구 글래스 설계 보고서 THE DISH

TEAM . B E _ 개발자



CONTENTS TEAM. BE_개발자

01

패키지 다이어그램

02

클래스 다이어그램

03

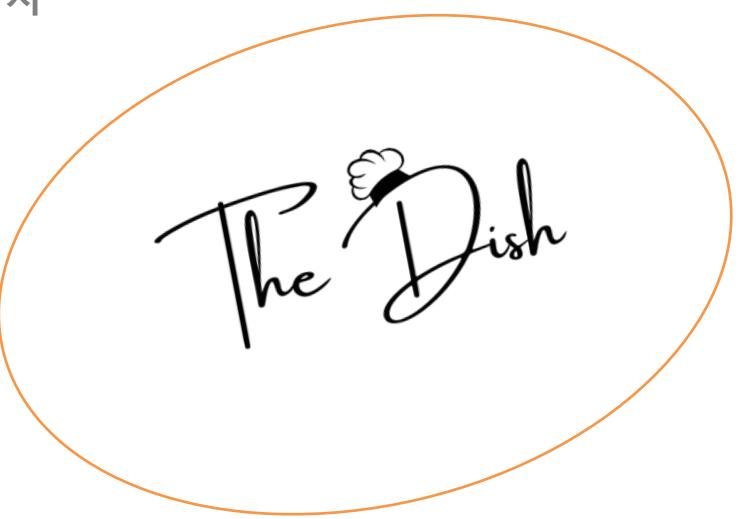
시퀀스 다이어그램



CONTENTS TEAM. BE_개발자

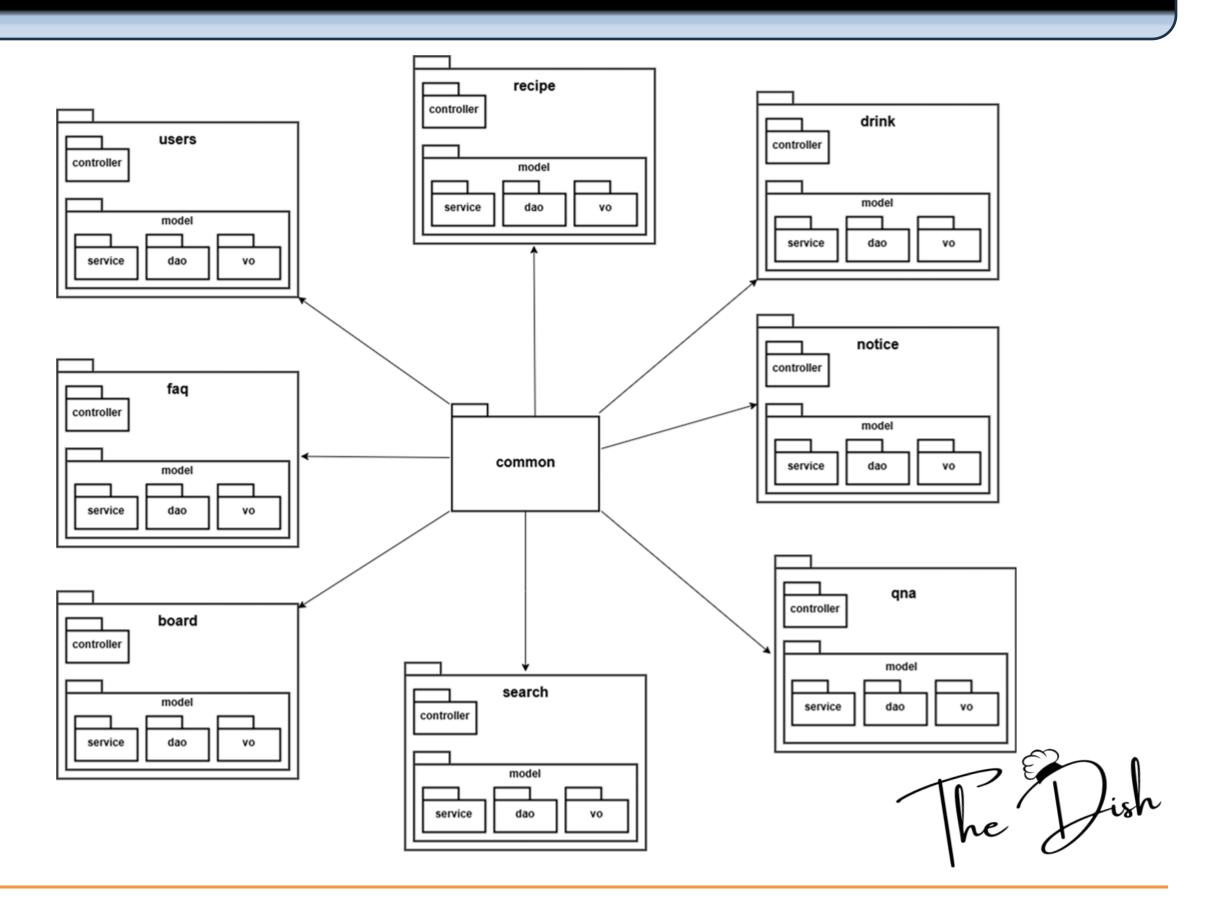
클래스 다이어그램

시퀀스 다이어그램



패키지 다이어그램

패키지 분리 & 계층화



CONTENTS TEAM. BE_개발자

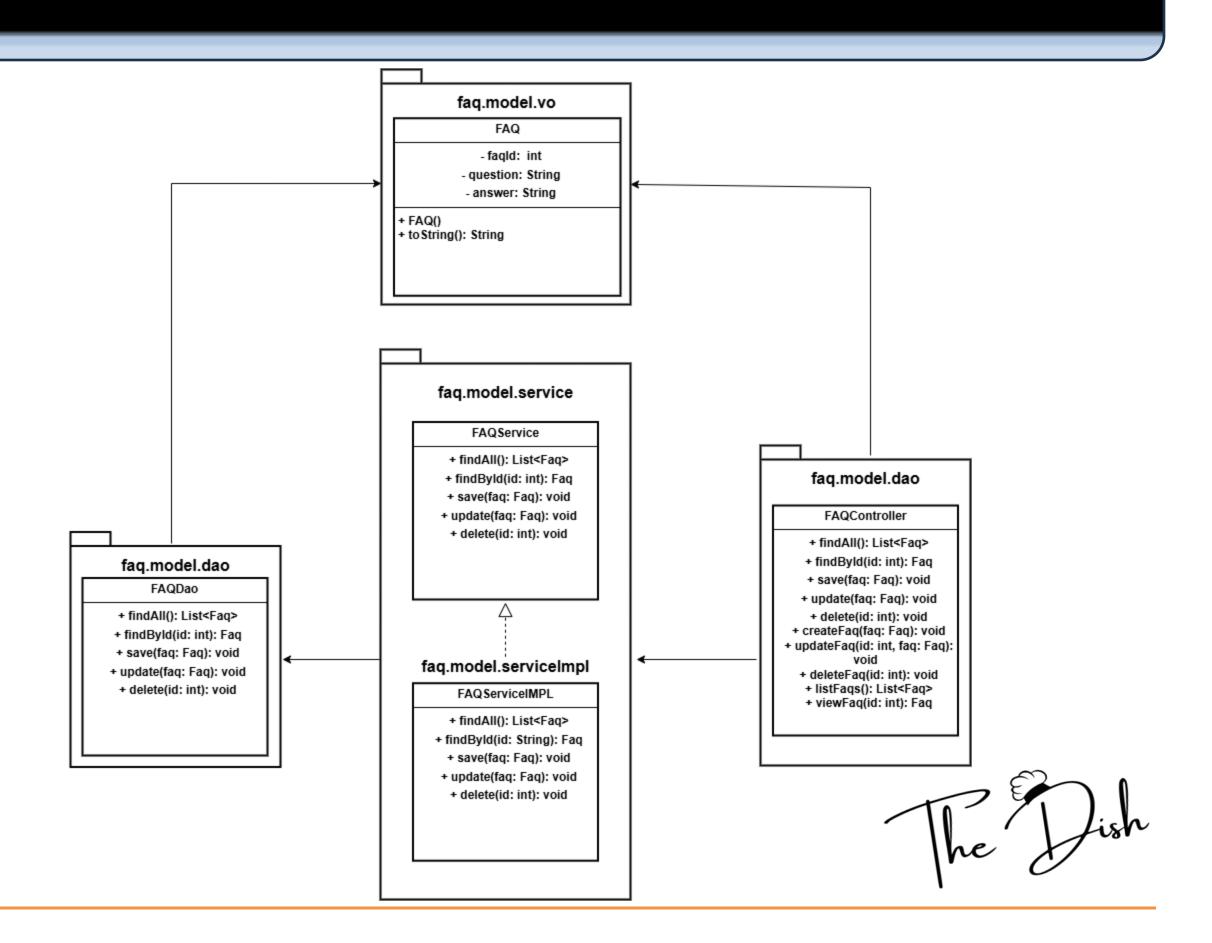
패키지 다이어그램

클래스 다이어그램

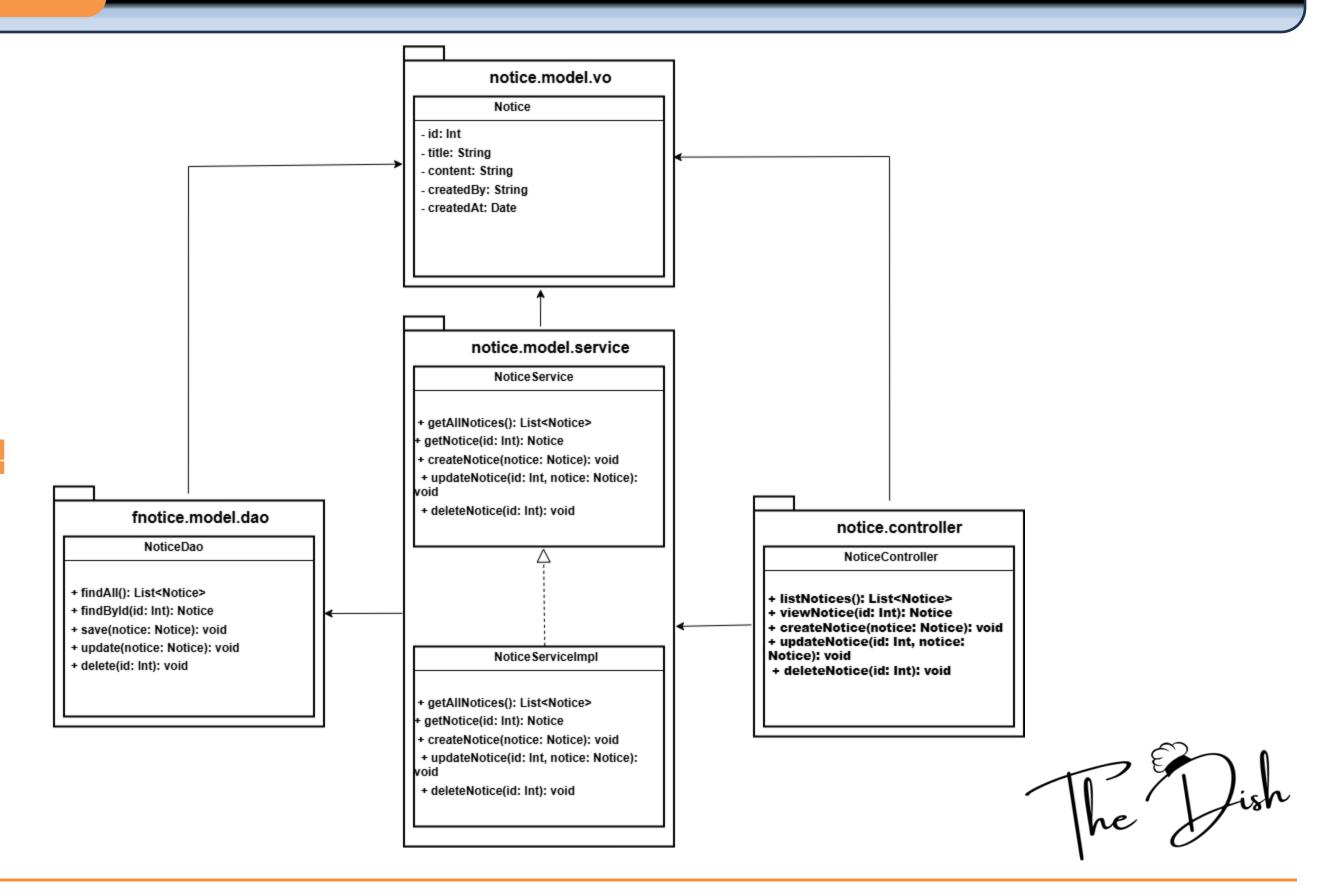
시퀀스 다이어그램



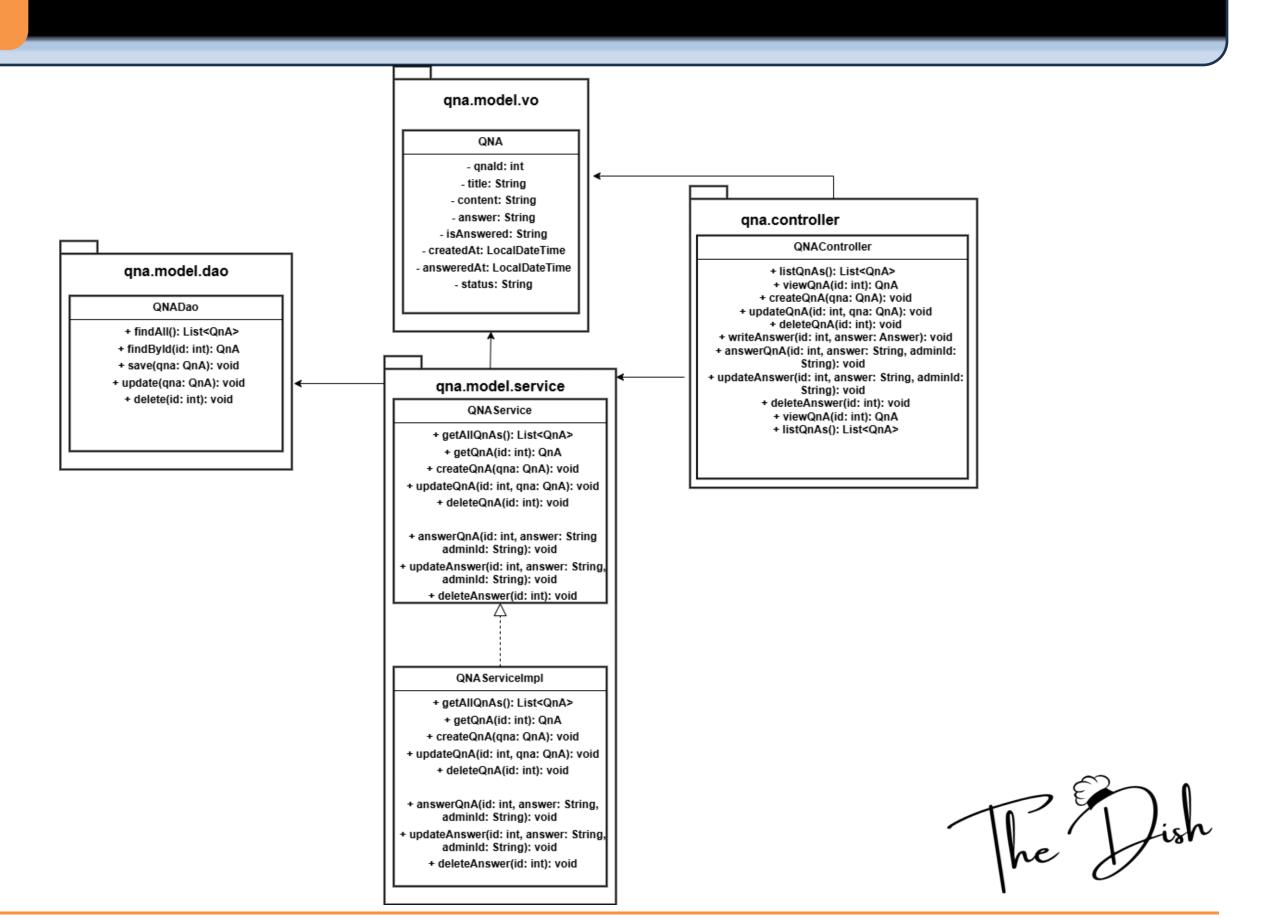
자주묻는 질문(FAQ)

