

```
public async Task Handle(UserDeletedDomainEvent domainEvent, CancellationToken ct)
{
    var teams = await _teamRepository.GetTeamsByUserIdAsync(domainEvent.User.Id, ct);

    foreach (var team in teams)
    {
        team.GetTeamMemberByUserId(domainEvent.User.Id)
            .Ensure(TeamRules.MemberCanChangeOwnership)
            .Tap(initiator =>
            {
                if (team.Members.Count == 1)
                {
                    //remove team if user that is being removed is the only member
                    _teamRepository.RemoveTeam(team);
                }
                else
                {
                    //change ownership when removing user that is owner of the team
                    var newOwner = team.GetHighestNonOwnerTeamMember();
                    initiator.UpdateRole(TeamRole.Admin);
                    newOwner.UpdateRole(TeamRole.Owner);
                    team.AddDomainEvent(new OwnerChangedEvent(initiator, newOwner));
                }
            });

        //db will cascade delete member, number of members has to be updated manually
        team.DecreaseNumberOfMembers();
    }

    _userRepository.RemoveUser(domainEvent.User);
}
```