된 설명연세대

위의 이미지는 연세대가 동아리를 신청하는 방식을 나타낸 것이다.

(<https://dongyon.yonsei.ac.kr/dongyon/hanaul.do>)

* + - 1. **신청 방식**
         1. 전화번호만 던져놨다.

1. **보면서 느낀 것들**
   1. 동아리 관련 정보가 동아리 팀장들의 전화번호 밖에 없다
      1. ( 전화번호가 어디에 쓰일까? )

**3. 핵심 기술**

3 – 1. Python3의 Streamlit: 웹 디자인, 웹 페이지 기능 제공

설명: 기본적으로 모든 프레임은 파이썬의 streamlit으로 작성되었습니다. (원래 node.js, mysql 및 HTML5로 작성할 생각이였으나 배우지 않은 것(Node.js, HTML5, Mysql)에 대한 팀원들의 스트레스가 가중되는 것을 보았으므로 인터넷으로 서핑한 결과 streamlit이 그래도 난이도는 쉬우면서 파이썬으로 구동되기 때문에 이를 선택함, 하지만 이것도 하기 싫었는지, 알려준 임무에 대한 git을 아무도 하지 않음)

3 – 2. Goormide(원격 서버)에 있는 Mysql

설명: 무료 원격 서버입니다. 해당 서버에서 Mysql 서비스를 시작하고 각종 필요한 정보를 가져오면(서버주소, 포트번호, 사용자 명, 비밀번호) 웹이 요구하는 정보와 웹이 저장하기를 원하는 정보를 관리할 수 있습니다. (단, 무료이기 때문에 서버를 한번 꺼버리면 필수 정보를 코드에서 다시 재설정해야 다시 정보를 가져올 수 있습니다.)

3 – 3. Visual Studio Code

설명: 파이썬 코드를 작성하는데 필수적인 에디터들입니다. 오류를 잡아주는 기능은 별로 없었고 단지 내장된 기능들이 필요해서 사용했습니다. 기본적으로 팀장 단독으로 작성했다고 봐도 무방하기 때문에 많은 오류가 났었지만, 파이썬 자체 오류 검출 기능으로 잡았던 것들이 대부분이였습니다. (IDE가 오래 가만히 놔두면 컴파일 기능이 중지된 경우가 많았었음)

3 – 4. kate( text editor )

설명: Mysql코드를 작성하는데 필요한 에디터입니다. 눈을 편안하게 해주는 기능이 있고 오류를 잡아주는 기능은 없으며 goormide에 있는 Mysql에 있는 자체 오류 검출 기능을 이용했습니다.

3 – 5. drawSQL( <https://drawsql.app/> )

전에 SQL 구조를 그렸던 사이트( <https://aquerytool.com/> )는 ERD그리는 횟수가 5개로 제한 되어있어서 사이트를 바꾸었다. ERD의 개수가 제한이 없는 것이 매우 좋았다. 또한 나중에 DB를 그리고 바로 적용될 수 있도록 DB를 선택하고 해당 DB에 맞게 export할 수 있어서 써볼 생각이다.

**4. 요구사항 개발 현황**

**UC-1: 로그인 및 로그아웃**

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 ID | 요구사항 명칭 | 요구사항 설명 | 개발현황 |
| UC-1-Require1 | 로그인 조건 | 로그인을 할려면 반드시 데이터베이스에 해당 회원의 아이디와 비밀번호가 등록이 되어 있어야 한다 | O |
| UC-1-Require2 | 로그인 실패 | 로그인에 실패하였다면 로그인 실패 원인을 알려준다(일치하지 않거나 존재하지 않거나) | O |
| UC-1-Require3 | 로그인 실패 초과 | 여러 번 로그인이 실패한다면 제한 시간을 둔다 | X |

**UC-2: 회원가입 및 탈퇴**

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 ID | 요구사항 명칭 | 요구사항 설명 | 개발현황 |
| UC-2-Require1 | 처음 회원가입 및 상태 | 회원가입은 외부에 있는 어떤 사용자라도 가입할 수 있으며 처음 가입하는 경우 동아리는 null값이다 | O |
| UC-2-Require2 | 탈퇴 계정 정보 처리 | 탈퇴는 본인 계정 페이지에서 설정하게 하며 본인의 게시글, 댓글이 모두 삭제되도록 한다(단, 게시글은 그냥 삭제하되, 댓글은 삭제된 게시글입니다. 라고 표시) | X |
| UC-2-Require3 | 회원가입 제약사항 | 회원가입은 여러 사용자의 정보가 겹치지 않게 하며 반드시 데이터베이스에 기록해야 한다. | O |

