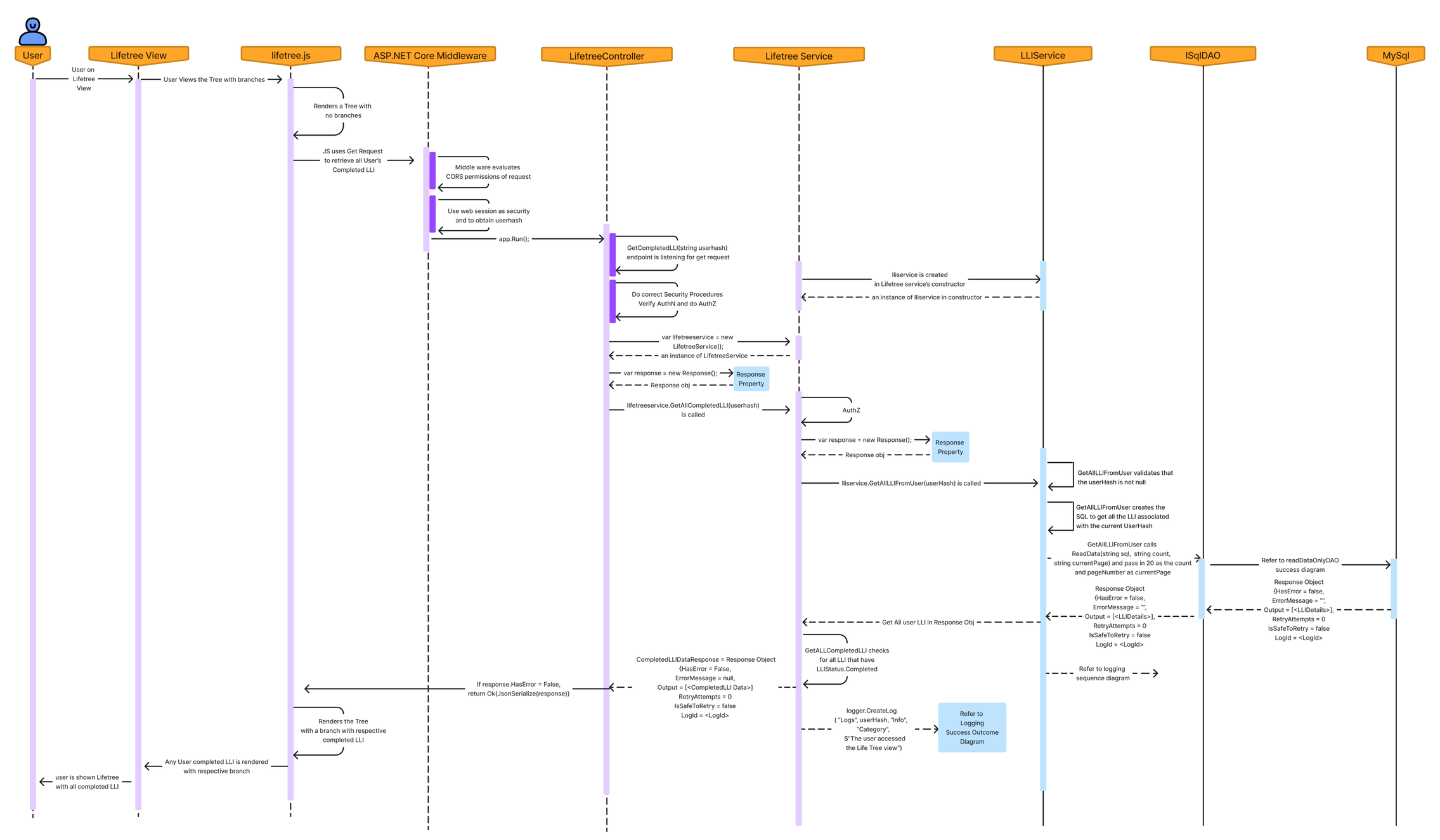
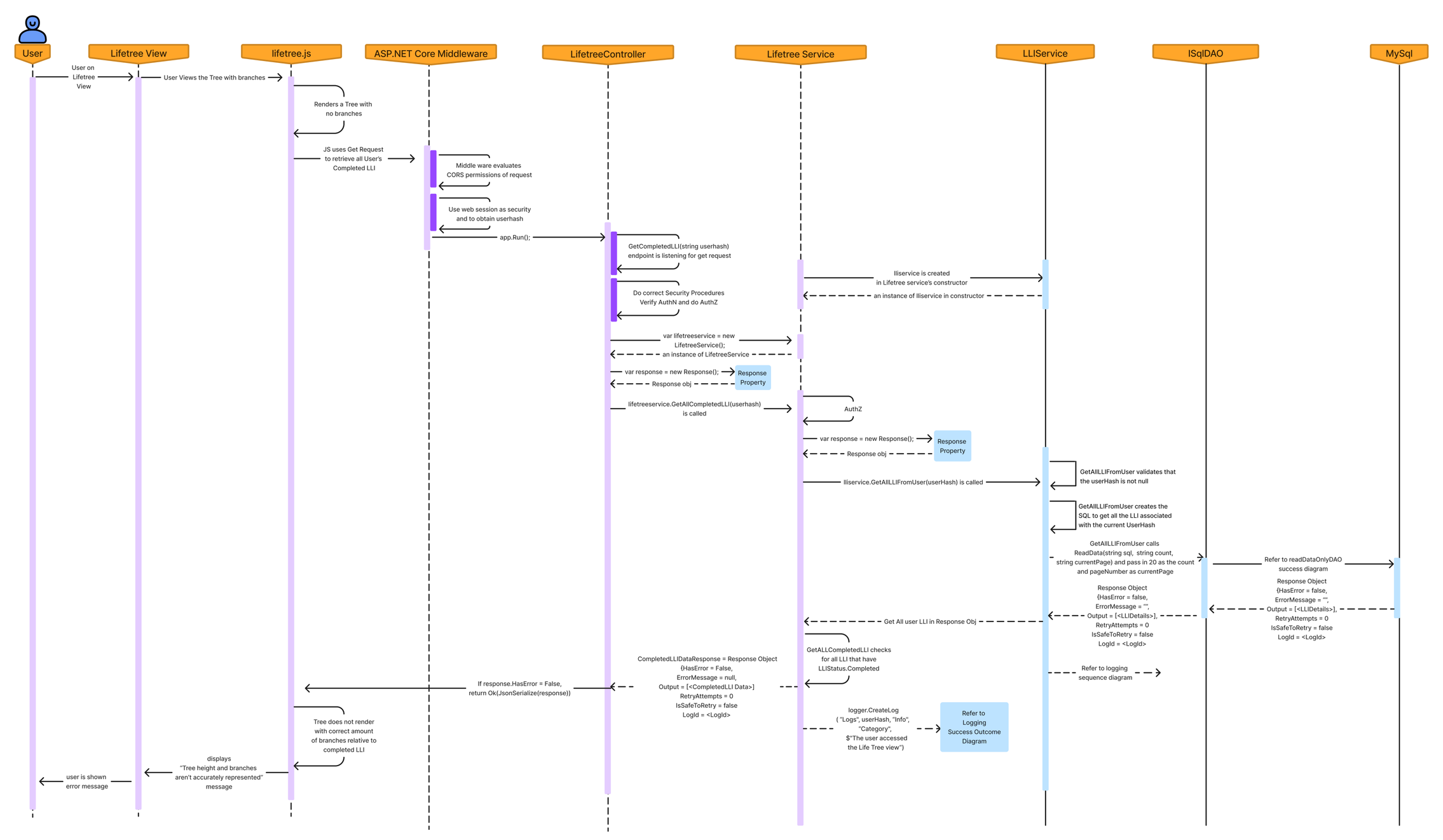
Life Tree

LT1 User Story: View Tree

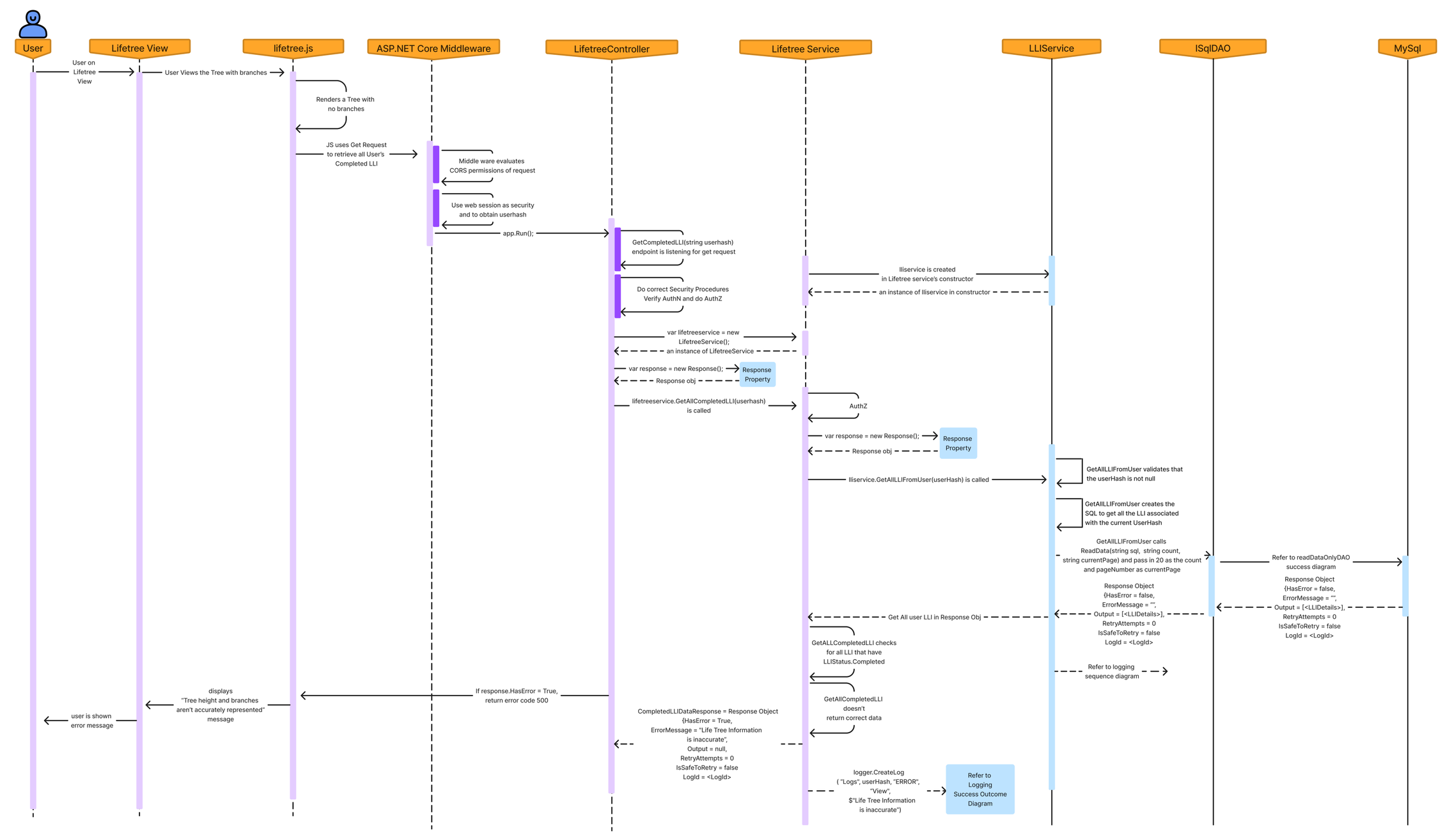
View Tree Success Outcome


User
Lifetree View
lifetree.js
ASP.NET Core Middleware
LifetreeController
Lifetree Service
LLIService
ISqlDAO
MySql
User on Lifetree View
User Views the Tree with branches
Renders a Tree with no branches
JS uses Get Request to retrieve all User's Completed LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
GetCompletedLLI(string userhash) endpoint is listening for get request
Do correct Security Procedures Verify AuthN and do AuthZ
llisservice is created in Lifetree service's constructor
