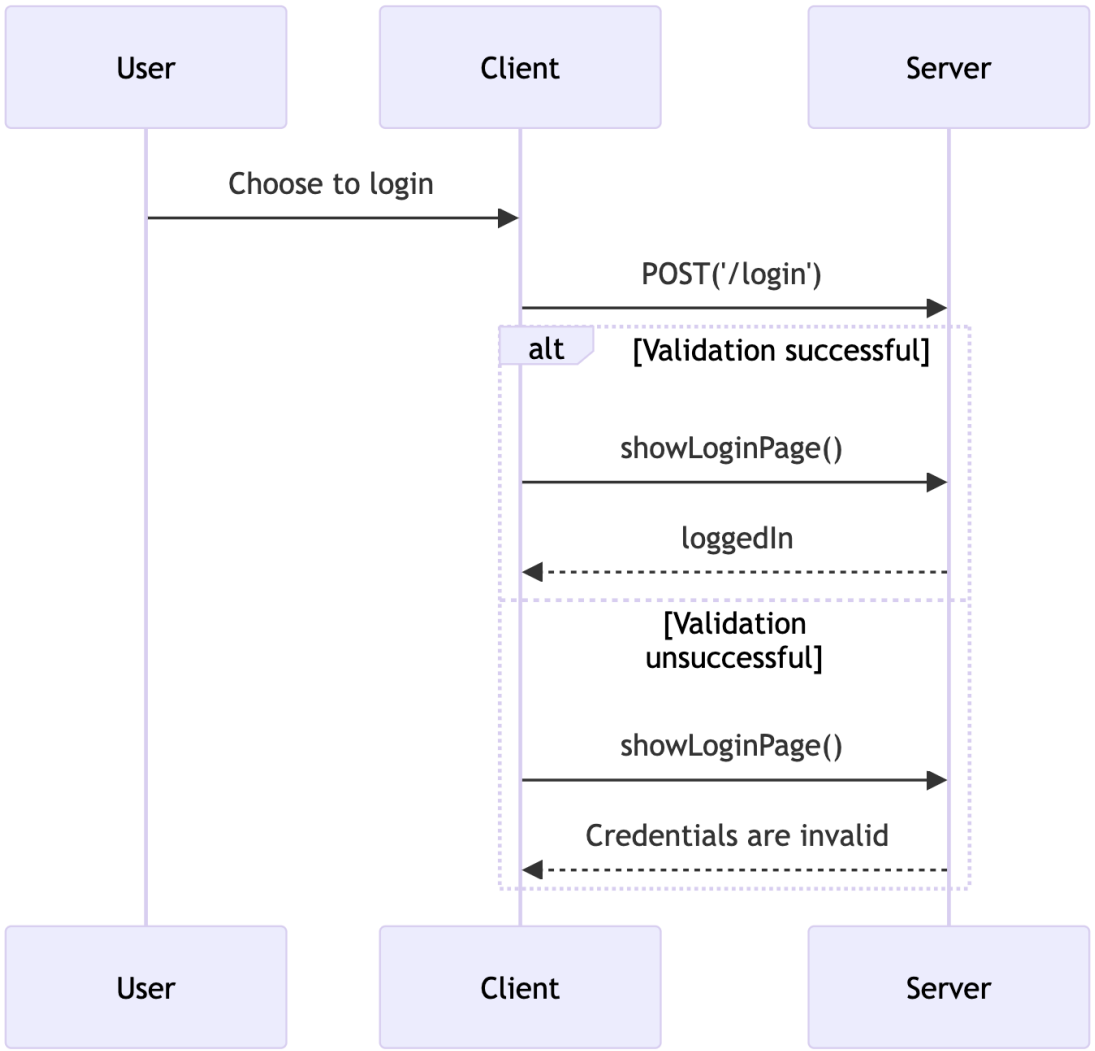
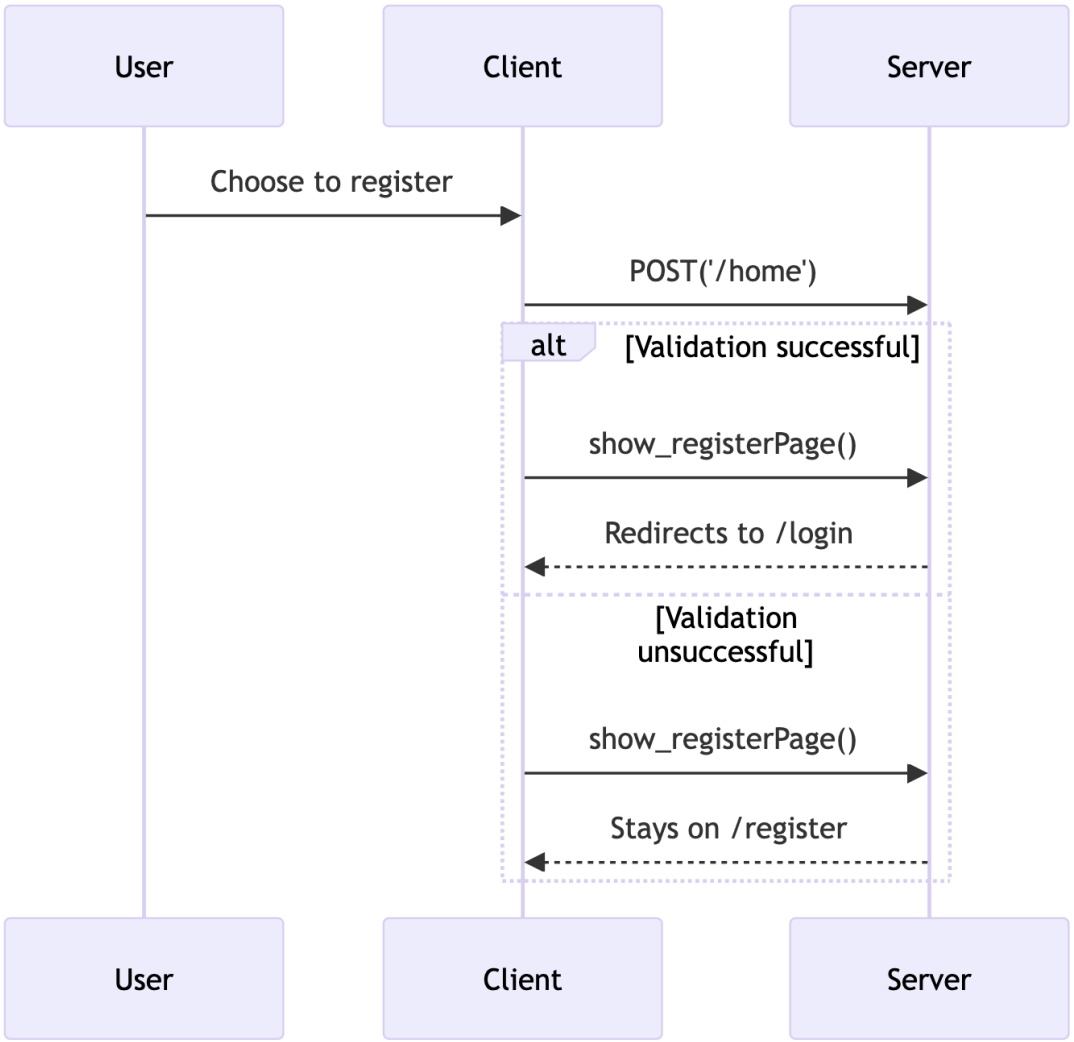
