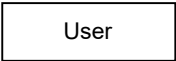
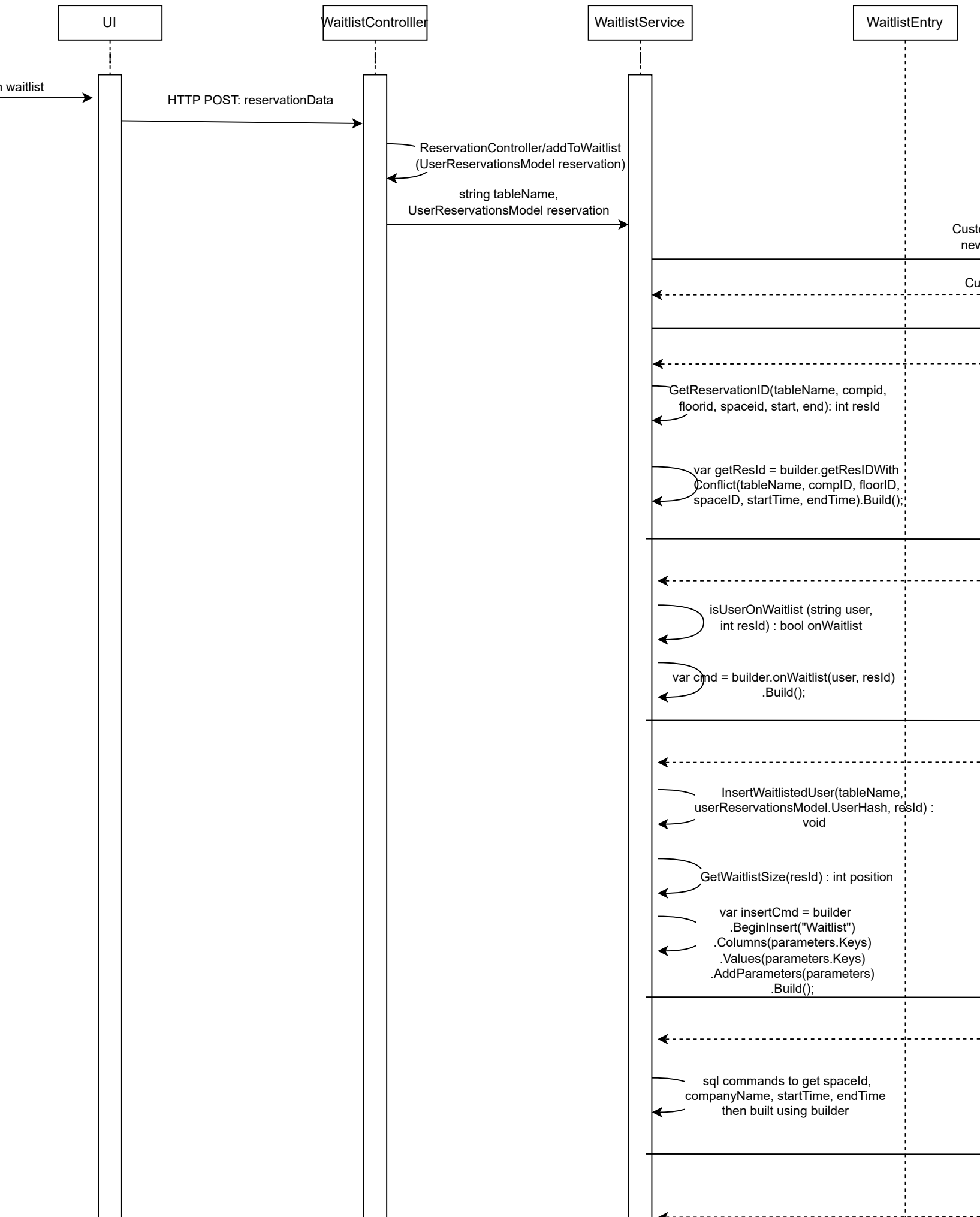
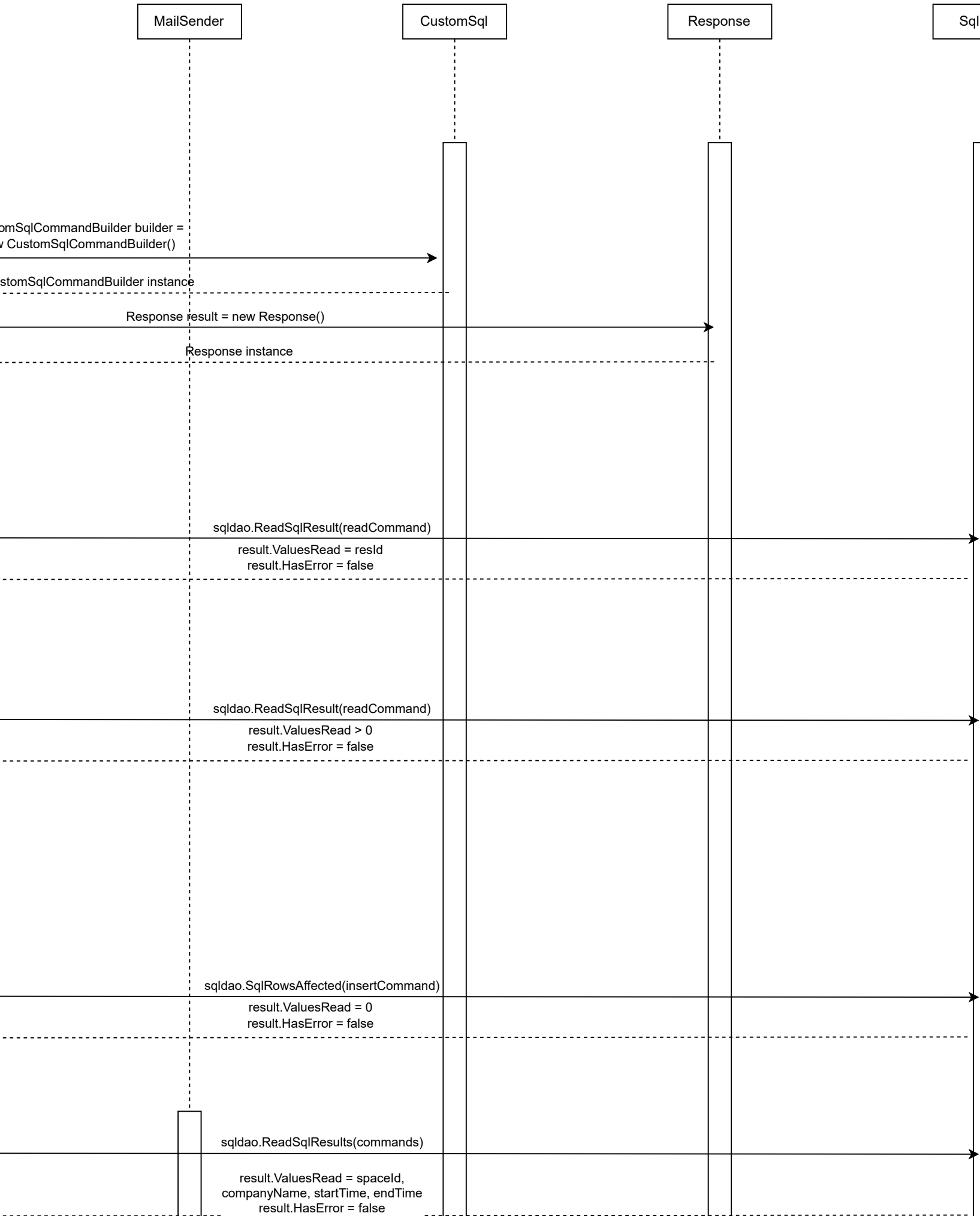


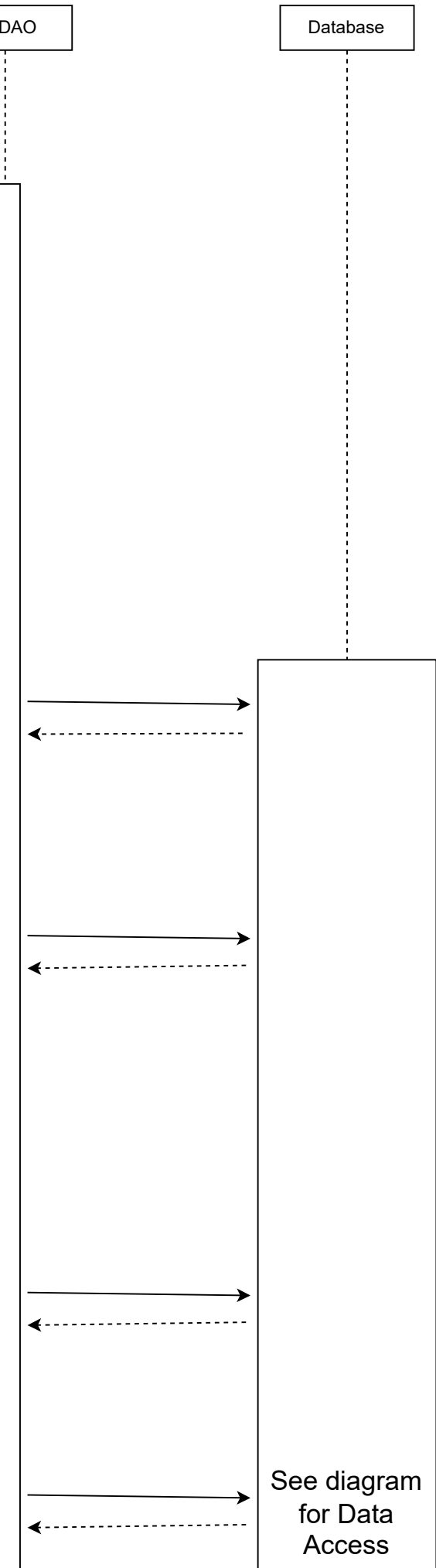

```
reservationData = {
  companyId: int companyId
  floorPlanId: int floorPlanId
  spaceId: string spaceId'
  reservationStartTime: datetime reservation-startTime
  reservationEndTime: datetime reservation-endTime')
  userHash: string username
}
```