**UC-3: 동아리 가입 및 탈퇴 신청**

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 ID | 요구사항 명칭 | 요구사항 설명 | 개발현황 |
| UC-3-Require1 | 가입 과정 | 가입은 신청을 먼저 해야 하며 관련된 서류가 있다면 제출하도록 한다(파일은 일단 로컬로 저장한다) | O |
| UC-3-Require2 | 탈퇴 과정 | 탈퇴도 동일하게 신청을 해야 하며 동아리 장이 신청을 받을 경우, 이를 처리하도록 한다 | X |
| UC-3-Require3 | 가입 및 탈퇴 관리 | 두 조건 모두 관리자 또는 동아리 장에 의해 거부되거나 승인될 수 있다. | X |

**UC-4: 동아리 개설 신청 및 폐쇄 신청**

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 ID | 요구사항 명칭 | 요구사항 설명 | 개발현황 |
| UC-4-Require1 | 동아리 개설 신청 | 동아리 개설은 어떤 회원도 신청할 수 있으며, 관리자의 승인이 있어야 한다(관리자는 동아리 신청을 승인 또는 거부할 수 있다) | O |
| UC-4-Require2 | 동아리 폐쇄 신청 | 동아리 폐쇄는 동아리 장만 신청할 수 있으며 관리자가 승인해야 한다. | X |
| UC-4-Require3 | 동아리 개설 서류 | 동아리 개설 신청에 필요한 서류를 제출해야 한다. | O |

**UC-5: 게시글 작성 및 삭제**

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 ID | 요구사항 명칭 | 요구사항 설명 | 우선순위 |
| UC-5-Require1 | 제한된 게시글 작성 | 게시글은 어떤 회원도 작성할 수 있다. 하지만 동아리 장 또는 관리자만 적을 수 있는 게시글은 일반 회원은 작성할 수 없도록 한다. | △ |
| UC-5-Require2 | 게시글 삭제 | 게시글 삭제는 작성자 본인이 할 수 있으며 다른 사용자의 게시물을 삭제할 수 없다. | O |
| UC-5-Require3 | 동아리 게시글 작성 | 현재 동아리에 속한 사용자 또는 아직 동아리에 가입하지 않은 사용자는 다른 동아리에 게시글을 작성할 수 없다.(단, 자유게시판 같이 동아리 제한이 없으면 작성 가능) | △ |

**UC-6: 댓글 작성 및 삭제**

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 ID | 요구사항 명칭 | 요구사항 설명 | 개발현황 |
| UC-6-Require1 | 댓글 글자 길이 제한 | 댓글은 어떤 회원도 작성할 수 있으며 댓글의 최대 글자 수는 1000자로 제한한다 | △ |
| UC-6-Require2 | 다른 동아리 게시글 댓글 금지 | 현재 동아리에 속한 사용자 또는 아직 동아리에 가입하지 않은 사용자는 다른 동아리에 댓글을 작성할 수 없도록 한다. (단, 자유게시판 같이 동아리 제한이 없는 경우 작성이 가능) | X |
| UC-6-Require3 | 삭제된 아이디의 댓글 | 댓글에 대한 사용자는 별명(아이디)로 표시하며 삭제된 아이디에 대한 정보는 표시하지 않는다. | X |

**UC-7: 개인정보 수정**

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 ID | 요구사항 명칭 | 요구사항 설명 | 개발현황 |
| UC-7-Require1 | 다른 회원에 대한 게시글 제한 조건 | 어떤 회원도 본인의 개인정보를 수정할 수 있으며 다른 사람들은 게시글 정보의 일부분만 읽을 수 있고 수정할 수 없다. | X |
| UC-7-Require2 | 수정 데이터의 제한 조건 | 수정할 수 있는 개인정보는 비밀번호, 전화번호, 이메일, 소속학과, 별명 등이며 그 이외의 정보는 수정할 수 없도록 한다. 나중에 정보가 더 추가되면 이 요구사항은 변경될 수 있다. | X |
| UC-7-Require3 | 수정사항 데이터베이스 기록 | 수정사항은 반드시 데이터베이스에 저장되어야 한다 | X |