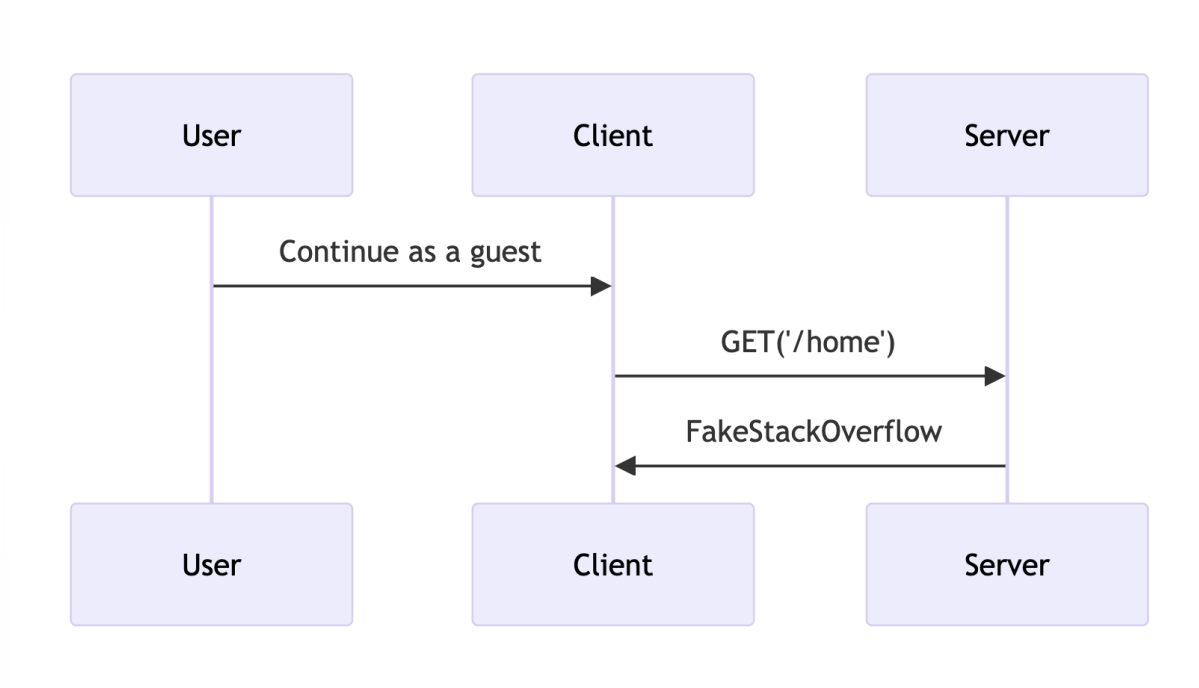
Login Scenario