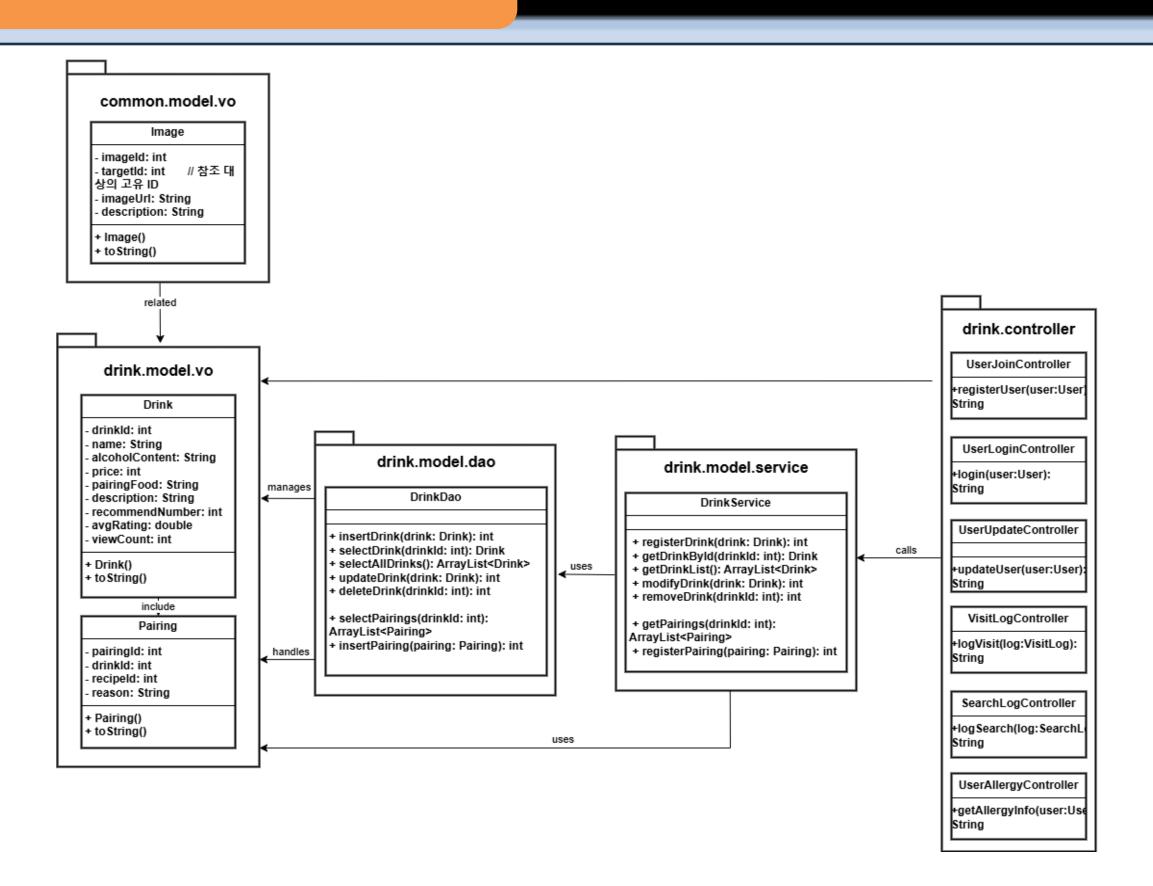


게시판(Notice)



질의응답(Q&A)



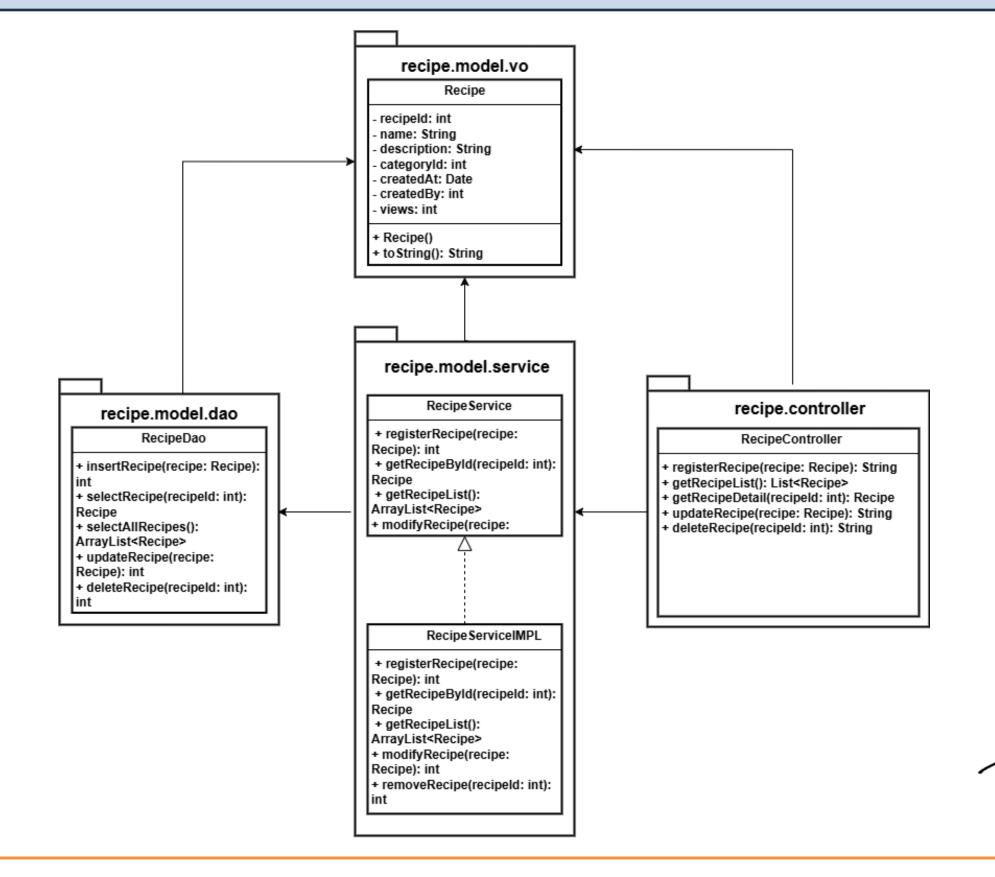


음료 & 술 서치

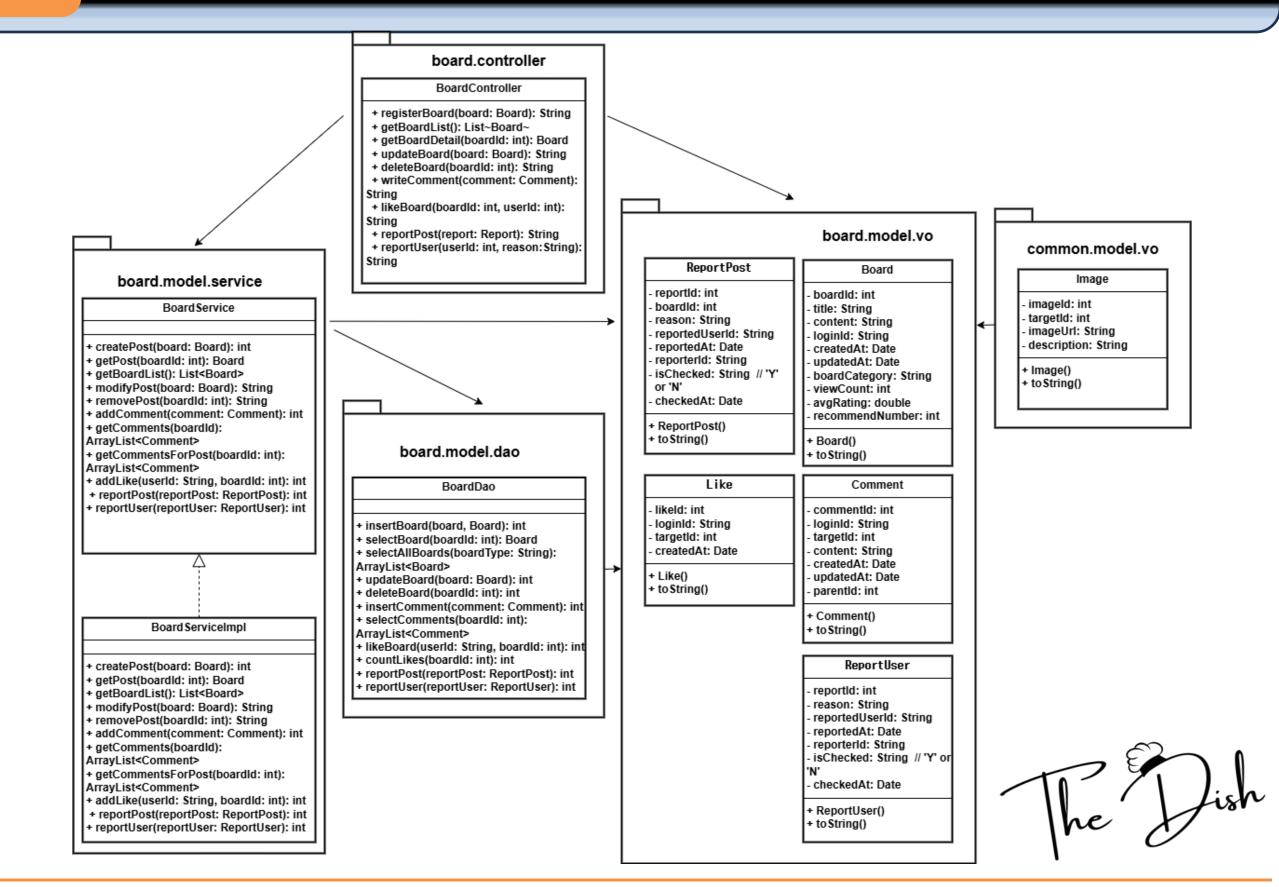
기능 클래스 다이어그램

The Dish

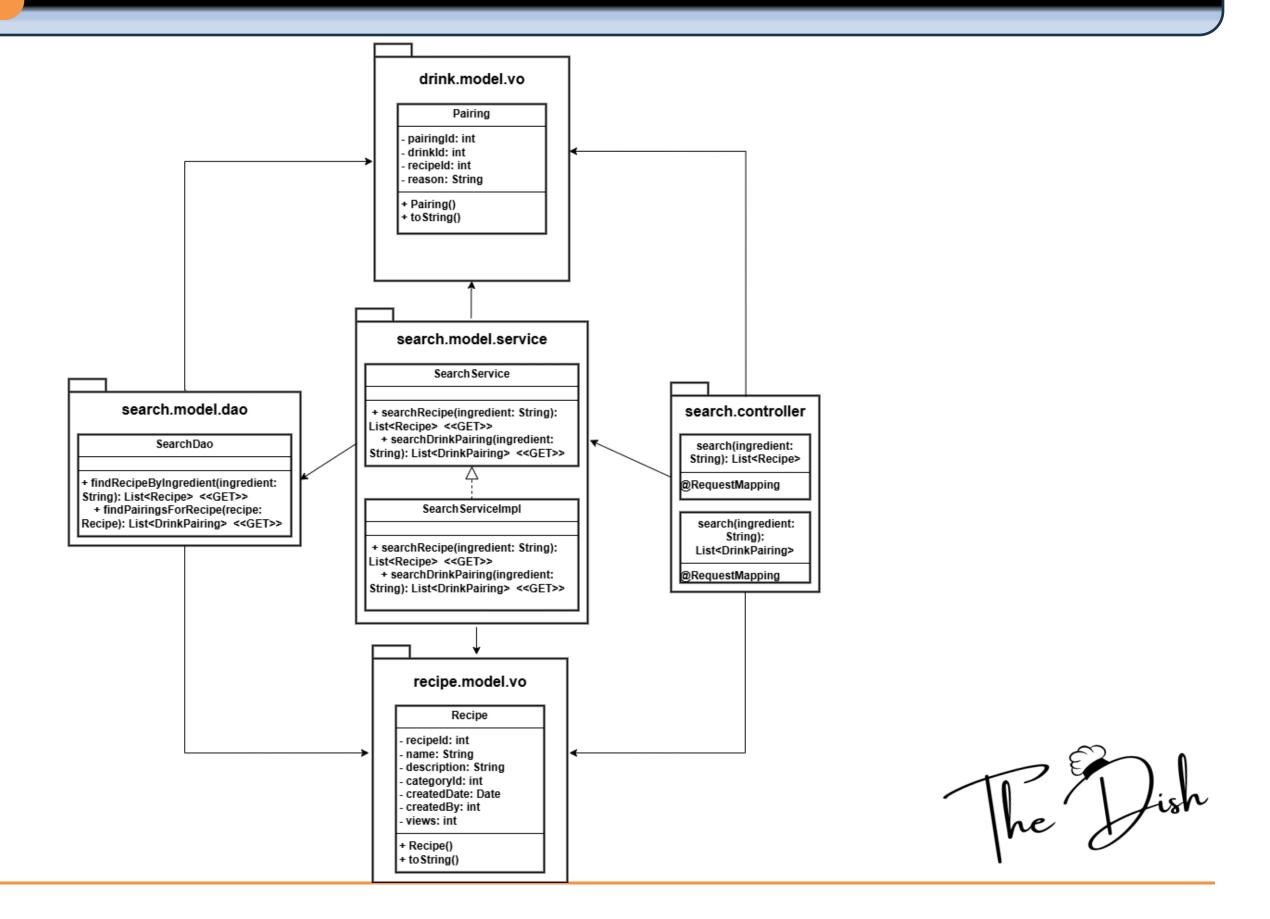
레시피(Recipe)



보드(Board)



