DB 기말

프로젝트 계획

평가 기준

기본 평가 기준

기말 발표 자료

게임 개발 과정에서 사용되는 데이터베이스를 프로그램과 연동하여 데이터를 입출력 하는 코드를 작성한다.

[제출물]
데이터베이스 설계 문서
데이터베이스 관련 소스 코드
발표자료

[평가 기준]
문서화, 설계의 깊이

코드 품질
- 코드 라인과 함수가 간결
- 변수, 클래스, 함수명 등이 명확
- 주석이 필요한 곳에 코드 의도를 잘 드러나도록

구현 완성도

수치적 분석 결과가 있다면 탁월
- 성능 향상의 요소

수업에서 제시된 평가 기준은 위와 같음

평가 기준 재해석

평가 기준은 아래와 같음

- 코드 퀄리티

    - 코드가 직관적인가?
    - 네이밍이 직관적이고 올바른가?
    - 주석이 코드의 의도를 잘 드러내는가?
- 시스템 문서화
    - 전체 시스템의 구조를 직관적으로 알 수 있게 나타내었는가?
    - 부분적인 시스템의 구조를 직관적으로 알 수 있게 나타내었는가?
    - 각 부분적인 시스템 간의 연관성 등을 알 수 있게 나타내었는가?
    - 위 항목을 시스템을 UML, 시퀀스 다이어그램 등으로 나타내었는가?
- 구현 완성도
    - 시스템의 깊이
        - 복잡성: 3-tier 구조처럼 어느정도의 단계와 구성으로 이루어져 있는가?
        - 안정성: 메모리 릭, 셧다운, 무결성 등
        - 사용성: 시스템이 얼마나 의미있게 설계되었는가?
- 수치 분석
    - 성능 분석 결과를 그래프 등으로 수치화
    - 성능 향상 결과를 그래프 등으로 수치화

프로젝트 목표

평가 기준 집중 우선순위

재해석된 평가 기준 중 다음 항목의 순서대로 집중적으로 다룰 예정

1. 구현 완성도
2. 문서화

구현 완성도

구현 완성도가 최우선 순위로 모든 기능의 정상 동작을 목표로 한다.

문서화

문서화는 구현 기능, 구조 등을 UML, 시퀀스 다이어그램을 통해 작성하고
시스템에 대한 흐름을 설명한다.시나리오

또한 어떤 시나리오를 위한 프로젝트였는지 설명한다.

시나리오

MMORPG 게임을 상정하여 일부 시스템을 구현한다.

- 플레이어 인벤토리를 구현한다.
- 상점을 구현한다.

플레이어 동작

플레이어는 다음과 같은 동작이 가능하다.

- 자신의 인벤토리 열람 (읽기, 쓰기)
- 상점에서 아이템 구매 가능

상점

플레이어는 다음과 같은 동작이 가능하다.

- 상점에서 판매하는 아이템을 구매
- 상점에서 판매하는 아이템은 수량이 한정되어 있음. 수량이 0이 되면 구매 불가
- 상점에서 판매하는 아이템 목록은 일정주기마다 랜덤으로 바뀜

동시접속

1명 이상의 플레이어가 한 공간에서 존재할 수 있다.

성능

대규모 인원(100명)의 처리를 동시에 할 수 있을 정도의 성능을 목표로함

성능에 대한 평가 기준은 다음과 같음

- 요청 건당 응답속도
    - 클라이언트에서 요청을 서버로 보냈을 때 서버가 클라이언트에 응답을 보내는데 까지 걸리는 시간

시스템 설계

네트워크 구조

클라이언트-서버-RDBMS, 3-Tier 구조를 사용한다.

클라이언트에서 쿼리문 위변조를 막기 위함

DBMS

DBMS는 RDBMS인 MySql를 사용한다.

트랜잭션 및 스토어 프로시저

각 쿼리/업데이트 명령을 서버에서 각각의 쿼리문으로 보내면

팬텀 리드 등이 일어날 수 있기 때문에 위 각 명령들은 트랜잭션으로 묶어 처리한다.

또한 재사용성 향상과 휴먼 에러 방지를 위해 모든 명령은 스토어 프로시저로 만들어 서버에서 프로시저를 호출하는 형태로 사용한다.