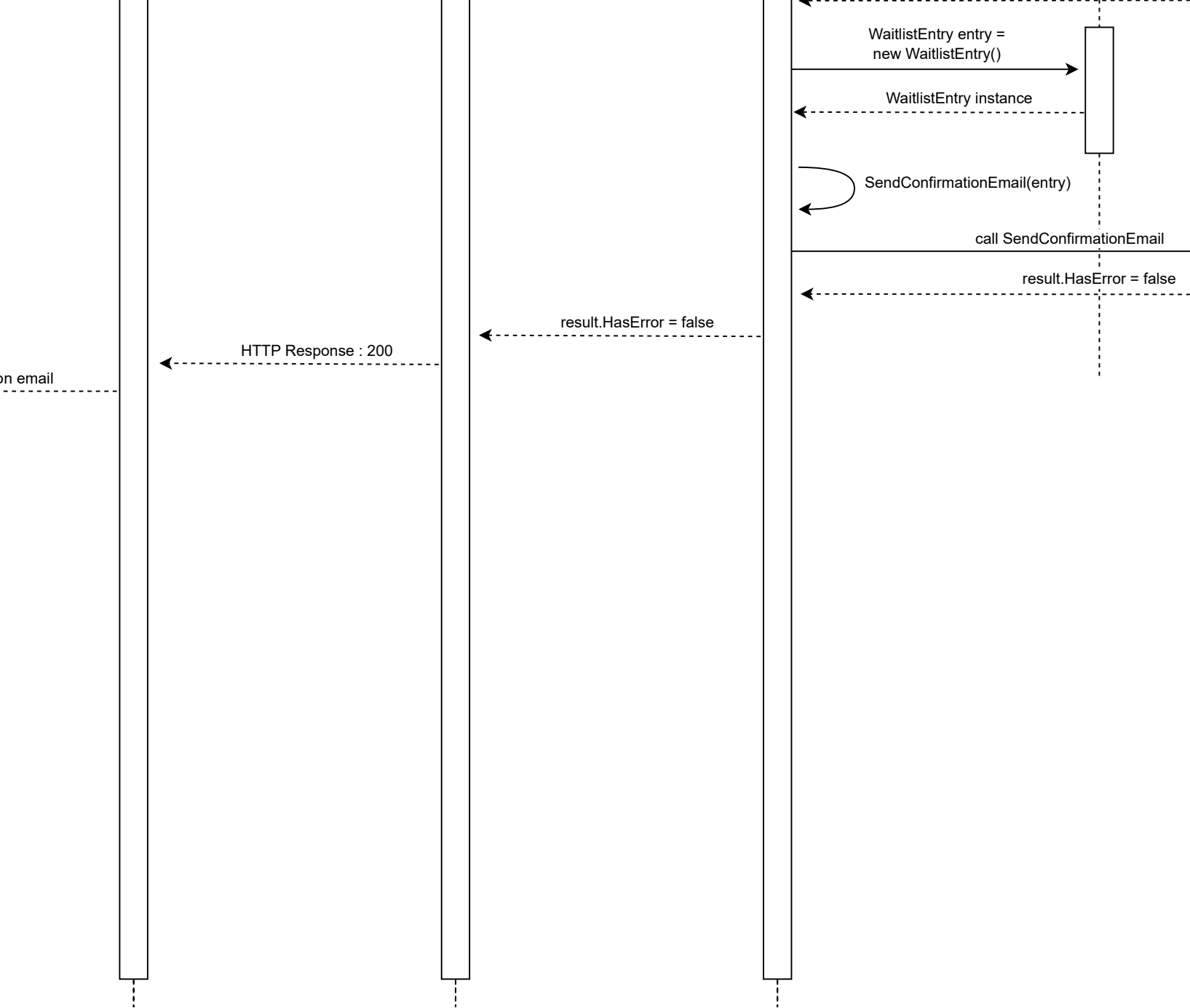
Request to join

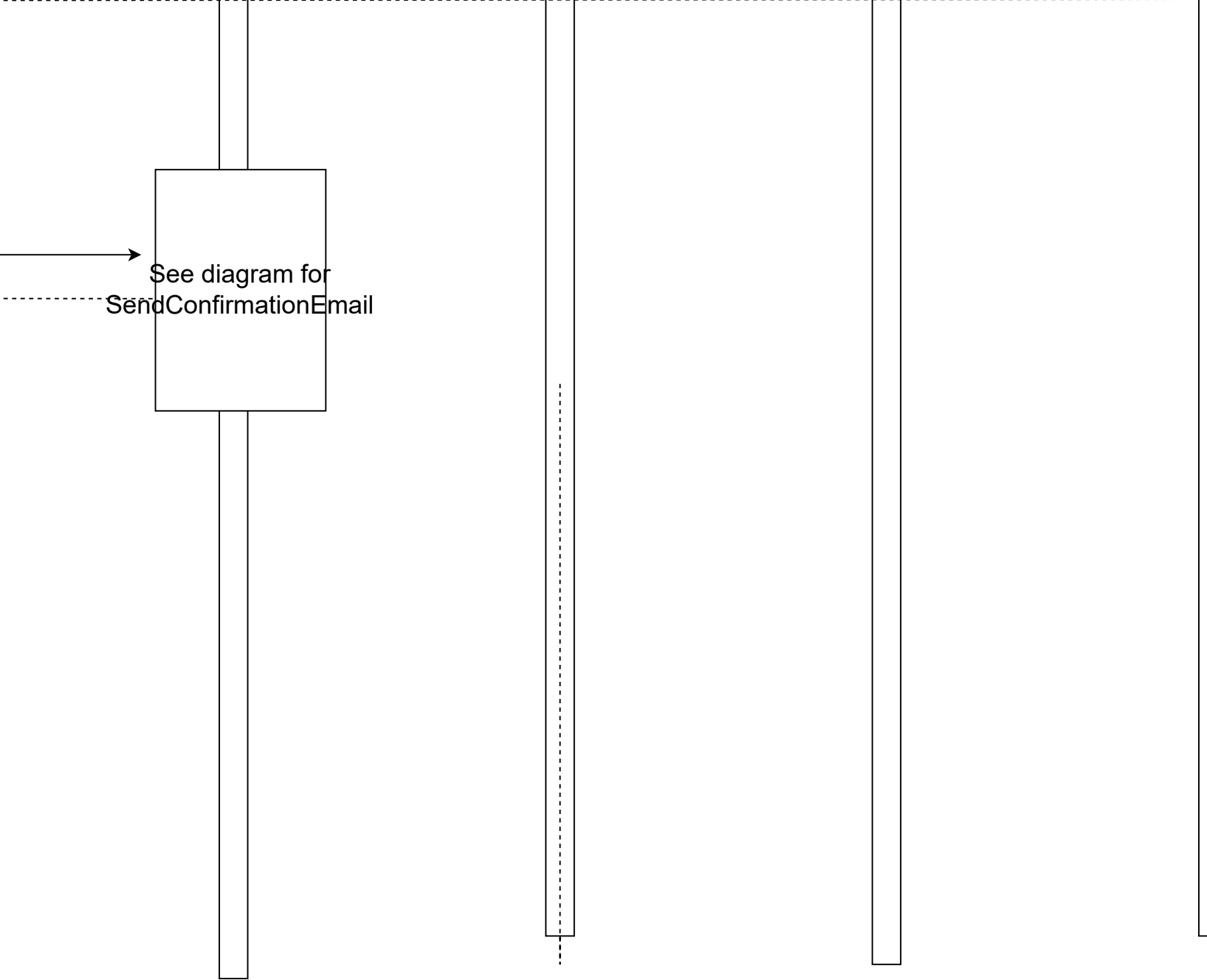


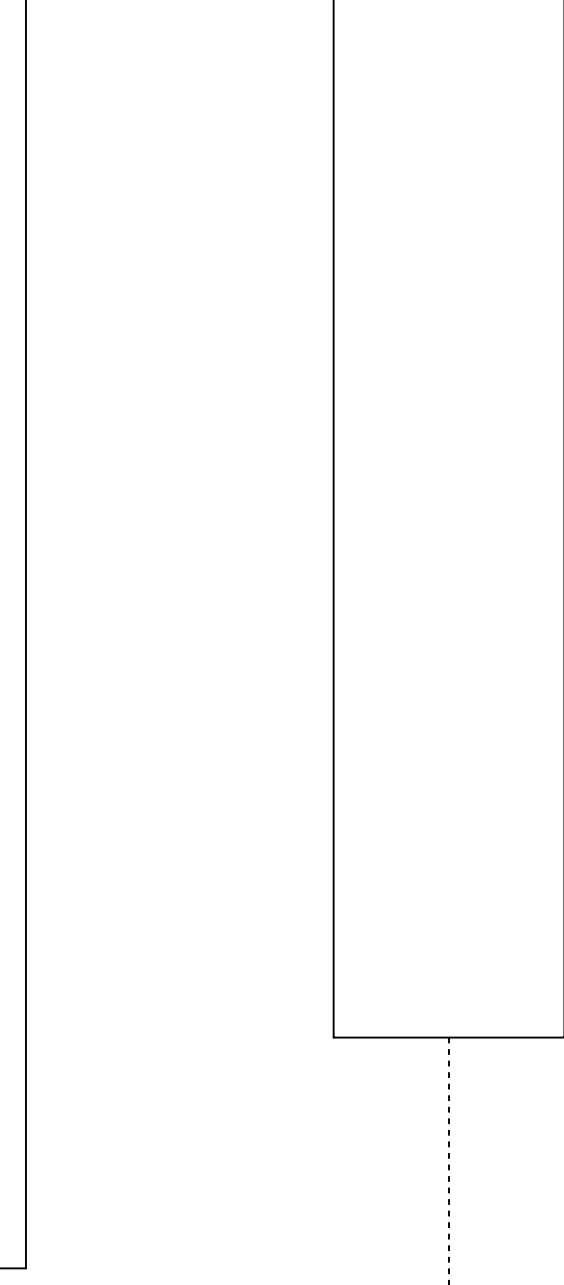




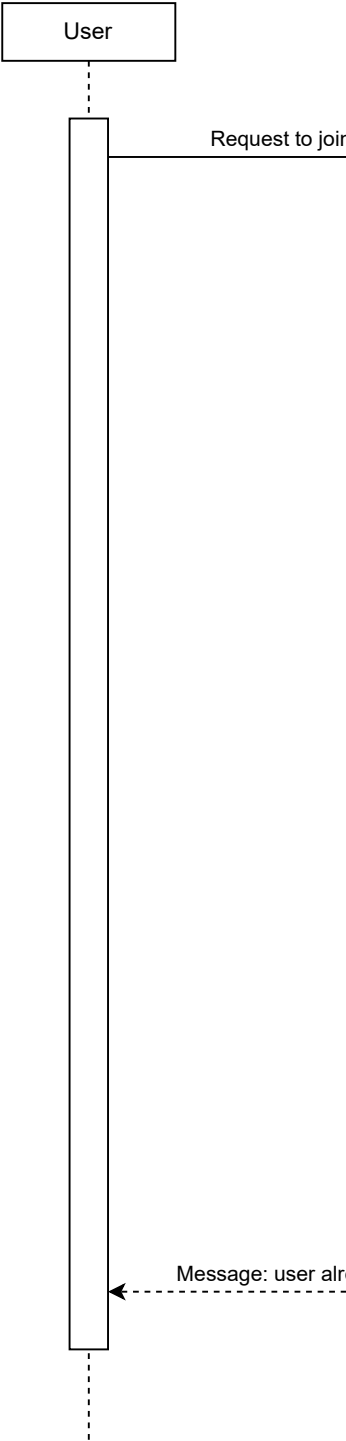
← Confirmation

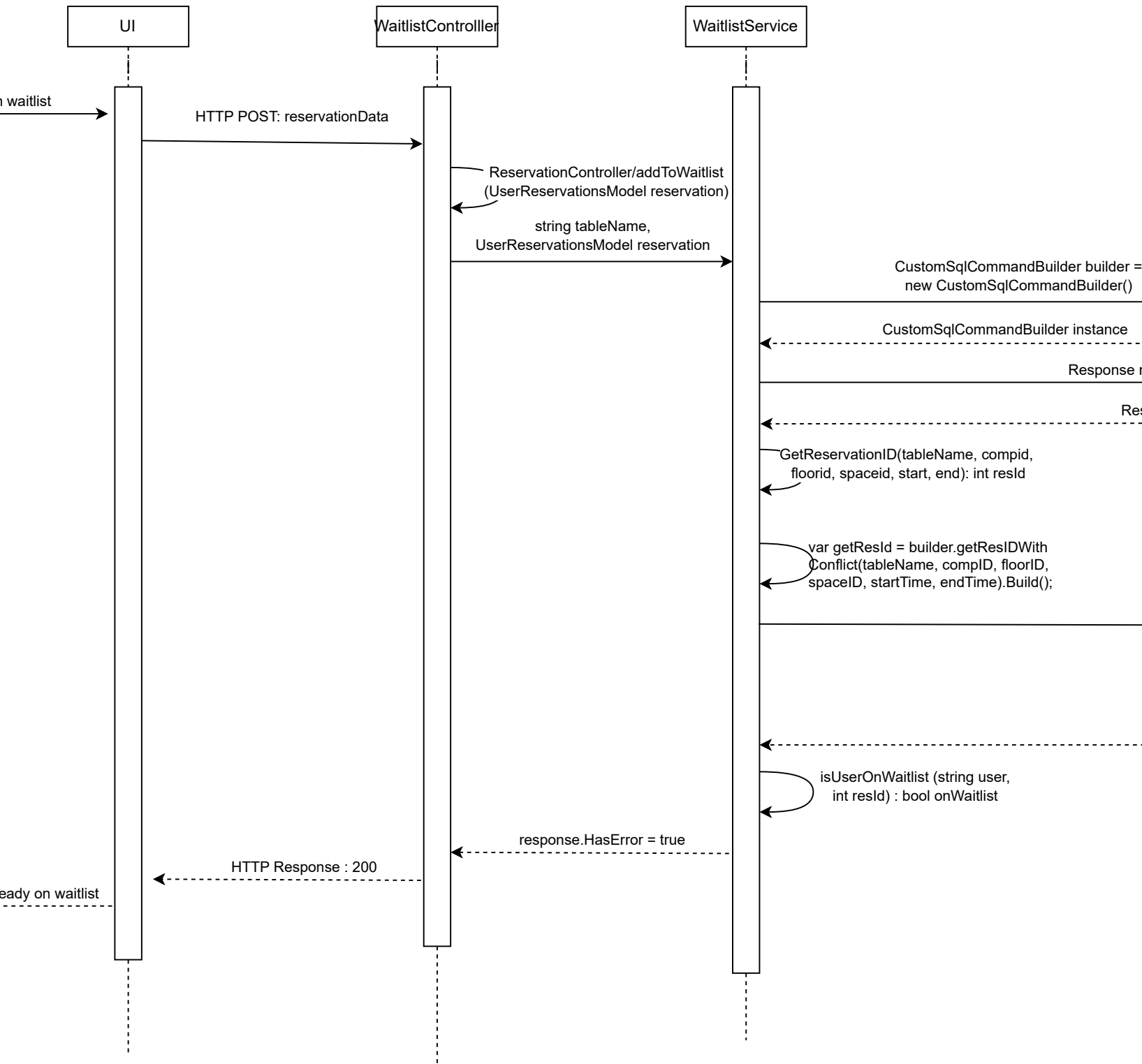


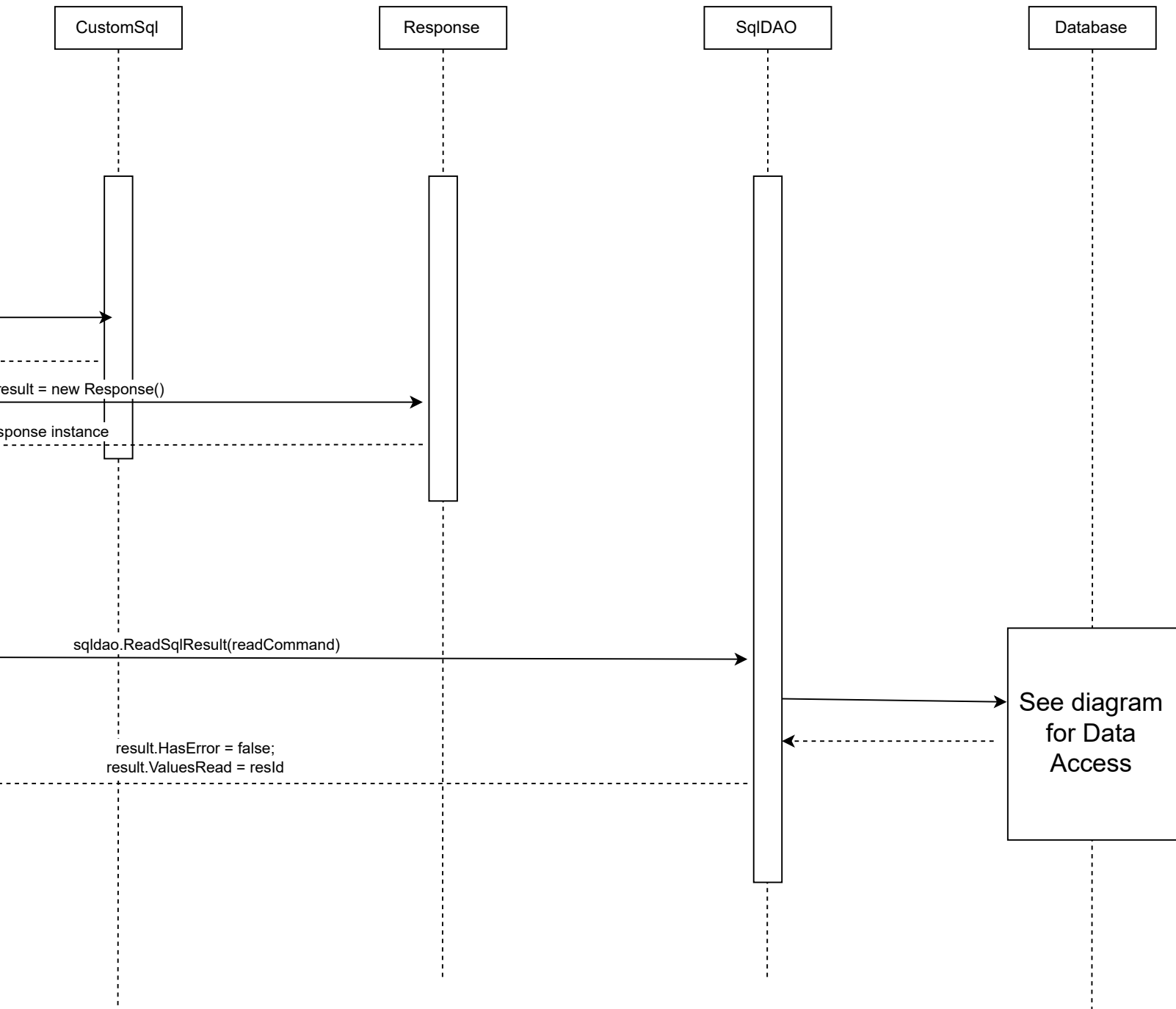





```
reservationData = {
  companyId: int companyId
  floorPlanId: int floorPlanId
  spaceId: string spaceId'
  reservationStartTime: datetime reservation-startTime
  reservationEndTime: datetime reservation-endTime')
  userHash: string username
}
```



