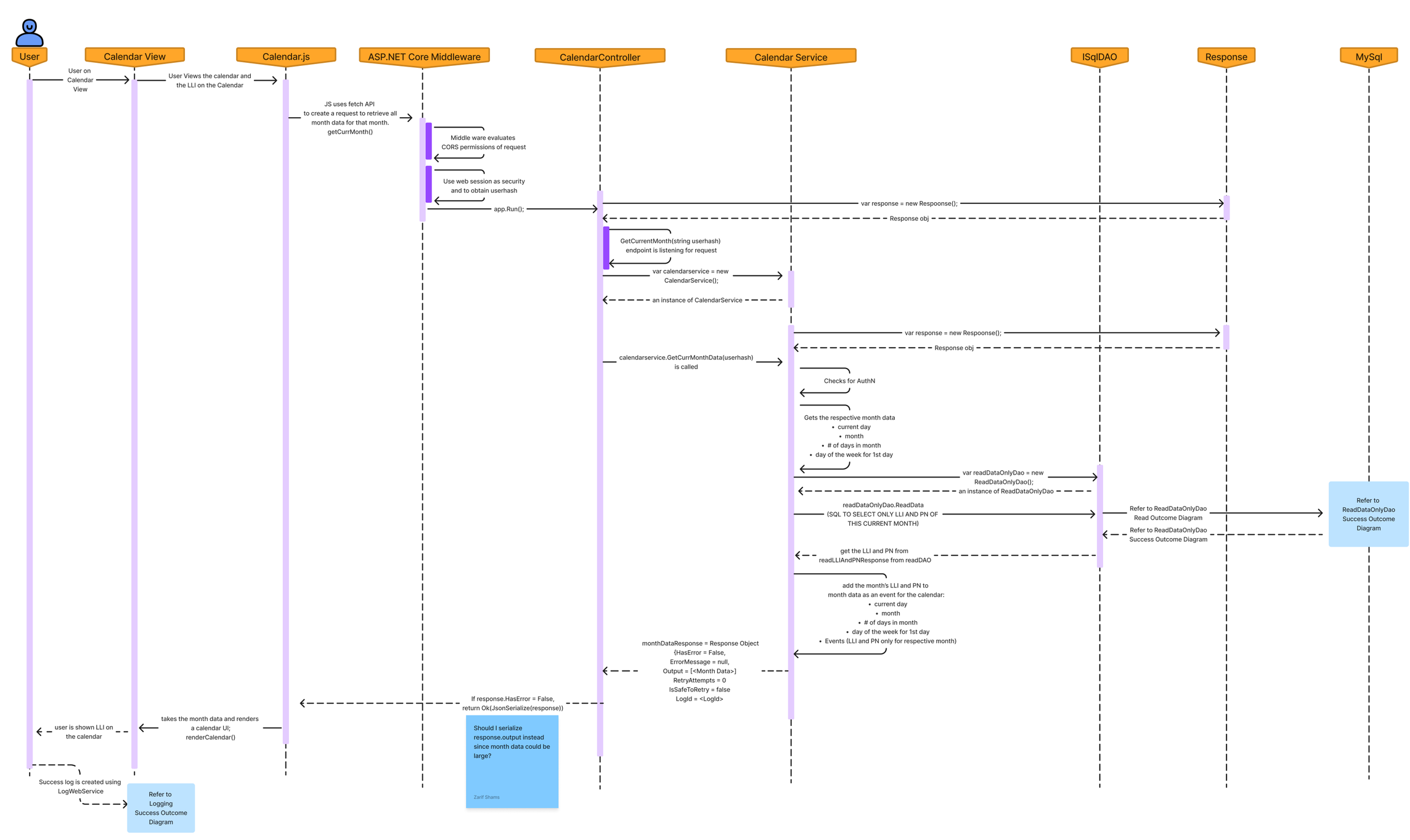
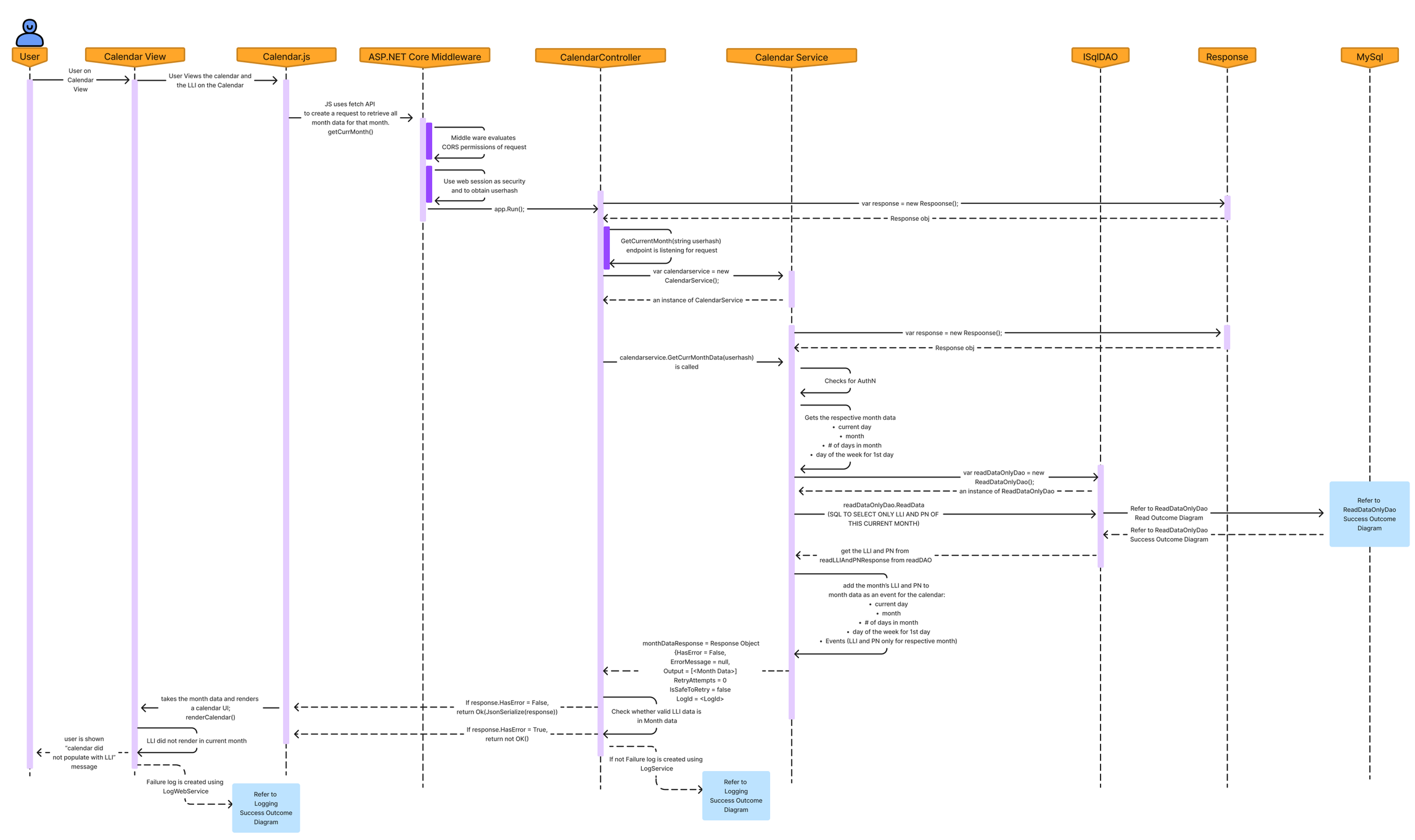
# Lifelog Calendar

## LC1 User Story: View Calendar

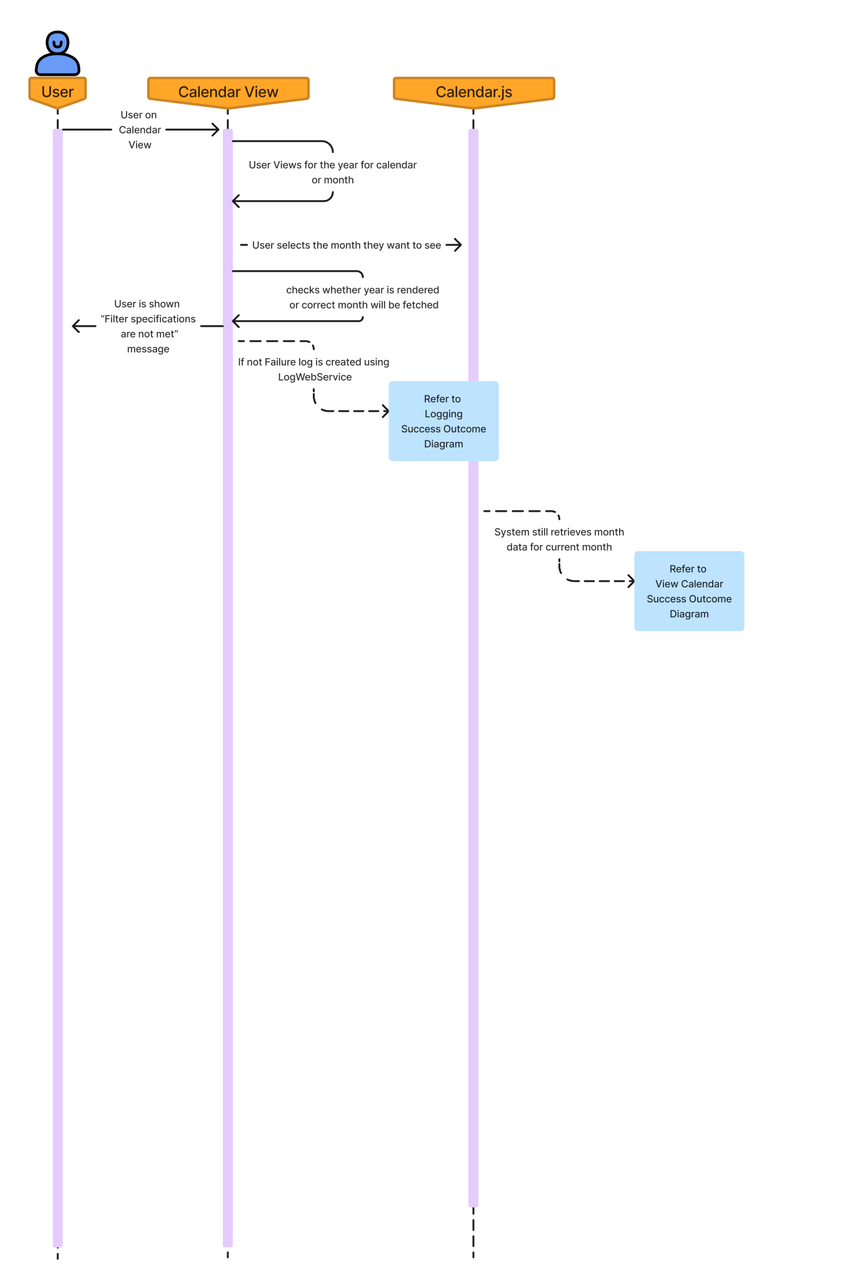
View Calendar Success Outcome


User
Calendar View
Calendar.js
ASP.NET Core Middleware
CalendarController
Calendar Service
ISqlDAO
Response
MySql
User on Calendar View
User Views the calendar and the LLI on the Calendar
JS uses fetch API to create a request to retrieve all month data for that month. getCurrMonth()
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
var response = new Respoonse();
Response obj
GetCurrentMonth(string userhash) endpoint is listening for request
var calendarservice = new CalendarService();
an instance of CalendarService
var response = new Respoonse();
Response obj
calendarservice.GetCurrMonthData(userhash) is called
Checks for AuthN
Gets the respective month data • current day • month • # of days in month • day of the week for 1st day
var readDataOnlyDao = new ReadDataOnlyDao();
an instance of ReadDataOnlyDao
readDataOnlyDao.ReadData (SQL TO SELECT ONLY LLI AND PN OF THIS CURRENT MONTH)
Refer to ReadDataOnlyDao Read Outcome Diagram
Refer to ReadDataOnlyDao Success Outcome Diagram
Refer to ReadDataOnlyDao Success Outcome Diagram
get the LLI and PN from readLLIAndPNResponse from readDAO
add the month's LLI and PN to month data as an event for the calendar: • current day • month • # of days in month • day of the week for 1st day • Events (LLI and PN only for respective month)
monthDataResponse = Response Object {HasError = False, ErrorMessage = null, Output = [<Month Data>] RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
If response.HasError = False, return Ok(JsonSerialize(response))
takes the month data and renders a calendar UI; renderCalendar()
user is shown LLI on the calendar
Success log is created using LogWebService
Refer to Logging Success Outcome Diagram
Should I serialize response.output instead since month data could be large?

