

<도연>

1. SellServiceFacade 클래스

```
public interface SellServiceFacade {  
    public List<SellItem> getSellList();  
    public SellItem getSellByItemId(int itemId);  
    public int getCountById(int memberId);  
    public int getMemberIdbyItemId(int itemId);  
    public List<SellItem> getSellListByItemName(String itemName);  
    public List<SellItem> getSellListByMemberId(int memberId);  
    public void insertSellItem(SellItem item);  
    public void updateSellItem(SellItem item);  
    public void deleteSellItem(int itemId);  
    public void updateDealStatus(SellItem item);  
    public int getDealStatus(int itemId);  
    public void insertWishlist(Wishlist wish);  
    public List<Wishlist> selectWishlist(int memberId);  
    public void deleteWishlist(int wishId);  
    public int getMemberIdbyId(String id);  
    public void insertBuyFin(BuyFin buy);  
    public List<BuyFin> selectBuyFin(int memberId);  
}
```

2. SellServiceImpl 클래스

```
@Service("sellServiceImpl")  
public class SellServiceImpl implements SellServiceFacade {  
    @Autowired  
    private SellDao sellDao;  
  
    public List<SellItem> getSellList() {  
        return sellDao.getSellList();  
    }  
    public SellItem getSellByItemId(int itemId) {  
        return sellDao.getSellByItemId(itemId);  
    }  
    public int getCountById(int memberId) {  
        return sellDao.getCountById(memberId);  
    }  
    public int getMemberIdbyItemId(int itemId) {  
        return sellDao.getMemberIdbyItemId(itemId);  
    }  
    public List<SellItem> getSellListByItemName(String itemName) {  
        return sellDao.getSellListByItemName(itemName);  
    }  
    public List<SellItem> getSellListByMemberId(int memberId) {  
        return sellDao.getSellListByMemberId(memberId);  
    }  
}
```

```

    }
    public void insertSellItem(SellItem item) {
        sellDao.insertSellItem(item);
    }
    public void updateSellItem(SellItem item) {
        sellDao.updateSellItem(item);
    }
    public void deleteSellItem(int itemId) {
        sellDao.deleteSellItem(itemId);
    }
    public void updateDealStatus(SellItem item) {
        sellDao.updateDealStatus(item);
    }
    public int getDealStatus(int itemId) {
        return sellDao.getDealStatus(itemId);
    }
    public void insertWishlist(Wishlist wish) {
        sellDao.insertWishlist(wish);
    }
    public List<Wishlist> selectWishlist(int memberId) {
        return sellDao.selectWishlist(memberId);
    }
    public void deleteWishlist(int wishId) {
        sellDao.deleteWishlist(wishId);
    }
    public int getMemberIdbyId(String id) {
        return sellDao.getMemberIdbyId(id);
    }
    public void insertBuyFin(BuyFin buy) {
        sellDao.insertBuyFin(buy);
    }
    public List<BuyFin> selectBuyFin(int memberId) {
        return sellDao.selectBuyFin(memberId);
    }
}

```

<지우>

1. **MemberService** 클래스

```
public interface MemberService{
    public void insertMember(Member member); // 회원 가입

    public boolean isValidUser(String username, String password); // 로그인 체크
}
```

2. **ChatService** 클래스

```
public interface ChatService{
    public void insertChatMessage(ChatMessage message); // 채팅 등록

    public List<ChatMessage> getChatMessageList(int sellId); // 채팅 목록 받아오기
}
```

3. **StudyService** 클래스

```
public interface StudyService{
    public void insertStudy(Study study); // 스터디 모집 등록

    public List<Study> getStudyList(); // 스터디 목록 가져오기
}
```