**UC-8: 회원 관리**

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 ID | 요구사항 명칭 | 요구사항 설명 | 개발현황 |
| UC-8-Require1 | 동아리 가입상태 관리 | 어떤 동아리에 있는 동아리 장은 해당 동아리에 있는 모든 회원에 대한 가입상태를 관리할 수 있다.(단 사이트에서 밴을 시키지 못하며 동아리에 대해서만 밴을 시킬수 있다.) | X |
| UC-8-Require2 | 사이트 가입 상태 관리 | 사이트에 있는 모든 동아리 회원은 관리자가 관리할 수 있다(사이트 자체에서 밴을 시킬 수 있지만 사용자는 다시 가입을 할 수 있으며 밴 당한 다음 게시글, 댓글은 모두 삭제된다.) | X |
| UC-8-Require3 | 회원 데이터베이스 기록 | 회원에 대한 가입 상태는 반드시 데이터베이스에 기록되어야 한다. | X |

**UC-9: 게시글, 댓글 관리**

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 ID | 요구사항 명칭 | 요구사항 설명 | 개발현황 |
| UC-9-Require1 | 게시글 관리 조건 | 작성자 이외의 사용자가 게시글, 댓글 내용을 수정할 수 없다. | X |
| UC-9-Require2 | 동아리 게시글, 댓글 관리 | 동아리에 있는 게시글, 댓글은 동아리 장이 삭제할 수 있다. | X |
| UC-9-Require3 | 사이트 게시글, 댓글 관리 | 사이트에 있는 모든 동아리의 게시글, 댓글은 관리자가 삭제할 수 있다. | X |

**UC-10: 검색**

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항 ID | 요구사항 명칭 | 요구사항 설명 | 개발현황 |
| UC-10-Require1 | 게시글 검색 | 검색을 통해서 게시글을 검색할 수 있으며 어떤 회원도 가능하다 | X |
| UC-10-Require2 | 일치하지 않은 게시글 검색 | 검색어 대한 게시글이 존재하지 않으면 하얀색 화면을 뿌려준다. | X |
| UC-10-Require3 | 일치하는 게시글 | 검색어 대한 게시글이 존재한다면 해당 게시물의 제목(만약 존재한다면 소제목까지도)출력한다. | X |
| UC-10-Require4 | 검색 이용 데이터 | 검색은 데이터베이스를 이용해야 한다. | X |

**5. 시스템 구성도**

원, 블랙, 흑백, 스크린샷이(가) 표시된 사진

자동 생성된 설명

위 그림은 COM 시스템의 순서도를 보여주는 그림이다. 영어로 되어 있는데 모두 확장자를 뺀 파일 이름이다.

설명

모든 시작은 navigator에서부터 시작한다. 로그인을 먼저 해야만 다른 곳으로도 접근이 가능하도록 프로그래밍되어 있다. (단, 동아리에 대한 설명하는 site\_introduce는 제외하며 그 이외에 파일을 접근하려고 한다면 로그인하라고 사용자에게 명시)

**6. 시스템 순서도**

텍스트, 스크린샷, 폰트, 그래픽이(가) 표시된 사진

자동 생성된 설명

위 그림은 COM 시스템의 순서도를 보여주는 그림이다.

설명

사용자가 로그인을 한 상태에서 동아리 개설 및 가입 신청을 누르면 서버에서 해당 사용자의 상태를 데이터 베이스에 저장한다. 그리고 업로드된 파일은 해당 사용자의 아이디의 이름으로 폴더가 생성되며 각각 개설, 가입으로 저장되게 된다.

7. 주요 기능 및 알고리즘

주요 기능

사용자 로그인, 회원가입, 게시글 쓰기, 동아리 개설 신청하기, 동아리 가입 신청하기, 게시글에 대한 댓글 쓰기, 게시글 삭제하기, 개설된 동아리 목록 보기, 사이트에 대한 소개 보기

알고리즘

X

**8. 데이터베이스 구조 또는 파일 구성도**

텍스트, 번호, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명8 – 1. 이전의 ERD

위 이미지는 이전 시스템에서의 ERD를 보여주고 있다.