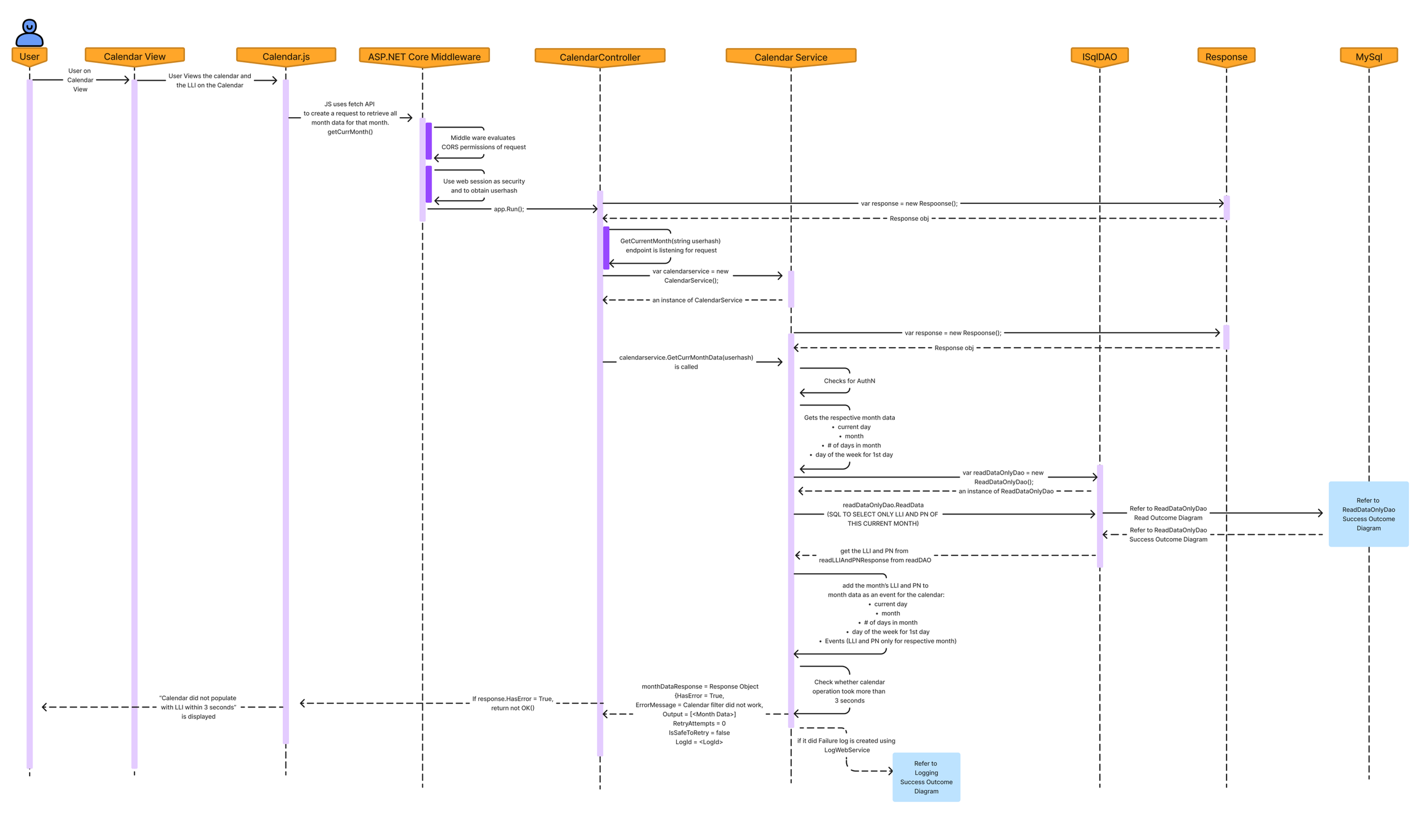
View Calendar Failure Outcome: Calendar did not populate with LLI


User
Calendar View
Calendar.js
ASP.NET Core Middleware
CalendarController
Calendar Service
ISqlDAO
Response
MySql
User on Calendar View
User Views the calendar and the LLI on the Calendar
JS uses fetch API to create a request to retrieve all month data for that month. getCurrMonth()
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
var response = new Respoonse();
Response obj
GetCurrentMonth(string userhash) endpoint is listening for request
var calendarservice = new CalendarService();
an instance of CalendarService
var response = new Respoonse();
Response obj
calendarservice.GetCurrMonthData(userhash) is called
Checks for AuthN
Gets the respective month data • current day • month • # of days in month • day of the week for 1st day
var readDataOnlyDao = new ReadDataOnlyDao();
an instance of ReadDataOnlyDao
readDataOnlyDao.ReadData (SQL TO SELECT ONLY LLI AND PN OF THIS CURRENT MONTH)
Refer to ReadDataOnlyDao Read Outcome Diagram
Refer to ReadDataOnlyDao Success Outcome Diagram
Refer to ReadDataOnlyDao Success Outcome Diagram
get the LLI and PN from readLLIAndPNResponse from readDAO
add the month's LLI and PN to month data as an event for the calendar: • current day • month • # of days in month • day of the week for 1st day • Events (LLI and PN only for respective month)
monthDataResponse = Response Object {HasError = False, ErrorMessage = null, Output = [<Month Data>] RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
If response.HasError = False, return Ok(JsonSerialize(response))
Check whether valid LLI data is in Month data
If response.HasError = True, return not OK()
If not Failure log is created using LogService
Refer to Logging Success Outcome Diagram
takes the month data and renders a calendar UI; renderCalendar()
LLI did not render in current month
user is shown "calendar did not populate with LLI" message
Failure log is created using LogWebService
Refer to Logging Success Outcome Diagram

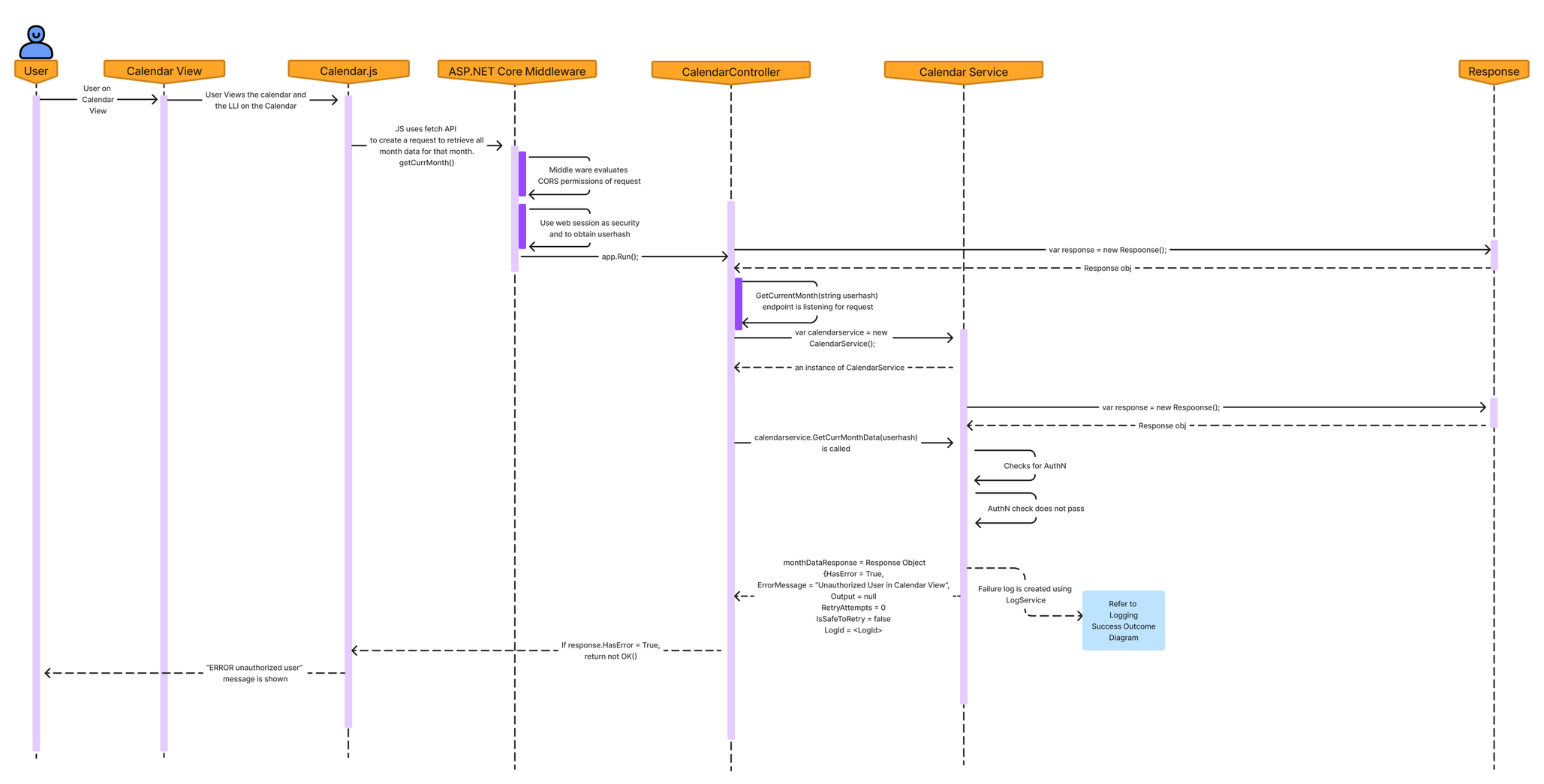
View Calendar Failure Outcome: Calendar did not meet filter specifications


User
Calendar View
Calendar.js
User on Calendar View
User Views for the year for calendar or month
User selects the month they want to see
checks whether year is rendered or correct month will be fetched
User is shown "Filter specifications are not met" message
If not Failure log is created using LogWebService
Refer to Logging Success Outcome Diagram
System still retrieves month data for current month
Refer to View Calendar Success Outcome Diagram

View Calendar Failure Outcome: Calendar display took more than 3 seconds

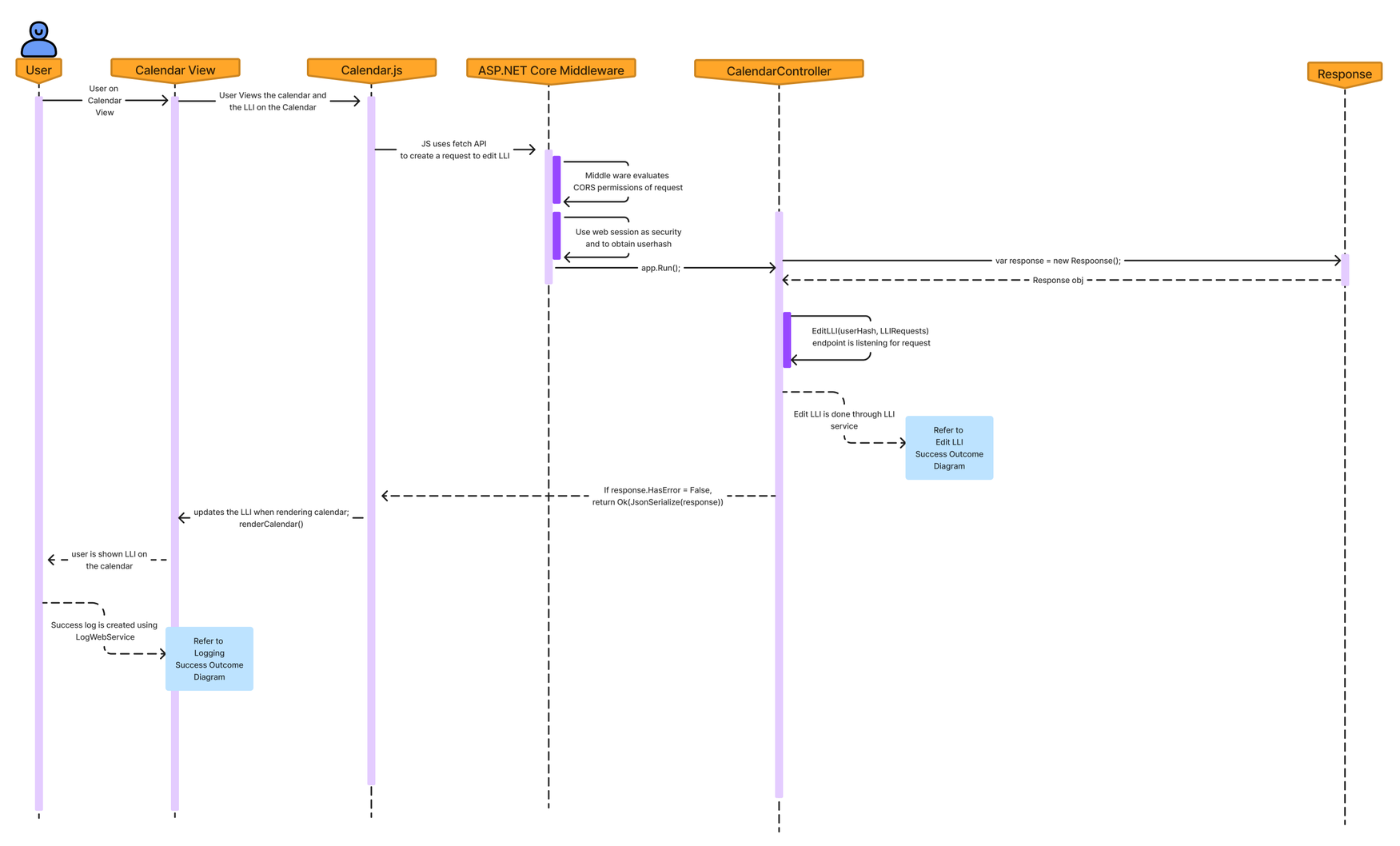

User
Calendar View
Calendar.js
ASP.NET Core Middleware
CalendarController
Calendar Service
ISqlDAO
Response
MySql
User on Calendar View
User Views the calendar and the LLI on the Calendar
JS uses fetch API to create a request to retrieve all month data for that month. getCurrMonth()
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
var response = new Respoonse();
Response obj
GetCurrentMonth(string userhash) endpoint is listening for request
var calendarservice = new CalendarService();
an instance of CalendarService
var response = new Respoonse();
Response obj
calendarservice.GetCurrMonthData(userhash) is called
Checks for AuthN
Gets the respective month data • current day • month • # of days in month • day of the week for 1st day
var readDataOnlyDao = new ReadDataOnlyDao();
an instance of ReadDataOnlyDao
readDataOnlyDao.ReadData (SQL TO SELECT ONLY LLI AND PN OF THIS CURRENT MONTH)
Refer to ReadDataOnlyDao Read Outcome Diagram
Refer to ReadDataOnlyDao Success Outcome Diagram
Refer to ReadDataOnlyDao Success Outcome Diagram
get the LLI and PN from readLLIAndPNResponse from readDAO
add the month's LLI and PN to month data as an event for the calendar: • current day • month • # of days in month • day of the week for 1st day • Events (LLI and PN only for respective month)
Check whether calendar operation took more than 3 seconds
monthDataResponse = Response Object {HasError = True, ErrorMessage = Calendar filter did not work, Output = [<Month Data>] RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
If response.HasError = True, return not OK()
"Calendar did not populate with LLI within 3 seconds" is displayed
If it did Failure log is created using LogWebService
Refer to Logging Success Outcome Diagram