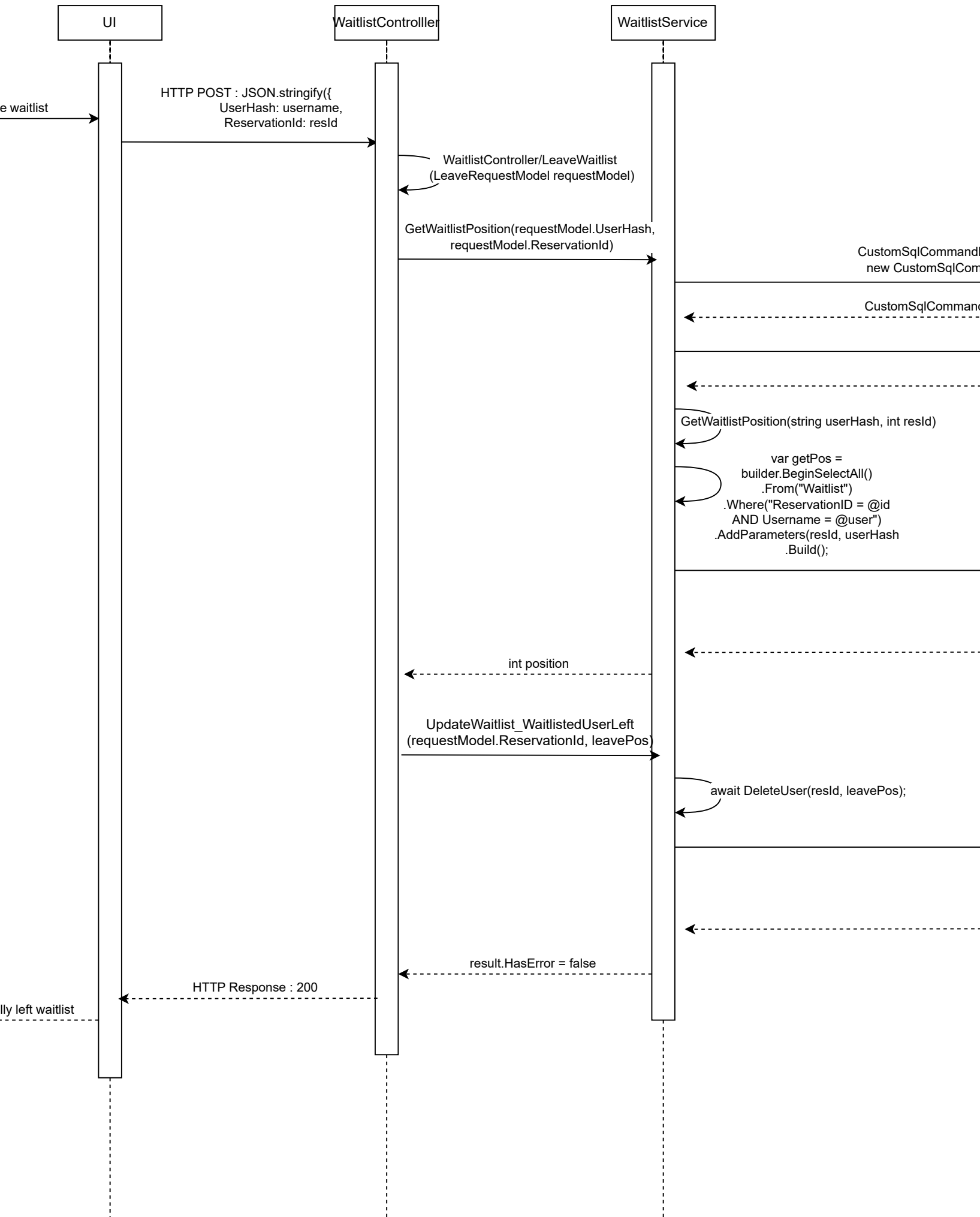


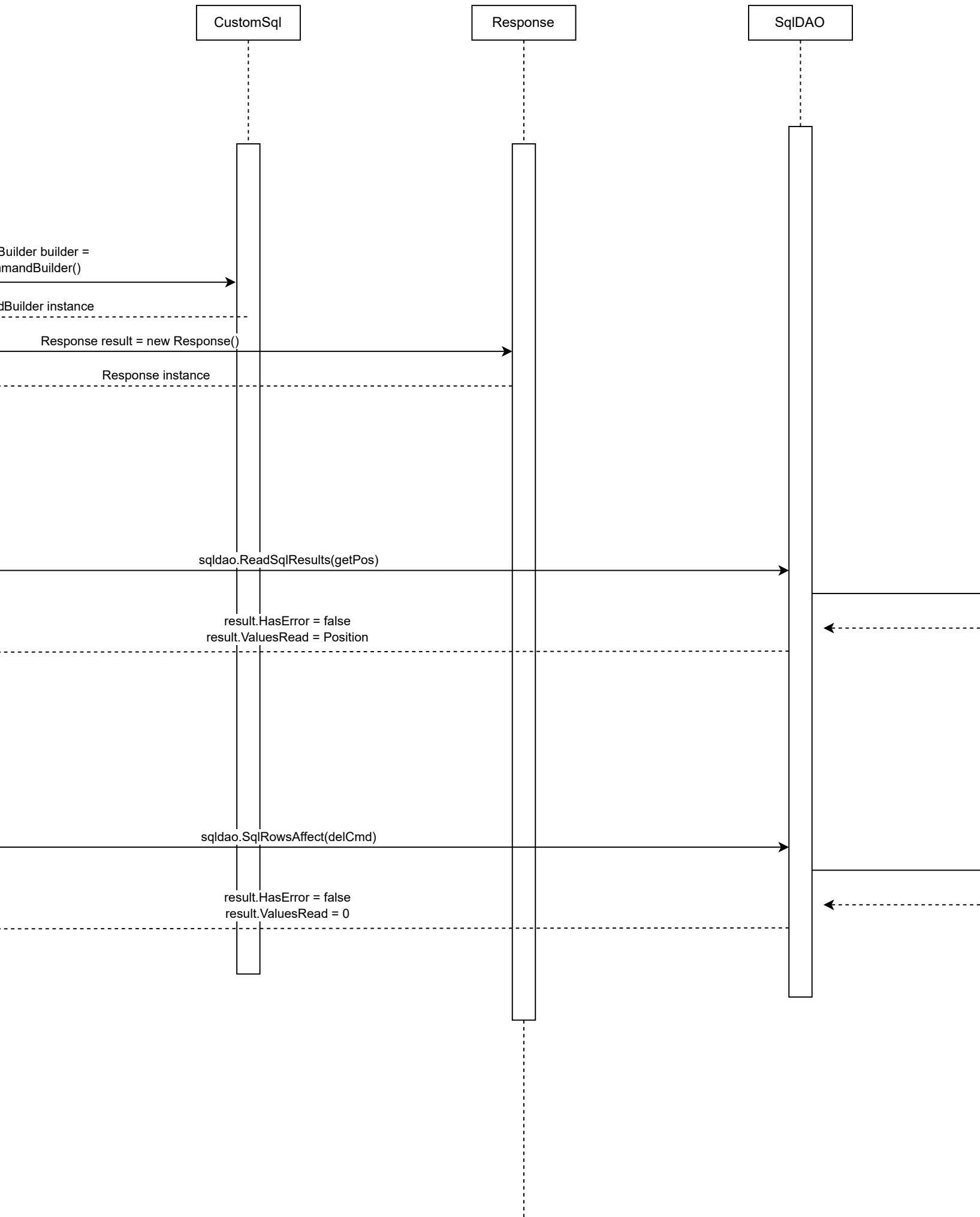


User

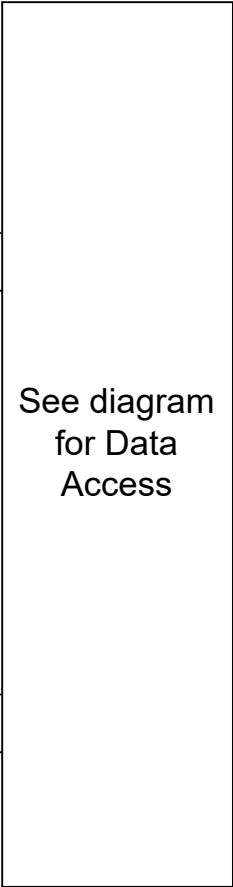
Request to leave

Message: successful





Database



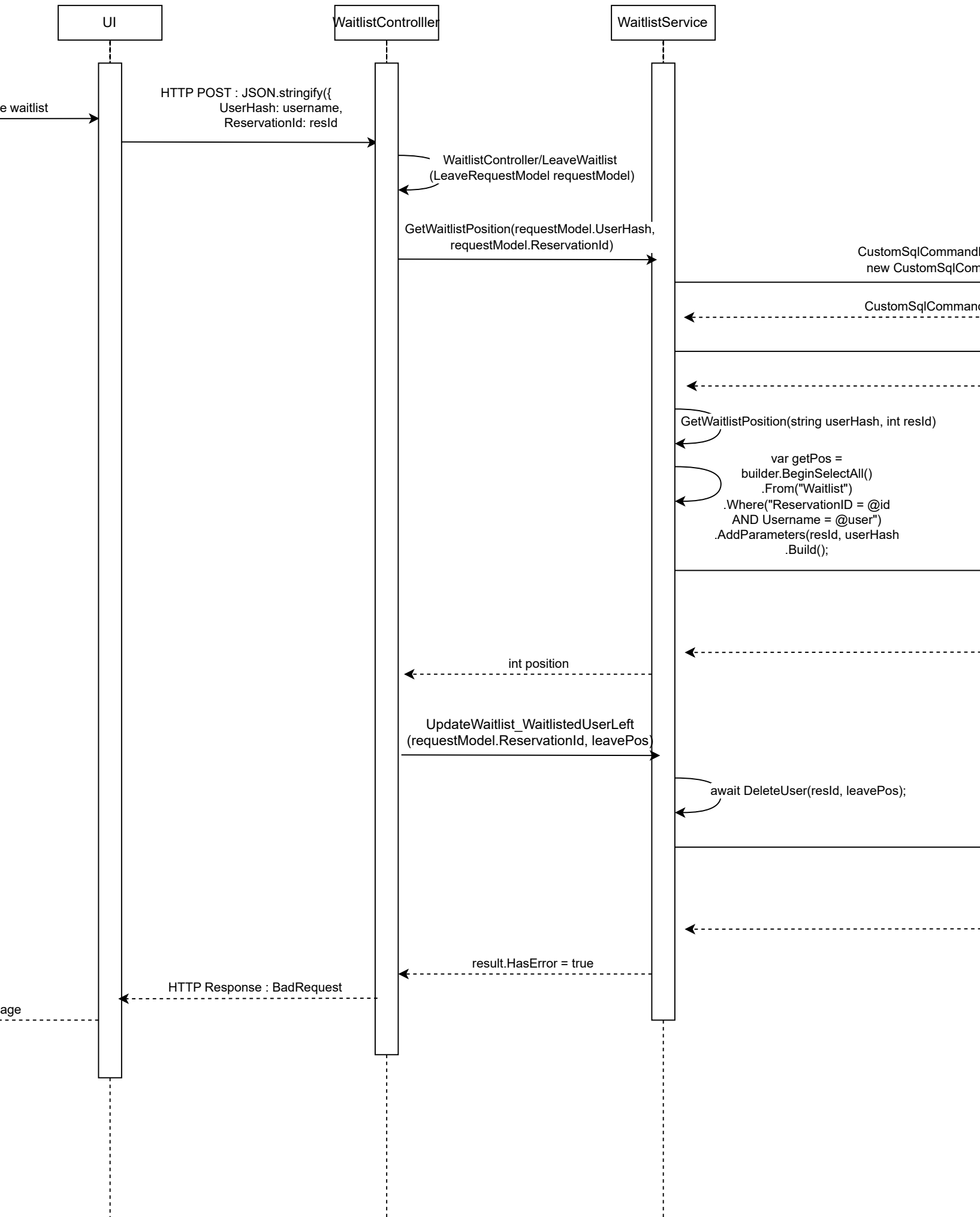
See diagram
for Data
Access

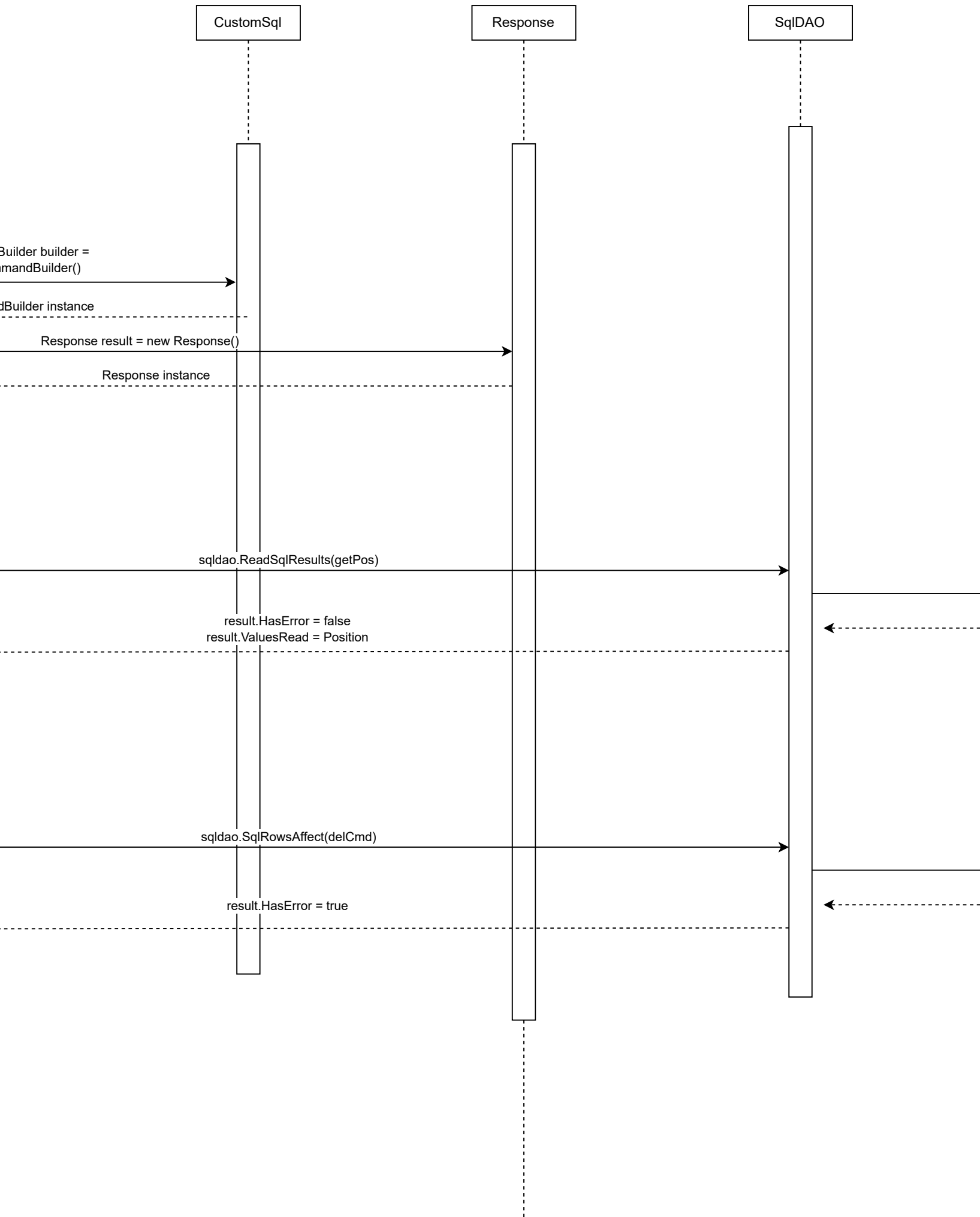
User



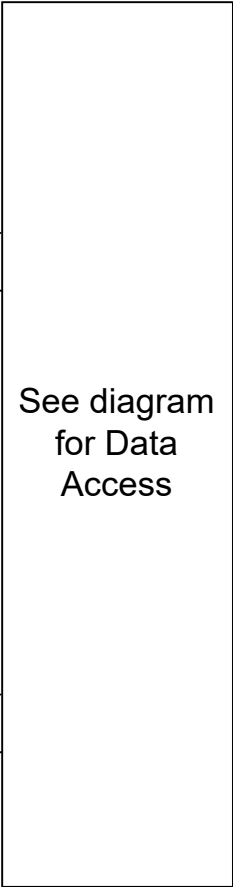
Request to leave

Error Message





Database



See diagram
for Data
Access

Leave Waitlist - Top of waitlist leaves

When a user reserves a space, they will be added to the top of the waitlist for that space and time. If they cancel their reservation, they will be removed from the waitlist, which will allow for the next user on the list to be notified that the space has opened up.