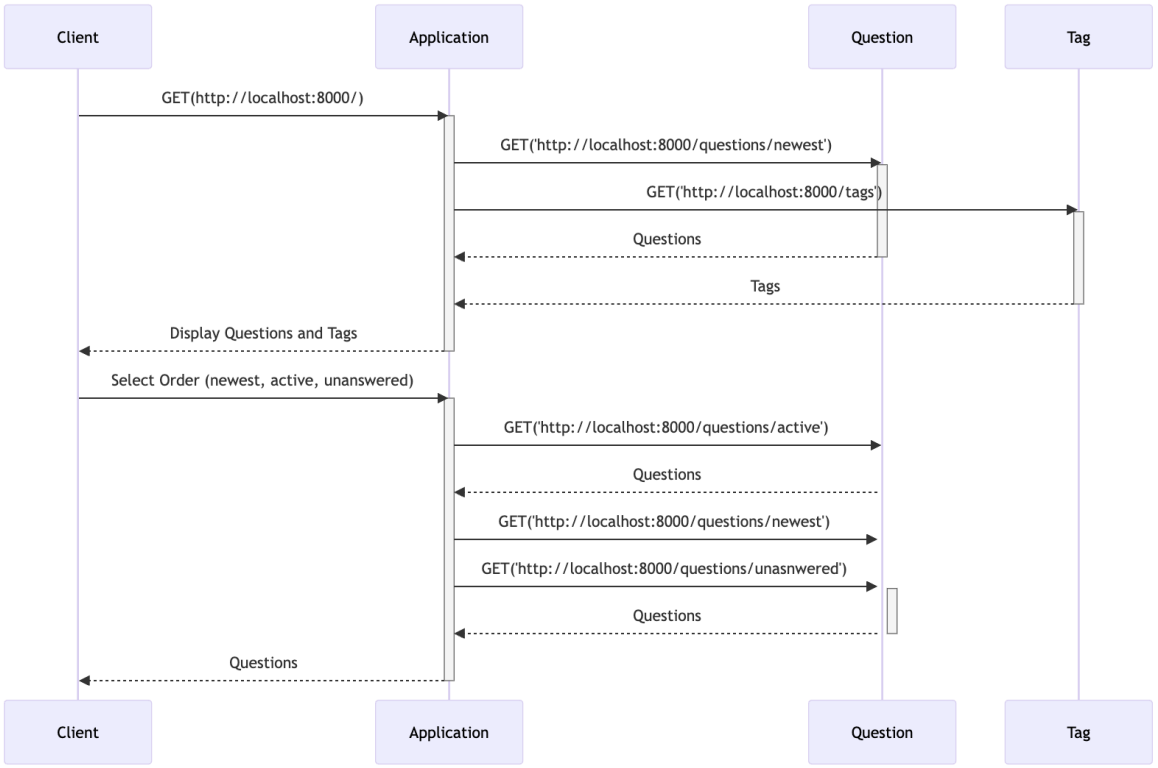
Register Scenario



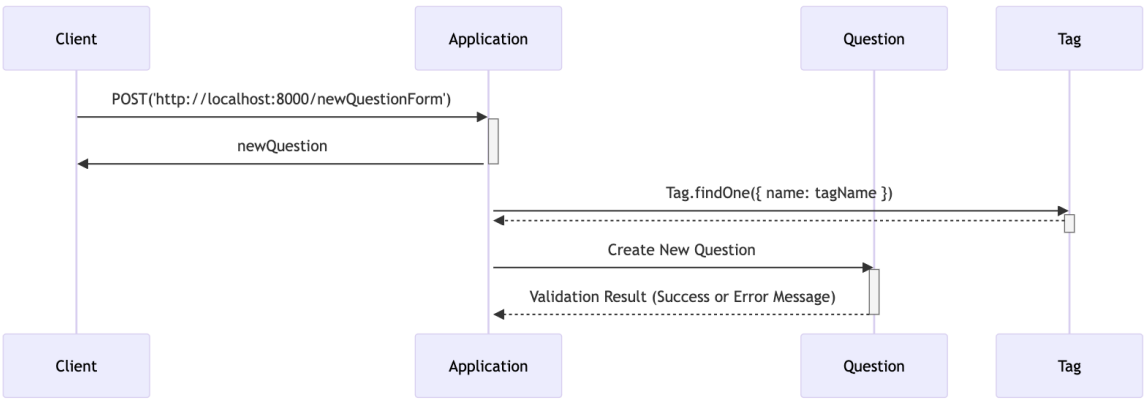
Continue as a guest scenario



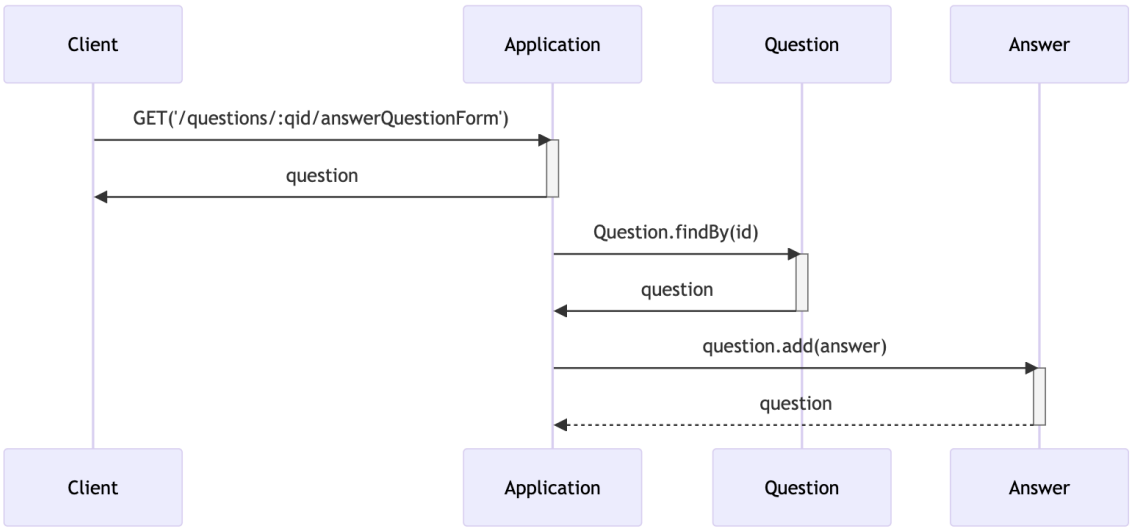
Landing Page



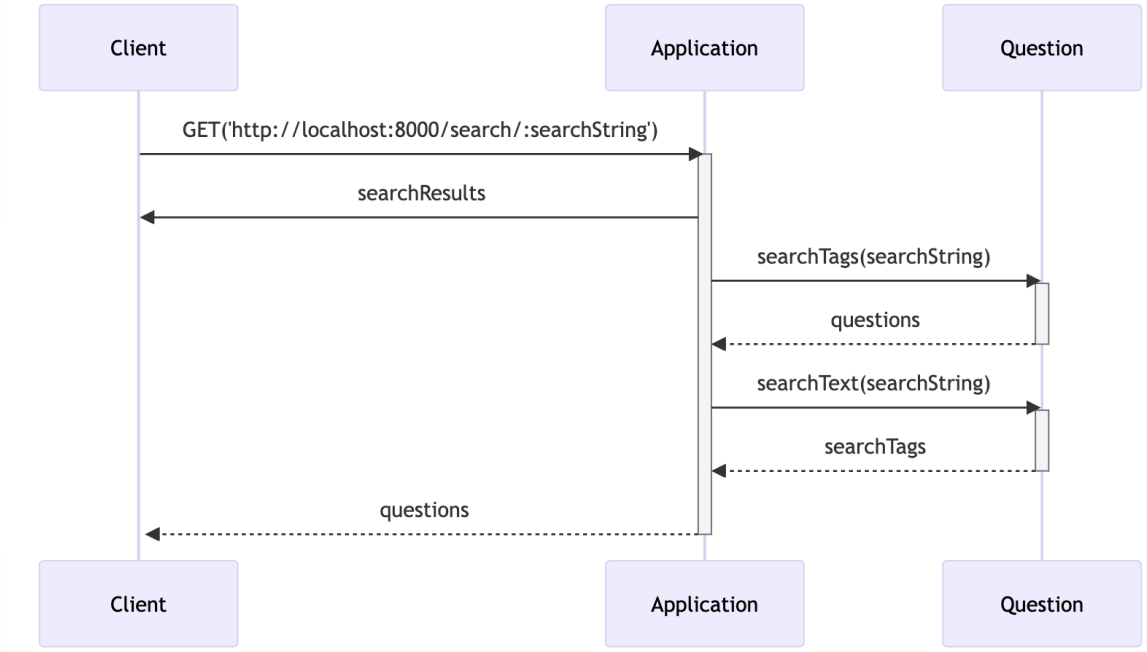
New Question



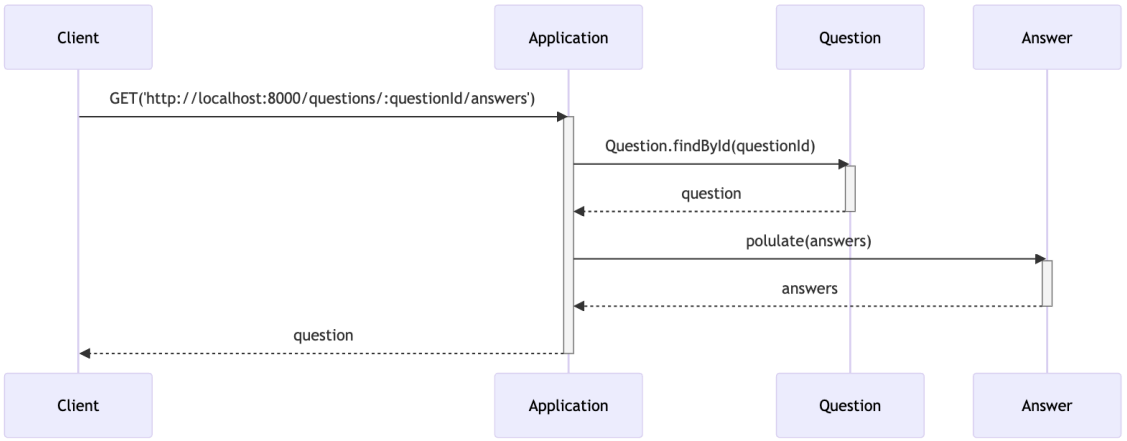
Answer Form



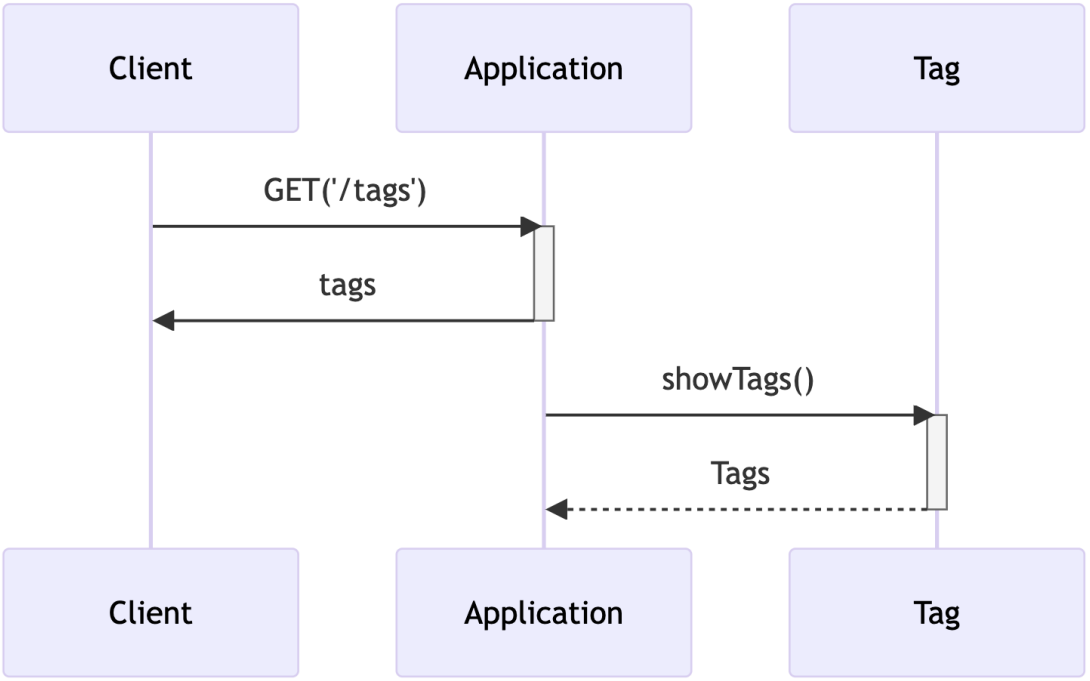
Search Question

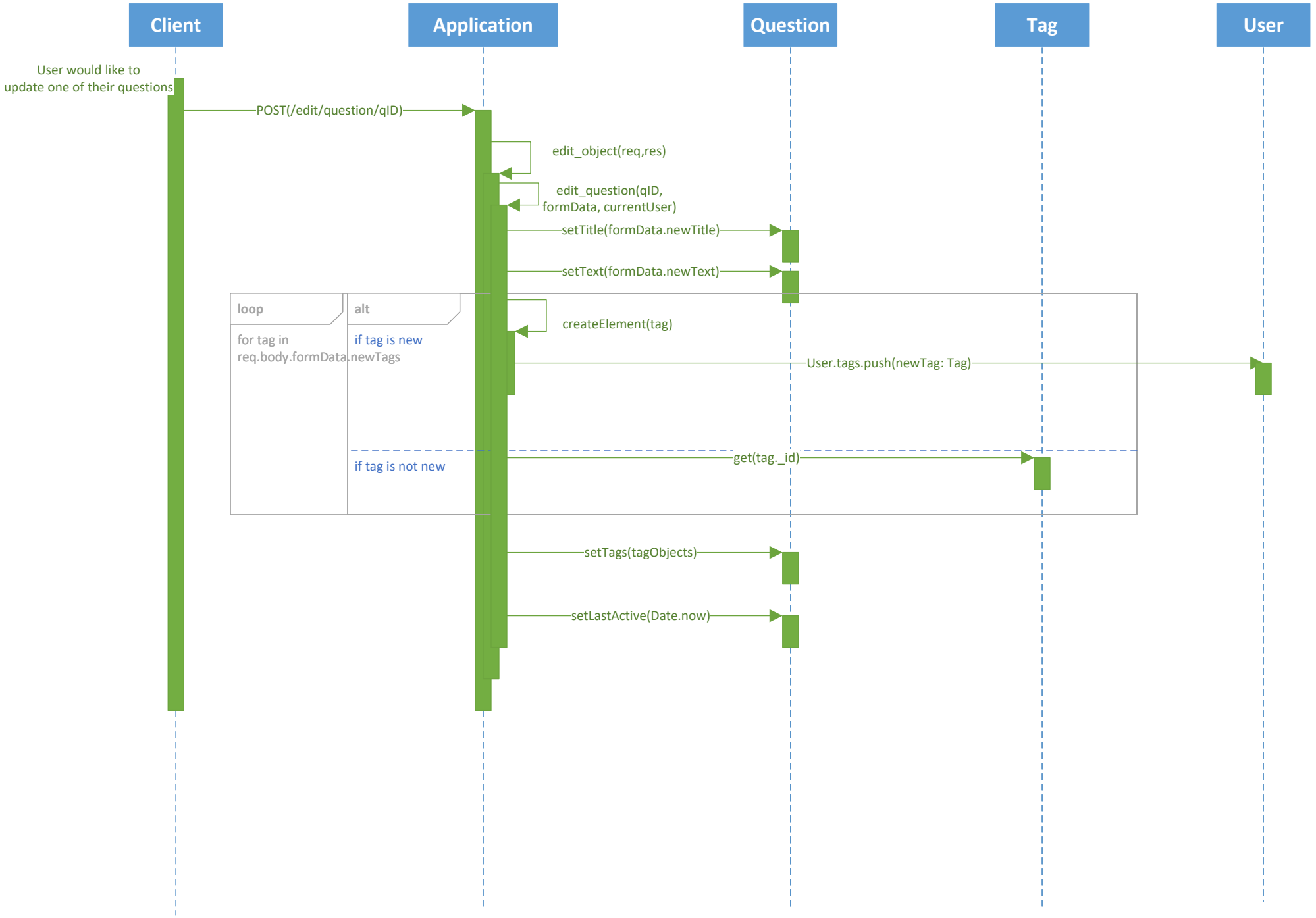


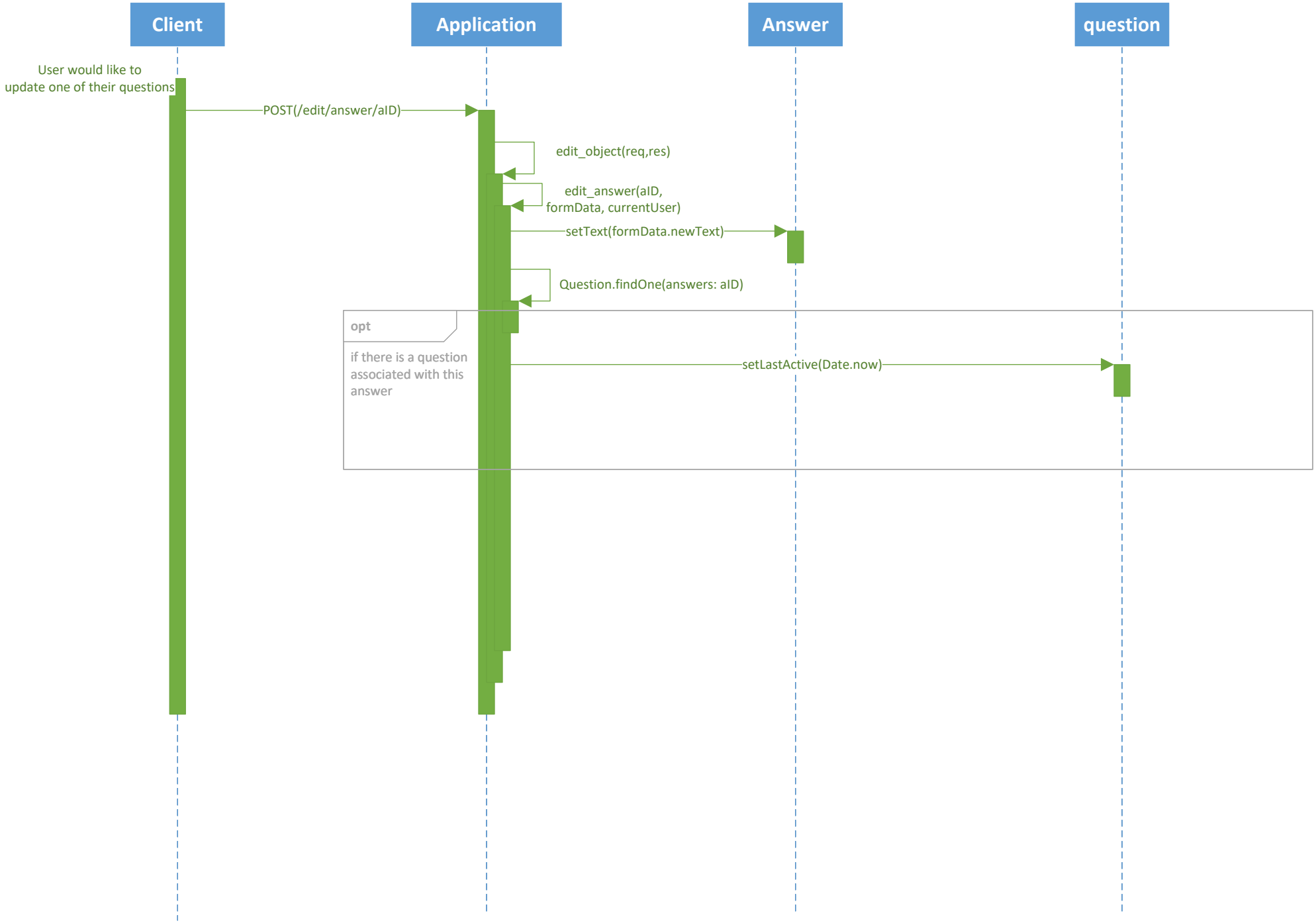
View Question

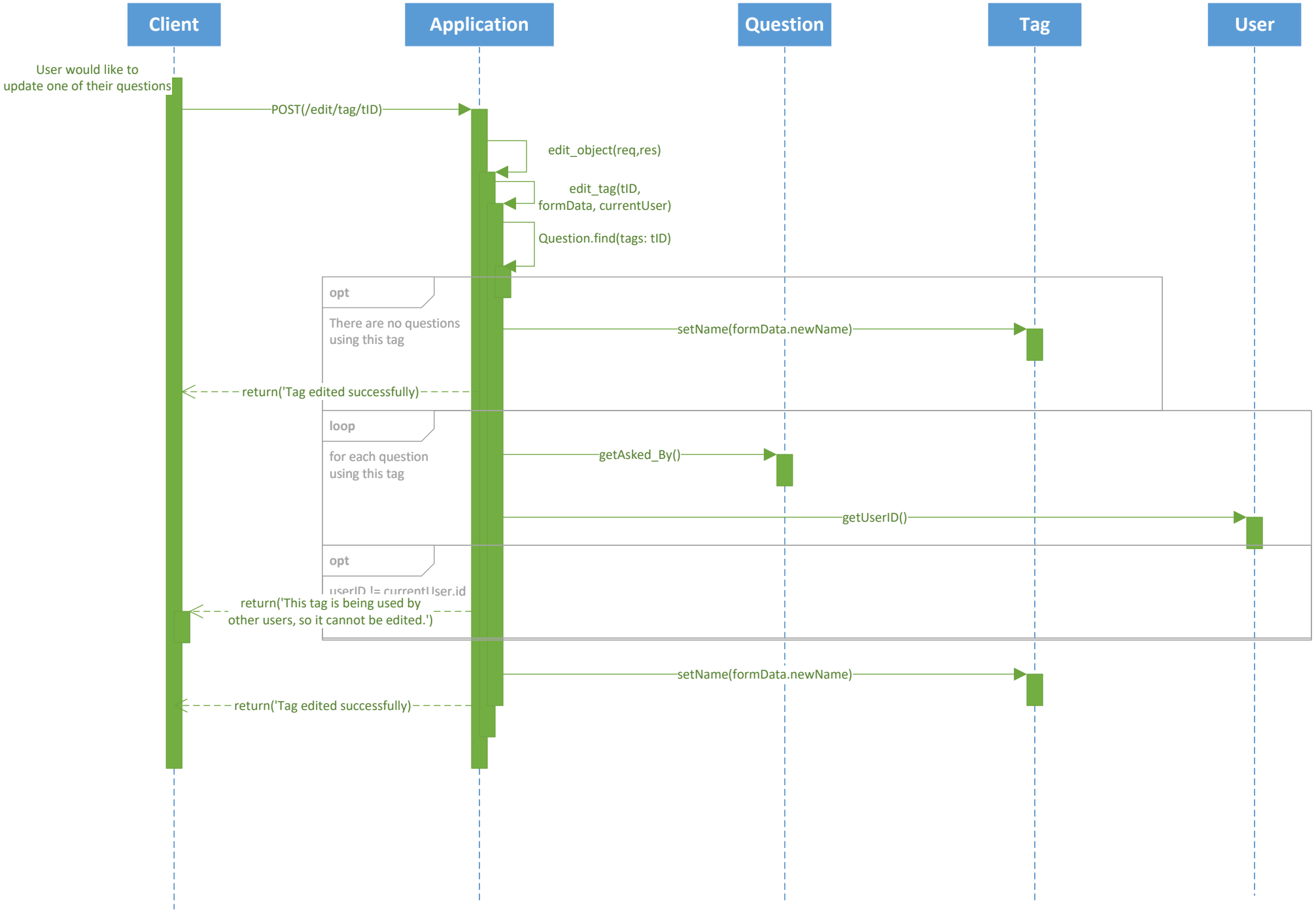


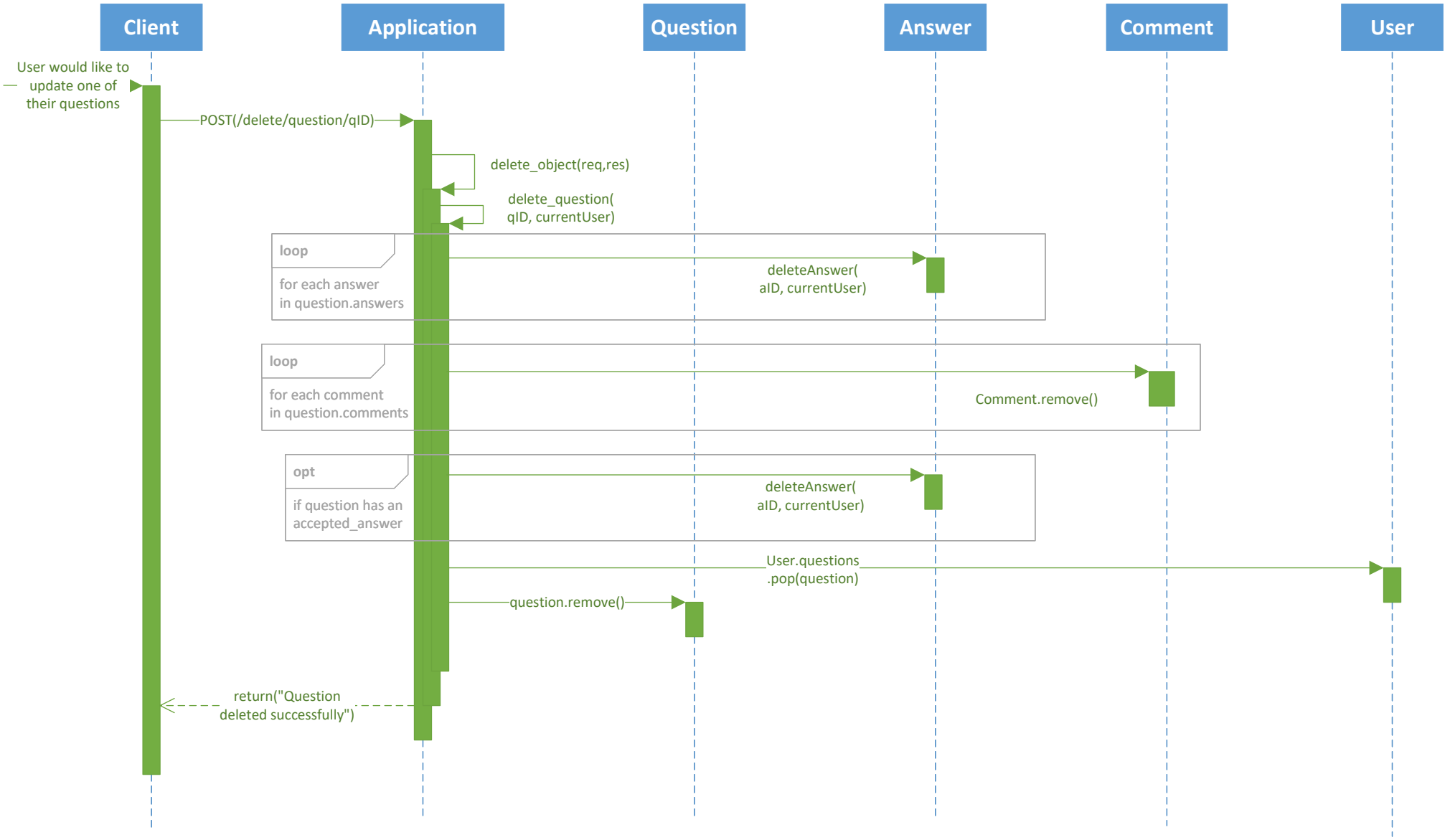
Tags Page