서치(Search)



users.model.dao 유저(Users) UserDao 기능 클래스 다이어그램 + selectUserByld(userld: String): User + loginUser(loginId: String, password: String): User users.controller + insertUser(user: User): int + updateUser(user: User): int UserJoinController + selectUserAllergies(loginId: String): + registerUser(user: users.model.service ArrayList<UserAllergy> User): String + insertUserAllergy(userAllergy: **UserService** UserAllergy): int UserLoginController + logVisit(visitLog: VisitLog): int + getUserByld(userId: int): User + log Search(searchLog: SearchLog): int + loginUser(loginId: String, password: + login(user: User): String): User String + registerUser(user: User): int + updateUser(user: User): int + getUserAllergies(userId: int): UserUpdateController ArrayList<UserAllergy> + registerUserAllergy(userAllergy: UserAllergy): int + updateUser(user: users.model.vo + logVisit(visitLog: VisitLog): int User): String UserAllergy UserAllergyController userld: int userld: int User ServiceImpl + getAllergyInfo(userId: email: String allergyld: int SearchLog int): String - nickname: String + UserAllergy() + getUserByld(userId: int): User - password: String logld: int + to String() + loginUser(loginId: String, password: - provider: String keyword: String - role: String // CHECK: String): User VisitLogController searchType: String / VisitLog USER, ADMIN + registerUser(user: User): int CHECK: 레시피, 술 등 + logVisit(log: VisitLog): - status: String // + updateUser(user: User): int loginld: String logld: int + getUserAllergies(userId: int): CHECK: ACTIVE, String searchedAt: Date ipAddress: String BLOCKED, PENDING ArrayList<UserAllergy> pageUrl: String - loginld: String + registerUserAllergy(userAllergy: SearchLog() loginld: String - username: String UserAllergy): int SearchLogController + to String() - visitedAt: Date - createdAt: Date + logVisit(visitLog: VisitLog): int + log Search(searchLog: SearchLog): + log Search(log: + VisitLog() + User() SearchLog): String

+ to String()

+ to String()

CONTENTS TEAM. BE_개발자

01

패키지 다이어그램

02

클래스 다이어그램

03

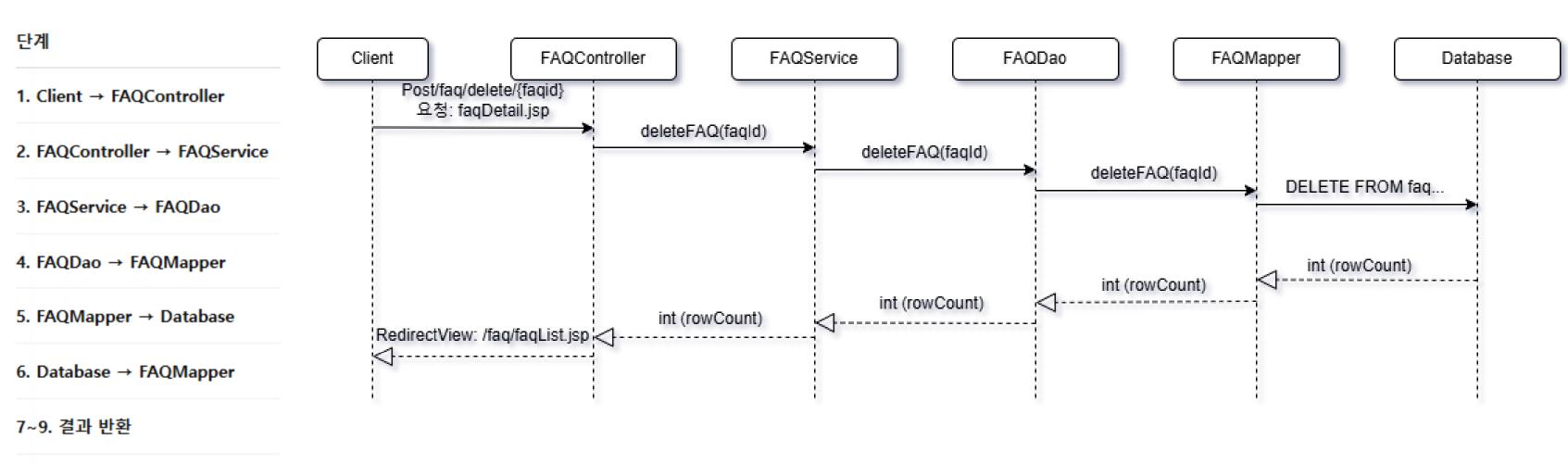
시퀀스 다이어그램



자주묻는 질문(FAQ)

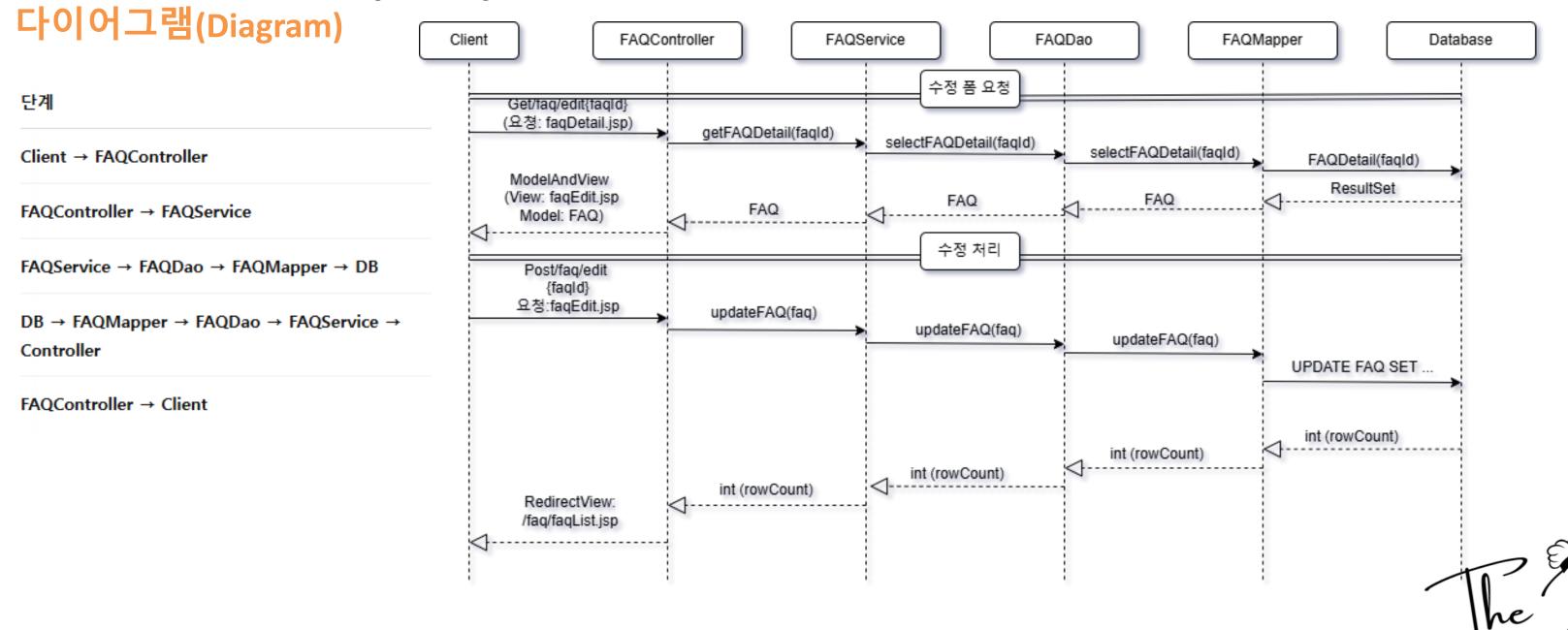
다이어그램(Diagram)

10. Controller → Client



The Dish

자주묻는 질문(FAQ)



자주묻는 질문(FAQ)

다이어그램(Diagram)

단계

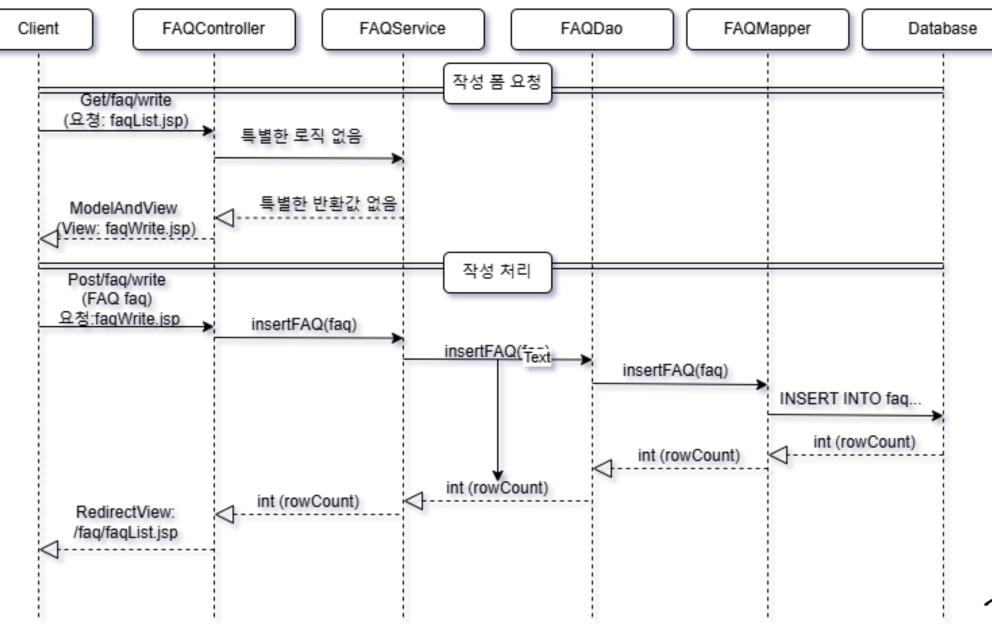
Client → FAQController

FAQController → FAQService

FAQService → FAQDao → FAQMapper → Database

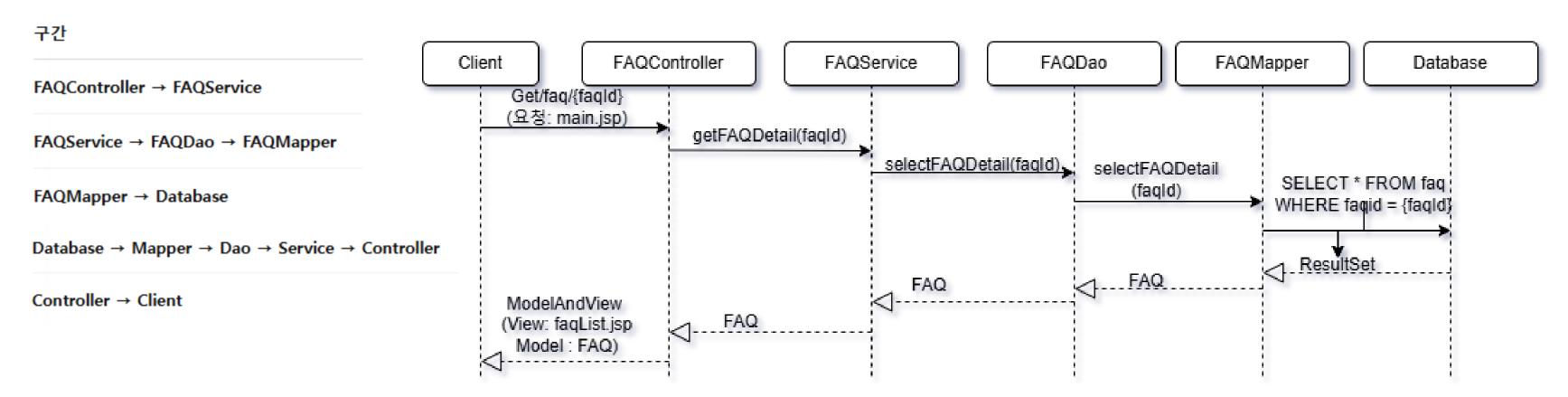
Database → Mapper → Dao → Service → Controller

FAQController → Client



The Dish

자주묻는 질문(FAQ)





질의응답(Q&A)

다이어그램(Diagram)

단계

Client → QNAController

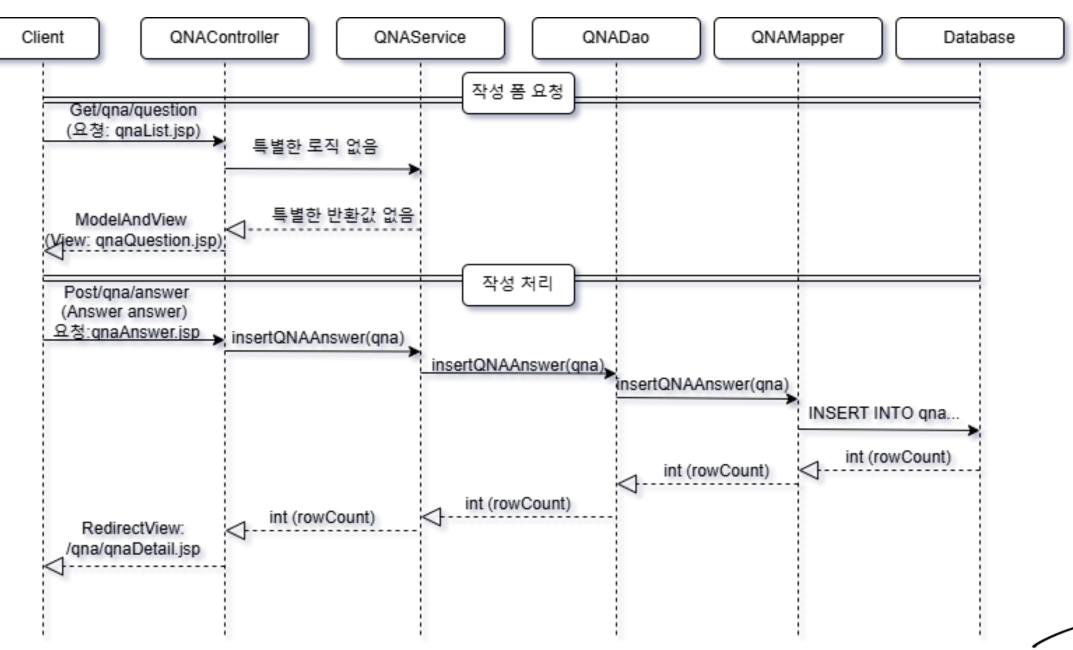
QNAController → QNAService

QNAService → QNADao → QNAMapper

QNAMapper → Database

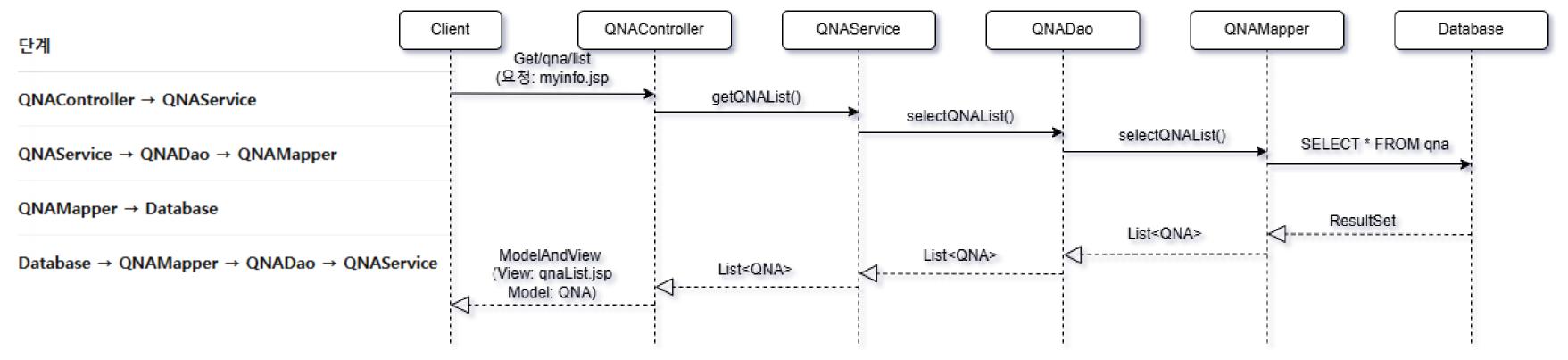
Database → Mapper → Dao → Service → Controller