View Calendar Failure Outcome: Unauthorized user in Calendar View

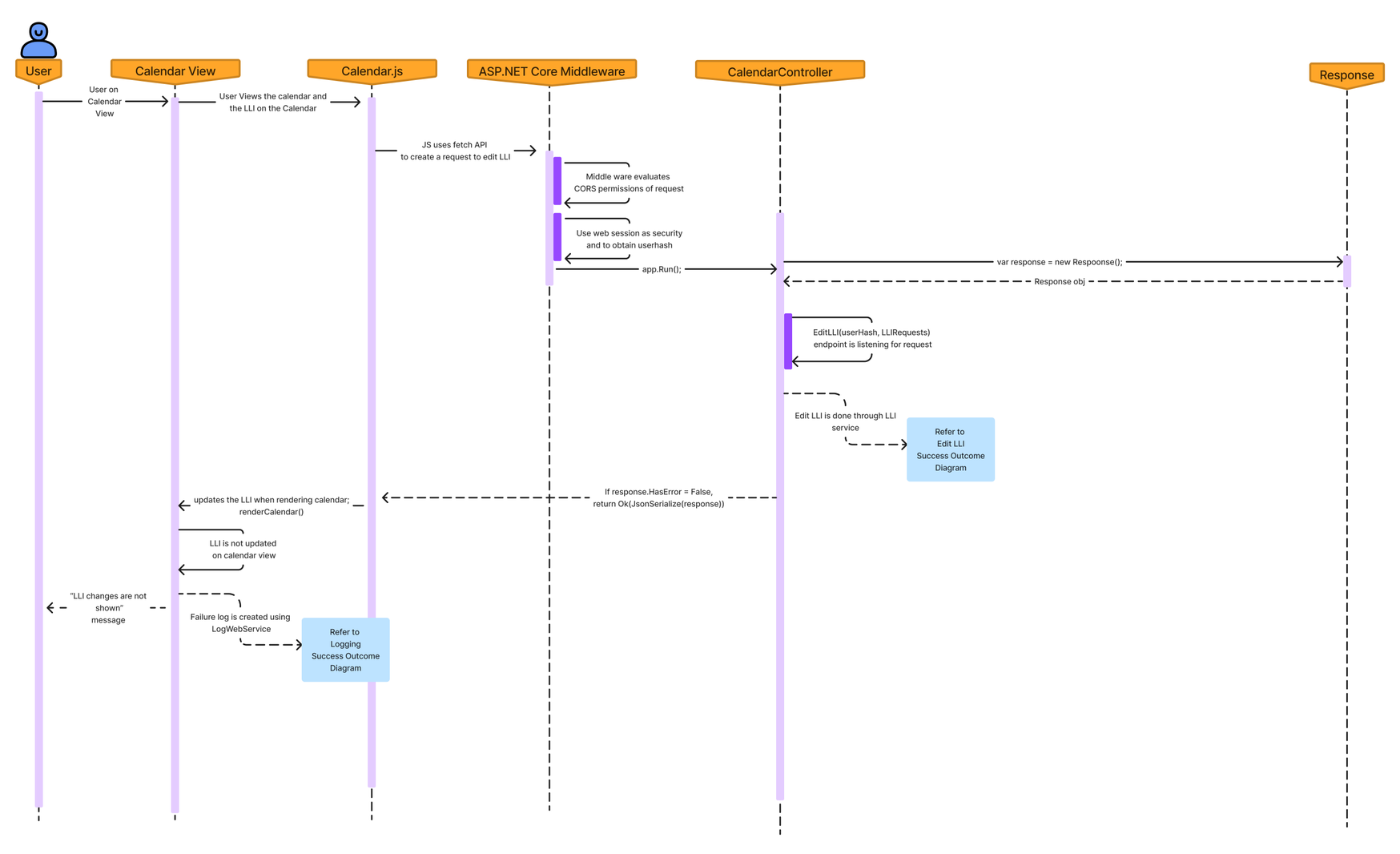

User
Calendar View
Calendar.js
ASP.NET Core Middleware
CalendarController
Calendar Service
Response
User on Calendar View
User Views the calendar and the LLI on the Calendar
JS uses fetch API to create a request to retrieve all month data for that month. getCurrMonth()
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
var response = new Respoonse();
Response obj
GetCurrentMonth(string userhash) endpoint is listening for request
var calendarservice = new CalendarService();
an instance of CalendarService
var response = new Respoonse();
Response obj
calendarservice.GetCurrMonthData(userhash) is called
Checks for AuthN
AuthN check does not pass
monthDataResponse = Response Object {HasError = True, ErrorMessage = "Unauthorized User in Calendar View", Output = null RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
Failure log is created using LogService
Refer to Logging Success Outcome Diagram
If response.HasError = True, return not OKO
"ERROR unauthorized user" message is shown

## LC2 User Story: Edit LLI using Calendar

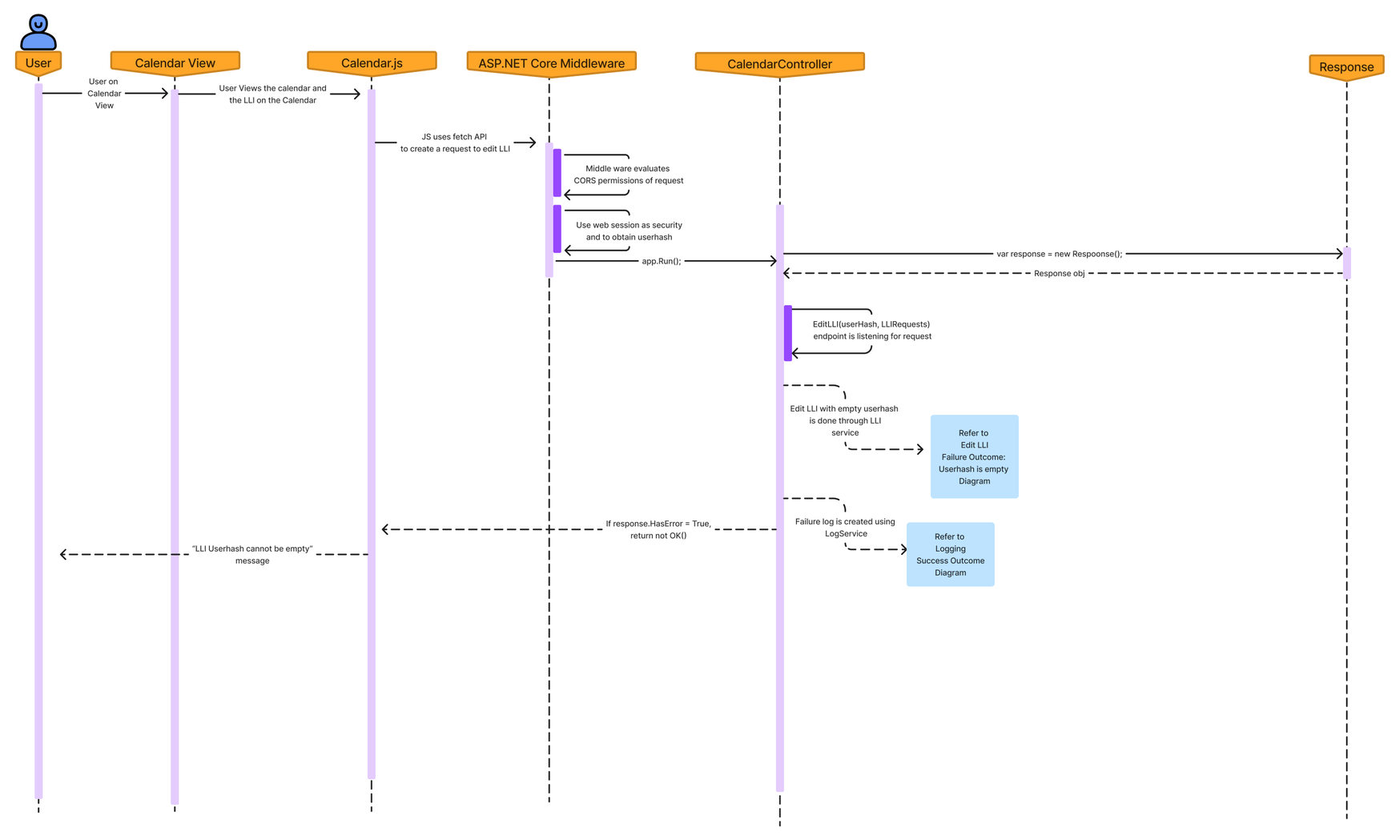
LC2 Success Outcome: Edit LLI using Calendar


User
Calendar View
Calendar.js
ASP.NET Core Middleware
CalendarController
Response
User on Calendar View
User Views the calendar and the LLI on the Calendar
JS uses fetch API to create a request to edit LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
var response = new Respoonse();
Response obj
EditLLI(userHash, LLIRequests) endpoint is listening for request
Edit LLI is done through LLI service
Refer to Edit LLI Success Outcome Diagram
If response.HasError = False, return Ok(JsonSerialize(response))
updates the LLI when rendering calendar; renderCalendar()
user is shown LLI on the calendar
Success log is created using LogWebService
Refer to Logging Success Outcome Diagram

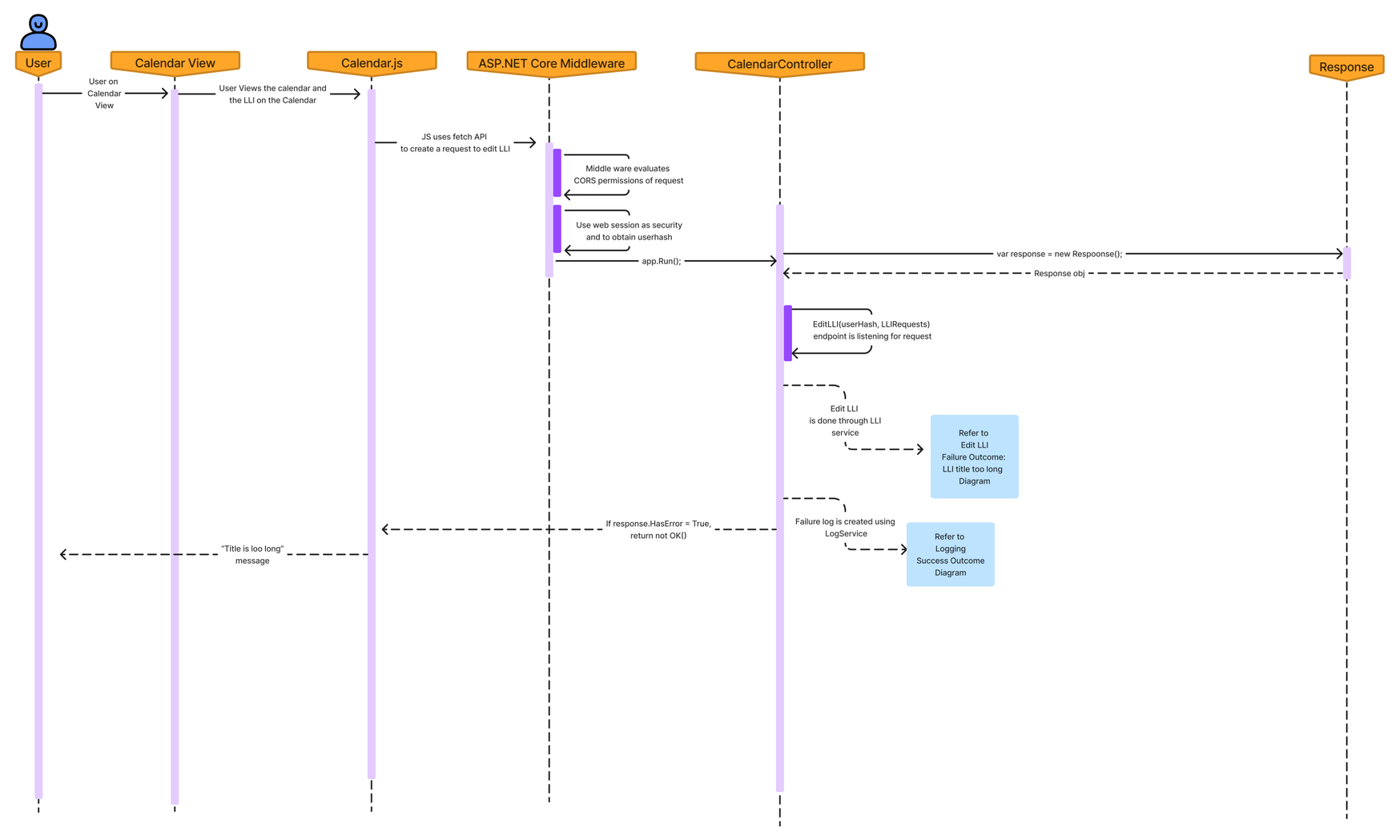
LC2 Failure Outcome: Calendar doesn't reflect updated LLI