위 이미지에서 몇 가지를 수정하고 추가해야만 했었다. ( 특히 파일 경로명을 저장해서 적용한 다음에 보여주어야 하는 부분 ) 또한 추가로 데이터베이스를 수정하는 부분에서 코드를 수정해야 하는 오버헤드가 막중해져서 코드에서 일부 코드가 중복되는 문제가 발생하였다.

8 – 2. 수정된 ERD

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

위의 이미지는 수정된 ERD의 모습을 보여주고 있다.

추가된 부분은 다음과 같다

1. Club에서 Users저장 내용은 데이터베이스 성능 때문에 뺌,
2. 댓글 내용, 게시글 내용은 not null로 바뀜, 이것도 성능 및 보안 때문
3. post\_title 추가됨
4. 소속된 동아리 사용자 아이디는 admin\_no => club\_member
5. 소속된 동아리 동아리 이름 rank => club\_name
6. cmt\_owner, post\_owner\_id의 이름이 삭제된 이름으로 뜨는 게 아니라 정보를 지워야 함(아니면 생성 못함)
7. club\_creator\_id는 null로 지정 아니면 동아리 전체가 날라감
8. Image\_Postpage 테이블이 추가됨

**9. 개발 규모**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 모듈명 | 파일명 | 기능 | LOC | 개발언어 |
| 로그인 | login.py | 로그인 하기 | 64 | Python3 |
| 회원가입 | register.py | 회원가입 하기 | 118 |
| 사이드바 | Navigator.py | 로그인 하면 할 수 있는 목록 보여주기 | 77 |
| 게시글 쓰기 | Write\_article.py | 사용자의 게시글 쓰기 | 24 |
| 게시글 삭제 | Board.py | 사용자의 게시글 삭제하기 | 170 |
| 댓글 쓰기 | 사용자의 댓글 쓰기 |
| 게시글 목록 보여주기 | 모든 사용자가 쓴 게시글 모두 보여주기 |
| 개설된 동아리 목록 보여주기 | Club\_list.py | 개설된 동아리 목록 보여주기 | 16 |
| 동아리 개설 | Apply\_for.py  Club\_make.py | 사용자가 일부 문서와 동아리 이름, 소개를 작성하고 신청하기를 누르면 개설하기 | 7 + 106 |
| 동아리 가입 | Apply\_for.py  Club\_register.py | 일부 문서와 신청할려고 하는 동아리를 클릭하면 가입이 되는 구조 | 7 + 54 |
| 사이트 소개 | Site\_introduce.py | 사이트에 대한 간단한 소개 | 20 |
| 결과 | 663 | | | |

**10. 주요 실행화면**

텍스트, 소프트웨어, 컴퓨터 아이콘, 운영 체제이(가) 표시된 사진

자동 생성된 설명

위 이미지는 사이트의 첫 화면에서 보여지는 화면이다.

텍스트, 소프트웨어, 컴퓨터 아이콘, 웹 페이지이(가) 표시된 사진

자동 생성된 설명

위 이미지는 로그인 버튼 밑에 회원가입 버튼을 누르면 보여지는 화면이다. 각 정보를 모두 입력해야만 회워가입이 되며 정보를 입력할 때, 형식이 맞는지 확인하는 과정이 있다.(길이가 충분한지, 형식에 맞는지(정규표현식, regex) ) 모두 통과하면 회원가입이 성공적으로 끝나면서 데이터베이스에 등록하게 된다.

텍스트, 스크린샷, 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진

자동 생성된 설명

위 이미지는 사이트에 대한 간단한 설명을 볼 수 있는 화면이다. 로그인 없이도 볼 수 있다.

텍스트, 스크린샷, 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진

자동 생성된 설명

위 이미지는 개설 성공한 동아리 목록을 보여주고 있다.

텍스트, 스크린샷, 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진

자동 생성된 설명

위 이미지는 동아리 개설, 동아리 가입의 화면을 보여주고 있다. 동아리를 신청하면 동아리에 대한 파일이 서버에 저장된다.(실무에서는 암호화해서 저장할 것 같다.)