Controller → Client



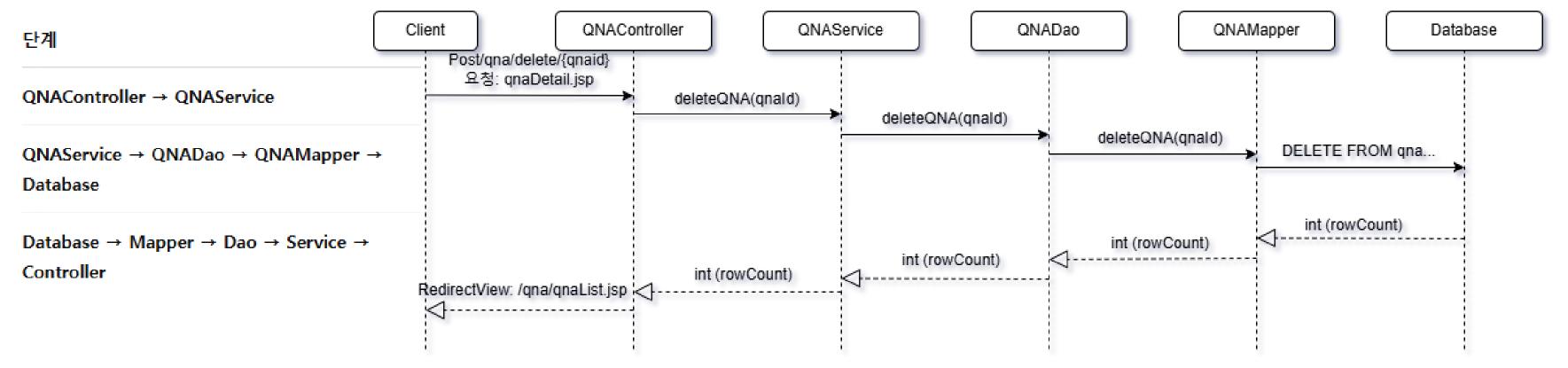
The Dish

질의응답(Q&A)





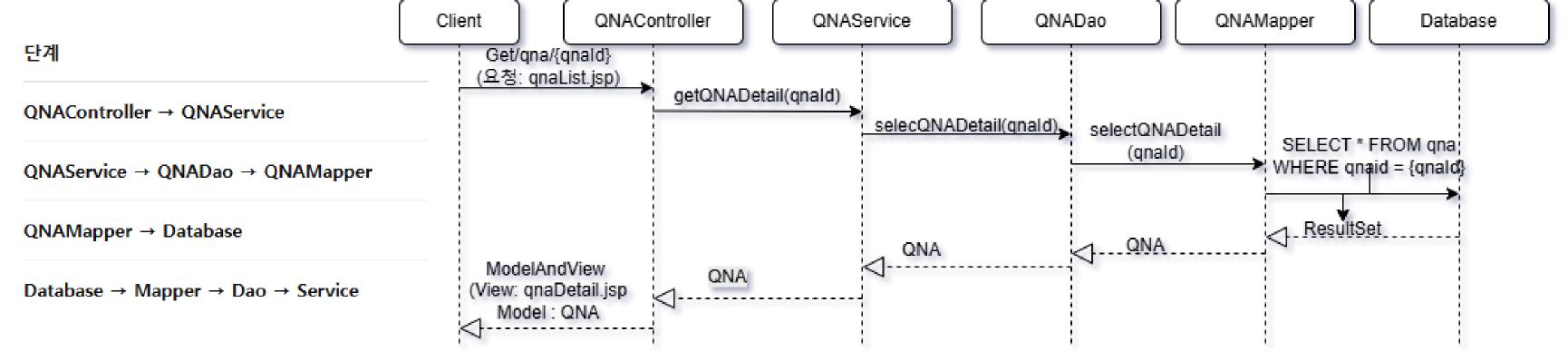
질의응답(Q&A)





질의응답(Q&A)

다이어그램(Diagram)



The Dish

질의응답(Q&A)

다이어그램(Diagram)

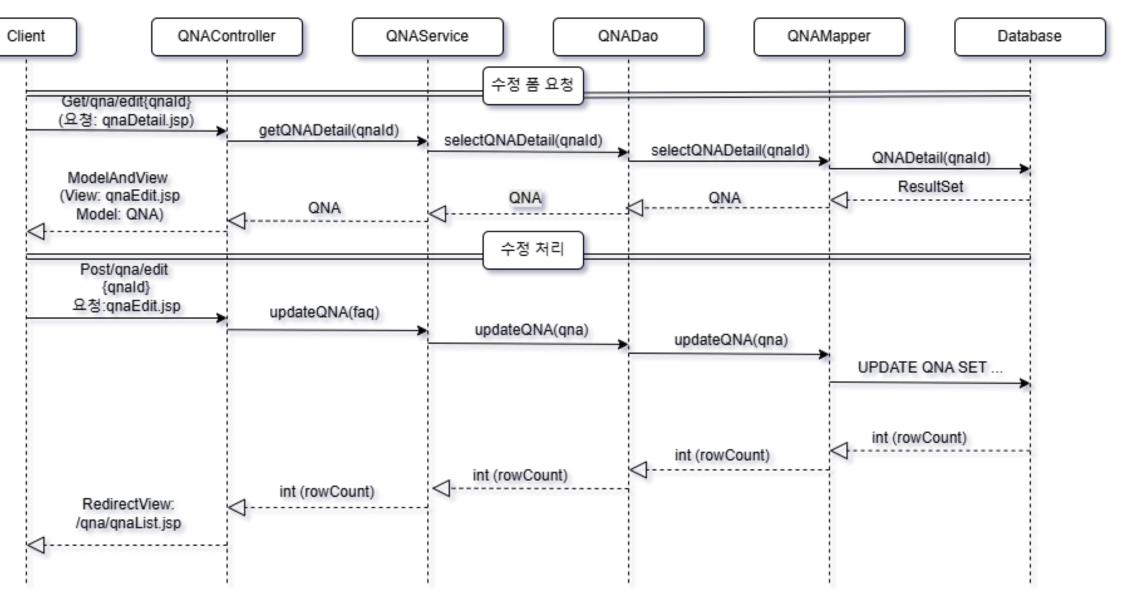
Client → QNAController

단계

QNAController → QNAService

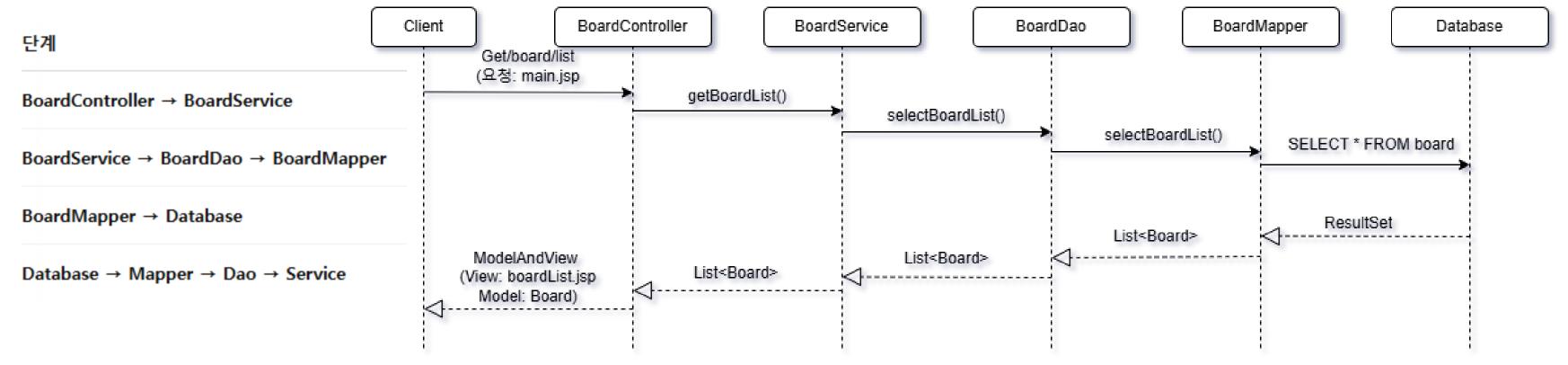
QNAService → QNADao → QNAMapper → DB

DB → Mapper → Dao → Service → Controller





보드(Board)





보드(Board)

다이어그램(Diagram)

게시글 조회 처리

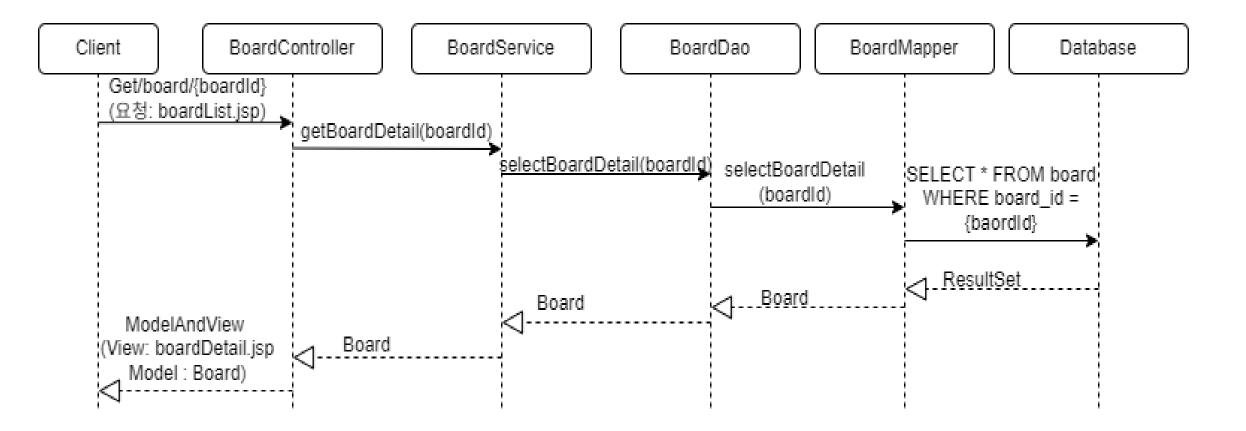
단계

Controller → Service

Service → DAO

DAO → Mapper → DB

결과 반환 및 화면 출력



단계

DB → Mapper → Dao → Service → Controller

Controller → Client



보드(Board)

다이어그램(Diagram)

게시글 등록 처리

단계

Client → BoardController

BoardController → BoardService

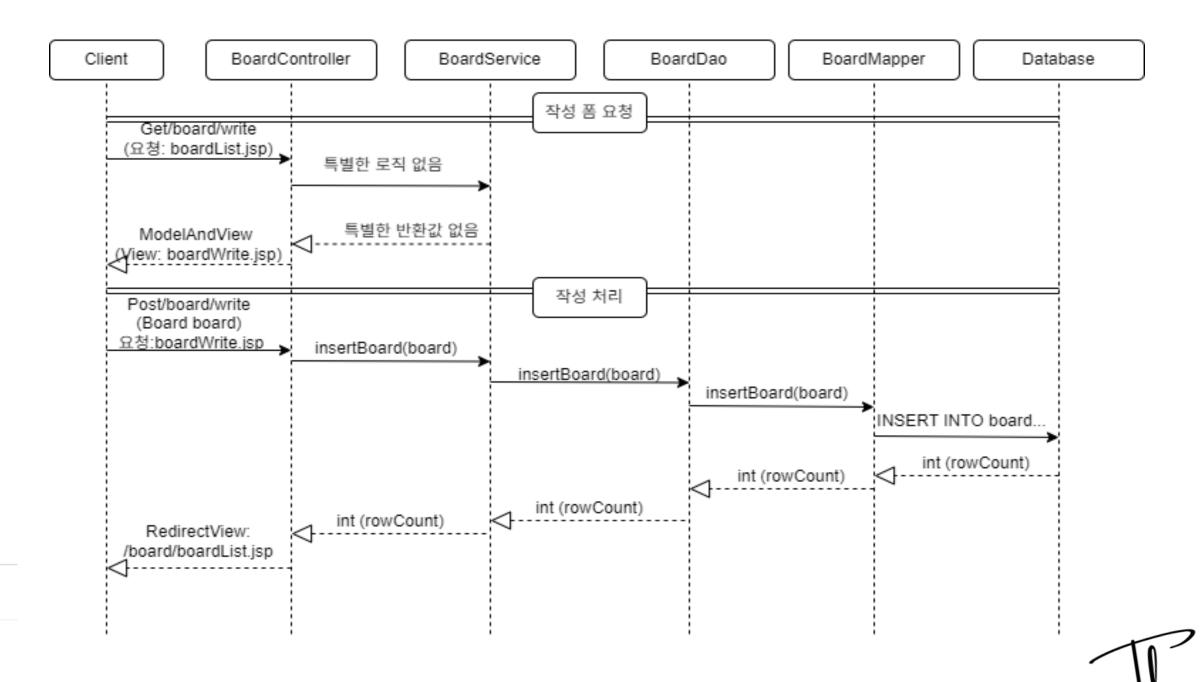
Service → DAO → Mapper → DB

결과 반환 및 이동 처리

단계

DB → Mapper → Dao → Service → Controller

Controller → Client



Client

보드(Board)

다이어그램(Diagram)

수정 폼 요청 과정 (GET)

Get/board/edit{boardId} (요쳥: boardDetail.jsp) getBoardDetail(boardId) → selectBoardDetail(boardId) 단계 selectBoardDetail(boardId) BoardDetail(boardId) ModelAndView ResultSet Client → GET /board/edit/{boardId} 요청 (View: boardEdit.jsp Board Board Board Model: Board) Controller → Service → DAO → Mapper → DB: 수정 처리 Post/board/edit SELECT * FROM board WHERE board_id = {boardId} {boardId} 요청:boardEdit.jsp updateBoard(board) 결과로 Board 객체 반환 updateBoard(board) updateBoard(board) UPDATE board SET ... 반환된 객체는 boardEdit.jsp 에 전달되어 수정폼 렌더링 int (rowCount) 단계 int (rowCount) int (rowCount) Client → POST /board/edit/{boardId} 요청 (boardEdit.jsp 에서 int (rowCount) RedirectView: /board/boardList.jsp Controller → Service → DAO → Mapper → DB: -----UPDATE board SET ... WHERE board_id = ? SQL 실행

