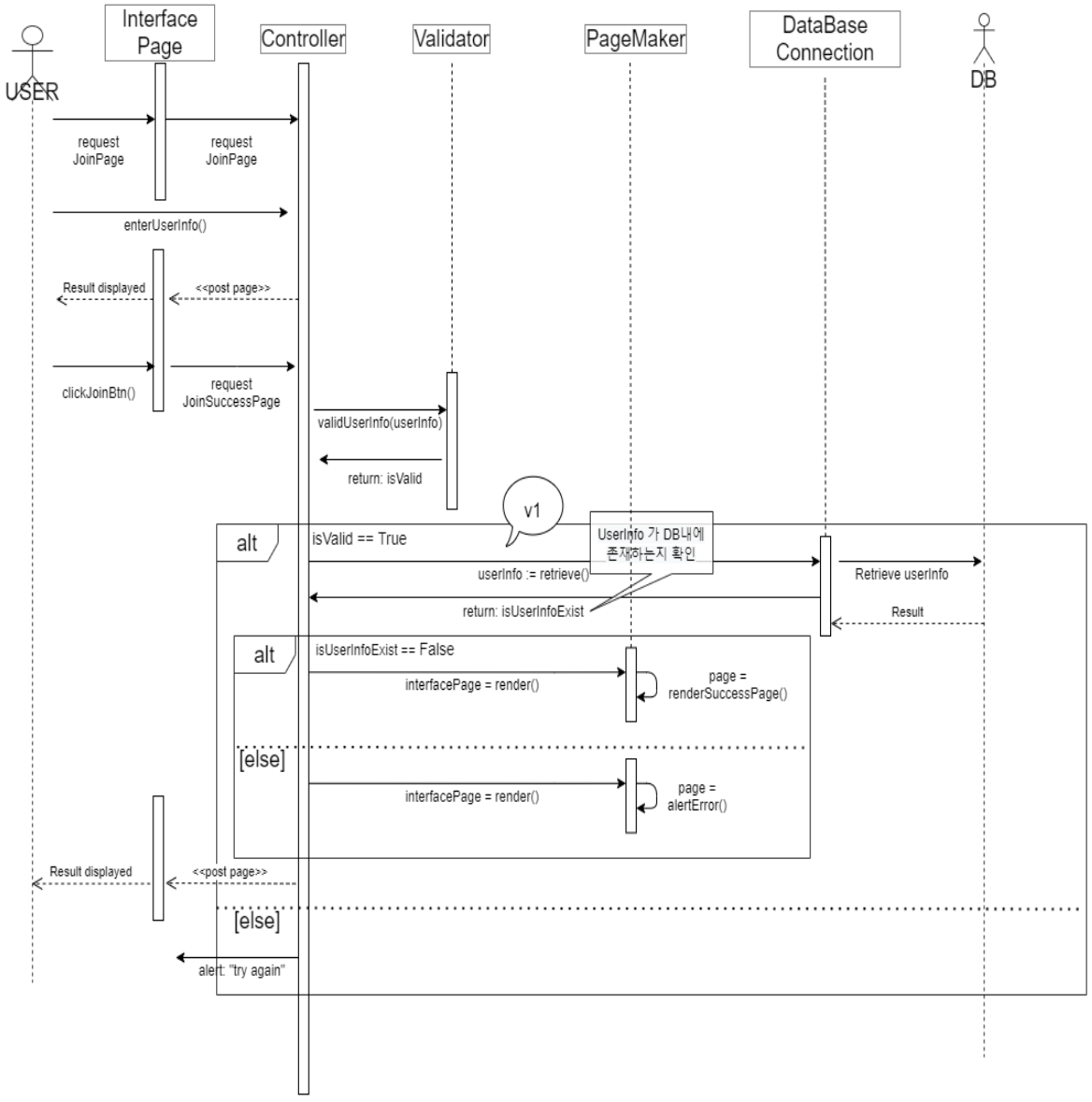
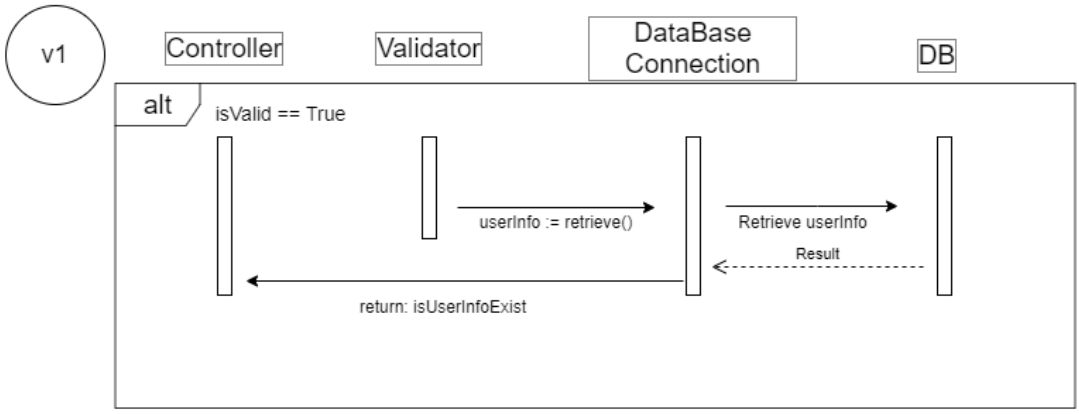
Sub group1