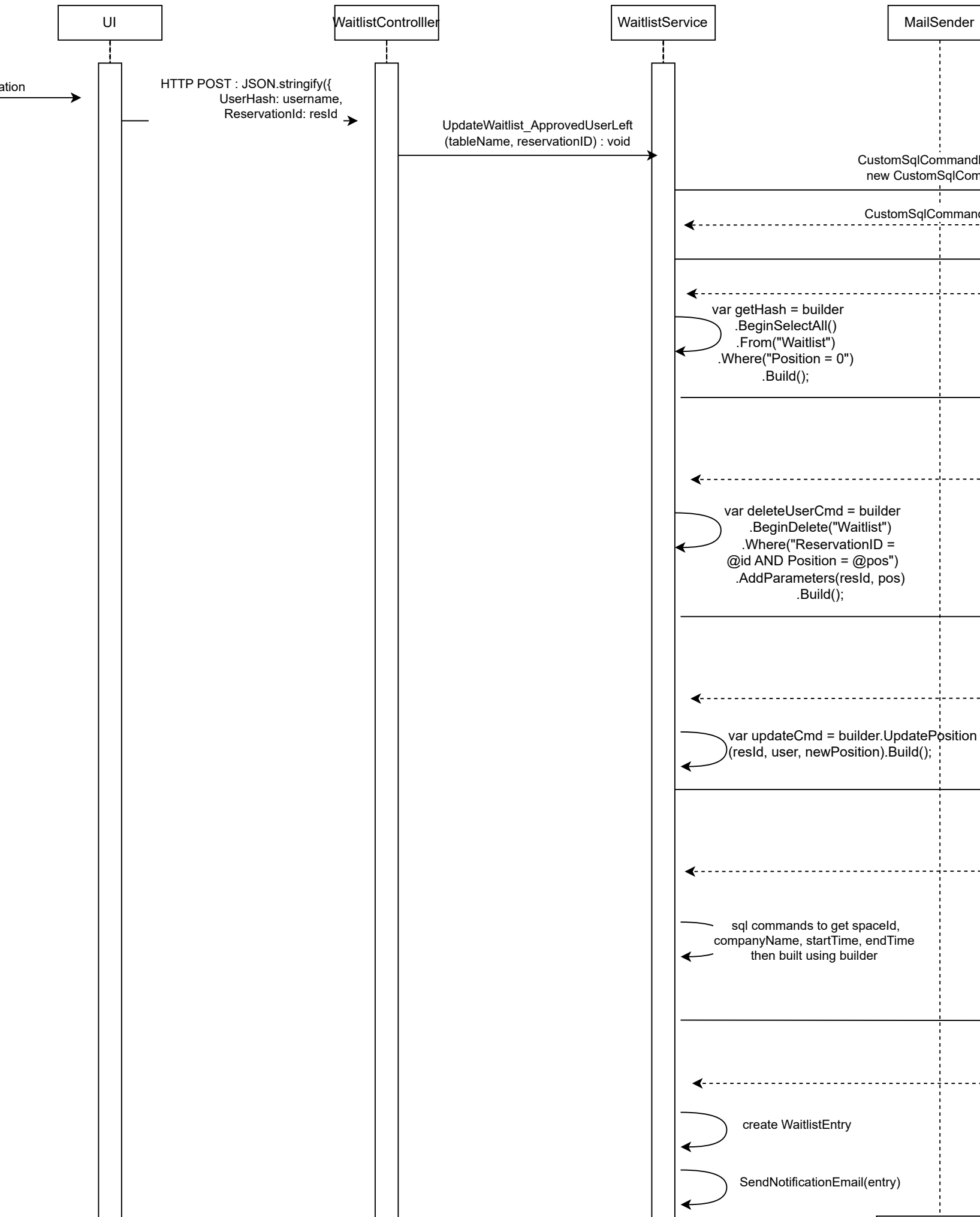
*user who is next in line

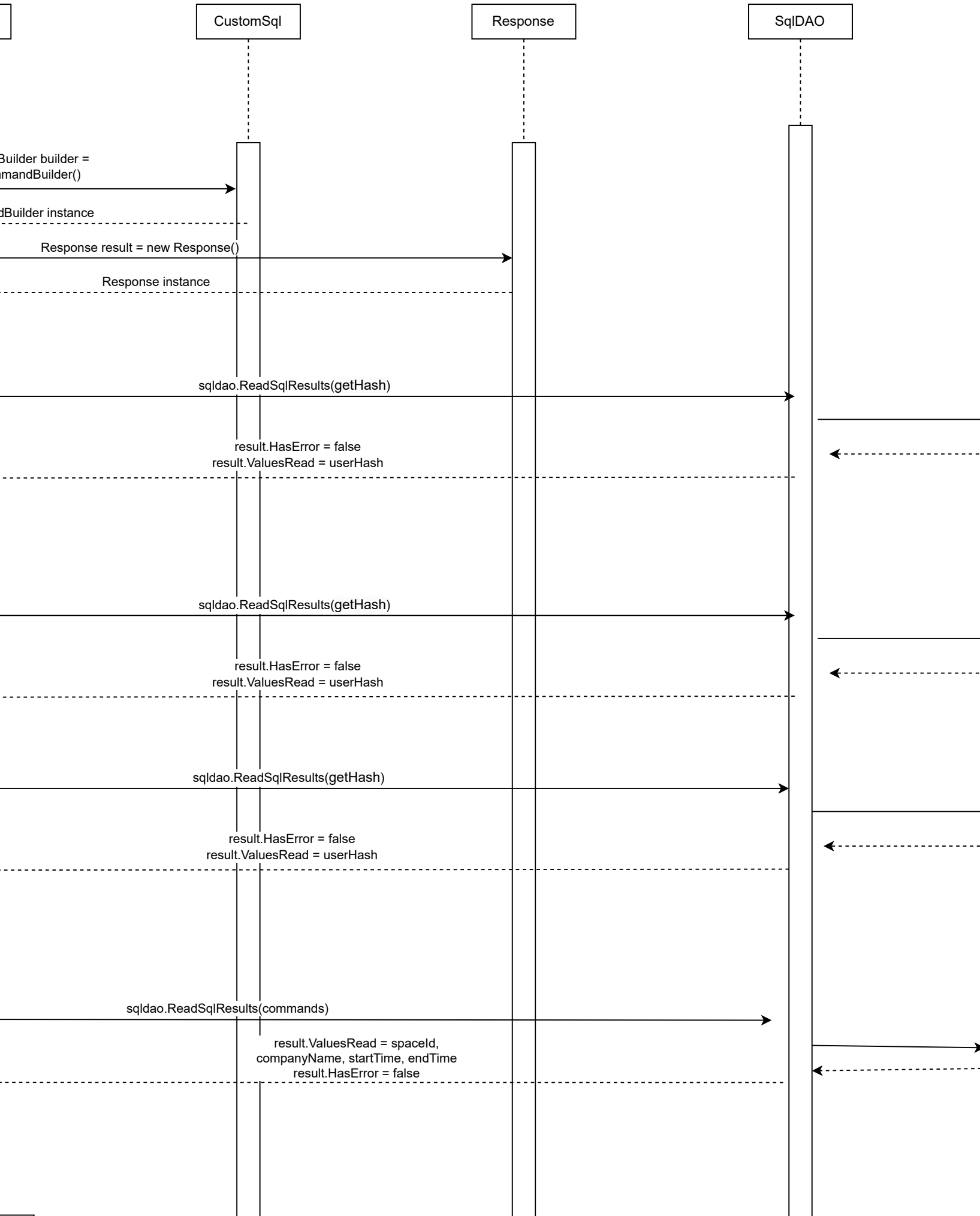
User

*user who cancels their reservation

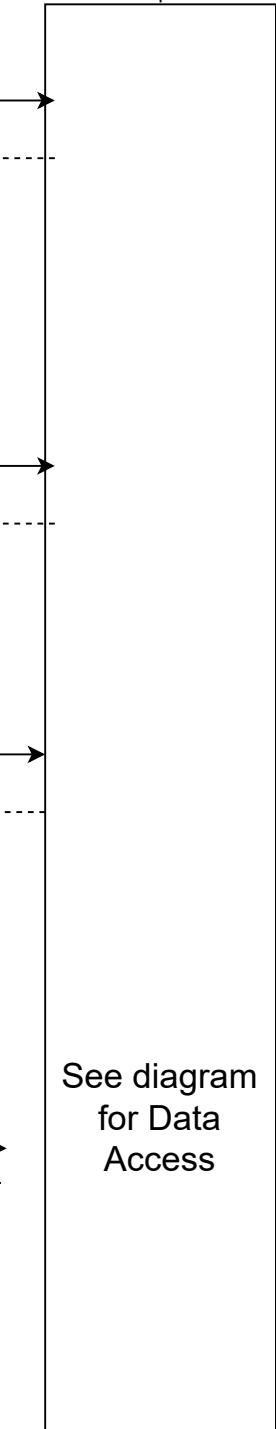
User

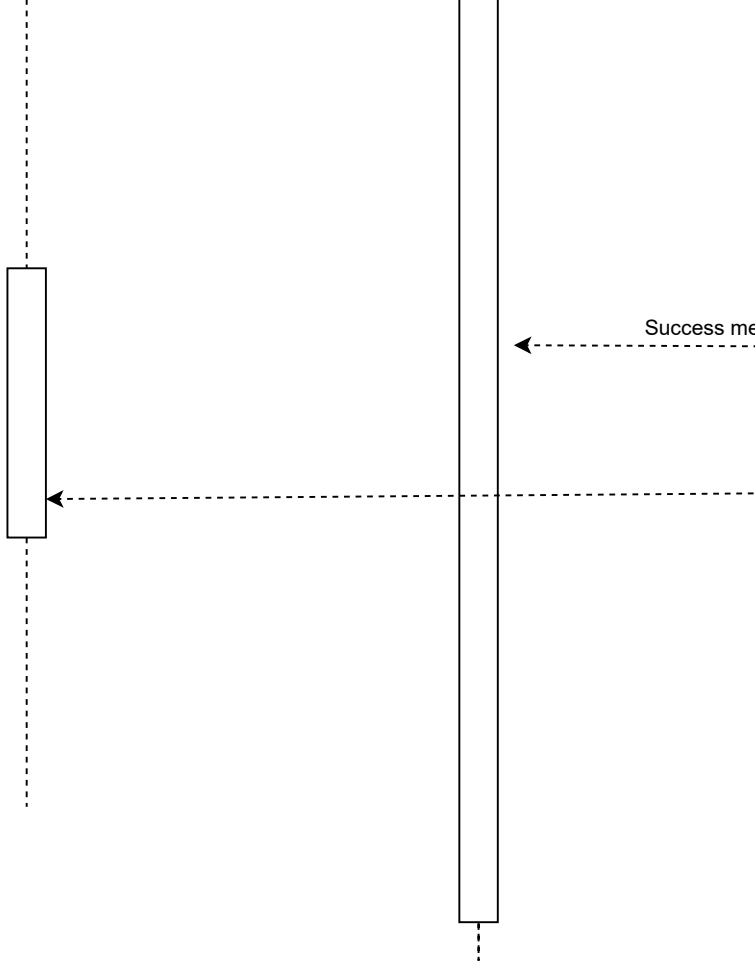
Cancels reserv.