User
Calendar View
Calendar.js
ASP.NET Core Middleware
CalendarController
Response
User on Calendar View
User Views the calendar and the LLI on the Calendar
JS uses fetch API to create a request to edit LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
var response = new Respoonse();
Response obj
EditLLI(userHash, LLIRequests) endpoint is listening for request
Edit LLI is done through LLI service
Refer to Edit LLI Success Outcome Diagram
If response.HasError = False, return Ok(JsonSerialize(response))
updates the LLI when rendering calendar; renderCalendar()
LLI is not updated on calendar view
"LLI changes are not shown" message
Failure log is created using LogWebService
Refer to Logging Success Outcome Diagram

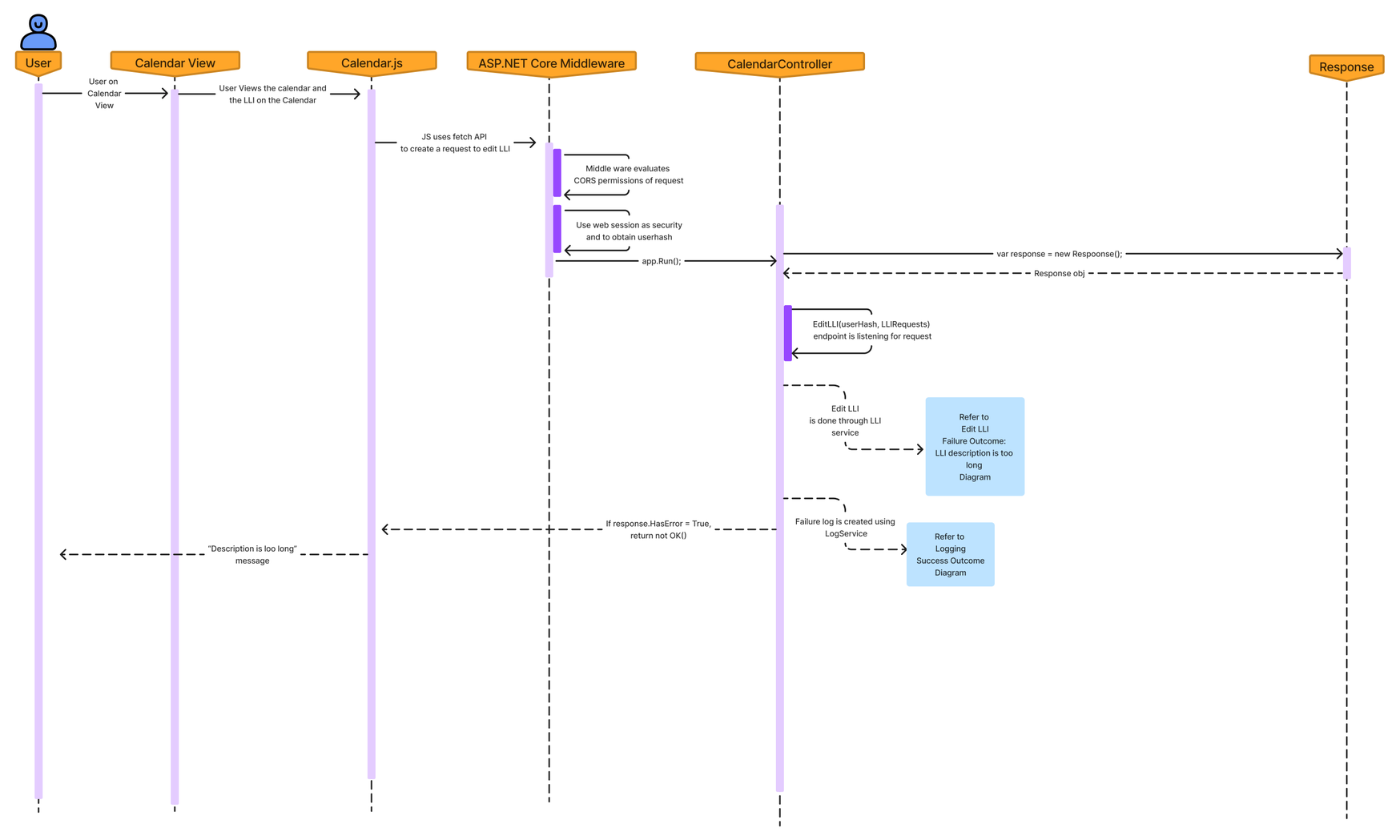
LC2 Failure Outcome: UserHash is empty


User
Calendar View
Calendar.js
ASP.NET Core Middleware
CalendarController
Response
User on Calendar View
User Views the calendar and the LLI on the Calendar
JS uses fetch API to create a request to edit LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
var response = new Respoonse();
Response obj
EditLLI(userHash, LLIRequests) endpoint is listening for request
Edit LLI with empty userhash is done through LLI service
Refer to Edit LLI Failure Outcome: Userhash is empty Diagram
If response.HasError = True, return not OK()
Failure log is created using LogService
Refer to Logging Success Outcome Diagram
"LLI Userhash cannot be empty" message

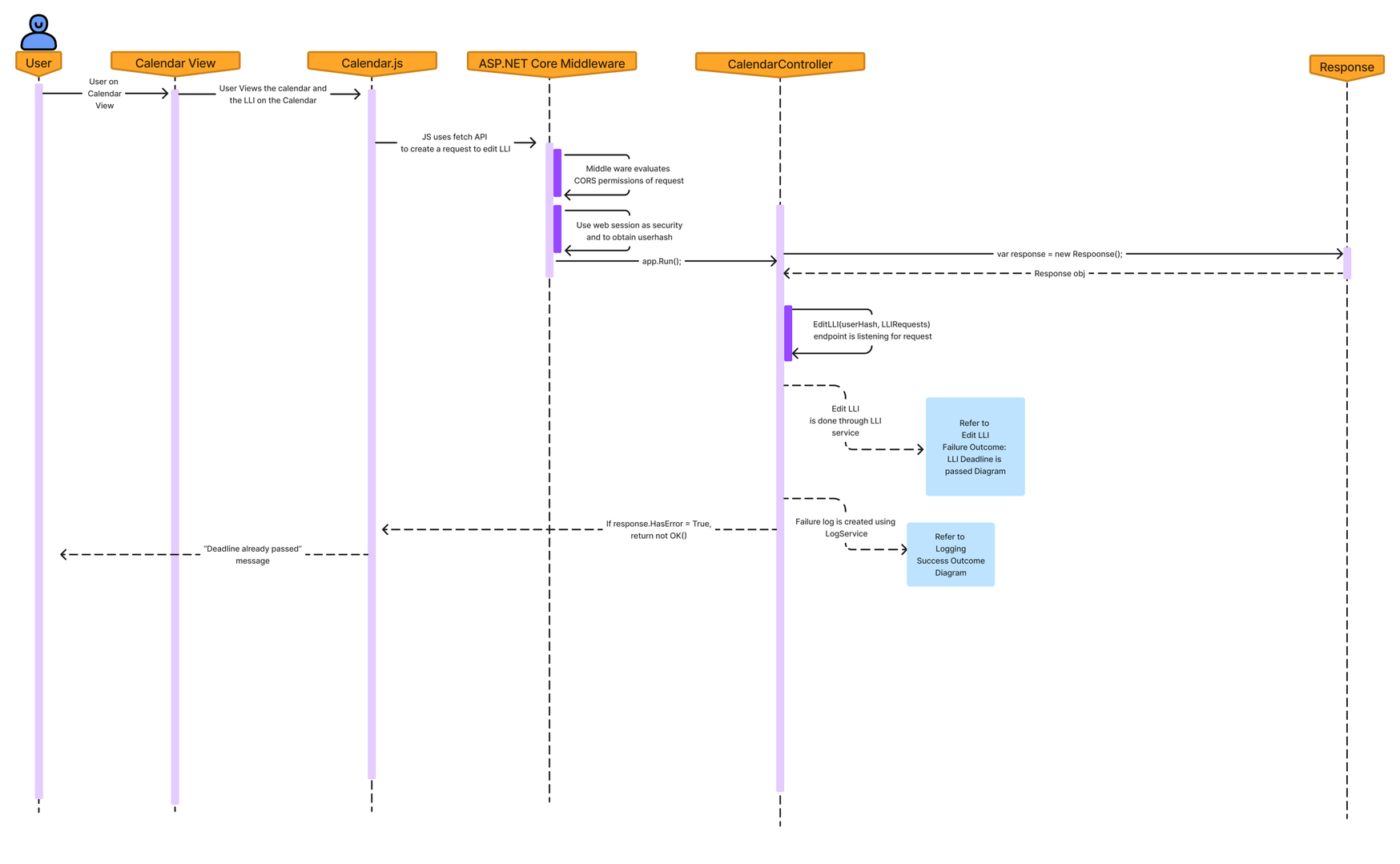
LC2 Failure Outcome: LLI title is too long


User
Calendar View
Calendar.js
ASP.NET Core Middleware
CalendarController
Response
User on Calendar View
User Views the calendar and the LLI on the Calendar
JS uses fetch API to create a request to edit LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
var response = new Respoonse();
Response obj
EditLLI(userHash, LLIRequests) endpoint is listening for request
Edit LLI is done through LLI service
Refer to Edit LLI Failure Outcome: LLI title too long Diagram
If response.HasError = True, return not OK()
Failure log is created using LogService
Refer to Logging Success Outcome Diagram
"Title is too long" message

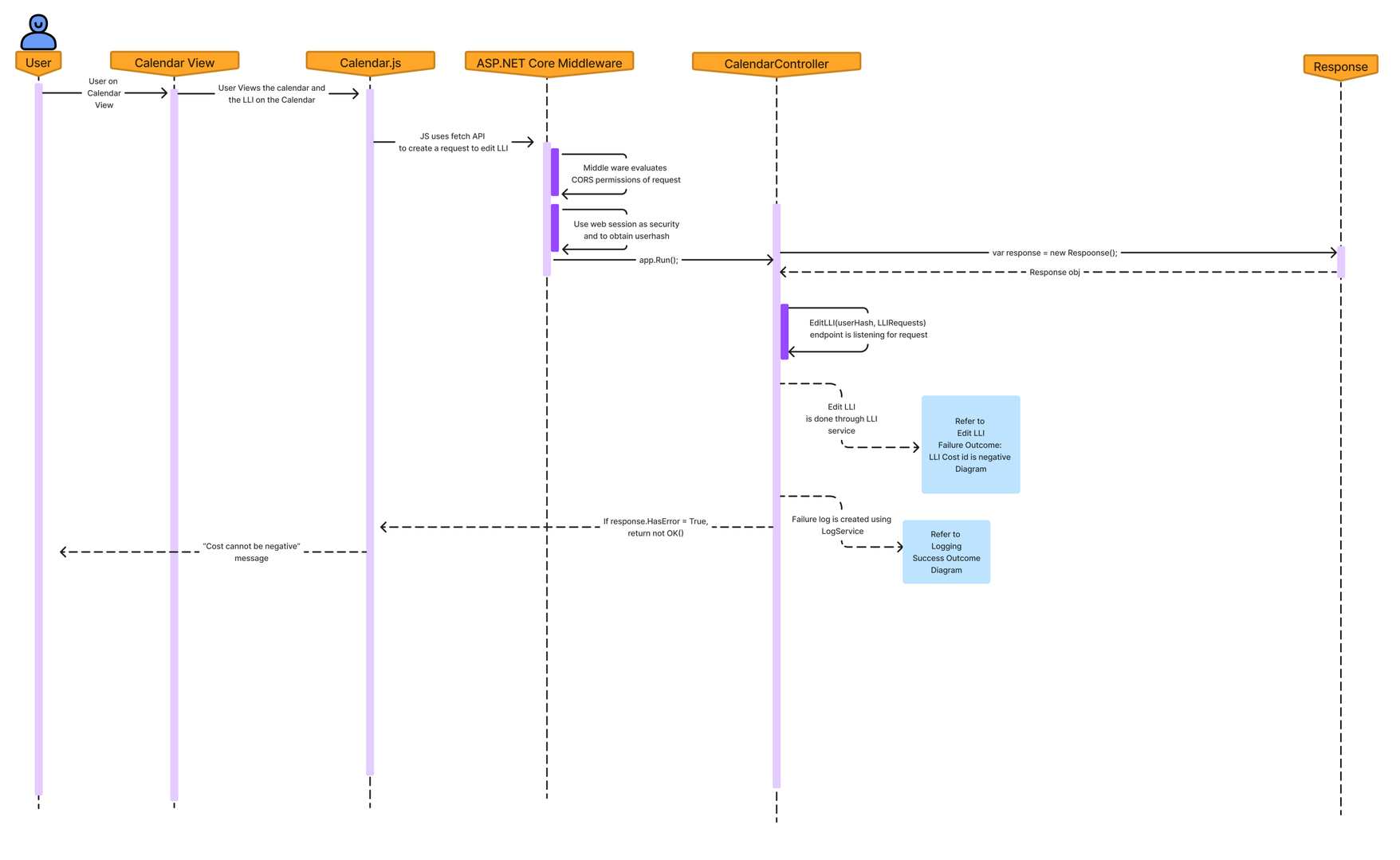
LC2 Failure Outcome: LLI description is too long