BoardService

BoardDao

수정 폼 요청

BoardMapper

Database

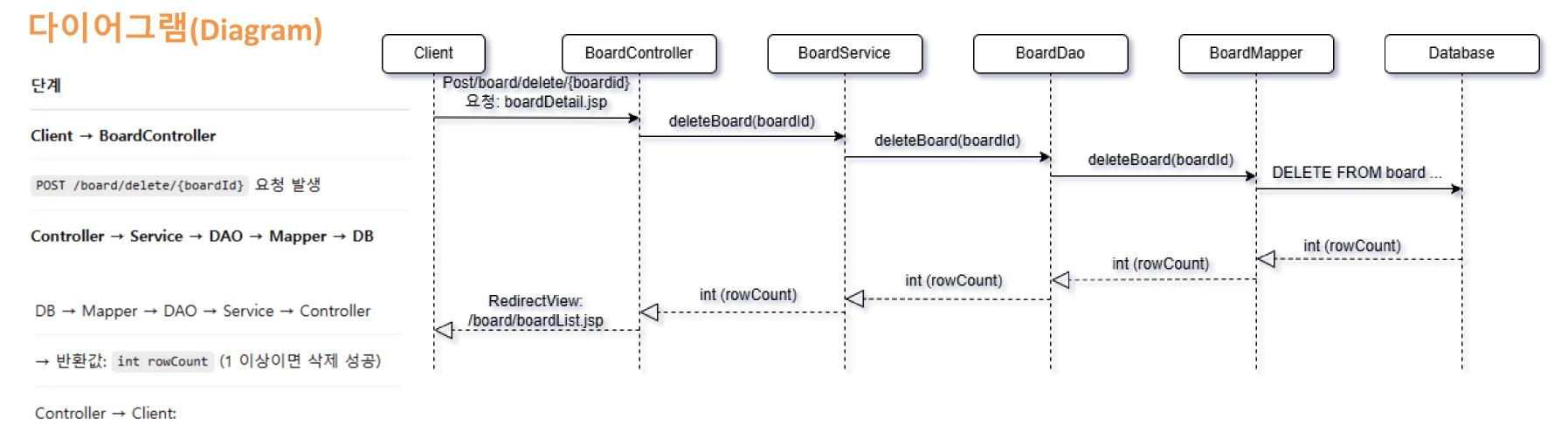
BoardController

결과: rowCount 반환

성공 시 → RedirectView: /board/boardList.jsp

보드(Board)

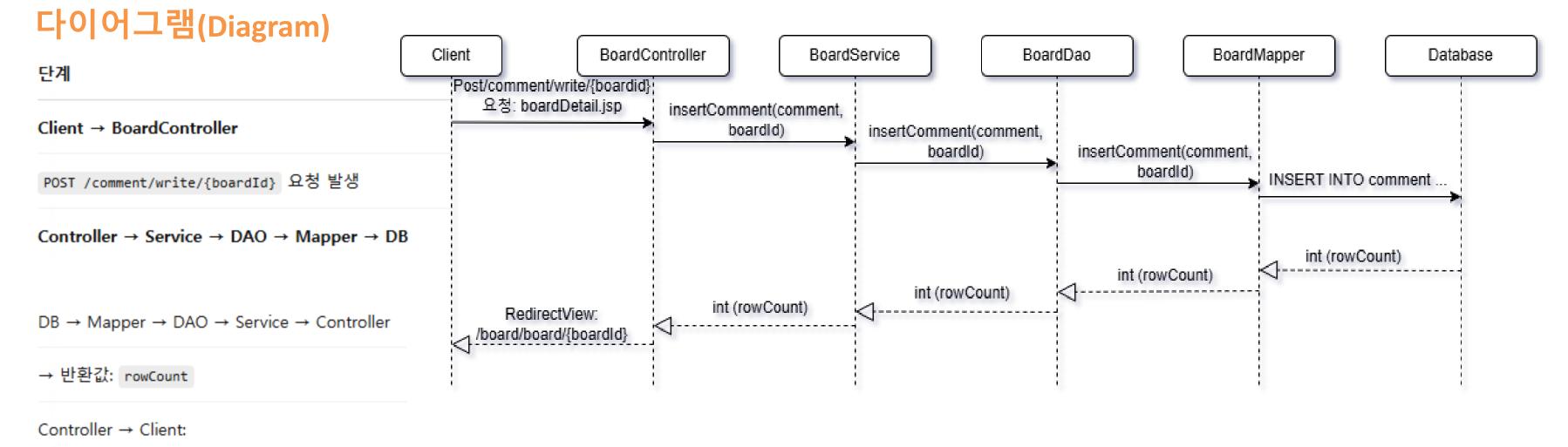
RedirectView: /board/boardList.jsp 로 이동



The Dish

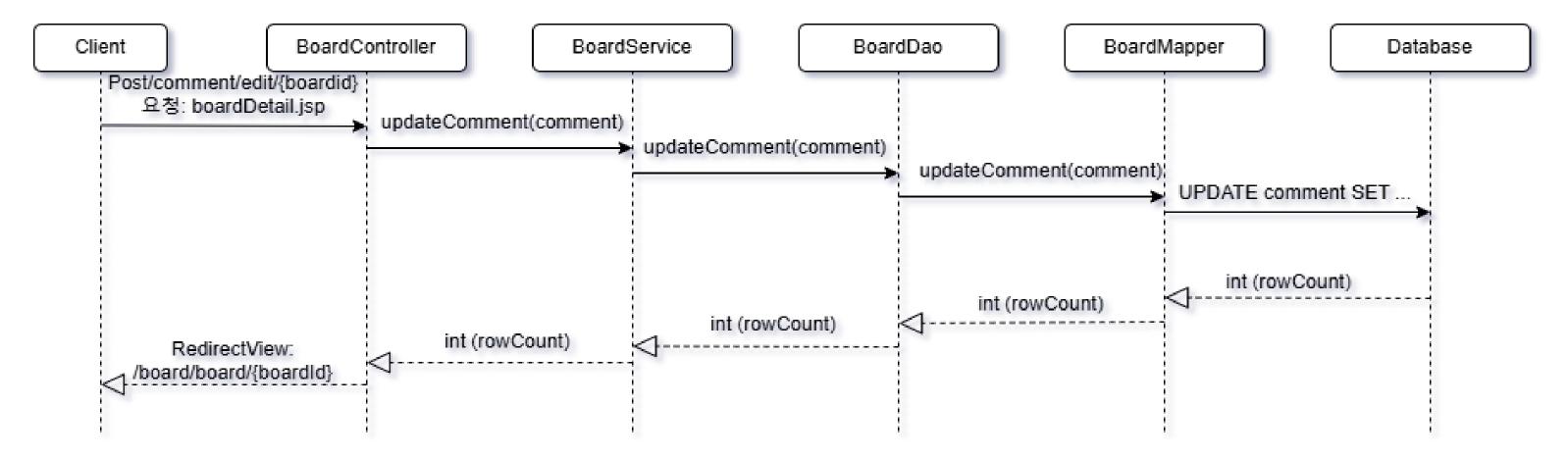
보드(Board)

RedirectView: /board/board/{boardId}



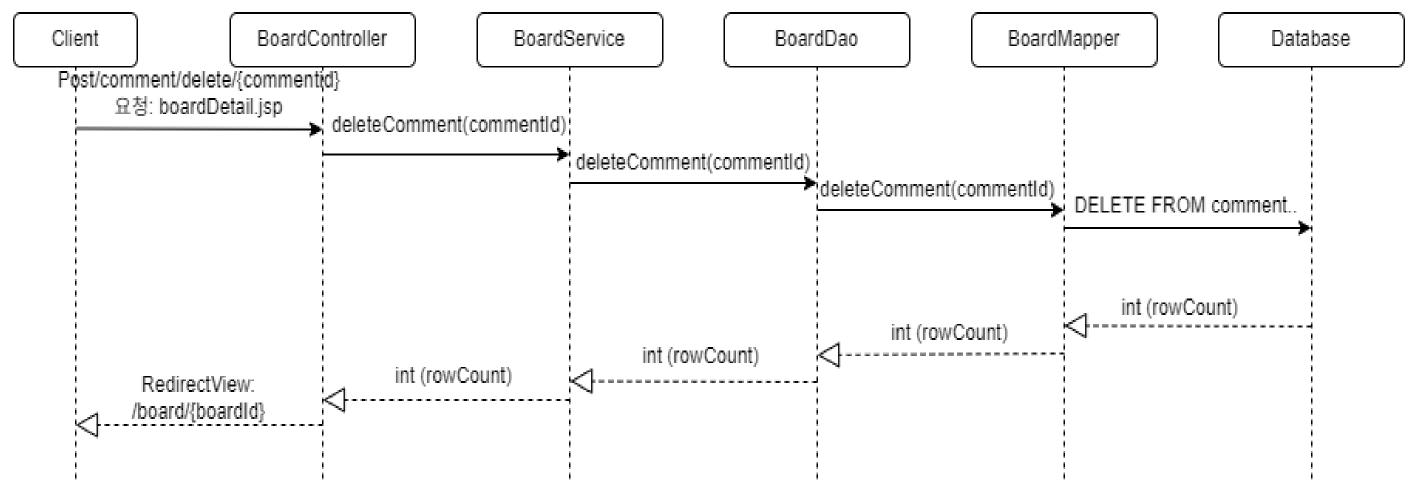
The Dish

보드(Board)



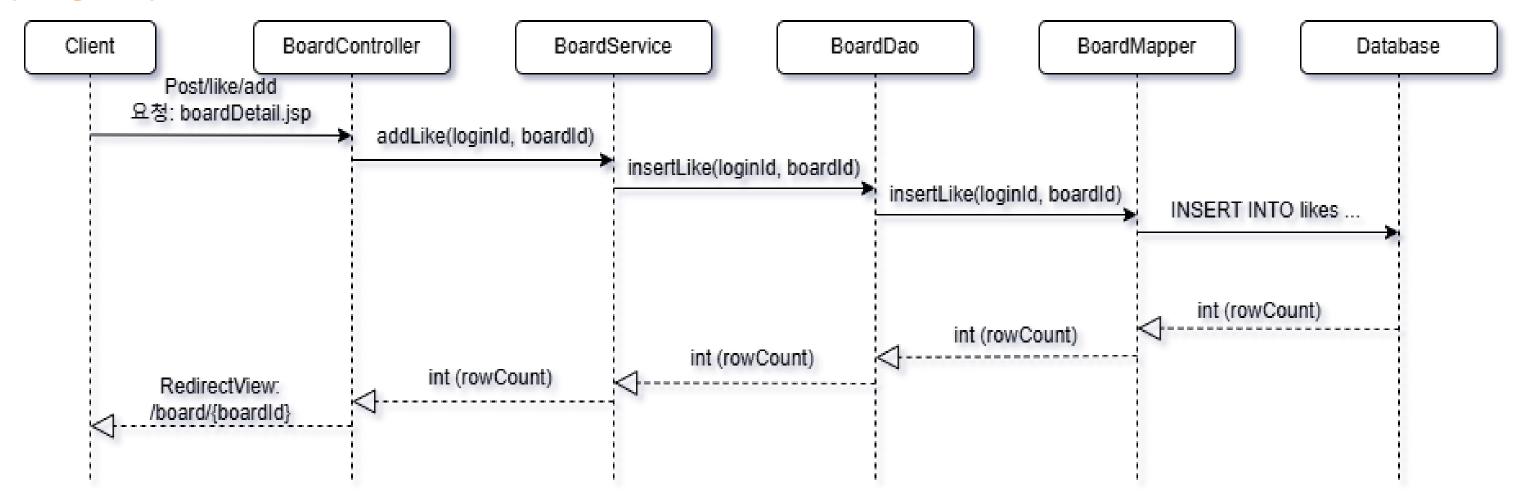


보드(Board)



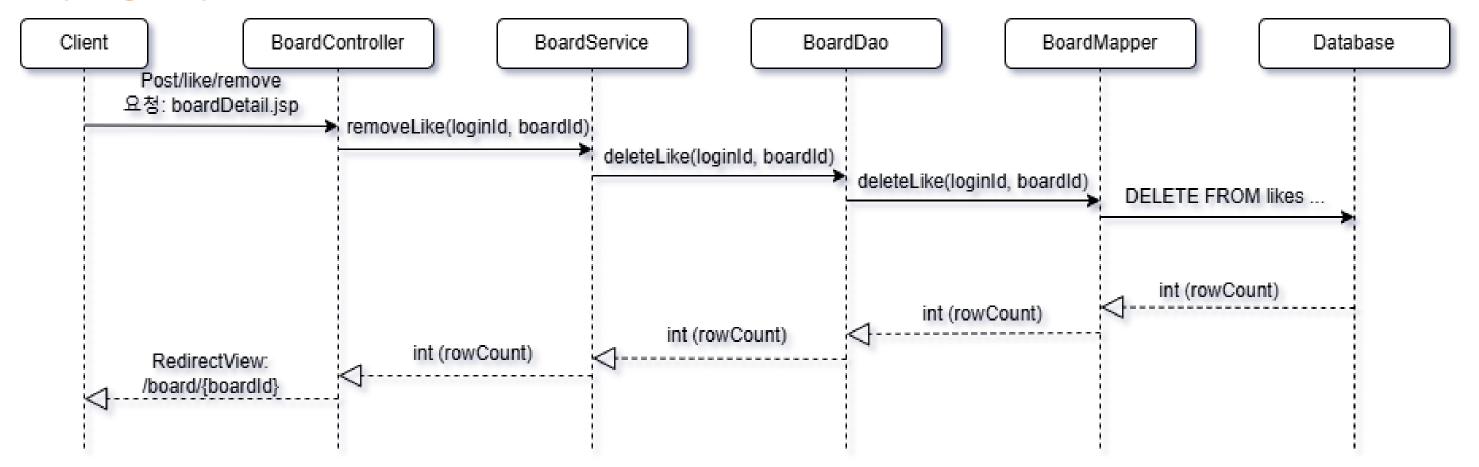


보드(Board)



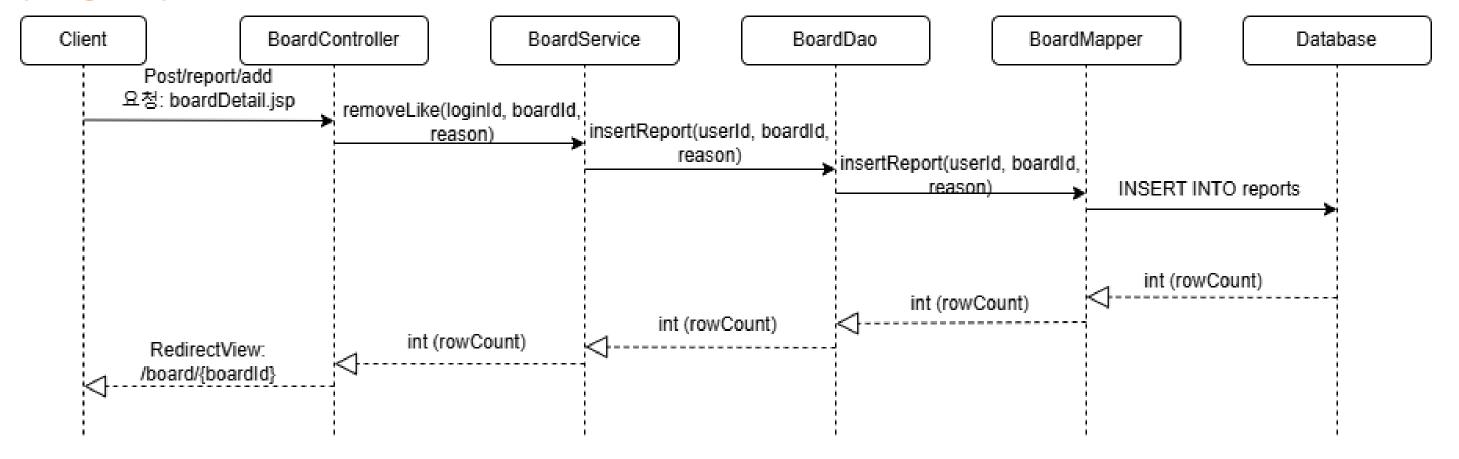


보드(Board)