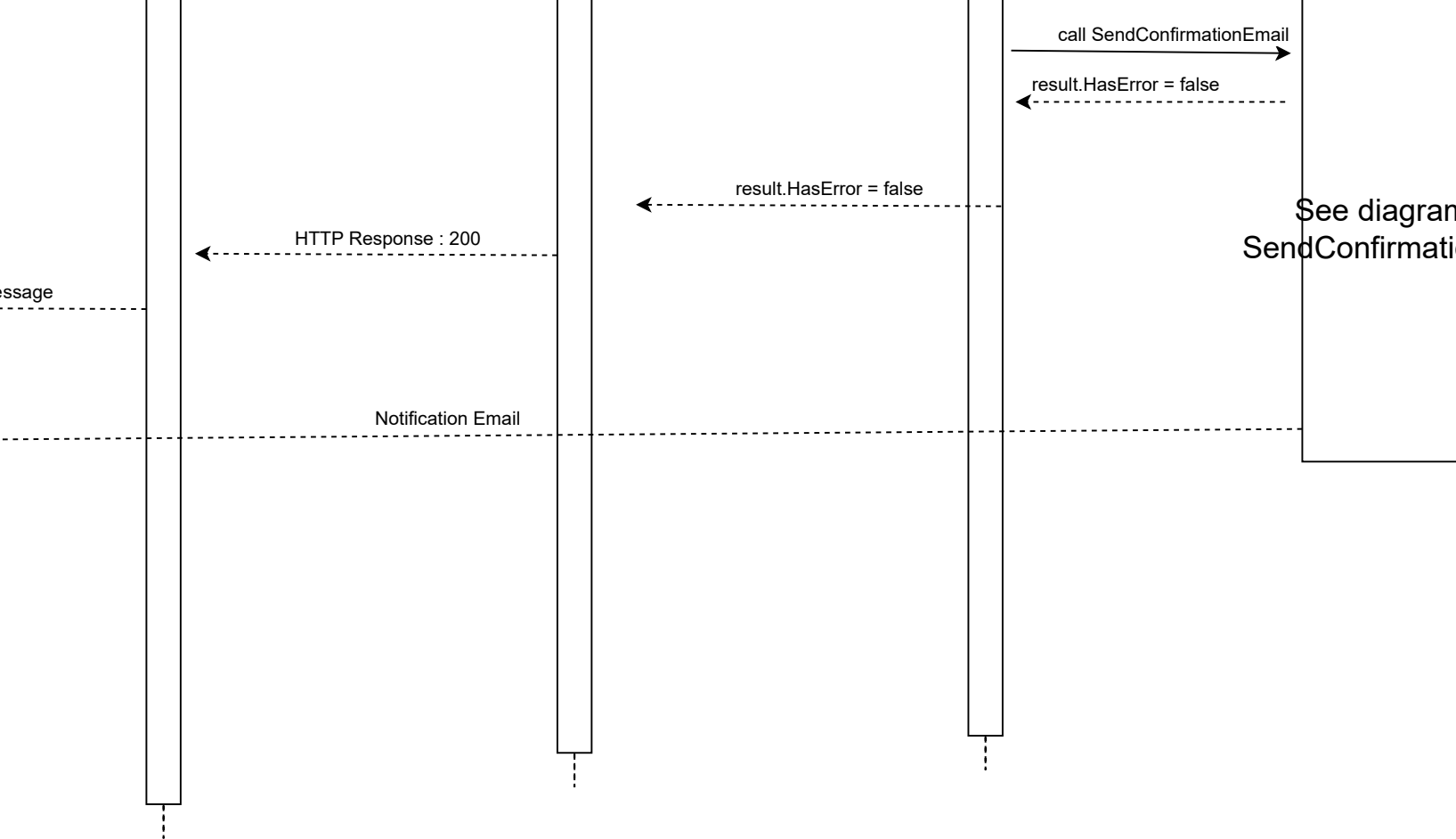




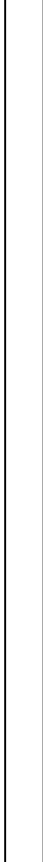
Database

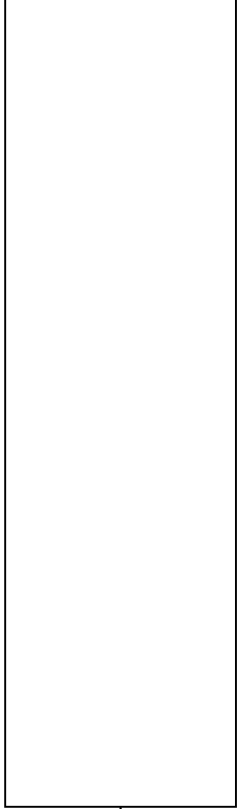






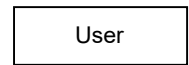
n for
onEmail



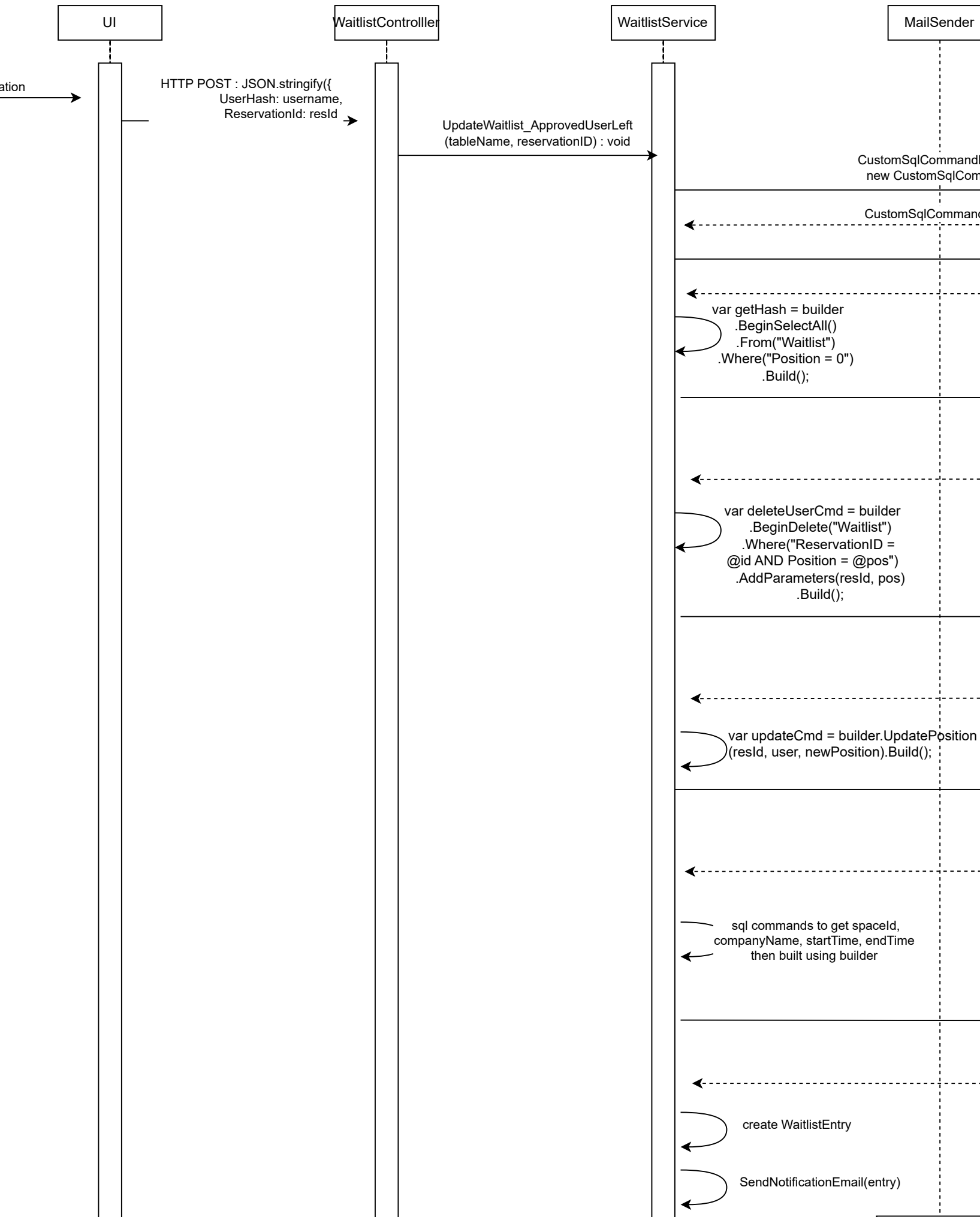


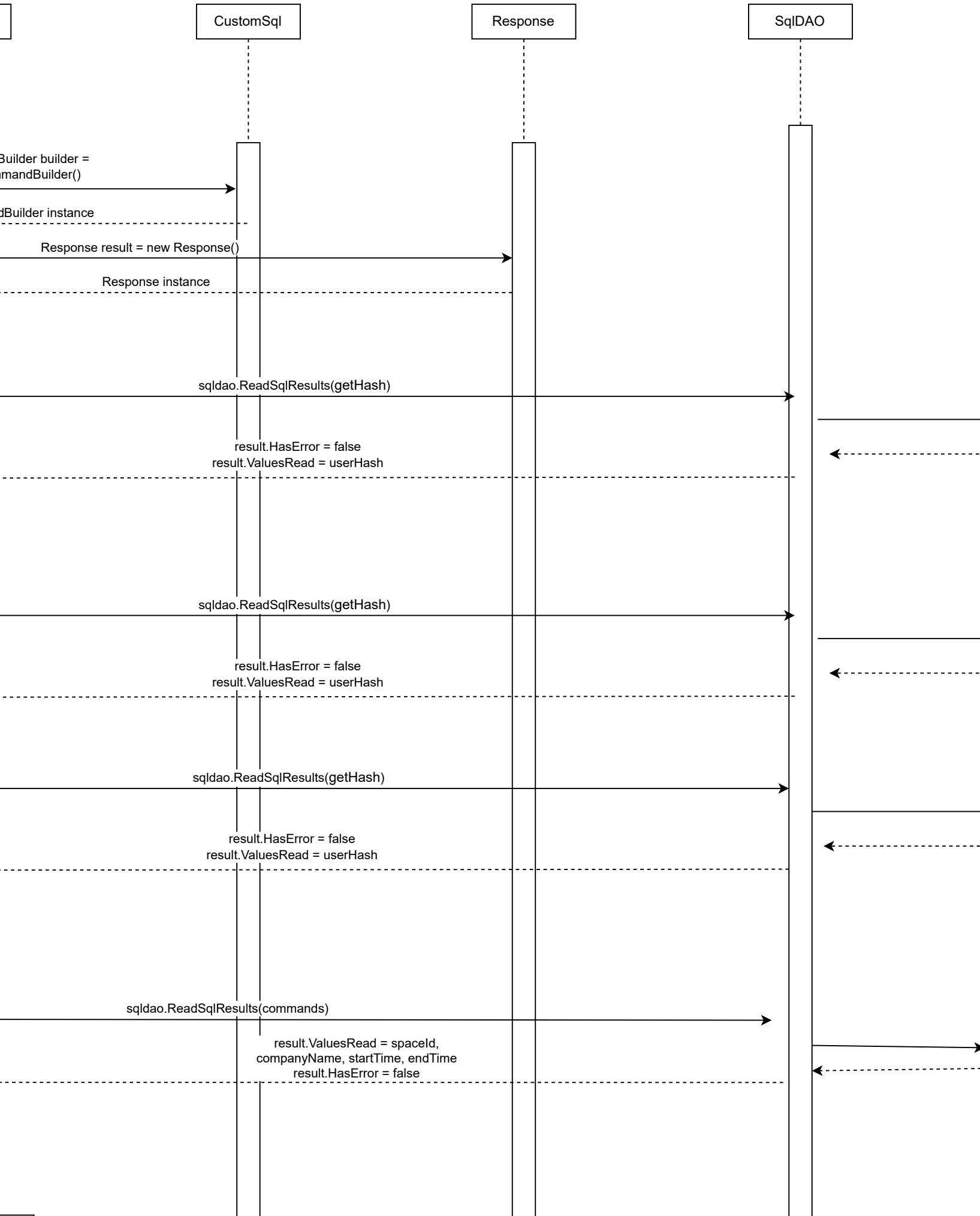
Leave Waitlist - Top of waitlist leaves - Failure

*user who cancels their reservation

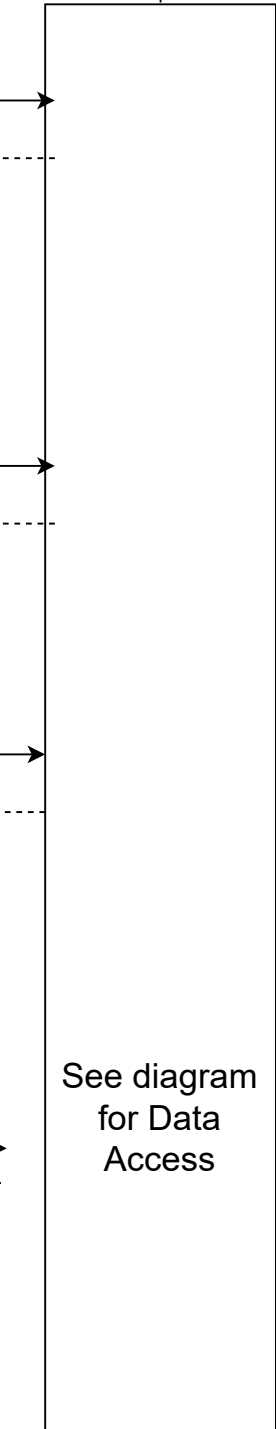


Cancels reserv





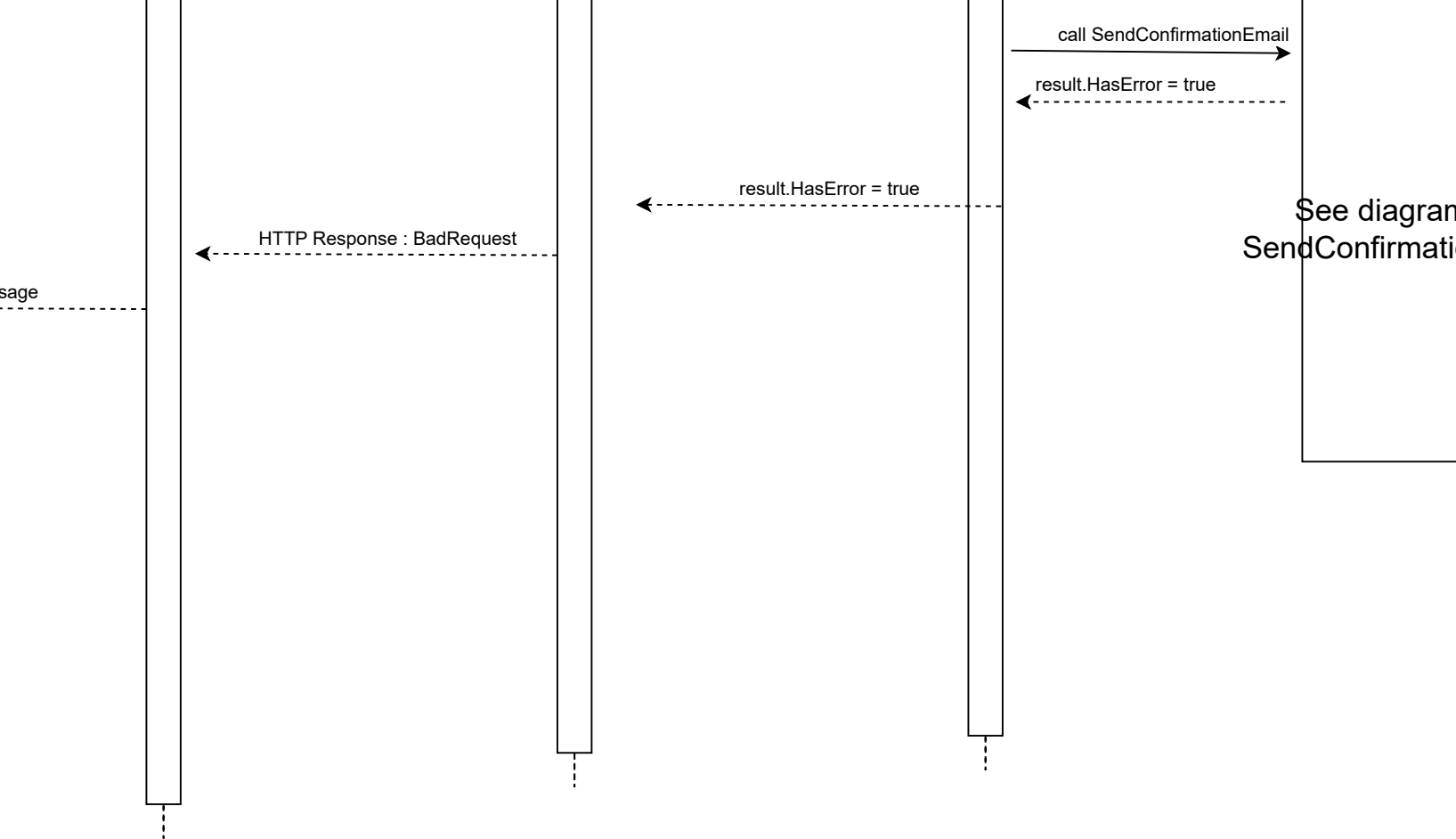
Database



See diagram
for Data
Access

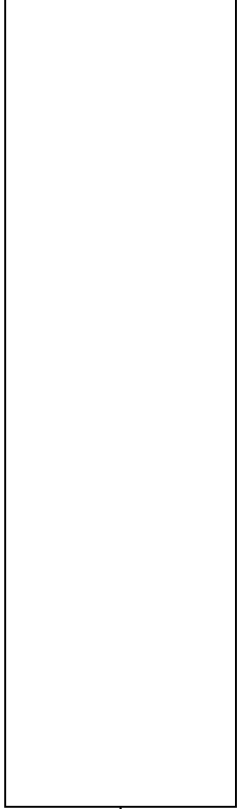
Error mes





n for
onEmail





User

WaitlistService



SendConfirmationEmail
(WaitlistEntry info): void



var getEmail = builder
.BeginSelectAll()
.From("userHash")
.Where(\$"hashedUsername
= '{info.userHash}'")
.Build();



var getName = builder
.BeginSelectAll()
.From("userProfile")
.Where(\$"hashedUsername
= '{info.userHash}'")
.Build();