User
Calendar View
Calendar.js
ASP.NET Core Middleware
CalendarController
Response
User on Calendar View
User Views the calendar and the LLI on the Calendar
JS uses fetch API to create a request to edit LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
var response = new Respoonse();
Response obj
EditLLI(userHash, LLIRequests) endpoint is listening for request
Edit LLI is done through LLI service
Refer to Edit LLI Failure Outcome: LLI description is too long Diagram
If response.HasError = True, return not OK()
Failure log is created using LogService
Refer to Logging Success Outcome Diagram
"Description is too long" message

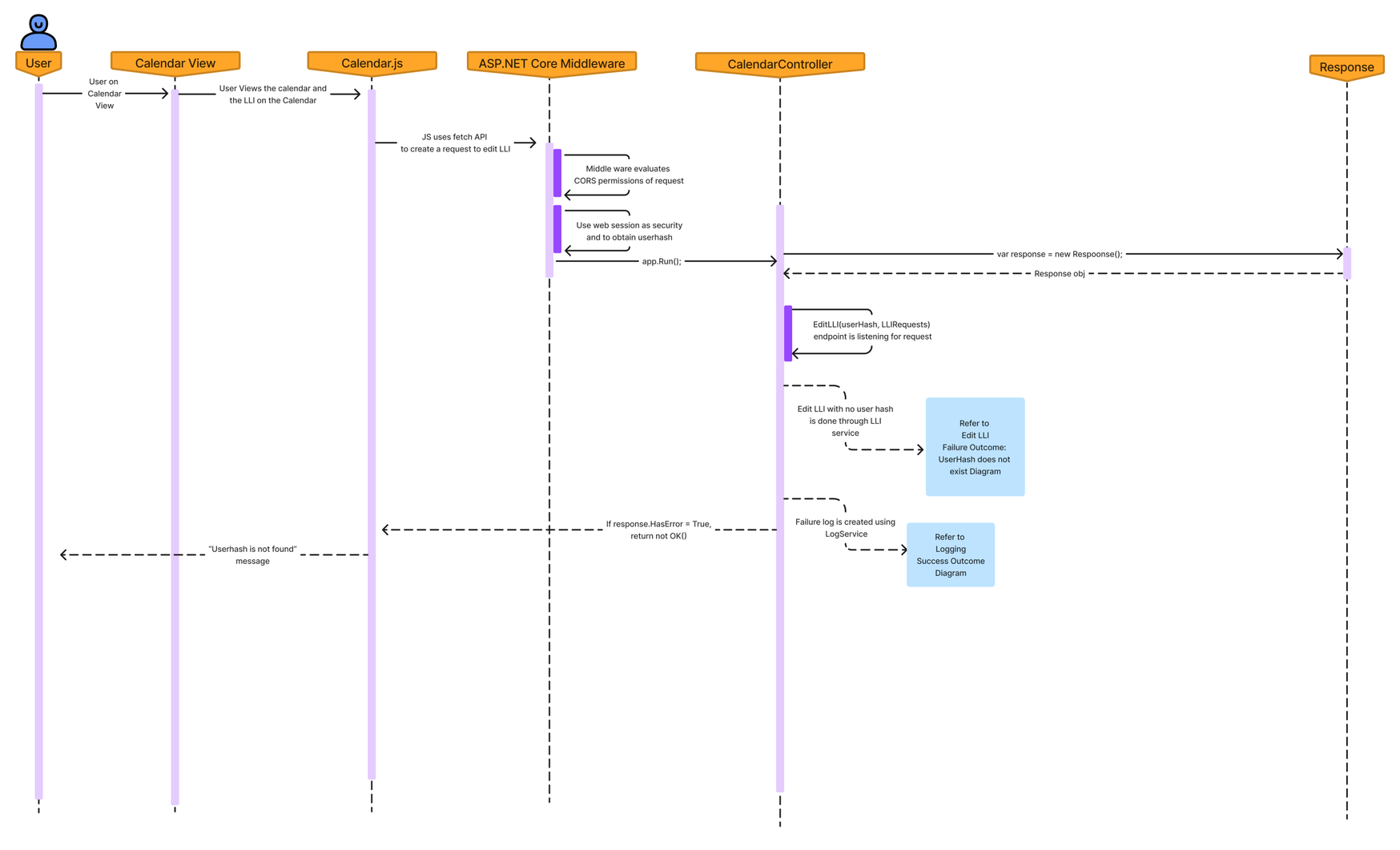
LC2 Failure Outcome: LLI deadline is passed


User
Calendar View
Calendar.js
ASP.NET Core Middleware
CalendarController
Response
User on Calendar View
User Views the calendar and the LLI on the Calendar
JS uses fetch API to create a request to edit LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
var response = new Respoonse();
Response obj
EditLLI(userHash, LLIRequests) endpoint is listening for request
Edit LLI is done through LLI service
Refer to Edit LLI Failure Outcome: LLI Deadline is passed Diagram
If response.HasError = True, return not OK()
Failure log is created using LogService
Refer to Logging Success Outcome Diagram
"Deadline already passed" message

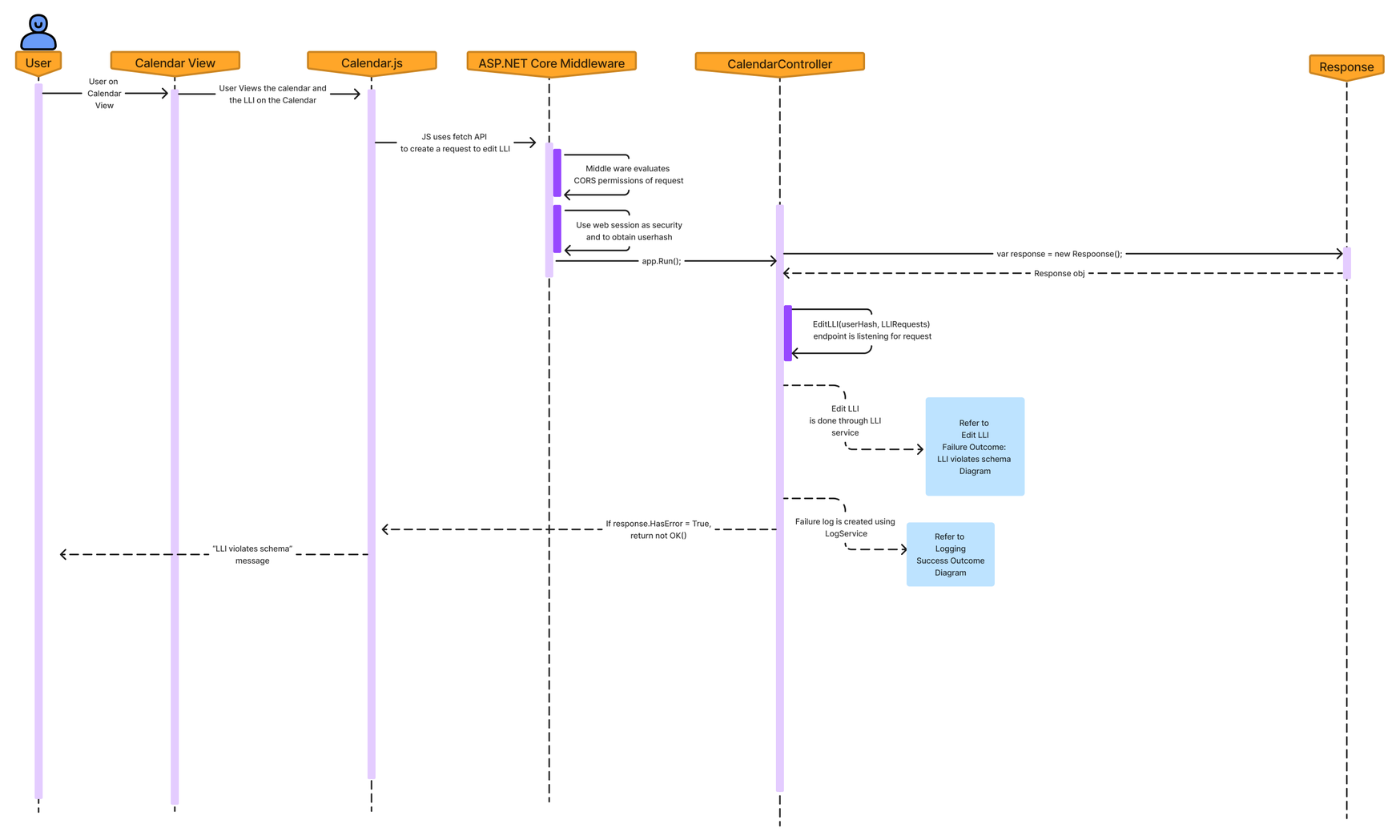
LC2 Failure Outcome: LLI cost is negative


User
Calendar View
Calendar.js
ASP.NET Core Middleware
CalendarController
Response
User on Calendar View
User Views the calendar and the LLI on the Calendar
JS uses fetch API to create a request to edit LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
var response = new Respoonse();
Response obj
EditLLI(userHash, LLIRequests) endpoint is listening for request
Edit LLI is done through LLI service
Refer to Edit LLI Failure Outcome: LLI Cost id is negative Diagram
If response.HasError = True, return not OK()
Failure log is created using LogService
Refer to Logging Success Outcome Diagram
"Cost cannot be negative" message

LC2 Failure Outcome: User hash does not exist

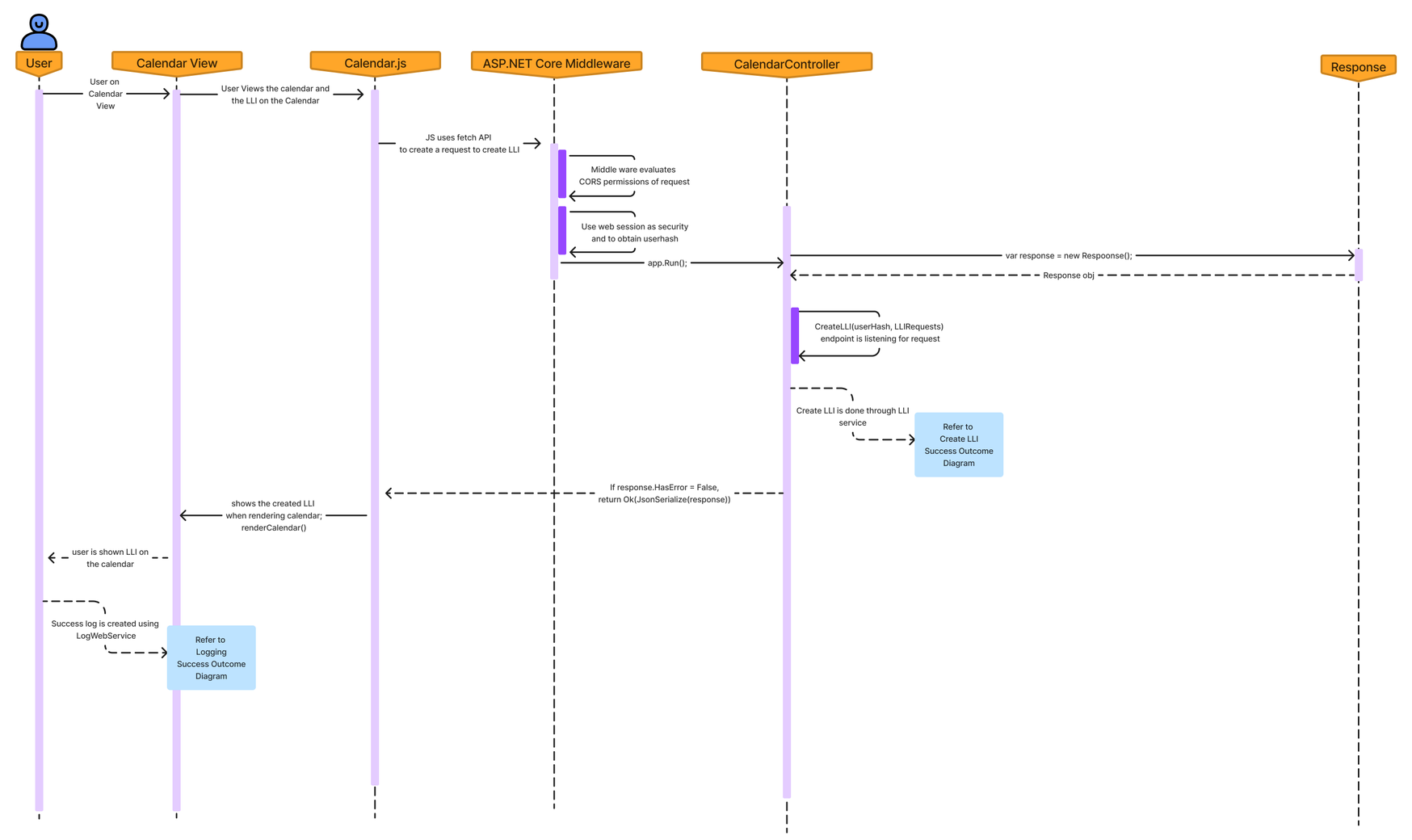

User
Calendar View
Calendar.js
ASP.NET Core Middleware
CalendarController
Response
User on Calendar View
User Views the calendar and the LLI on the Calendar
JS uses fetch API to create a request to edit LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
var response = new Respoonse();
Response obj
EditLLI(userHash, LLIRequests) endpoint is listening for request
Edit LLI with no user hash is done through LLI service
Refer to Edit LLI Failure Outcome: UserHash does not exist Diagram
If response.HasError = True, return not OK()
Failure log is created using LogService
Refer to Logging Success Outcome Diagram
"Userhash is not found" message