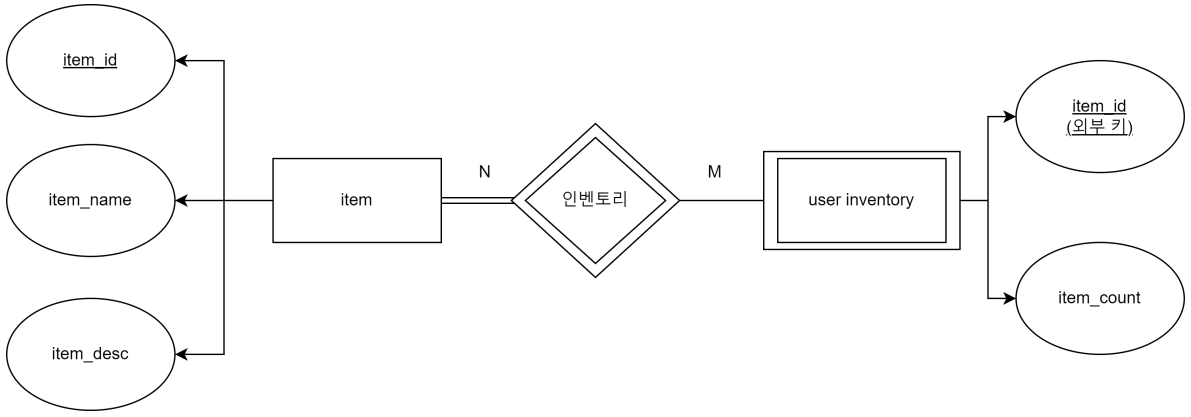
데이터

테이블

item 테이블과 플레이어 인벤토리 테이블(1개 이상) 이 존재한다.

item 테이블은 모든 아이템들에 대한 레코드로 구성되어있다.

플레이어 인벤토리 테이블은 플레이어 한 명당 테이블 하나가 할당되고,
item id를 외부키로 하는 테이블이다.

