|  |  |  |  |
| --- | --- | --- | --- |
| **SSAFY Project 제 29회 보고서** | | | |
| 팀명 | SOLAFY | 작성일자 | 2020-12-17 |
| 참여자 | 윤기현, 이주희, 이아영 | 작성자 | 서범석 |

**○ 기본내용**

|  |  |  |  |
| --- | --- | --- | --- |
| 회의 주제 | Back-End 개발 중 발생한 문제점 토의 | | |
| 회의 장소 | Discord | 회의 일시 | 2020-12-17 20:00 |

**○ 회의내용**

|  |
| --- |
| 1. 문제점 제기    1. MySQL에 멤버 테이블이 필요할 수 있다 (부제 : 그룹 가입과정)       1. 이유 및 해결 방안 제시:   **가입 신청을 하고 그룹 관리자가 가입 수락**하는 과정에서  그룹 관리자에게 알림 송신 or **그룹에 가입 신청을 저장하는 테이블 생성필요**  **or Firebase에 넘길 수도 있는데 그룹정보의 많은 부분이 넘어가야한다.**   * + 1. 가입 신청 할 때, 신청한 사람의 이름(닉네임), 가입 메세지를 보냄   이 때 이 메시지를 DB 같은 곳에 저장할 필요성 존재.  우리는 회원에 대한 정보를 리얼타임데이터베이스에 쭉 저장하려고 하고 있음   * + 1. GroupMember테이블에서 “0:관리자, 1:일반회원, 2:회원가입을 신청한 사람” 등으로 구분하여서, 가입을 승낙하거나 확인하는 작업은 회원의 등급을 이용하여 확인 할 수 있다는 의견   1. 닉네임이 MySQL에 있는 이유에 대해서 의문점을 가짐      1. 답변 : 보통 화면에 표시할 때 UID가 아닌 닉네임을 보여주는 쪽으로 생각했기때문에, 사용하려면 MySQL에서 join을 해서 사용할 수 있어야 하므로 userNickname테이블이 존재한다.   2. 여러가지 정보가 계속 회원 테이블에 들어가게 될 것 같다.      1. 예시 : 문제를 만들고 관리할 수 있게 하려면, 해당 문제를 제출한 정보를 또 회원정보에 넣어 주어야하는 상황      2. 일단 진행해보며 관련 사항이 확립되면 그렇게 진행하는 쪽으로 결정   3. 메시지를 보내는 기능은 지원한다고 하는데, 삭제하는 기능은?      1. 일단 웹에서도 메시지를 보낼 수 있는지는 확인 필요(앱은 가능)      2. 해결 방안 제시 : 메세지를 보내지만, 지우는 기능은 굳이 필요하지 않을 것 같다는 의견 ( “가입 신청이 들어왔습니다” 라는 알림만 띄우고, 확인했을 때 이미 처리가 완료되어 있으면 그냥 넘어가는 형식)   4. 가입 신청과 수락은 어디에서 관리할 것인가?      1. 이것 또한 GroupMember의 등급을 통해서 관리 (회원등급 0, 1, 2중 2의 등급을 가입 승인 과정으로 관리)      2. 혹시 신청한 멤버(2의 등급을 가진 회원)가 가입 요청을 취소했다면, 해당 그룹의 테이블에서 삭제하는 방식으로 개발 진행   5. MySQL과 Firebase에 두는 데이터들의 밸런스에 대한 문제      1. 3가지 방안         1. 확실히 Member에 대한 정보만 Firebase에 고정할 건지         2. 적절히 분산할 건지         3. Firebase에선 인증/로그인만 시켜주고, 나머지 정보는 MySQL로 할 건지      2. 아직 개발은 안 해봐서 확실한 판단이 힘들다.      3. 혹시 문제가 될 가능성이 있다면, 미리 방지하는 것이 좋을 수 있다는 의견      4. 이 또한 일단 진행해보며 관련 사항이 확립되면 그렇게 진행하는 쪽으로 결정   6. 가입 메시지, 상태 메시지      1. 그룹원이 개인 상태 메시지를 바꿨을 때,   Group내에서의 상태 메시지와 동기화를 어떻게 시킬 수 있을지.   * + 1. 회원가입 메시지는 일시적으로만 필요하니까 회원가입 테이블을 만들어서 따로 관리하는 것도 좋을 것 같다. (기각)     2. 메시지는 전송하면, 받는 사람이 읽을 때까지 저장하고 있어야 하기 때문에 일시적으로 무작정 지울 수는 없다.   1. 데이터 베이스에 Listener가 필요한 상황이 있을 수 있다      1. 예시 : 문제를 수정하는데, 동 시간에 문제를 풀고 있던 사람들에 대한 처리         1. 해결책 : 수정 사항을 유저의 활동이 적은 시간대에 일괄 적용하는 방안      2. 모든 데이터를 MySQL로 한다면, 데이터의 변경을 전혀 감지할 수 없고,   모든 데이터를 Firebase로 해도 부담이 커질 것 같다.   * + - 1. 배운 것을 활용하기 위해서 spring은 사용해보자 : 모든 데이터를 firebase에 올리는 것은 피하기.   1. 기본 기능을 넘어서서 유저의 편의성을 고려하는 데에는 많은 고민이 필요하다.      1. **프로젝트 기간이 너무 길어질 수 있으므로,**   **일단 현재까지 정해진대로 진행하도록 한다.**   * 1. ProblemDto의 다른 데이터 포함여부 관련      1. HashtagDto, AnswerDto, CategoryNo는 ProblemDto의 외부에 두고 참조해서 가져오는 것으로 결정   2. ProblemMapper의 category관련 기능 분리 여부 관련      1. 문제에 대한 카테고리를 가져오는 기능,   카테고리를 통해서 문제를 가져오는 기능   * + 1. 기능이 너무 많아진다면 분리하는 것이 맞다고 판단  1. DB 수정    1. GroupMember 컬럼 추가       1. statusMessage(Datatype : TEXT)       2. regiMessage(Datatype :TEXT)    2. categoryLarge -> categorylarge : 테이블명 수정    3. problem 테이블       1. categoryNo          1. VARCHAR타입으로 변경. 2-3-5로 하기로 했으므로 크기는 10   **+) 차후 일정**   * **뷰 만들면 모듈테스트 하면서 단위테스트 진행** * **테스트해야 할 목록들도 곧 만들기 (QA)** |

**○ 회의성과**

|  |
| --- |
| 1. 문제 토의 및 해결 |

**○ 다음날 일정**

|  |
| --- |
| 1. 개발 및 진행 상황 토의 |

**○개인별 소견**

|  |
| --- |
| 범석 : 개발중에 침대금지 정말중요 정신차렷  기현 : “**자고로 개발할 때는 공식문서를 보세요”**  주희 : 🖤🤎💜💙💚💛🧡❤홧팅❤🧡💛💚💙💜🤎🖤  아영 : 취업 박람회가 끝났네요! 이제 개발에 전념합시다 😊 |