LC2 Failure Outcome: LLI violates LLI schema

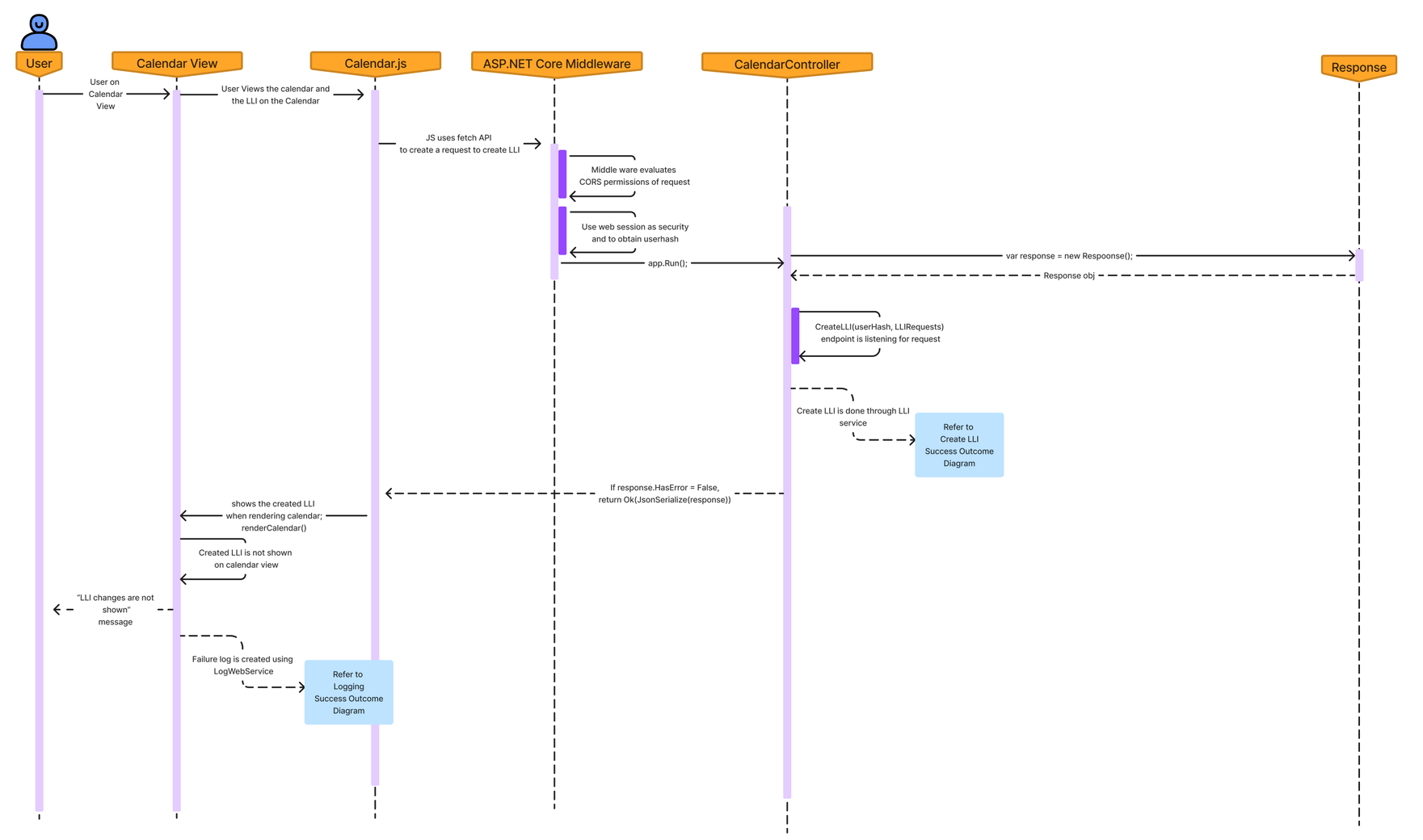

User
Calendar View
Calendar.js
ASP.NET Core Middleware
CalendarController
Response
User on Calendar View
User Views the calendar and the LLI on the Calendar
JS uses fetch API to create a request to edit LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
var response = new Respoonse();
Response obj
EditLLI(userHash, LLIRequests) endpoint is listening for request
Edit LLI is done through LLI service
Refer to Edit LLI Failure Outcome: LLI violates schema Diagram
If response.HasError = True, return not OK()
Failure log is created using LogService
Refer to Logging Success Outcome Diagram
"LLI violates schema" message

## LC3 User Story: Create LLI

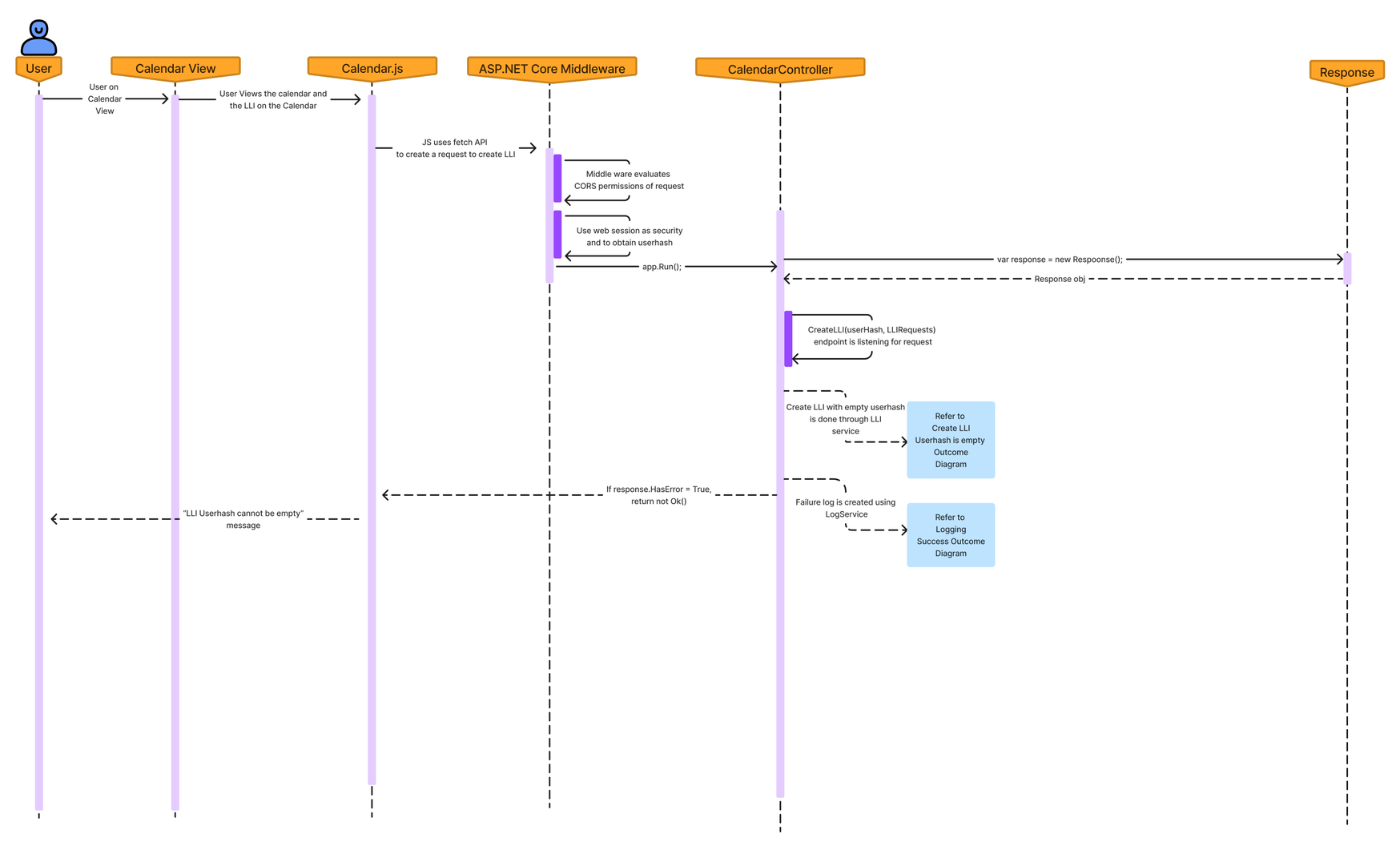
LC3 Success Outcome: Create LLI using Calendar


User
Calendar View
Calendar.js
ASP.NET Core Middleware
CalendarController
Response
User on Calendar View
User Views the calendar and the LLI on the Calendar
JS uses fetch API to create a request to create LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
var response = new Respoonse();
Response obj
CreateLLI(userHash, LLIRequests) endpoint is listening for request
Create LLI is done through LLI service
Refer to Create LLI Success Outcome Diagram
If response.HasError = False, return Ok(JsonSerialize(response))
shows the created LLI when rendering calendar; renderCalendar()
user is shown LLI on the calendar
Success log is created using LogWebService
Refer to Logging Success Outcome Diagram

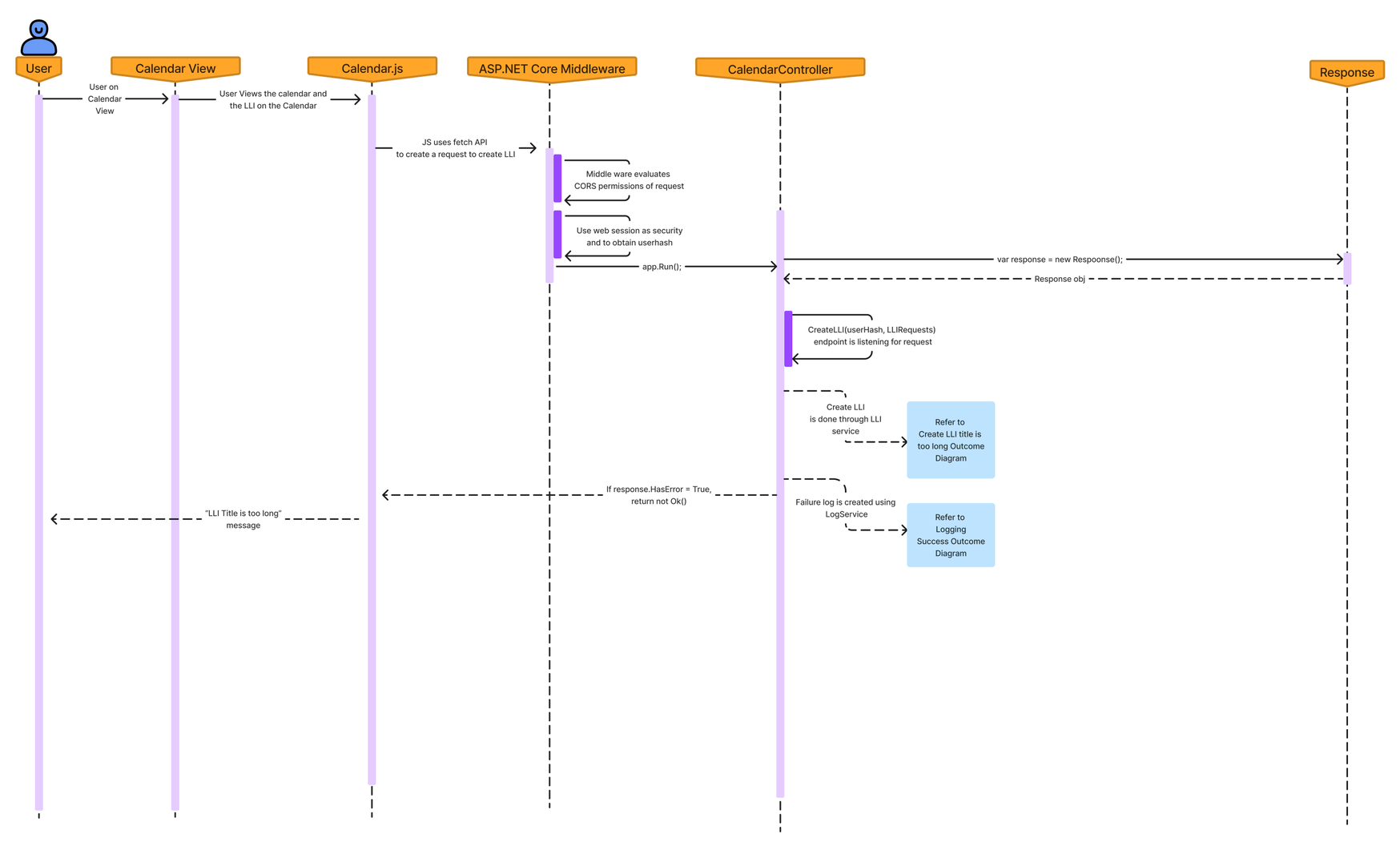
LC3 Failure Outcome: Calendar does not reflect created LLI


User
Calendar View
Calendar.js
ASP.NET Core Middleware
CalendarController
Response
User on Calendar View
User Views the calendar and the LLI on the Calendar
JS uses fetch API to create a request to create LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
var response = new Respoonse();
Response obj
CreateLLI(userHash, LLIRequests) endpoint is listening for request
Create LLI is done through LLI service
Refer to Create LLI Success Outcome Diagram
If response.HasError = False, return Ok(JsonSerialize(response))
shows the created LLI when rendering calendar; renderCalendar()
Created LLI is not shown on calendar view
"LLI changes are not shown" message
Failure log is created using LogWebService
Refer to Logging Success Outcome Diagram