[sequence diagram for UC-1]

UC1: JOIN

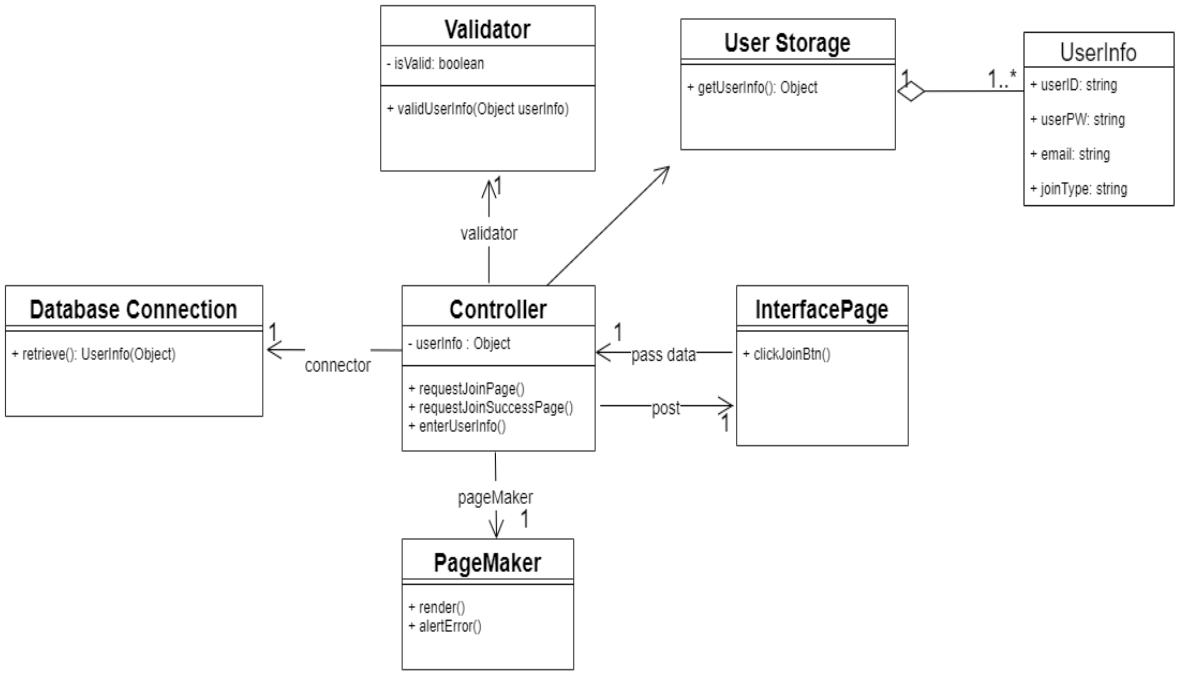


Variation 1



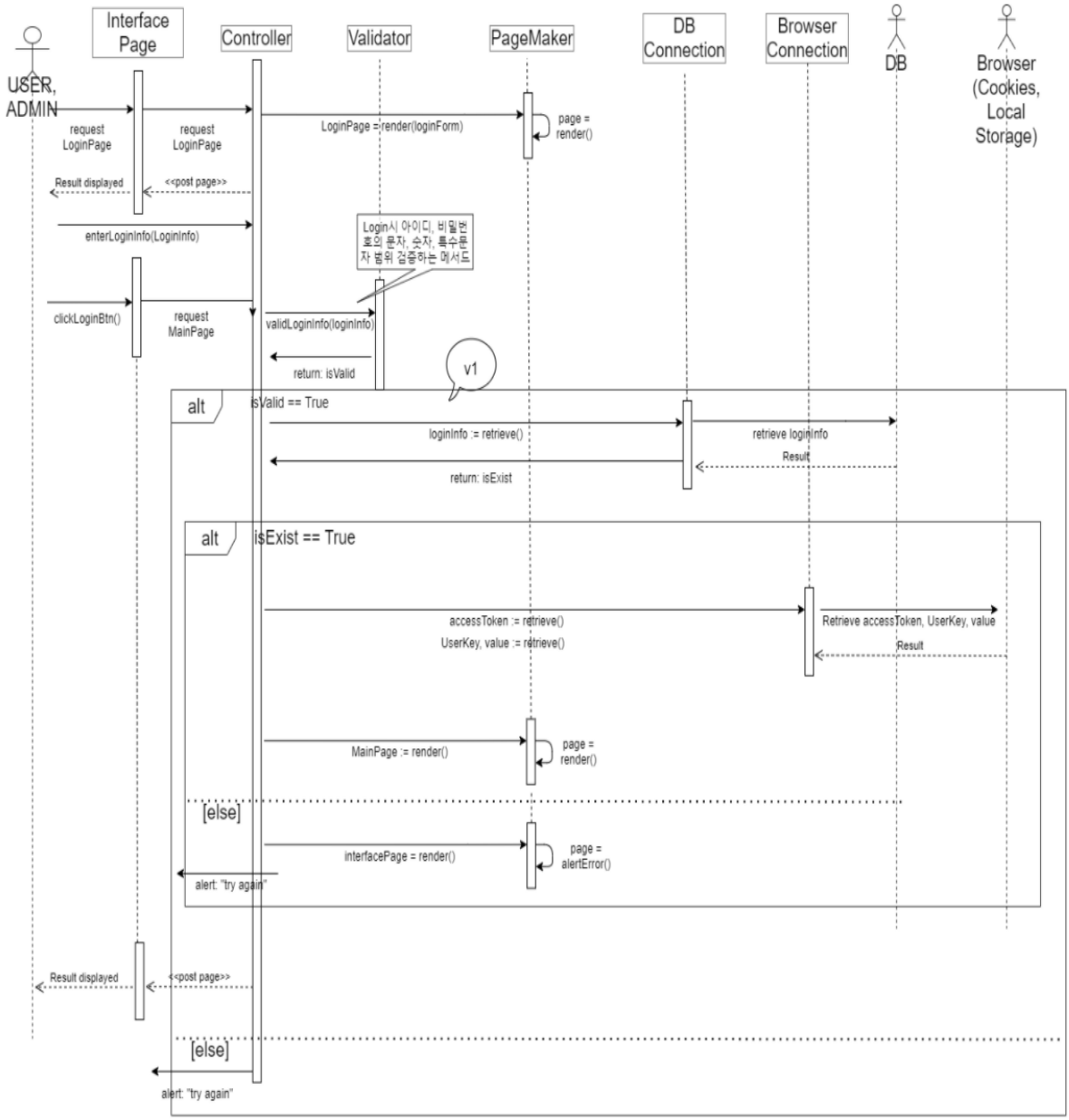
[Process]
초기 설계 당시에는 Controller 에 전문성을 부여하여 validator 에서 회원가입 양식의 form 이 validate 하다면 controller 에서 valid 한 회원가입 정보를 데이터베이스에 전달하는 방식이였다면 variation1 에서는 validator 에서 자체적으로 검증하고 controller 에 valid 를 던지는 방식이 아닌 해당 step 에서 바로 데이터베이스에 회원가입 정보를 전달하는 방식을 고안하였다. 이를 통해 얻게 되는 이점은 controller 에 하중된 책임들을 다른 domain concept 로 분산시킴으로써 workload 를 밸런스하게 맞출 수 있고 각 object 간의 short communication chain 이 가능하다.

[class diagram for UC-1]