MailSender sender = new MailS

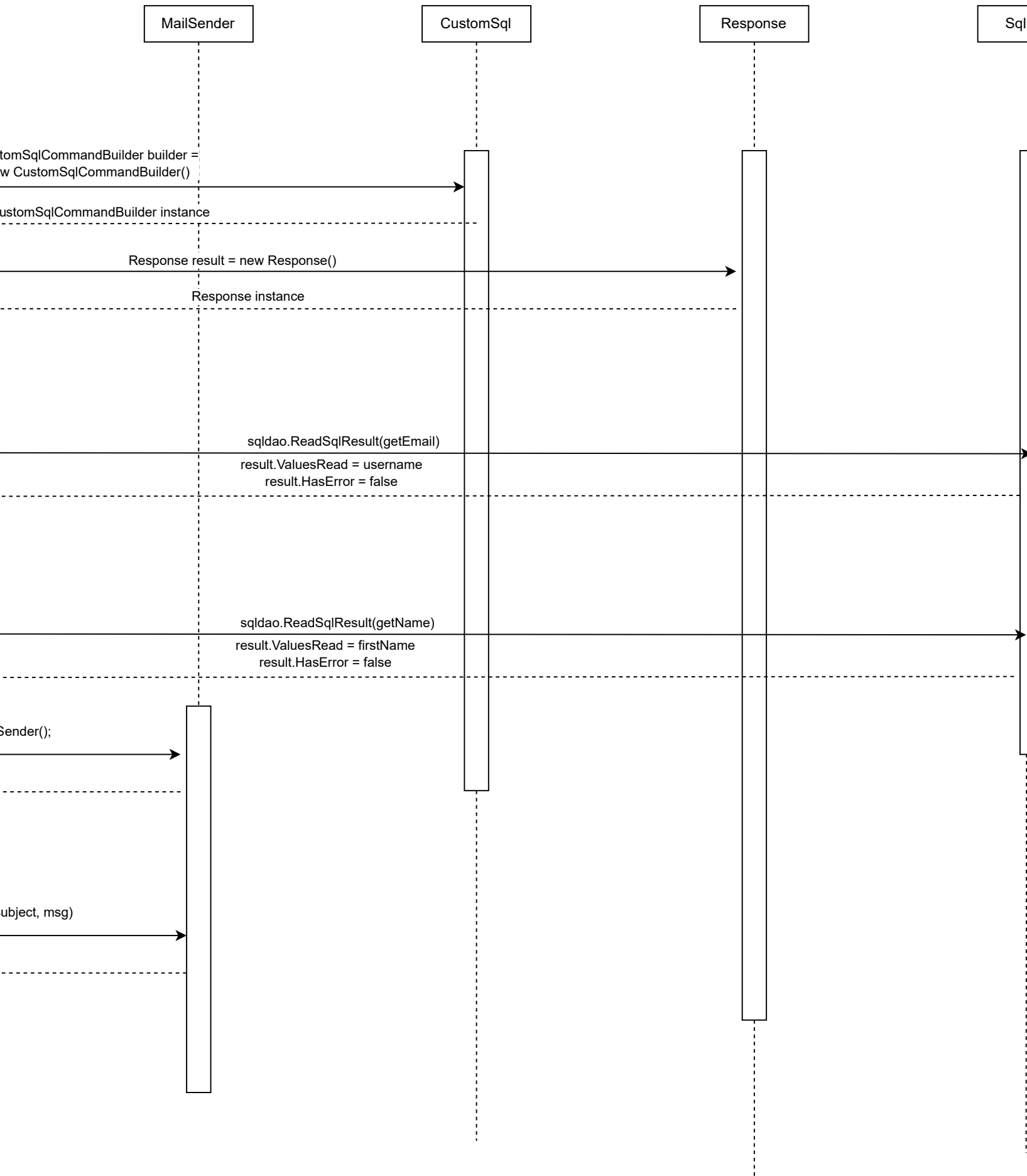


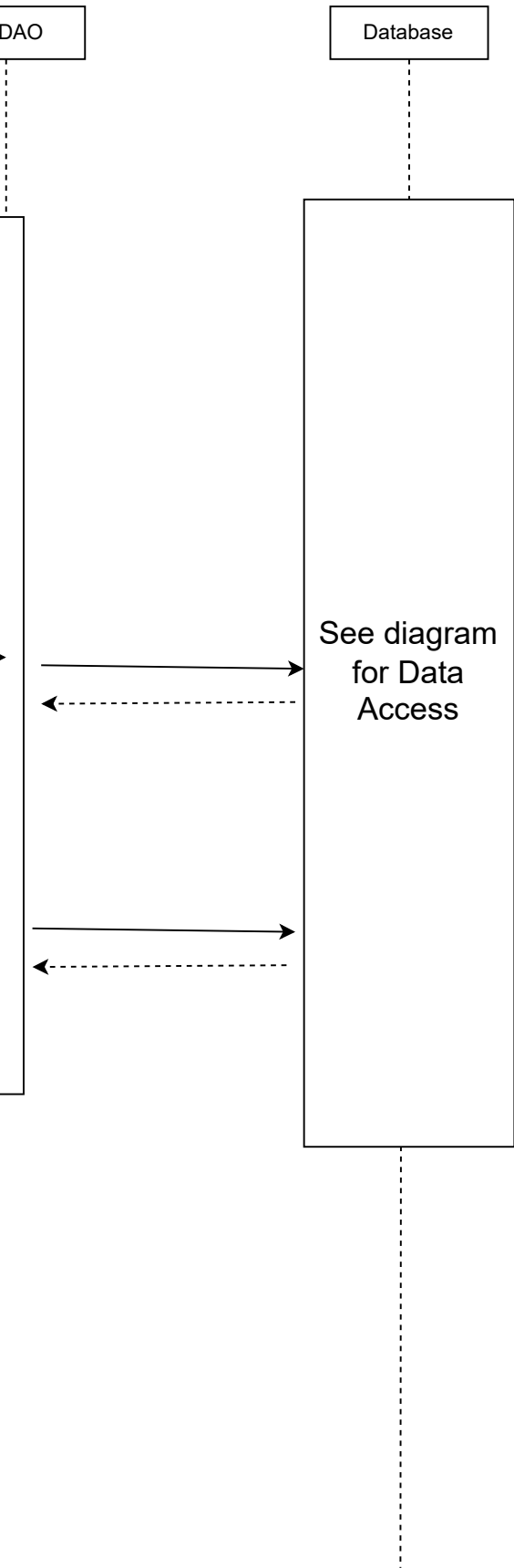
MailSender.SendEmail
(targetEmail, subject, msg) : void

sender.SendEmail(targetEmail, s

Confirmation email







User

WaitlistService



SendConfirmationEmail
(WaitlistEntry info): void

Cus
ne

C

var getEmail = builder
.BeginSelectAll()
.From("userHash")
.Where(\$"hashedUsername
= '{info.userHash}'")
.Build();

var getName = builder
.BeginSelectAll()
.From("userProfile")
.Where(\$"hashedUsername
= '{info.userHash}'")
.Build();

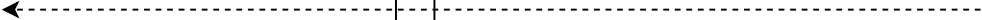
MailSender sender = new MailS

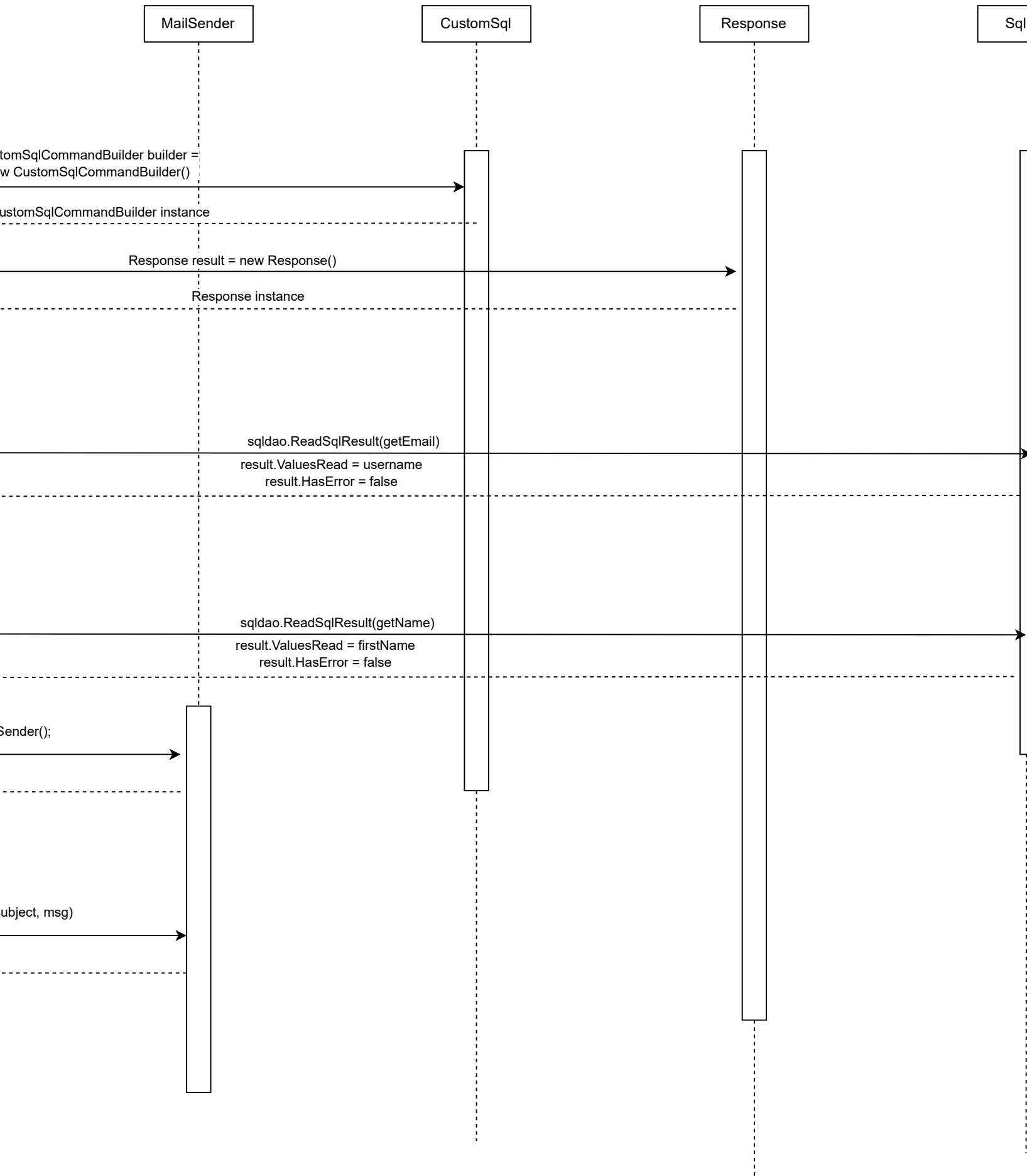
MailSender instance

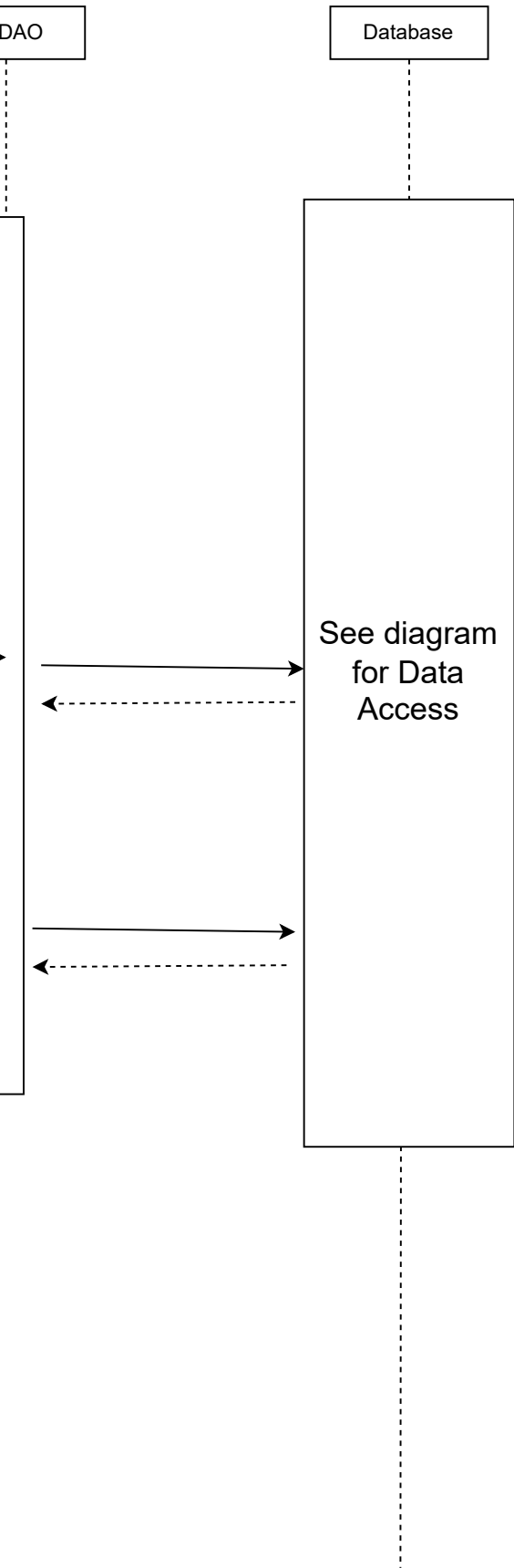
MailSender.SendEmail
(targetEmail, subject, msg) : void