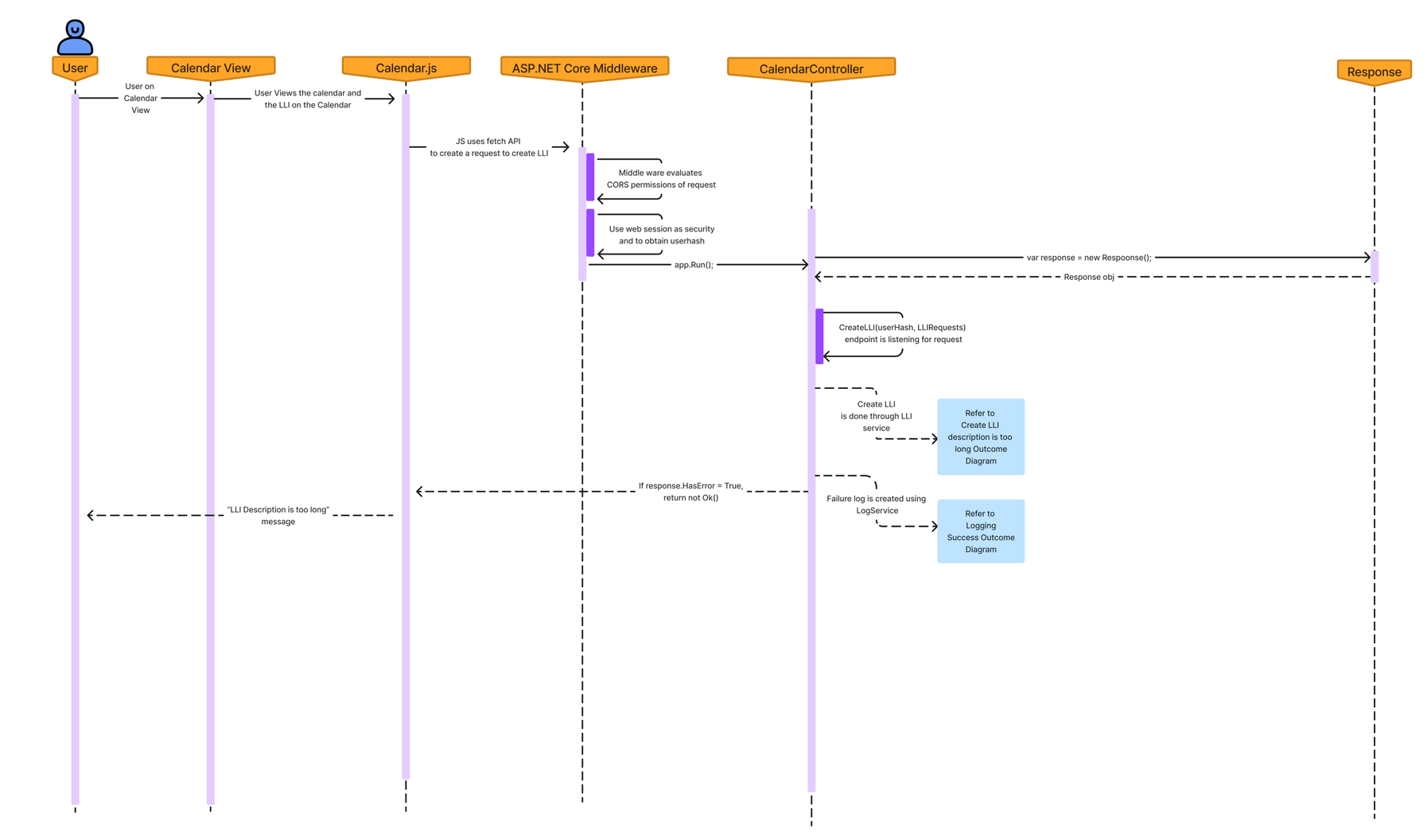
LC3 Failure Outcome: Userhash is empty


User
Calendar View
Calendar.js
ASP.NET Core Middleware
CalendarController
Response
User on Calendar View
User Views the calendar and the LLI on the Calendar
JS uses fetch API to create a request to create LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
var response = new Respoonse();
Response obj
CreateLLI(userHash, LLIRequests) endpoint is listening for request
Create LLI with empty userhash is done through LLI service
Refer to Create LLI Userhash is empty Outcome Diagram
If response.HasError = True, return not OK()
Failure log is created using LogService
Refer to Logging Success Outcome Diagram
"LLI Userhash cannot be empty" message

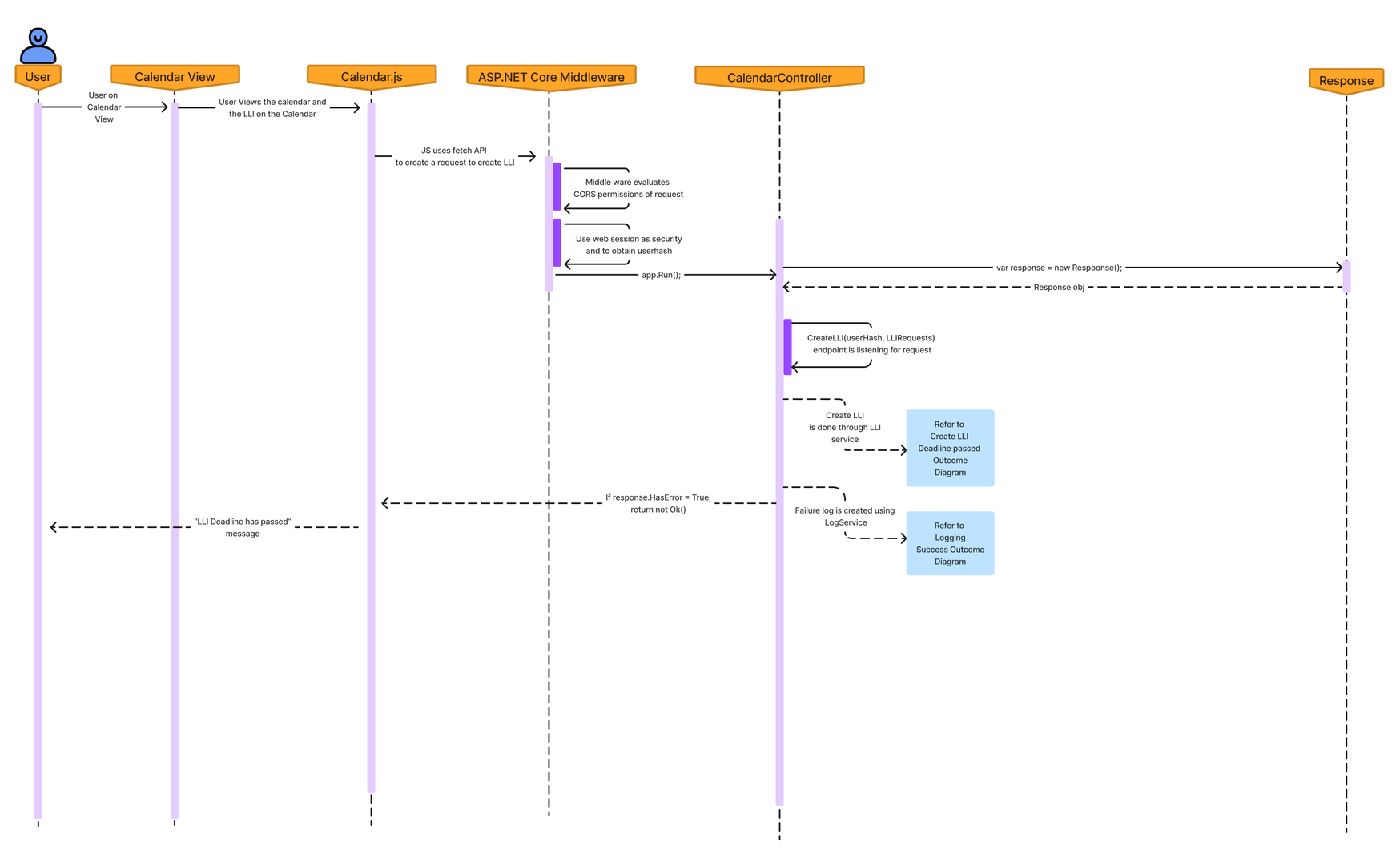
LC3 Failure Outcome: LLI title too long


User
Calendar View
Calendar.js
ASP.NET Core Middleware
CalendarController
Response
User on Calendar View
User Views the calendar and the LLI on the Calendar
JS uses fetch API to create a request to create LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
var response = new Respoonse();
Response obj
CreateLLI(userHash, LLIRequests) endpoint is listening for request
Create LLI is done through LLI service
Refer to Create LLI title is too long Outcome Diagram
If response.HasError = True, return not Ok()
Failure log is created using LogService
Refer to Logging Success Outcome Diagram
"LLI Title is too long" message

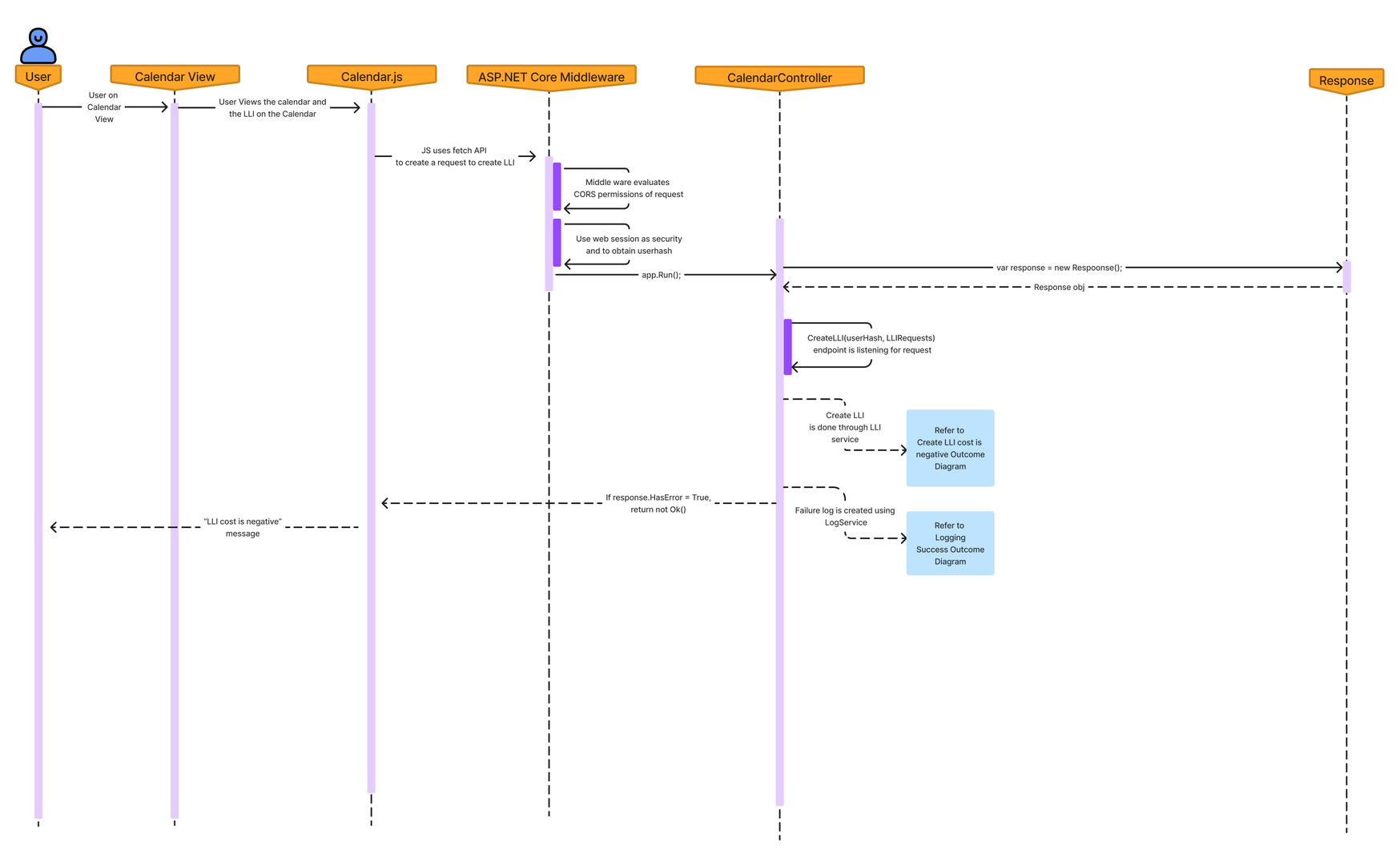
LC3 Failure Outcome: LLI description is too long


User
Calendar View
Calendar.js
ASP.NET Core Middleware
CalendarController
Response
User on Calendar View
User Views the calendar and the LLI on the Calendar
JS uses fetch API to create a request to create LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
var response = new Respoonse();
Response obj
CreateLLI(userHash, LLIRequests) endpoint is listening for request
Create LLI is done through LLI service
Refer to Create LLI description is too long Outcome Diagram
If response.HasError = True, return not Ok()
Failure log is created using LogService
Refer to Logging Success Outcome Diagram
"LLI Description is too long" message