<하윤>

1. **GroupPurchaseImpl** 클래스

```
@Service("groupPurchaseServiceImpl")
public class GroupPurchaseImpl implements GroupPurchaseDao {

    @Autowired
    private GroupPurchaseDao groupPurchaseDao;

    // 전체 게시글 불러오기(book_name==null) or 검색 book_name에 맞게 불러오기
    public List<GroupPurchase> getGroupPurchaseList(String book_name) {
        return groupPurchaseDao.getGroupPurchaseList(book_name);
    };

    // 게시글 자세히 보기
    public GroupPurchase Detail_GroupPurchase(int gp_postId) throws
    DataAccessException {
        return groupPurchaseDao.Detail_GroupPurchase(gp_postId);
    };
}
```

```

        // 공동구매 게시글 등록
        public void Upload_GroupPurchase(int book_id, int gp_postId, String
post_title, String post_contents, int readerId,
        Date date, int ing, int limit, int now, String account, int dis_price) {

            groupPurchaseDao.Upload_GroupPurchase(book_id, gp_postId, post_title,
post_contents, readerId,
                date, ing, limit, now, account, dis_price);

        };

        // 공동구매 신청
        public void Apply_GroupPurchase(int gp_postId, int ing, int limit, int now) {
            groupPurchaseDao.Apply_GroupPurchase(gp_postId, ing, limit, now);
        };

        // 공동구매 신청취소
        public void Cancel_GroupPurchase(int gp_postId, int ing, int limit, int now) {
            groupPurchaseDao.Cancel_GroupPurchase(gp_postId, ing, limit, now);
        };

        // 공동구매 게시글 검색
        public Book Search_GroupPurchase(String book_name) {
            return groupPurchaseDao.Search_GroupPurchase(book_name);
        }
    }
}

```

2. BookServiceImpl 클래스

```

@Service("bookServiceImpl")
public class BookServiceImpl {

    @Autowired
    private BookDao bookDao;

    // 책 제목으로 검색
    public List<Book> Serach_BookName(String book_name) {
        return bookDao.Serach_BookName(book_name);
    };

    // 책 아이디로 검색 (ISBN)
    public Book Serach_BookId(int book_id) {
        return bookDao.Serach_BookId(book_id);
    };

    // Book 정보 등록

```

```

        public void Resister_Book(String book_id, String book_name, String book_writer,
                                   String book_publisher, int book_price, String book_summary, String
book_photo) {
            bookDao.Resister_Book(book_id, book_name, book_writer, book_publisher,
book_price, book_summary, book_photo);
        }
    }
}

```

<지운>

1. AuctionServiceImpl 클래스

```

@Service("auctionServiceImpl")
public interface AuctionServiceImpl{
    @Autowired
    private AuctionDao auctionDao;

    //1) 경매 아이템 게시글 등록시 사용
    public void insertAuctionItem(AuctionItem ai) {
        auctionDao.insertAuctionItem(ai);
    }
    //2) 경매 게시글 수정 시 사용
    public void updateAuctionItem(AuctionItem ai) {
        auctionDao.updateAuctionItem(ai);
    }
    //3) 경매 게시글 삭제할 시 혹은 마감 시 사용.
    public void deleteAuctionItem(AuctionItem ai) {
        auctionDao.deleteAuctionItem(ai);
    }
    //4) 경매 아이템 이름으로 검색할 때 사용 return 값 : 해당 경매 게시글 아이디
    public String searchAuctionByItemName(String au_itemName) {
        return auctionDao.searchAuctionByItemName(au_itemName);
    }
    //5) 경매 아이템 작가 이름으로 검색할 때 사용. retrun : 해당 경매 게시글 아이디
    public String searchAuctionByAuthor(String author) {
        return auctionDao.searchAuctionByAuthor(author);
    }
    //6) 경매 아이템 게시글 상세보기로 불러올 때 사용.
    public AuctionItem selectAuctionItem(String au_itemId) {
        return auctionDao.selectAuctionItem(au_itemId);
    }
    //7) 경매 게시글 리스트 불러올 때 사용. (가장 먼저 불러오는)
    public List<AuctionItem> selectAuctionItemList() {
        return auctionDao.selectAuctionItemList();
    }
    //8) 경매 입찰시 사용.
    public void insertAuctionBid(AuctionBid ab) {
        auctionDao.insertAuctionBid(ab);
    }
}

```

```

//9) 경매 입찰 리스트 불러올 때
public List<AuctionBid> selectAuctionBidList() {
    return auctionDao.selectAuctionBidList();
}
//10) 경매 입찰 상태 수정하고 싶을 때 사용
public void updateAuctionBidDealStatus(AuctionItem ai, int bidStatus) {
    auctionDao.updateAuctionBidDealStatus(ai, bidStatus);
}
//11) 경매 아이템 상태 수정하고 싶을 때 사용.
public void updateAuctionItemDealStatus(AuctionItem ai, int itemStatus) {
    auctionDao.updateAuctionItemDealStatus(ai, itemStatus);
}
//12) 마감하고, 낙찰자한테 알림등록 시 사용.
public void insertNoticeMessageByAuction(int memberId) {
    auctionDao.insertNoticeMessageByAuction(memberId);
}
//13) 낙찰자 정보 불러올 때 사용.
public Member selectMemberByAuction(int memberId) {
    auctionDao.selectMemberByAuction(memberId);
}
}

```