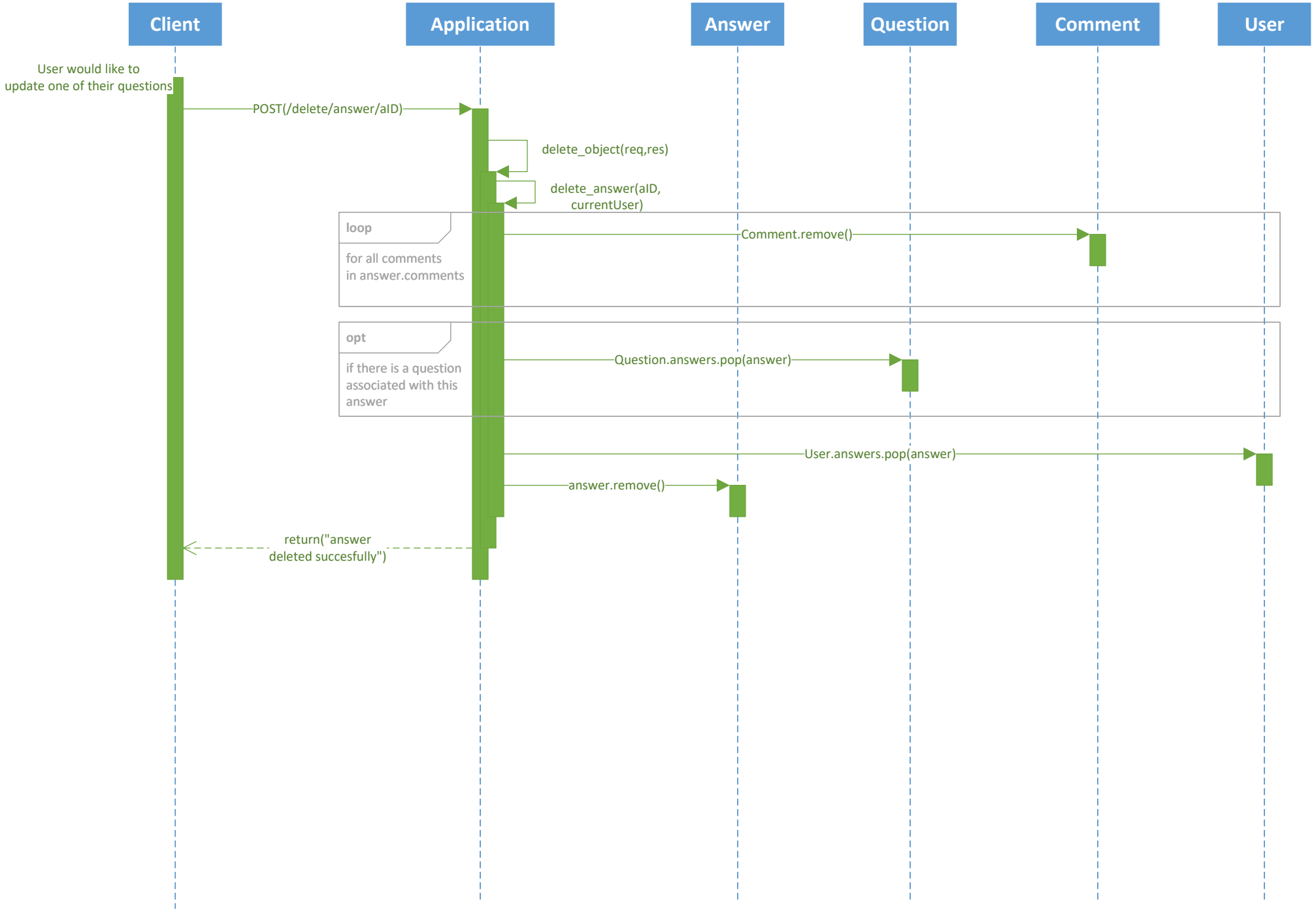

Client
Application
Tag
GET('/tags')
tags
showTags()
Tags
Client
Application
Tag

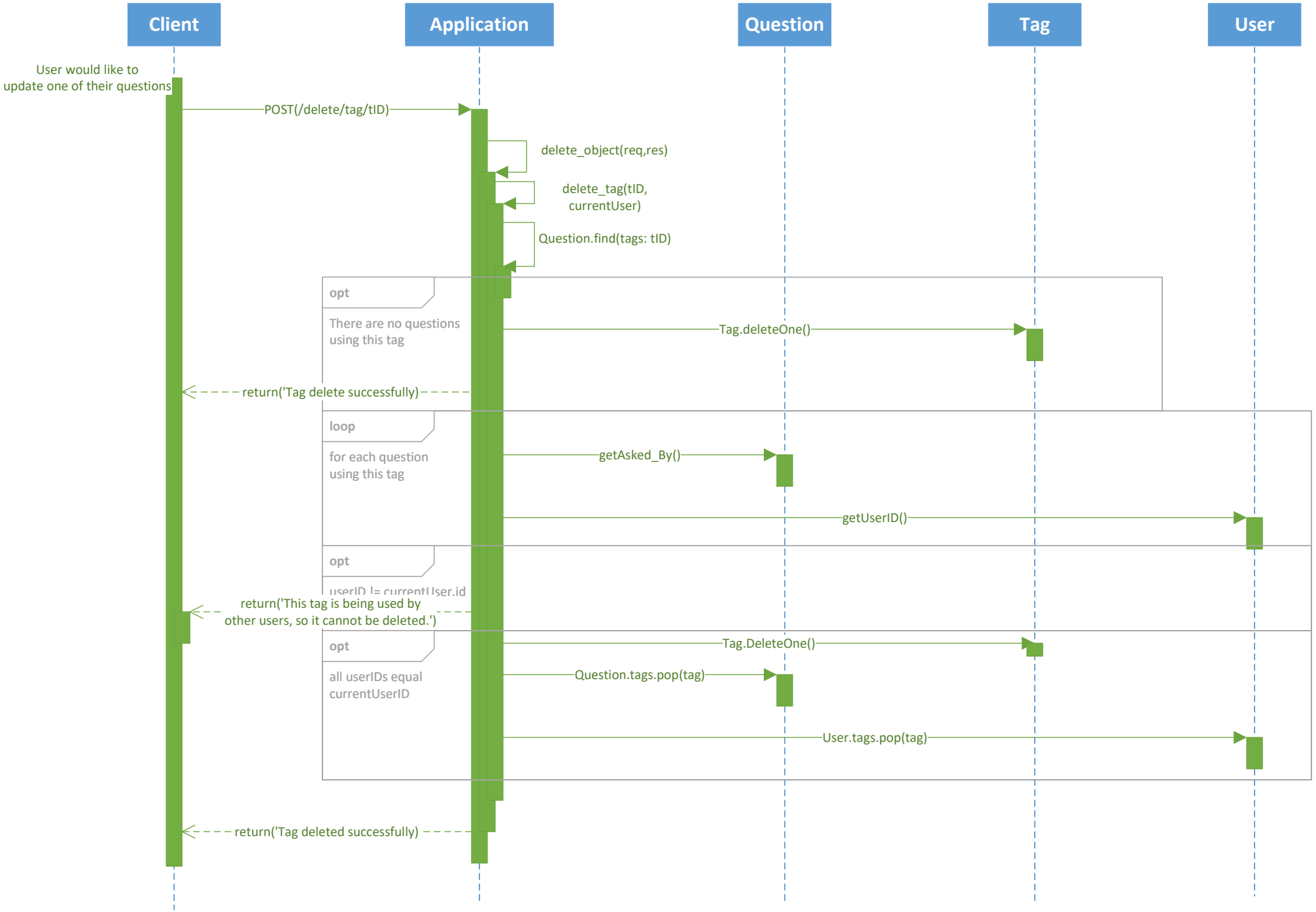

Client
Application
Question
Tag
User
User would like to update one of their questions
POST(/edit/question/qID)
edit_object(req,res)
edit_question(qID, formData, currentUser)
setTitle(formData.newTitle)
setText(formData.newText)
loop
for tag in req.body.formData.newTags
alt
if tag is new
createElement(tag)
User.tags.push(newTag: Tag)
if tag is not new
get(tag._id)
setTags(tagObjects)
setLastActive(Date.now)