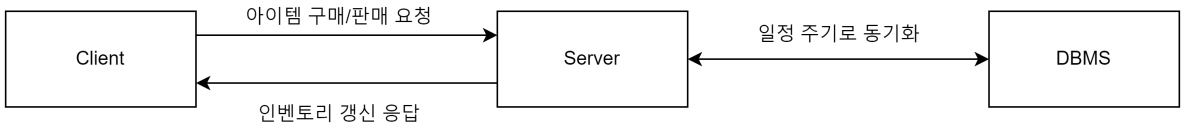


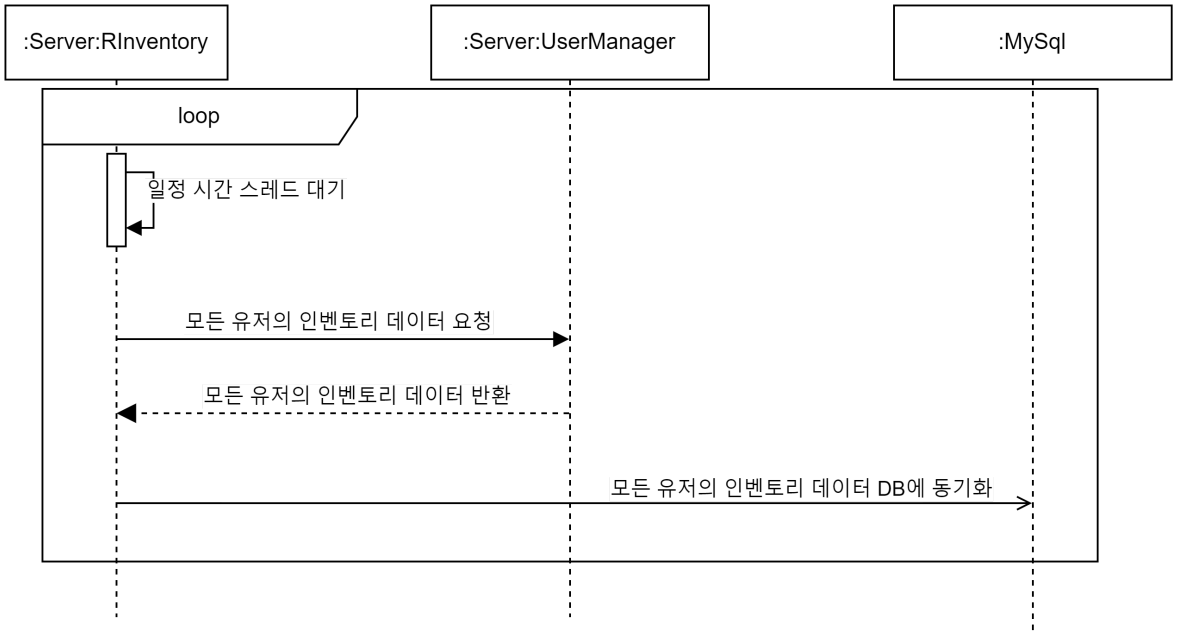
플레이어 인벤토리

write-back 방식을 차용한다. (https://yoongrammer.tistory.com/101: write-back 설명)

인벤토리 데이터는 기본적으로 서버에서 가지고 있는다.

인벤토리 데이터는 영구적으로 저장되어야 하므로 일정 주기마다 DB와 서버의 데이터를 동기화한다.

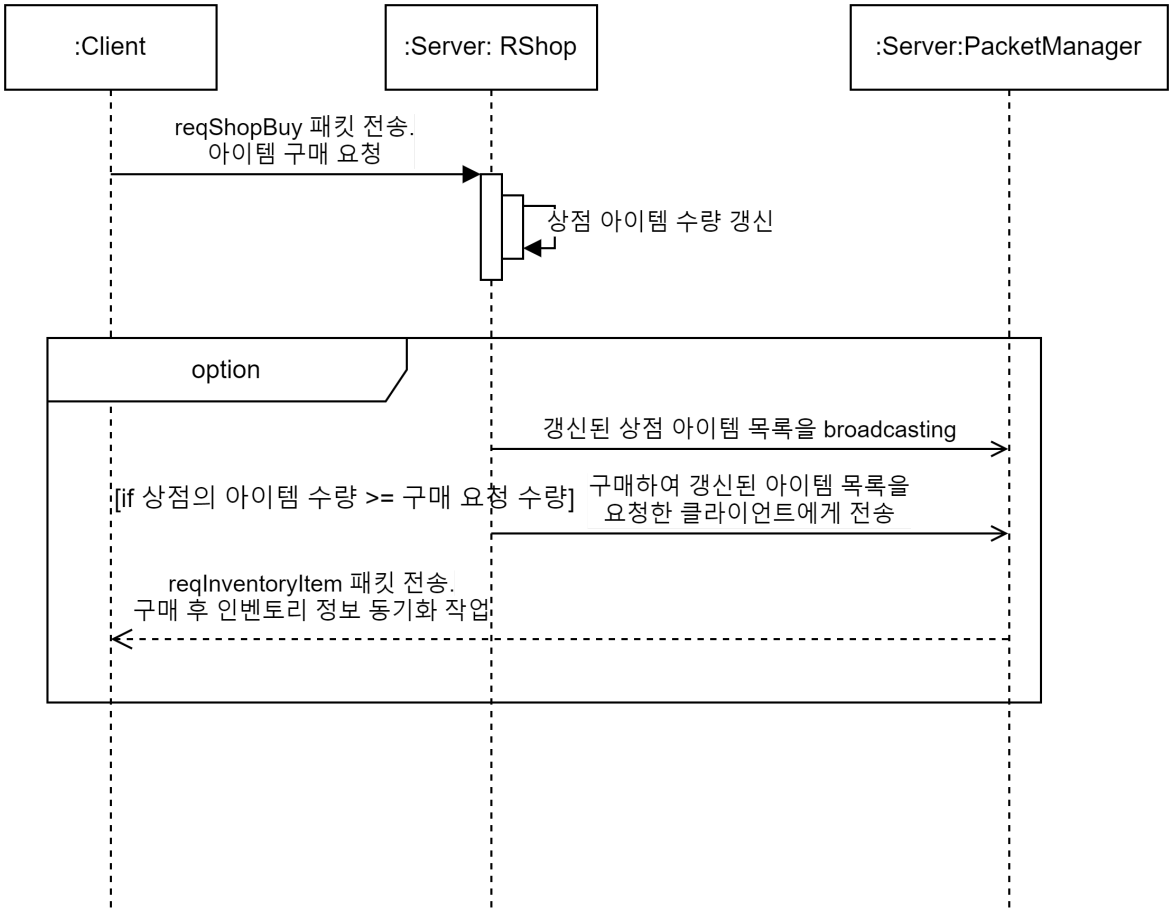




상점

현재 상점 정보는 서버가 유일하게 가지고 있는다.