보드(Board)

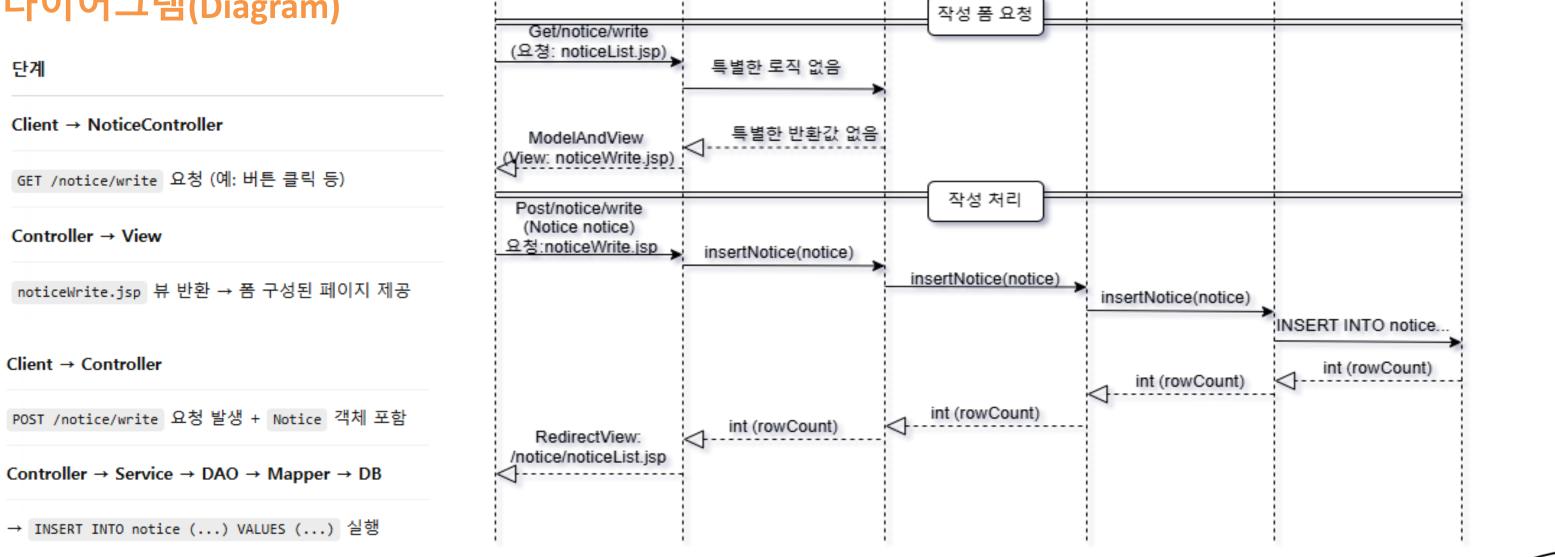




Client

게시판(Notice)

다이어그램(Diagram)



NoticeService

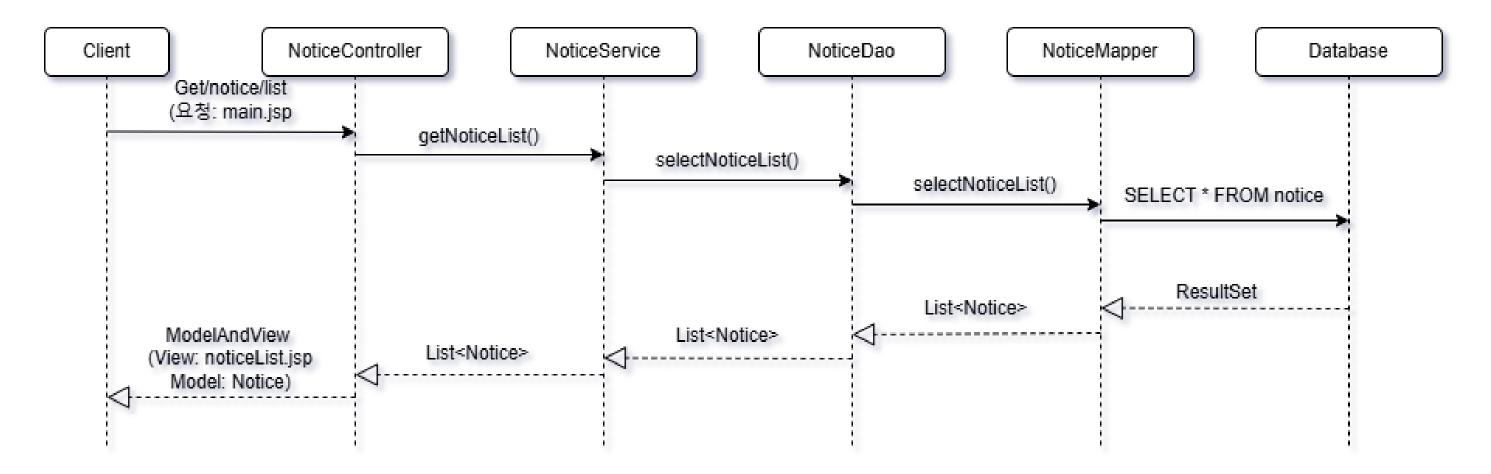
NoticeDao

NoticeMapper

Database

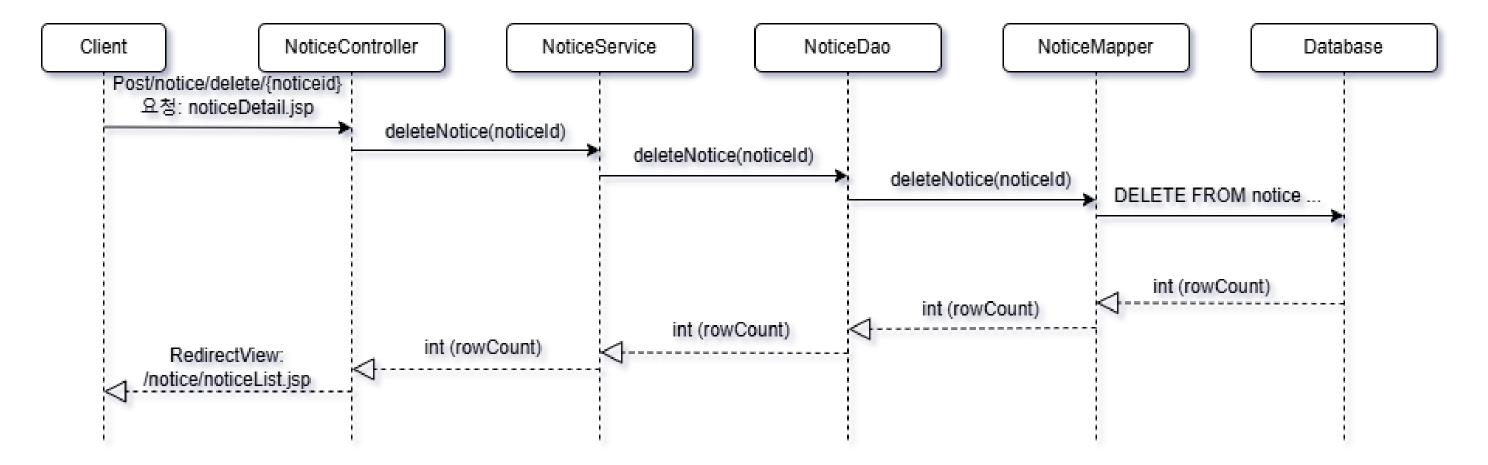
NoticeController

게시판(Notice)



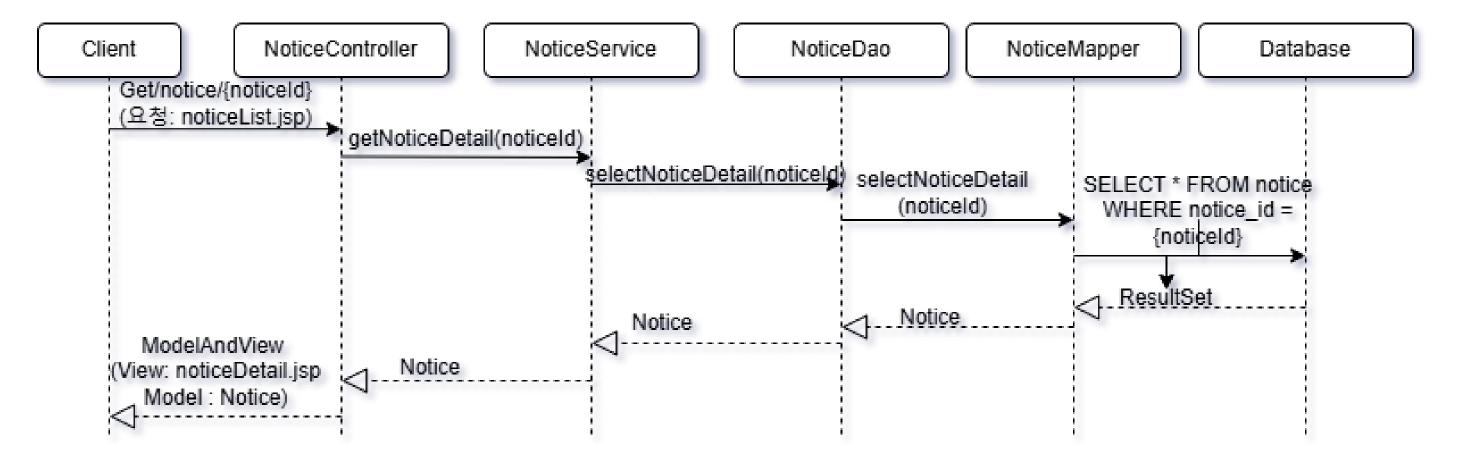


게시판(Notice)





게시판(Notice)

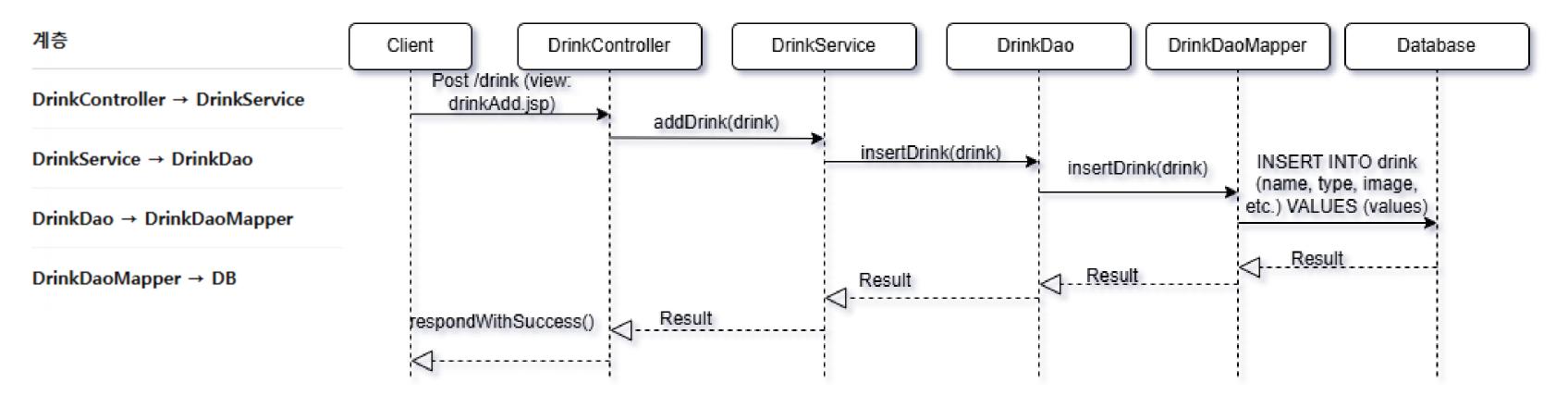




게시판(Notice)

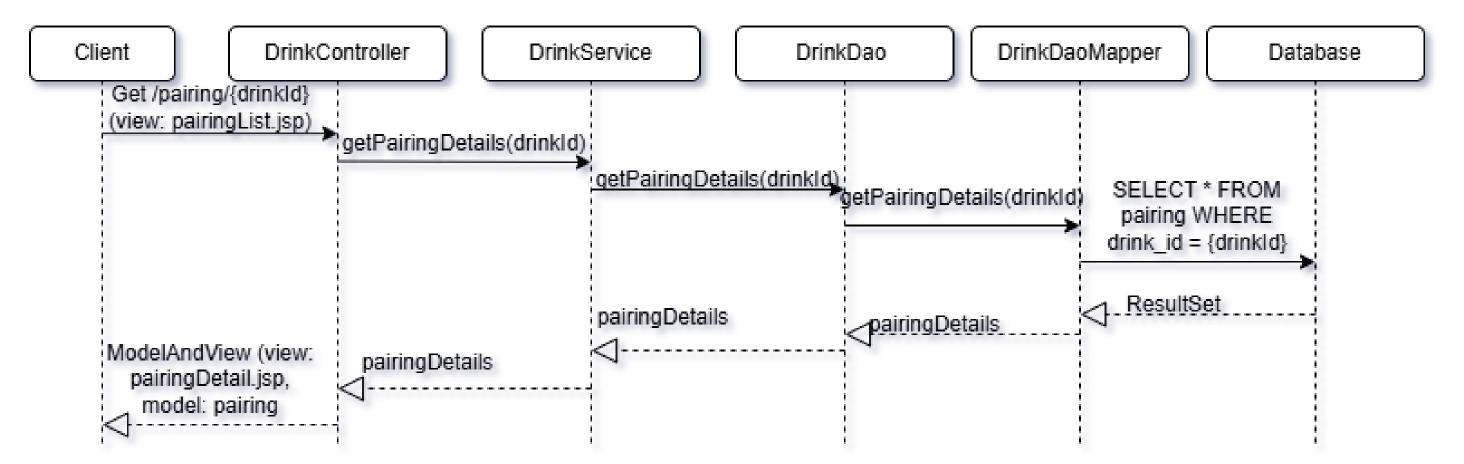
다이어그램(Diagram) NoticeDao NoticeMapper NoticeController NoticeService Database 수정 폼 요청 Get/notice/edit{noticeId} (요쳥: noticeDetail.jsp) getNoticeDetail(noticeId) selectNoticeDetail(noticeId) selectNoticeDetail(noticeId) NoticeDetail(noticeId) ModelAndView ResultSet (View: noticeEdit.jsp Notice Notice Notice Model: Notice) 수정 처리 Post/notice/edit {noticeId} 요청:noticeEdit.jsp updateNotice(notice) updateNotice(notice) updateNotice(notice) UPDATE notice SET int (rowCount) int (rowCount) int (rowCount) int (rowCount) RedirectView: /notice/noticeList.jsp **-----**