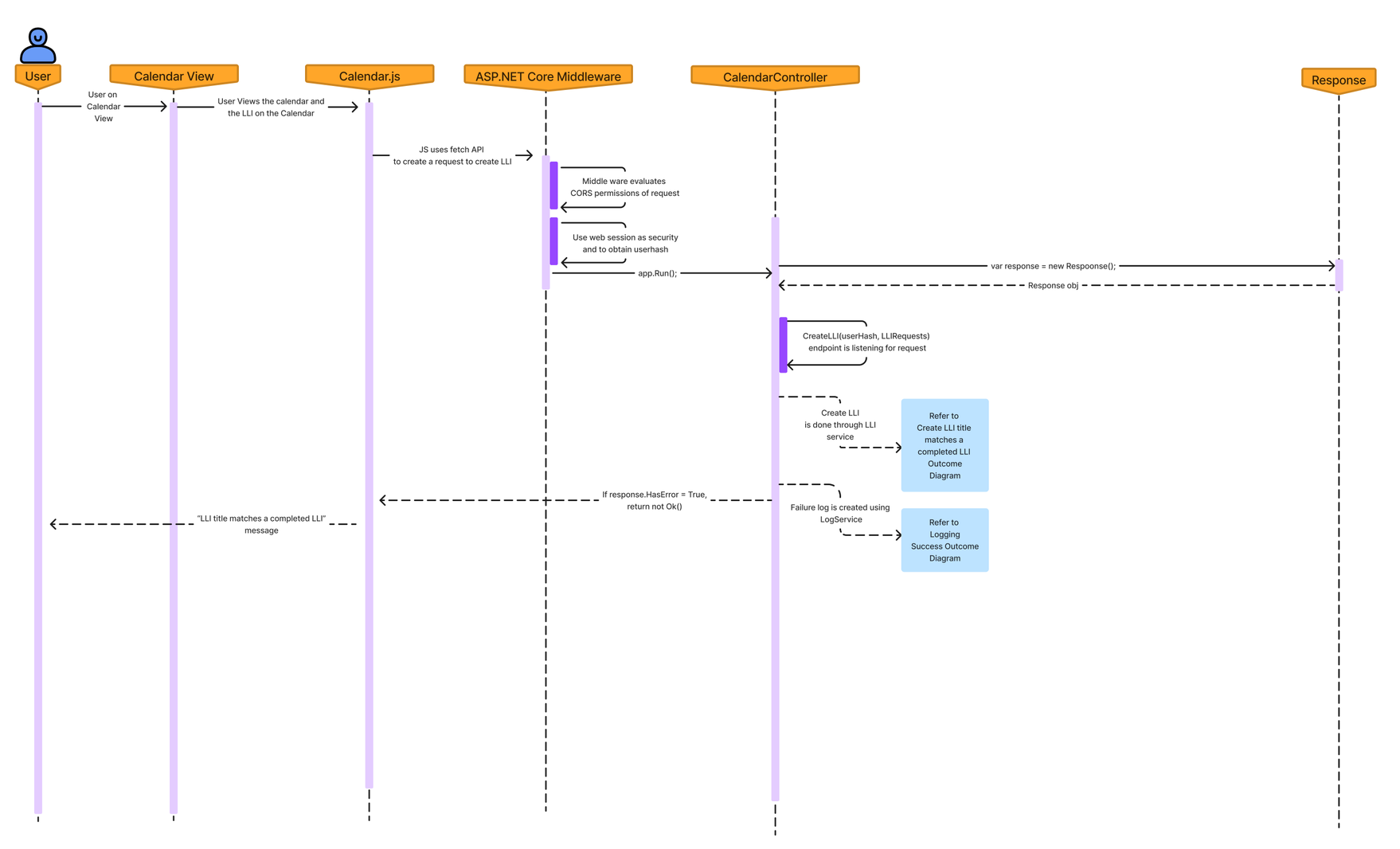
LC3 Failure Outcome: LLI deadline already passed


User
Calendar View
Calendar.js
ASP.NET Core Middleware
CalendarController
Response
User on Calendar View
User Views the calendar and the LLI on the Calendar
JS uses fetch API to create a request to create LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
var response = new Respoonse();
Response obj
CreateLLI(userHash, LLIRequests) endpoint is listening for request
Create LLI is done through LLI service
Refer to Create LLI Deadline passed Outcome Diagram
If response.HasError = True, return not Ok()
Failure log is created using LogService
Refer to Logging Success Outcome Diagram
"LLI Deadline has passed" message

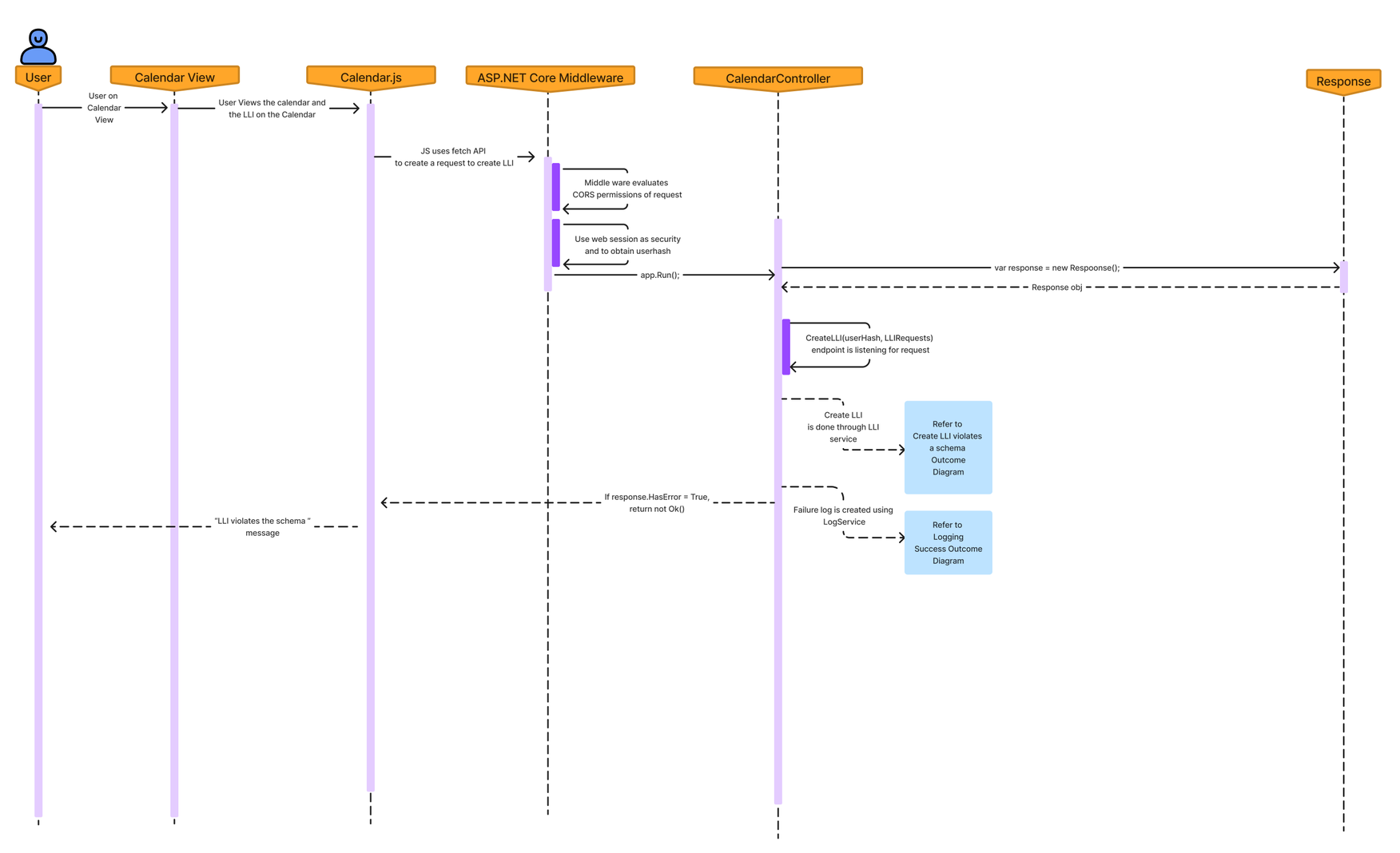
LC3 Failure Outcome: LLI cost is negative


User
Calendar View
Calendar.js
ASP.NET Core Middleware
CalendarController
Response
User on Calendar View
User Views the calendar and the LLI on the Calendar
JS uses fetch API to create a request to create LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
var response = new Respoonse();
Response obj
CreateLLI(userHash, LLIRequests) endpoint is listening for request
Create LLI is done through LLI service
Refer to Create LLI cost is negative Outcome Diagram
If response.HasError = True, return not Ok()
Failure log is created using LogService
Refer to Logging Success Outcome Diagram
"LLI cost is negative" message

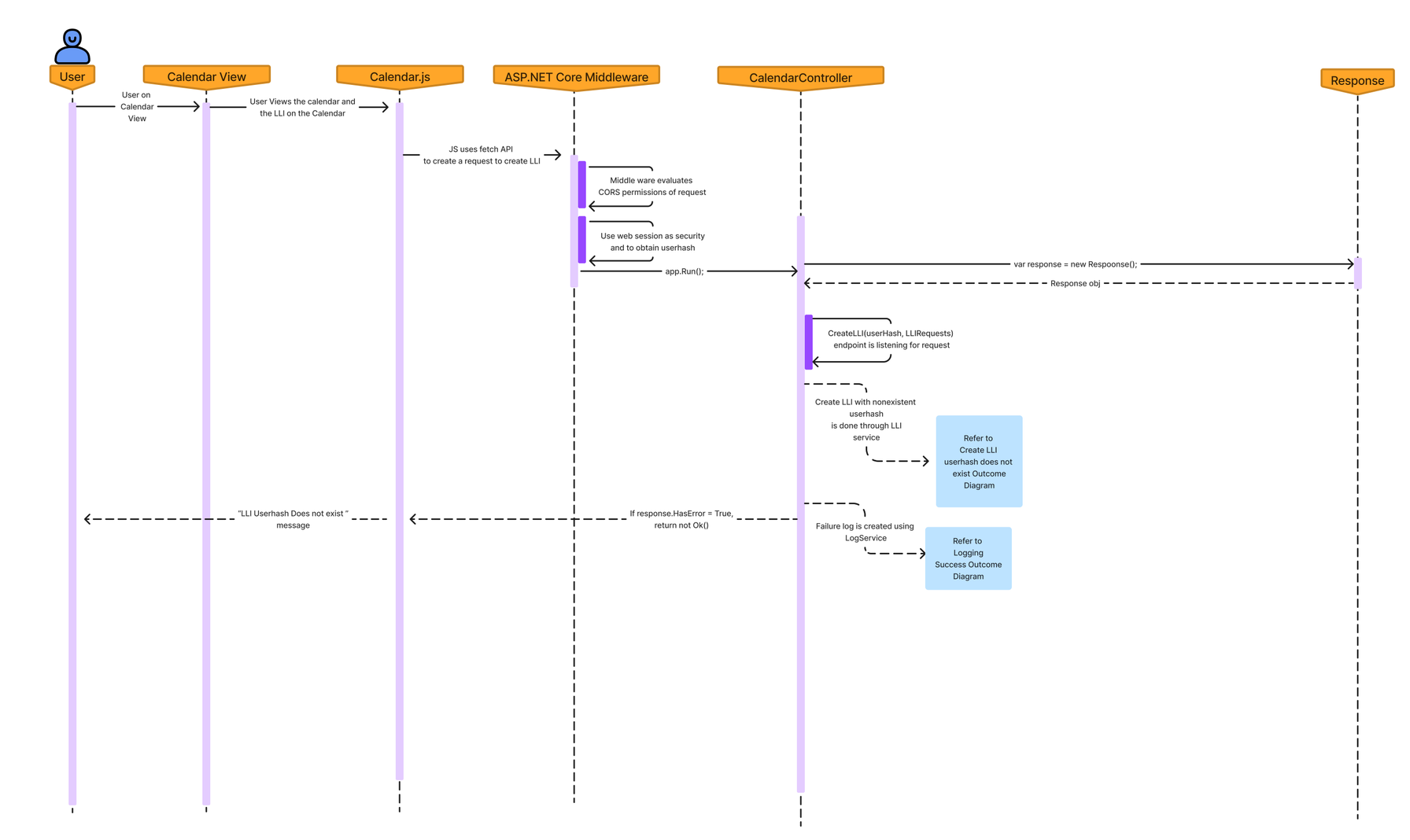
LC3 Failure Outcome: LLI title matches a completed LLI


User
Calendar View
Calendar.js
ASP.NET Core Middleware
CalendarController
Response
User on Calendar View
User Views the calendar and the LLI on the Calendar
JS uses fetch API to create a request to create LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
var response = new Respoonse();
Response obj
CreateLLI(userHash, LLIRequests) endpoint is listening for request
Create LLI is done through LLI service
Refer to Create LLI title matches a completed LLI Outcome Diagram
If response.HasError = True, return not Ok()
Failure log is created using LogService
Refer to Logging Success Outcome Diagram
"LLI title matches a completed LLI" message

LC3 Failure Outcome: LLI violates schema


User
Calendar View
Calendar.js
ASP.NET Core Middleware
CalendarController
Response
User on Calendar View
User Views the calendar and the LLI on the Calendar
JS uses fetch API to create a request to create LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
var response = new Respoonse();
Response obj
CreateLLI(userHash, LLIRequests) endpoint is listening for request
Create LLI is done through LLI service
Refer to Create LLI violates a schema Outcome Diagram
If response.HasError = True, return not Ok()
Failure log is created using LogService
Refer to Logging Success Outcome Diagram
"LLI violates the schema" message

LC3 Failure Outcome: userhash does not exist


User
Calendar View
Calendar.js
ASP.NET Core Middleware
CalendarController
Response
User on Calendar View
User Views the calendar and the LLI on the Calendar
JS uses fetch API to create a request to create LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
var response = new Respoonse();
Response obj
CreateLLI(userHash, LLIRequests) endpoint is listening for request
Create LLI with nonexistent userhash is done through LLI service
Refer to Create LLI userhash does not exist Outcome Diagram
If response.HasError = True, return not Ok()
Failure log is created using LogService
Refer to Logging Success Outcome Diagram
"LLI Userhash Does not exist" message