텍스트, 스크린샷, 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진

자동 생성된 설명

위 이미지는 게시글 목록을 확인할 수 있는 화면이다. 게시글 작성자 옆에 괄호 ()는 해당 사용자가 속한 동아리를 알려준다. 게시글 번호는 자동으로 증가되게끔 설정되어 있다.

텍스트, 스크린샷, 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진

자동 생성된 설명

위의 이미지는 댓글을 작성하고 댓글 작성 버튼을 누르니 댓글이 위 이미지와 같이 저장된다는 것을 보여주고 있다. 댓글은 길이가 충분해야 저장되는 방식으로 구현되어 있다.

실제로 구동되는 것이 궁금하다면 같이 첨부된 파일을 참고하시길 바랍니다.

**11. 팀원별 역할**

참고할 것

처음에 맡은 역할도 아닌 팀원들이 실제로 맡은 역할과 팀원들의 태도들을 보고 여기에 대한 팀장의 평가를 쓰는 공간, 팀원들이 실제로 결과를 낸 것들이 궁금하다면 git을 참고, 현재 이 내용도 팀장이 쓰는 중. 또한 아이디가 없는 사람은 실제로 한 것이 없다고 봐도 무방함

1. 박성국(팀장), 아이디: <https://github.com/Solomon-Psk153>
   1. 역할: 데이터베이스 구축, 웹 디자인, 웹 프로그래밍, 웹 기능 및 데이터베이스와 연동 구현, 문서(계획서, 설계도, 보고서, 발표자료(ppt) ) 작성 및 디자인, 테스팅(작동 확인 및 오류 고치기), 유지보수, 팀원들의 학습 지도, 팀원들 학습 상태 확인 및 조정(팀원들이 공부도 안하고 아예 모아 놓아도 이야기를 하지 않음)
2. 한태민, 아이디: <https://github.com/hanuelbawi>
   1. 역할: 웹 디자인, 및 화면 코딩, 결과는 많이 미흡하지만, 어떤 것이 있으면 같이 하려고 하며, 결과를 내려고 노력하는 모습이 있었다.
3. 김정현, 아이디: <https://github.com/Ownjeong00>
   1. 역할: 코딩(이지만 도대체 무엇을 하는지 알 수 없음. 결과를 낸 것이 거의 없었음. 그리고 코딩에 대한 열정도 없으며 오로지 학교에 있는 활동에 대한 관심만 가지고 있다. 그리고 게임을 너무 열심히 하고 그로 인한 피해는 상관하지 않는다.
4. 정유성, 아이디: X
   1. 역할: 초대를 했지만 이에 응하지 않아서 아이디가 없다. 무엇을 했는지 알 수 없으며 실제로도 하지 않았다. 팀에서 말이 제일 많았지만, 자신은 어떤 것도 손대지 않았으며 말 한 것들 중에서 실행에 옮겨진 것이 단 하나도 없었다. 방으로 급습했을 때는 약속 때문에 게임을 하겠다고 했을 때, 팀장으로서 정말 화가 많이 났었다. 물론 미안하다는 말은 단 한번도 하지 않았지만 말이다.
5. 김태한, 아이디: X
   1. 역할: 초대를 했지만 이에 응하지 않아서 아이디가 없다. 방 안에서 무엇을 했는지 알 수 없으며 실제로 무엇을 하려고 하는 의지가 없으며, 누군가와 대화하고자 하는 의지도 없다. 방안에서 급습을 하였지만, 방안에서 잠을 자고 막 일어나는 것을 발견하는 일이 매우 자주 있었다.

**12. 결론 및 느낀 점**

12 – 1. 결론

팀장: 나는 이걸 작업할 테니까, 너는 이런 작업해줘, 그리고 너는 이거 좀 작업해서 올려줘 (미팅: 카톡, 디스코드, 대면, 전화 등 각종 방법을 씀)