음료 & 술 서치



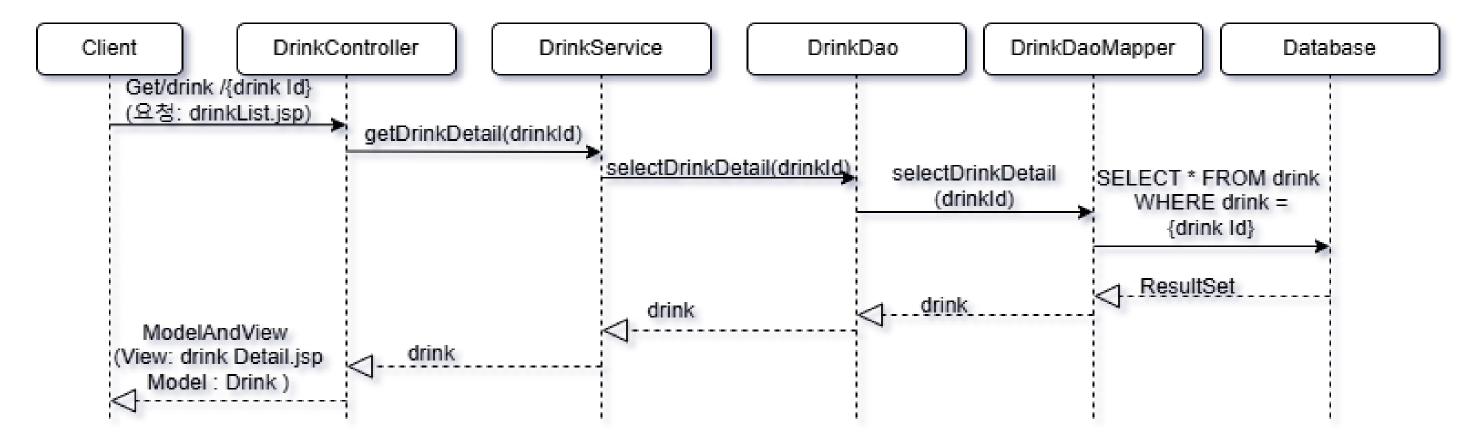


음료 & 술 서치



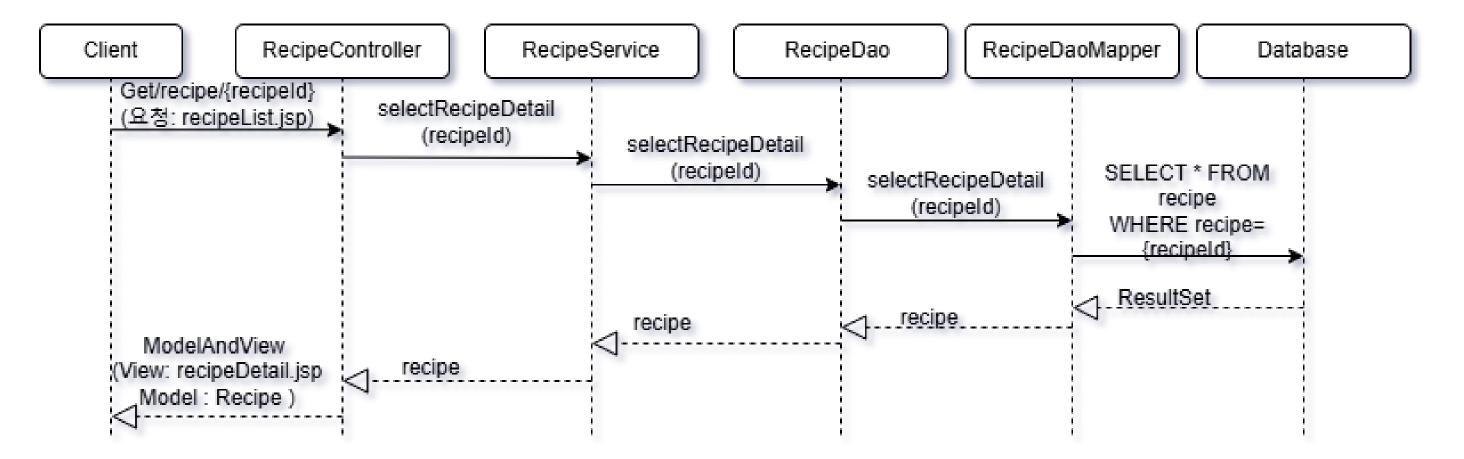


음료&술서치





레시피(Recipe)





레시피(Recipe)

다이어그램(Diagram)

전체 레시피 조회 처리

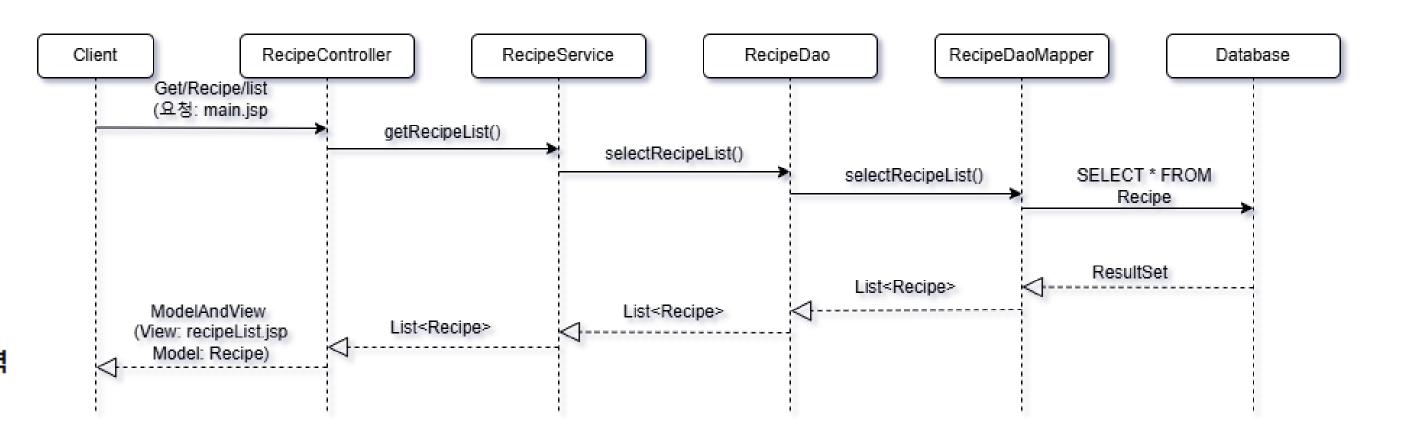
단계

Controller → Service

Service → DAO

DAO → Mapper → DB

데이터 반환 및 페이지 출력



단계

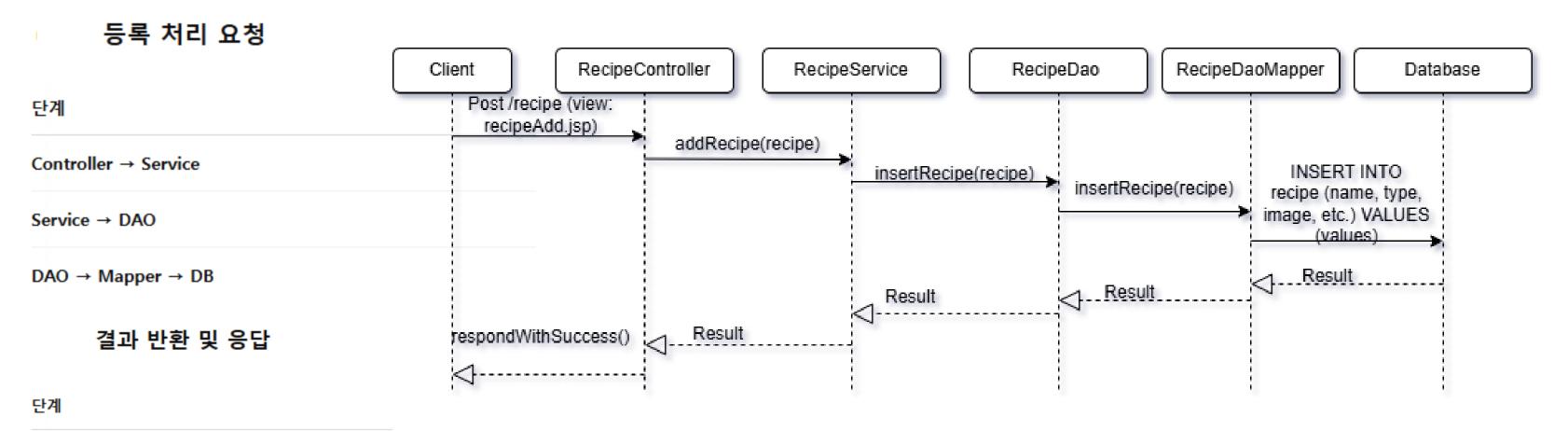
DB → Mapper → Dao → Service → Controller

Controller → Client



레시피(Recipe)

다이어그램(Diagram)



DB → Mapper → DAO → Service → Controller

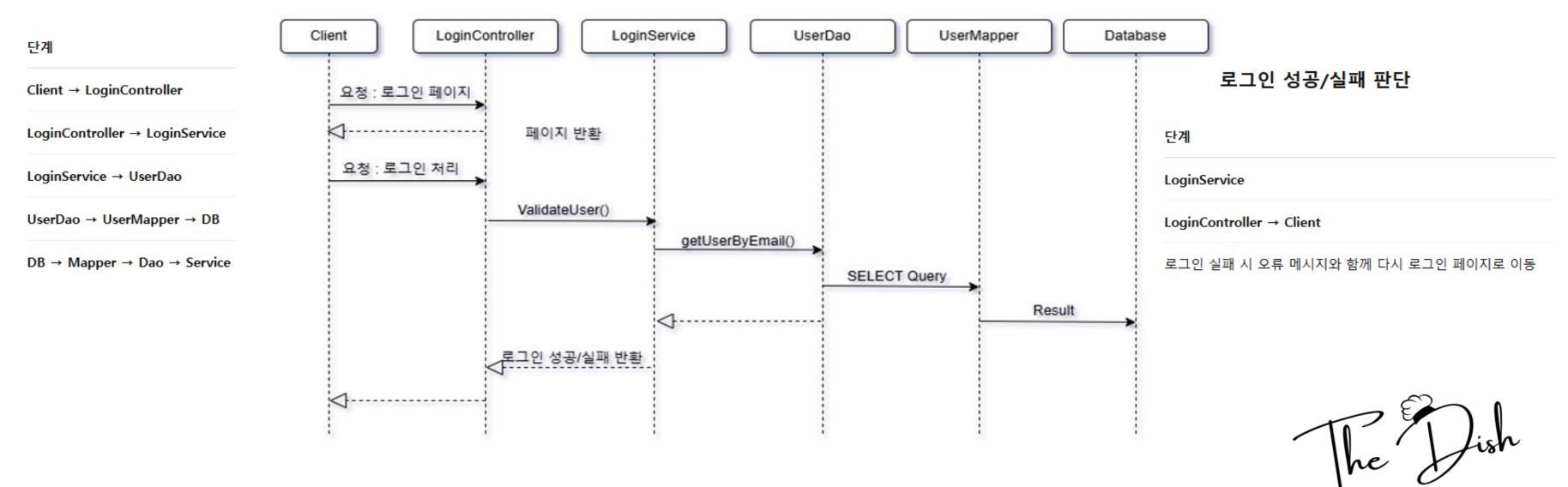
Controller → Client

The Dish

유저(Users)

다이어그램(Diagram)

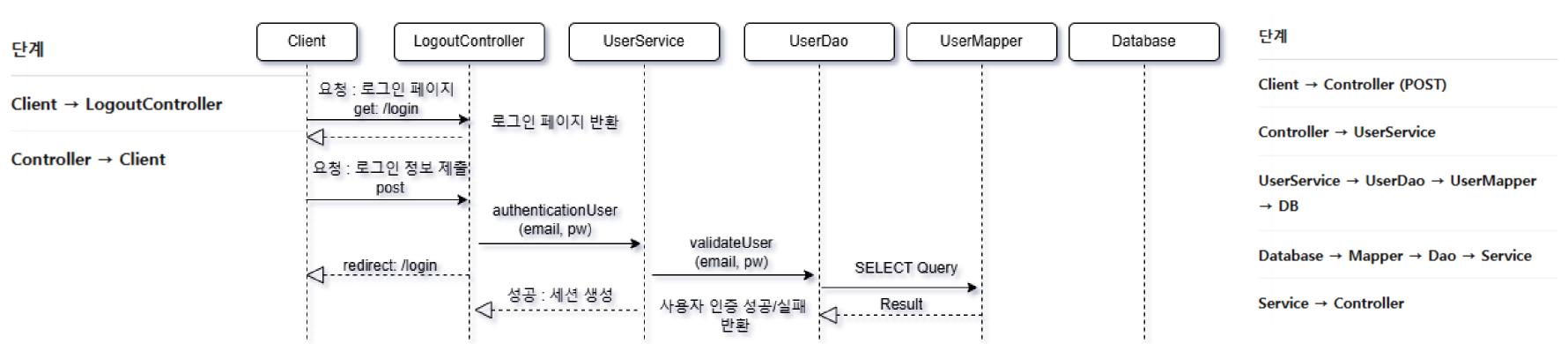
로그인 정보 제출 및 인증 처리



유저(Users)