Client
Application
Answer
question
User would like to update one of their questions
POST(/edit/answer/aID)
edit_object(req,res)
edit_answer(aID, formData, currentUser)
setText(formData.newText)
Question.findOne(answers: aID)
opt
if there is a question associated with this answer
setLastActive(Date.now)


Client
Application
Question
Tag
User
User would like to update one of their questions
POST(/edit/tag/tID)
edit_object(req,res)
edit_tag(tID, formData, currentUser)
Question.find(tags: tID)
opt
There are no questions using this tag
setName(formData.newName)
return('Tag edited successfully)
loop
for each question using this tag
getAsked_By()
getUserID()
opt
userID != currentUser.id
return('This tag is being used by other users, so it cannot be edited.')
setName(formData.newName)
return('Tag edited successfully)


Client
Application
Question
Answer
Comment
User
User would like to update one of their questions
POST(/delete/question/qID)
delete_object(req,res)
delete_question(qID, currentUser)
loop
for each answer in question.answers
deleteAnswer(aID, currentUser)
loop
for each comment in question.comments
Comment.remove()
opt
if question has an accepted_answer
deleteAnswer(aID, currentUser)
User.questions .pop(question)
question.remove()
return("Question deleted successfully")


Client
Application
Answer
Question
Comment
User
User would like to update one of their questions
POST(/delete/answer/aID)
delete_object(req,res)
delete_answer(aID, currentUser)
loop
for all comments in answer.comments
Comment.remove()
opt
if there is a question associated with this answer
Question.answers.pop(answer)
User.answers.pop(answer)
answer.remove()
return("answer deleted succesfully")


Client
Application
Question
Tag
User
User would like to update one of their questions
POST(/delete/tag/tID)
delete_object(req,res)
delete_tag(tID, currentUser)
Question.find(tags: tID)
opt
There are no questions using this tag
Tag.deleteOne()
return('Tag delete successfully)
loop
for each question using this tag
getAsked_By()
getUserID()
opt
userID != currentUser.id
return('This tag is being used by other users, so it cannot be deleted.')
opt
all userIDs equal currentUserID
Tag.DeleteOne()
Question.tags.pop(tag)
User.tags.pop(tag)
return('Tag deleted successfully)