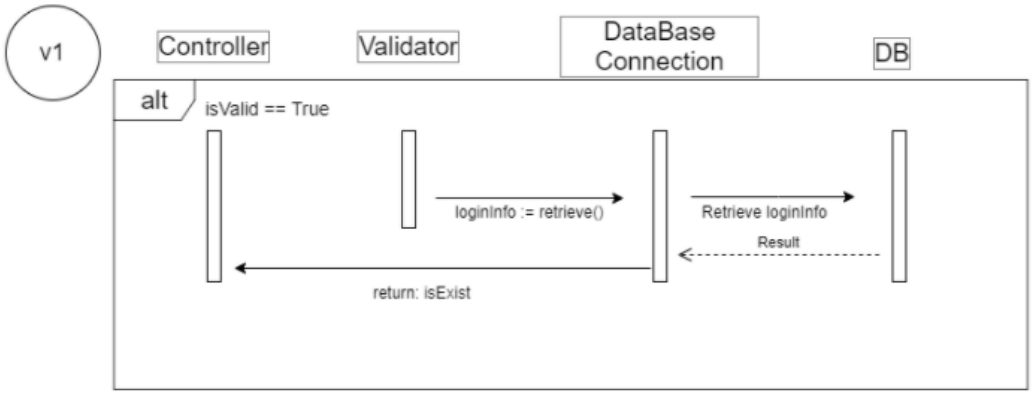


[sequence diagram for UC-2]

UC2: Login



Variation 1



[Process]
Usecase 1 과 마찬가지로 초기 설계 당시에는 Controller 에 전문성을 부여하여 validator 에서 로그인의 정규표현방식이 validate 하다면 controller 에서 valid 하다는 것을 리턴받고 valid 한 로그인 정보를 데이터베이스에 전달하는 방식이였다면 variation1 에서는 validator 에서 자체적으로 검사하고 controller 에 valid 를 던지는 방식이 아닌 해당 step 에서 바로 데이터베이스에 로그인 정보를 전달하는 방식을 고안하였다. 이를 통해 얻게 되는 이점은 마찬가지로 controller 에 하중된 책임들을 다른 domain concept 로 분산시킴으로써 workload 를 밸런스하게 맞출 수 있고 각 object 간의 short communication chain 이 가능하다.

[class diagram for UC-2]