상점 정보는 임시적인 것이며, 서버 인스턴스가 종료되면 함께 소멸된다.



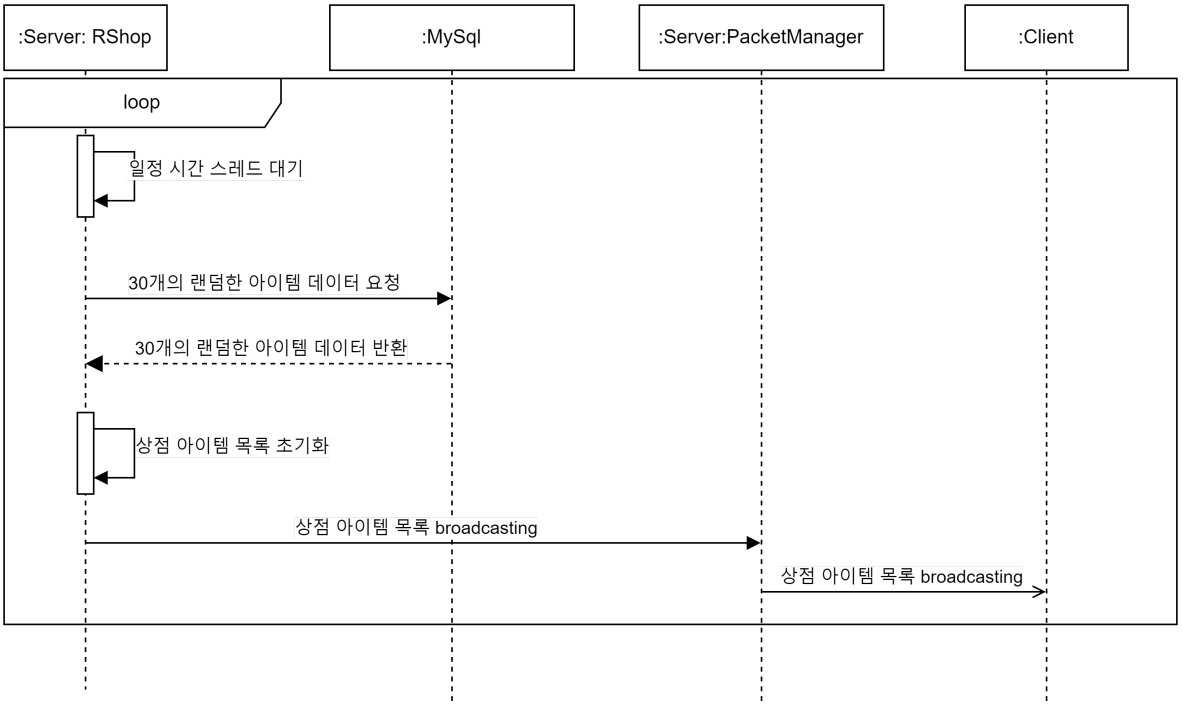
상점 정보를 갱신하기 위해 일정 주기마다 DB에서 특정 아이템 목록을 요청한다.
아래의
아이템 단락에서 설명한다.

아이템

상점에서 사고 팔 수 있는 아이템에 대한 정보다.

기본적으로 DB에서 관리하며 서버에서는 읽기 행동만을 수행한다.

아이템 정보는 DB 클라이언트에서 직접 생성하고,
서버가 아이템 정보를 수정하지 않는 것을 상정한다.


:Server: RShop
:MySql
:Server:PacketManager
:Client
loop
일정 시간 스레드 대기
30개의 랜덤한 아이템 데이터 요청
30개의 랜덤한 아이템 데이터 반환
상점 아이템 목록 초기화
상점 아이템 목록 broadcasting
상점 아이템 목록 broadcasting