팀원들 = 건들지 마 난 버스 타러 왔어 ^^

12 – 2. 느낀 점

팀장(박성국)이 쓰는 느낀 점

과목에 대한 공부도 해야 하는데, 이런 프로젝트를 왜 굳이 이 과목에서 해야 하는지 모르겠다. 팀 별로 차이가 너무 크다. 어떤 팀은 팀원이 차이가 너무 나서 팀장이 모든 일일을 다하는 팀이 있는 반면에(우리 팀처럼), 역할이 아주 잘 분담 되어 있어 매우 수월하게 하는 게 눈에 보였었다. 하지만 더욱 문제는 지방대의 고질적인 문제로 학생들이 공부를 안하고 서로 떠밀기만 한다는 것이다. 이러면 누군가는 희생해서 해야 하는데, 차라리 이럴 거면, 작은 프로젝트를 하나 개인적으로 개인 프로젝트를 하는 것이 더 나은 선택이 될지도 모르겠다. (밖에서는 하려고 하는 팀원들이 모여서 팀 프로젝트를 하고 결과를 내는 프로젝트가 정말 많다. <https://disquiet.io/clubs>, <https://www.beginmate.com/>, <https://letspl.me/>, <https://www.1cup.kr/project> ) 그리고 앞에서도 이야기 했지만, 과목을 하는 학생들의 태도가 정말 마음에 들지 않았었다. 물론 다른 과목에서도 마찬가지이지만, 공부를 하려고 하는 마음도 없고, 이야기 하려고 하는 마음도 없었다. 모아 놓으면 그냥 이야기 흘러가는 대로 그냥 가만히 앉아있거나 그냥 멍하니 쳐다보는 것이 전부이니 팀장으로써 이야기해줄 수 있는 것이 거의 없다. 4학년에 들을 캡스톤은 얼마나 팀원들이 안할 지 걱정이다. 물론 열심히 할 거란 보장이 없지만 말이다. 이 과목으로 인해서 피해를 받은 과목이 한 두개가 아니다. 다시는 이런 팀원들과 프로젝트를 하고싶지 않으며 이런 과목도 두 번 다시 듣고 싶지 않을 정도로 최악이다. 팀원들에게 아무리 말해도 듣지를 않고 잠수만 타고 화를 내도 그냥 모르는 척 자기는 열심히 했다는 듯 이야기 한다. Git한 건 속이지 못하니 git한걸 참조하길….

한태민의 느낀 점

짧은 시간이지만 처음으로 하는 팀 프로젝트 였었던 만큼 부족한 점이 더 많았다고 생각이 든다. 팀을 구성했을 때 당시에도 사전에 준비가 되어있지도 않은 상태로 덜컥 성국이 한테 찾아가서 같이 할래? 라고 물어봐서 팀이 구성이 되었는데 확실히 소통에 부족함이 있었던 것은 맞는거 같다. 웹을 하나 만드는데 조금만 공부하면 다 이해하고 쉽게 할 수 있을 거 같았지만 생각보다 더 어려웠다. 이제 캡스톤도 해야하는 시점에서 이 경험을 가지고 더 소통을 열심히 해서 다음에는 이런 실수를 하지 않아야 겠다고 생각이 들었다.

김정현의 느낀 점

이번 소프트웨어공학 팀프로젝트를 하면서 내가 가진 전공의 깊이가 얼마나 되는지 되돌아보게 되었다. 또한 팀원들 간의 소통이 잘 되지 않아 어려움을 겪었다. 팀플에서의 내가 부족하고 고쳐야 될 점을 찾은 것이 좋았다.

정유성의 느낀 점

이번 소프트웨어 공학 프로젝트를 하면서 느낀 점은 아이디어와 설계 그리고 코딩과 테스팅까지 전부 도움 없이 해결해 나가는 과정에서 팀원들과의 불화를 해결하고 다시한번 나아가는 경험이 굉장히 좋았다.

김태한의 느낀 점

이번 프로젝트를 시작할 때 가장 걱정이 많았던 것은 프로젝트의 내용을 따라갈 수 있는지 였습니다. 다른 팀원들에 비해 배움이 부족한만큼 제가 이 프로젝트에서 뭔가를 하기는 어렵다고 느꼈고 그 부분은 어떻게 든 시킨 일 만이라도 잘하자는 것으로 해결하고자 했습니다. 실제로도 저는 프로젝트 팀 내에서도 능동적이지 못하고 의견도 딱히 없었던데 다 진행하는 속도가 느려 팀원들에게 여러 가지 누를 끼쳤다고 생각됩니다.