sender.SendEmail(targetEmail, s

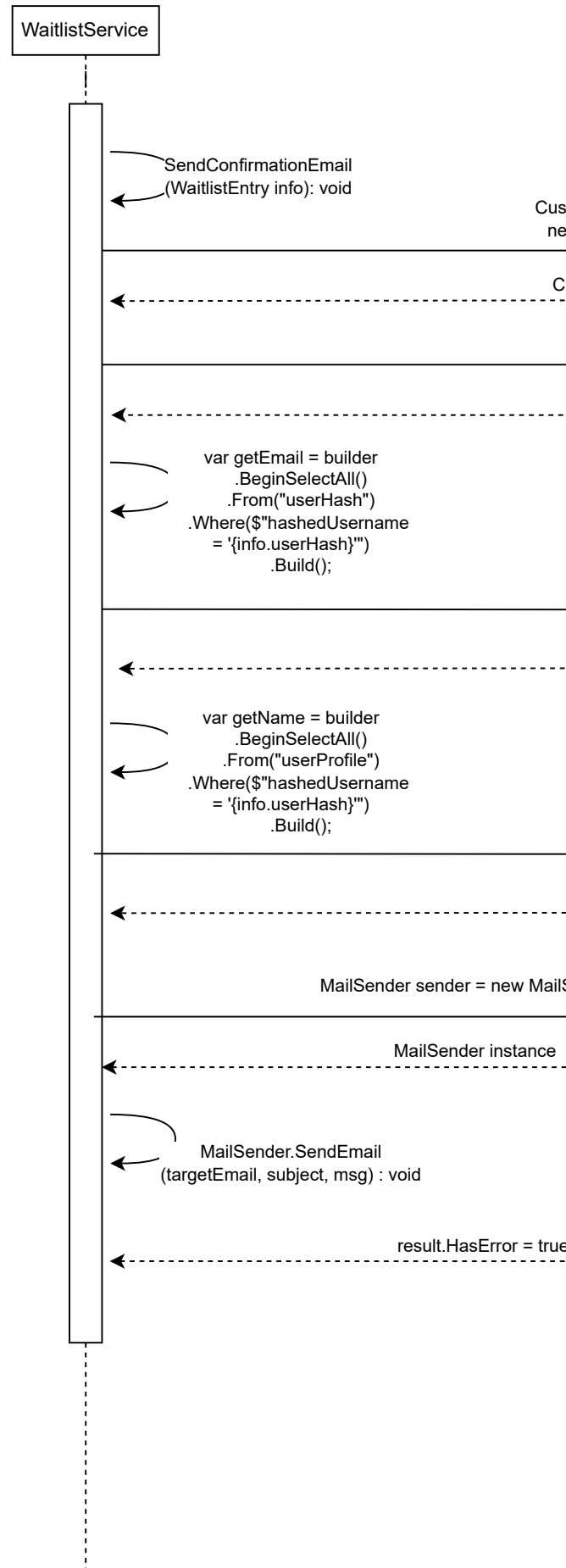
Confirmation email

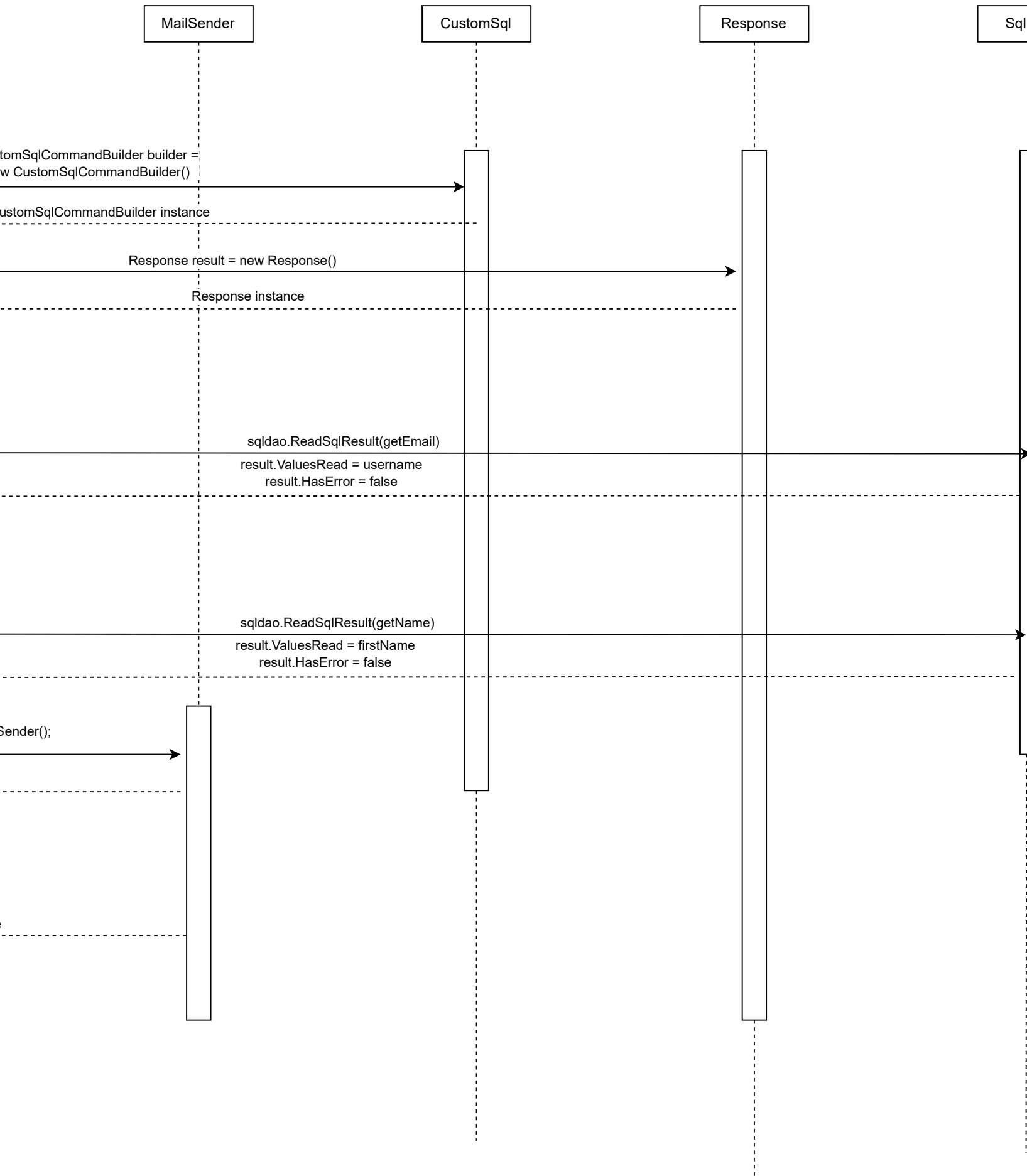


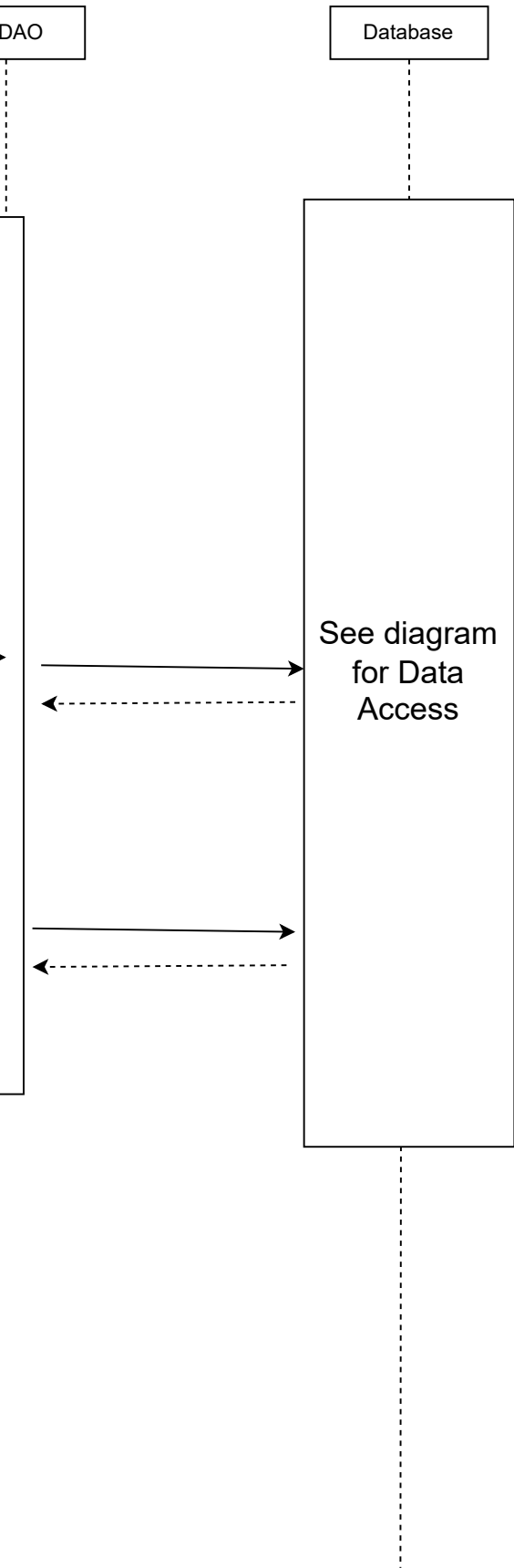




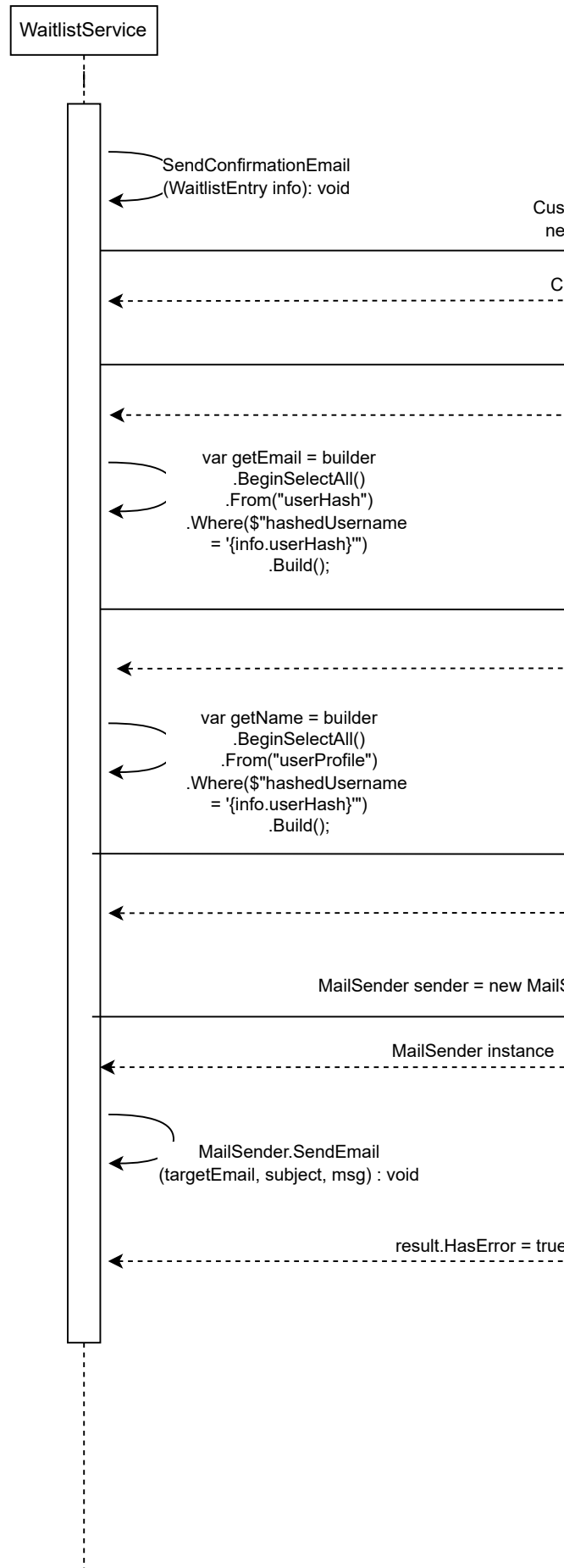
Confirmation Email - Success

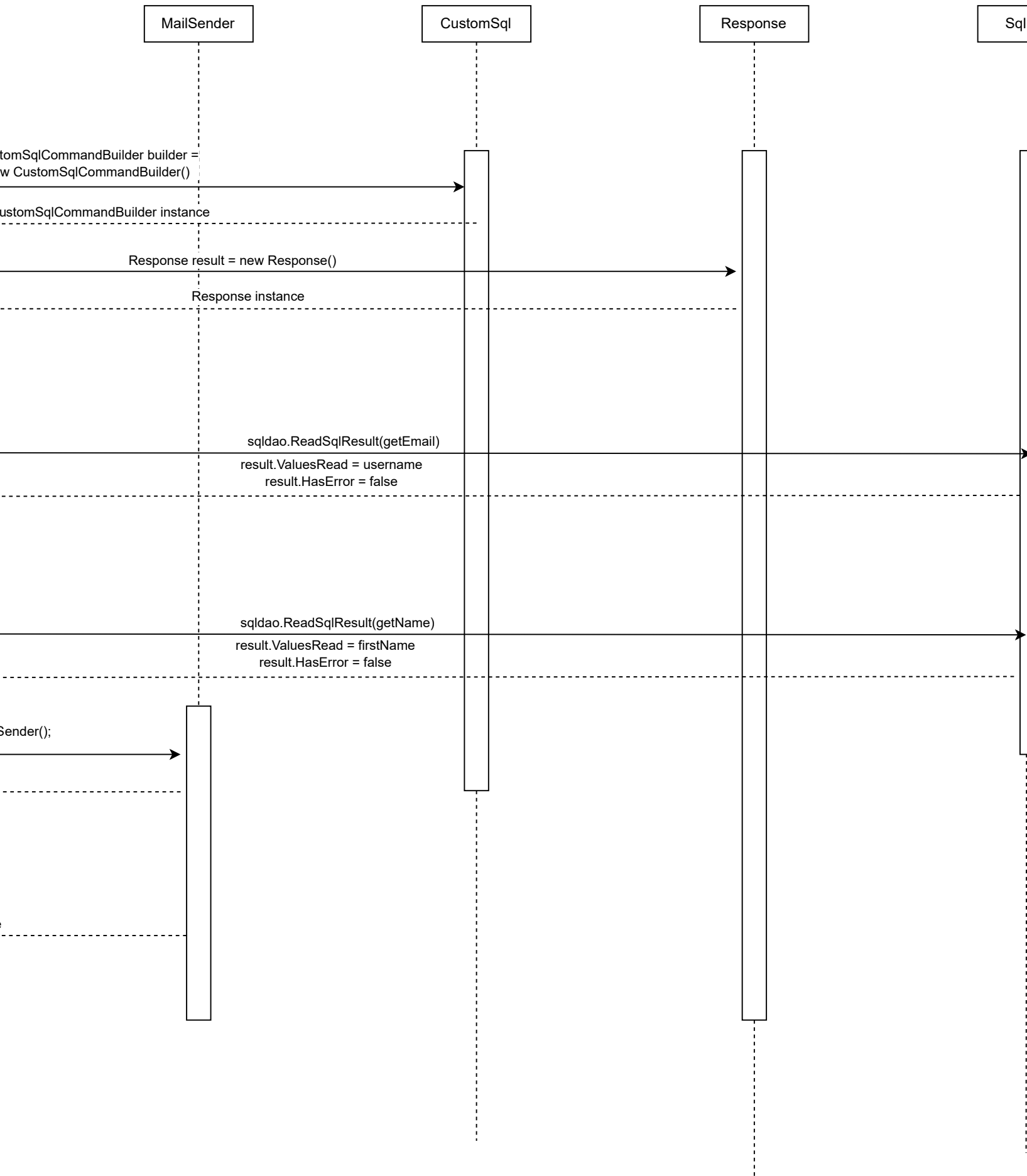






Notification Email - Failure





DAO

Database