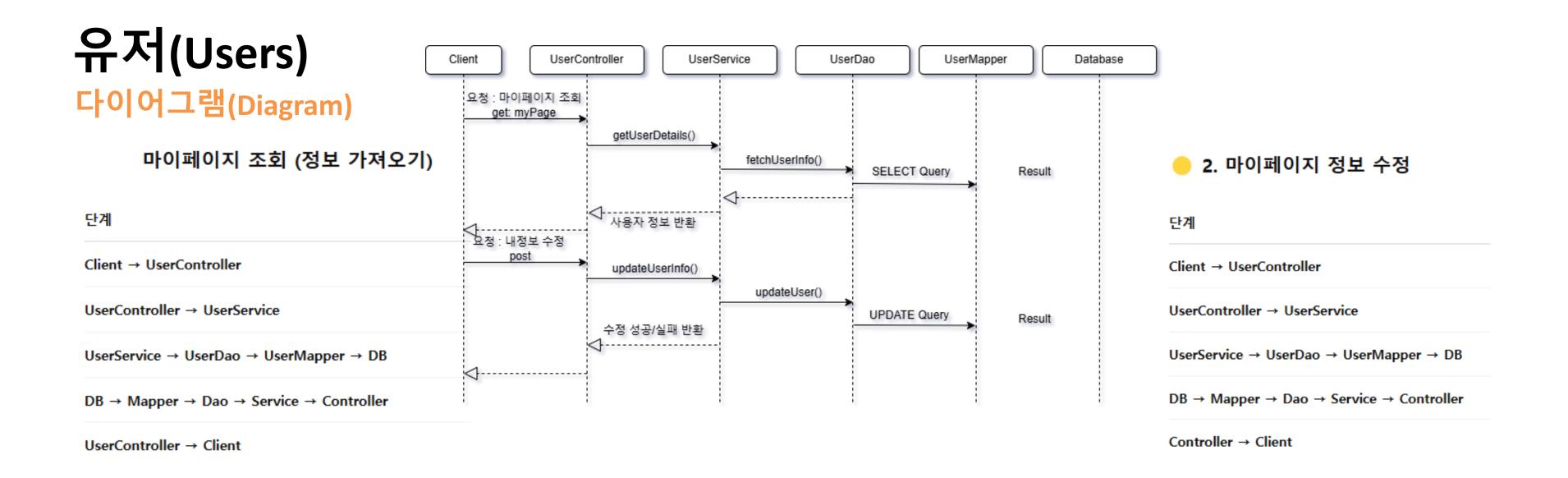
다이어그램(Diagram)

로그인 페이지 요청



T Dish

로그인 정보 제출





유저(Users)

다이어그램(Diagram)

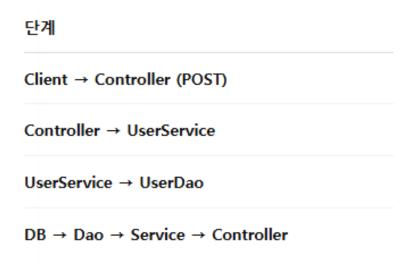
아이디 찾기 페이지 요청

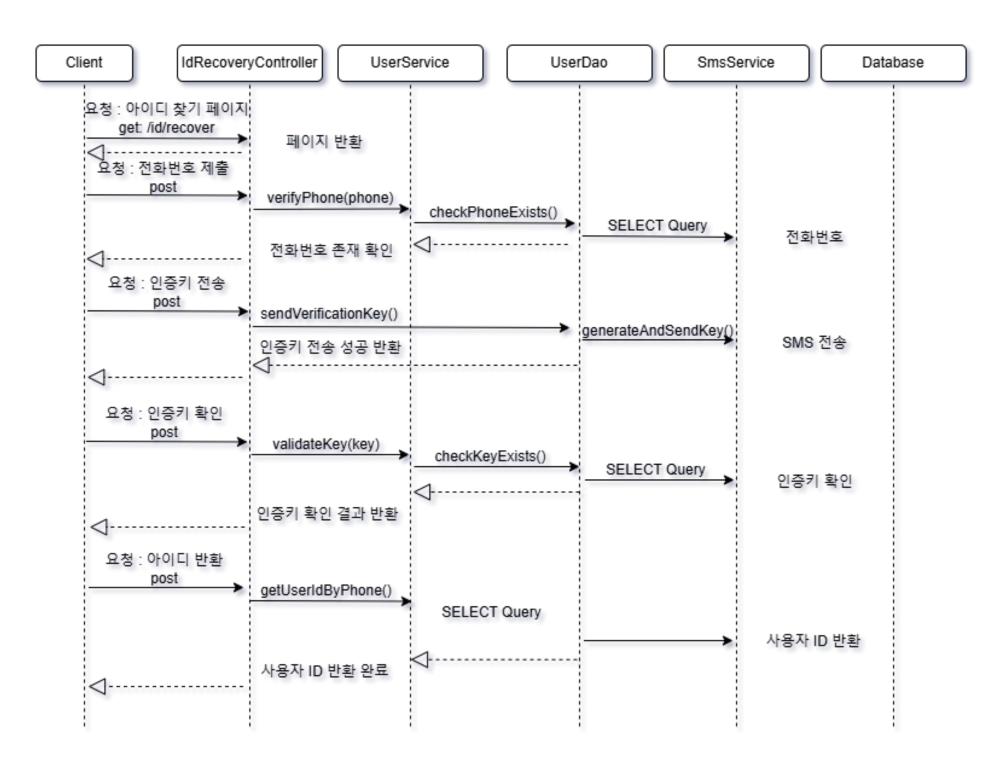
단계

Client → IdRecoveryController

Controller → Client

전화번호 제출 및 존재 확인





인증키 전송

단계

Client → Controller (POST)

Controller → UserService

UserService → SmsService

결과 반환

인증번호 입력 → 검증

단계

Client → Controller (POST)

Controller → Service → Dao → DB

결과 반환



유저(Users)

다이어그램(Diagram)

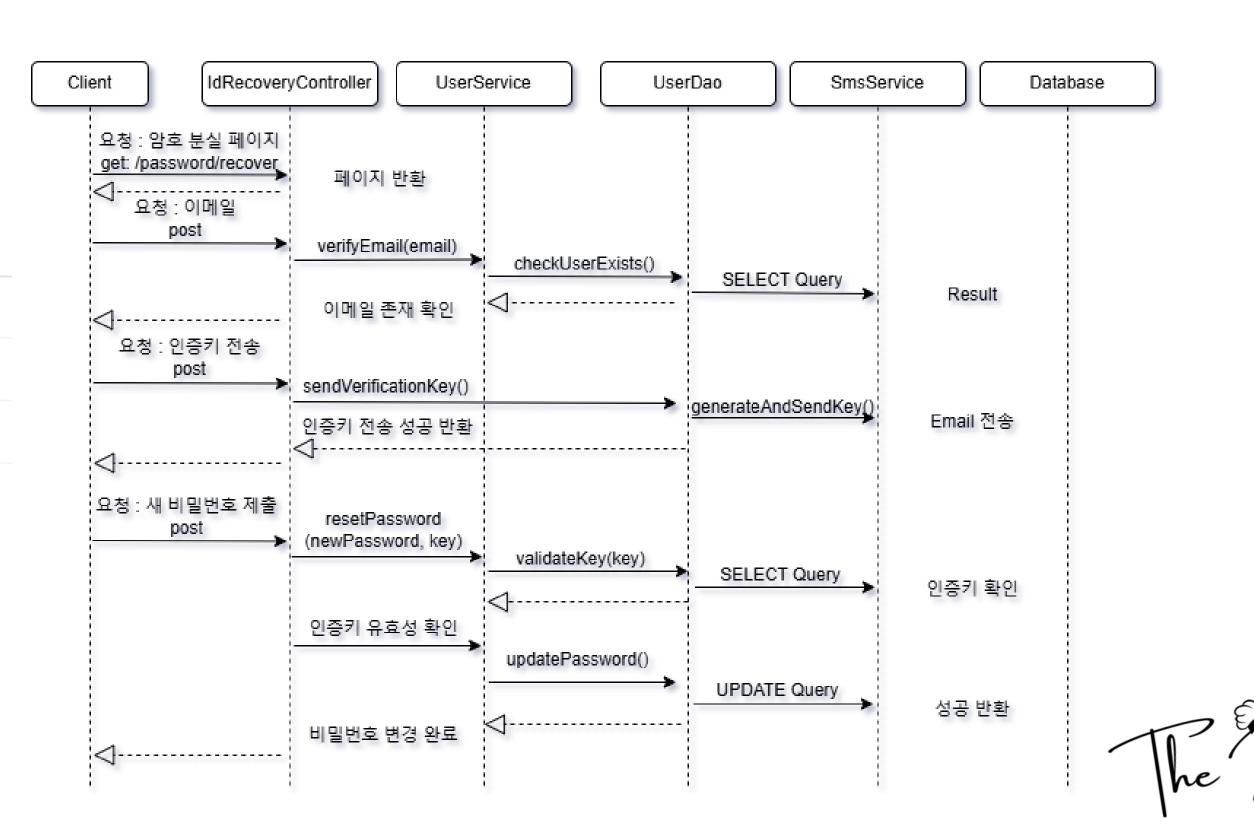
단계

Client → Controller (POST)

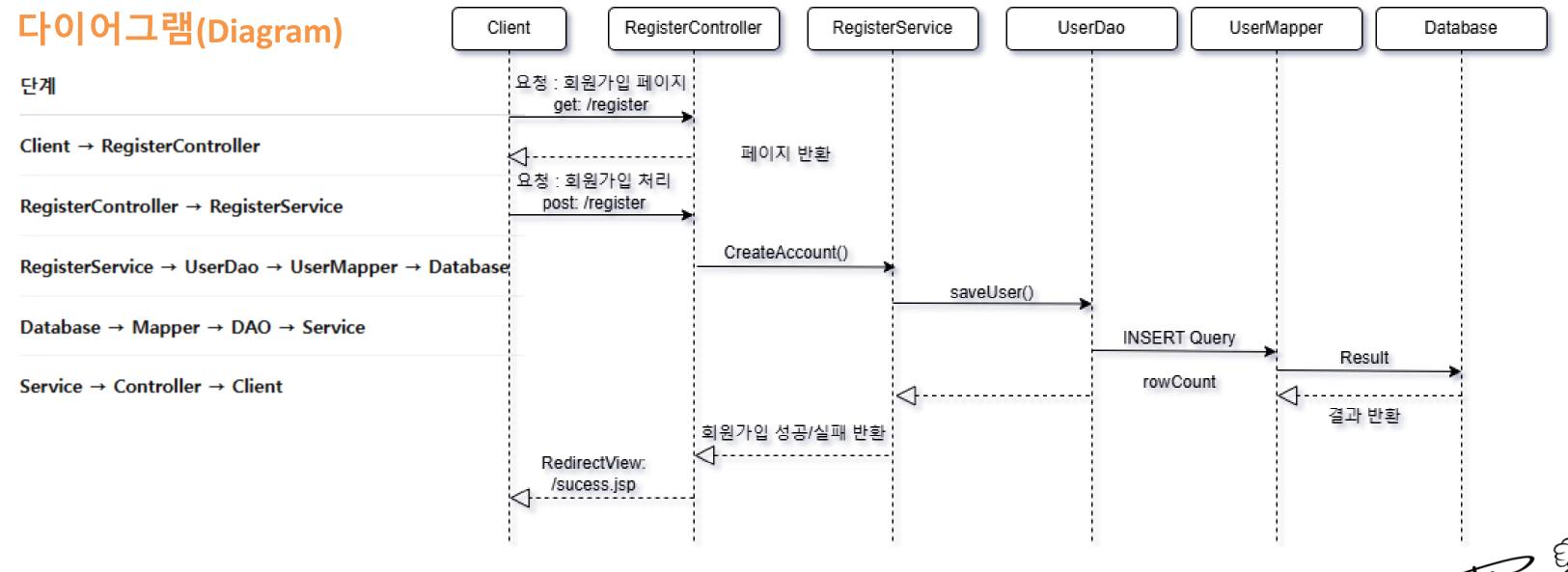
Controller → UserService

UserService → UserDao

DB → Dao → Service → Controller

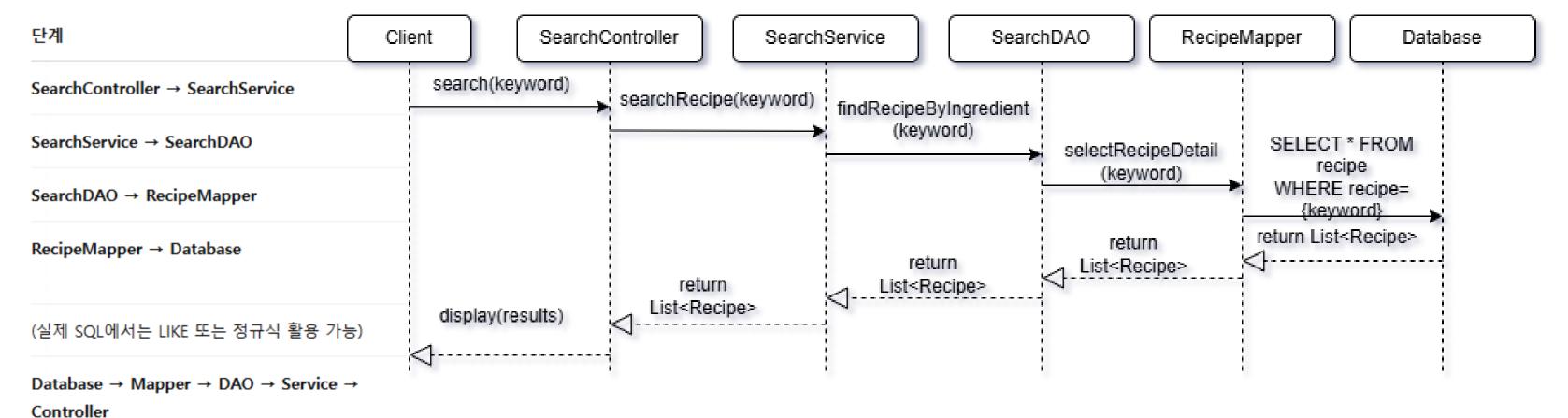


유저(Users)



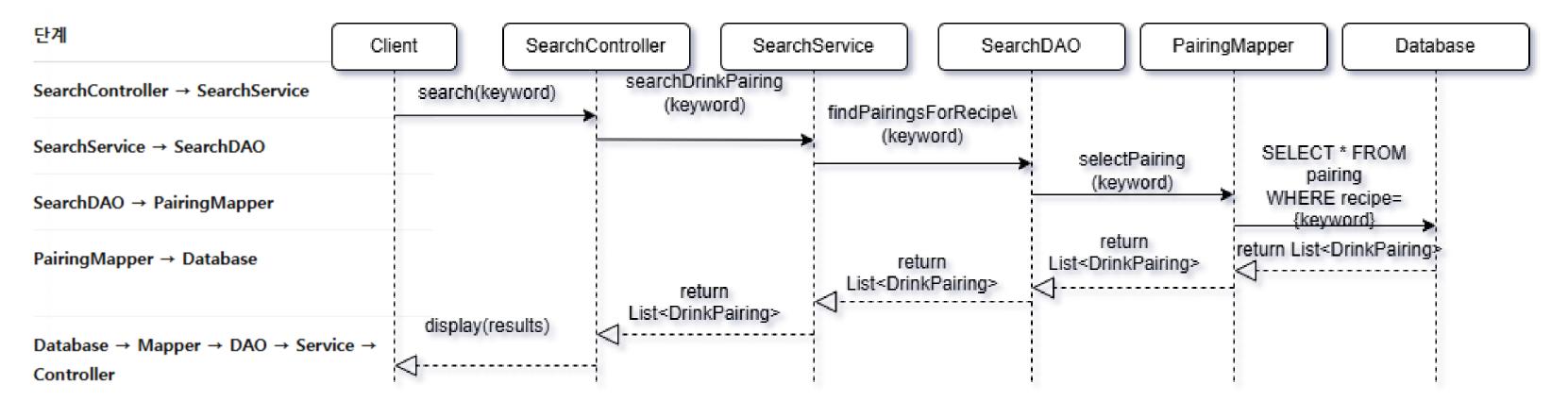
서치(Search)

다이어그램(Diagram)



The Dish

서치(Search)





Thank You

The Dish