an instance of lliservice in constructor
var lifetreeservice = new LifetreeService();
an instance of LifetreeService
var response = new Response();
Response obj
Response Property
lifetreeservice.GetAllCompletedLLI(userhash) is called
AuthZ
lliservice.GetAllLLIFromUser(userHash) is called
GetAllLLIFromUser validates that the userHash is not null
GetAllLLIFromUser creates the SQL to get all the LLI associated with the current UserHash
GetAllLLIFromUser calls ReadData(string sql, string count, string currentPage) and pass in 20 as the count and pageNumber as currentPage
Refer to readDataOnlyDAO success diagram
Response Object (HasError = false, ErrorMessage = "", Output = [<LLIDetails>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
Get All user LLI in Response Obj
GetALLCompletedLLI checks for all LLI that have LLIStatus.Completed
Refer to logging sequence diagram
CompletedLLIDataResponse = Response Object (HasError = False, ErrorMessage = null, Output = [<CompletedLLI Data>] RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
If response.HasError = False, return Ok(JsonSerialize(response))
logger.CreateLog ("Logs", userHash, "Info", "Category", $"The user accessed the Life Tree view")
Refer to Logging Success Outcome Diagram
Renders the Tree with a branch with respective completed LLI
Any User completed LLI is rendered with respective branch
user is shown Lifetree with all completed LLI

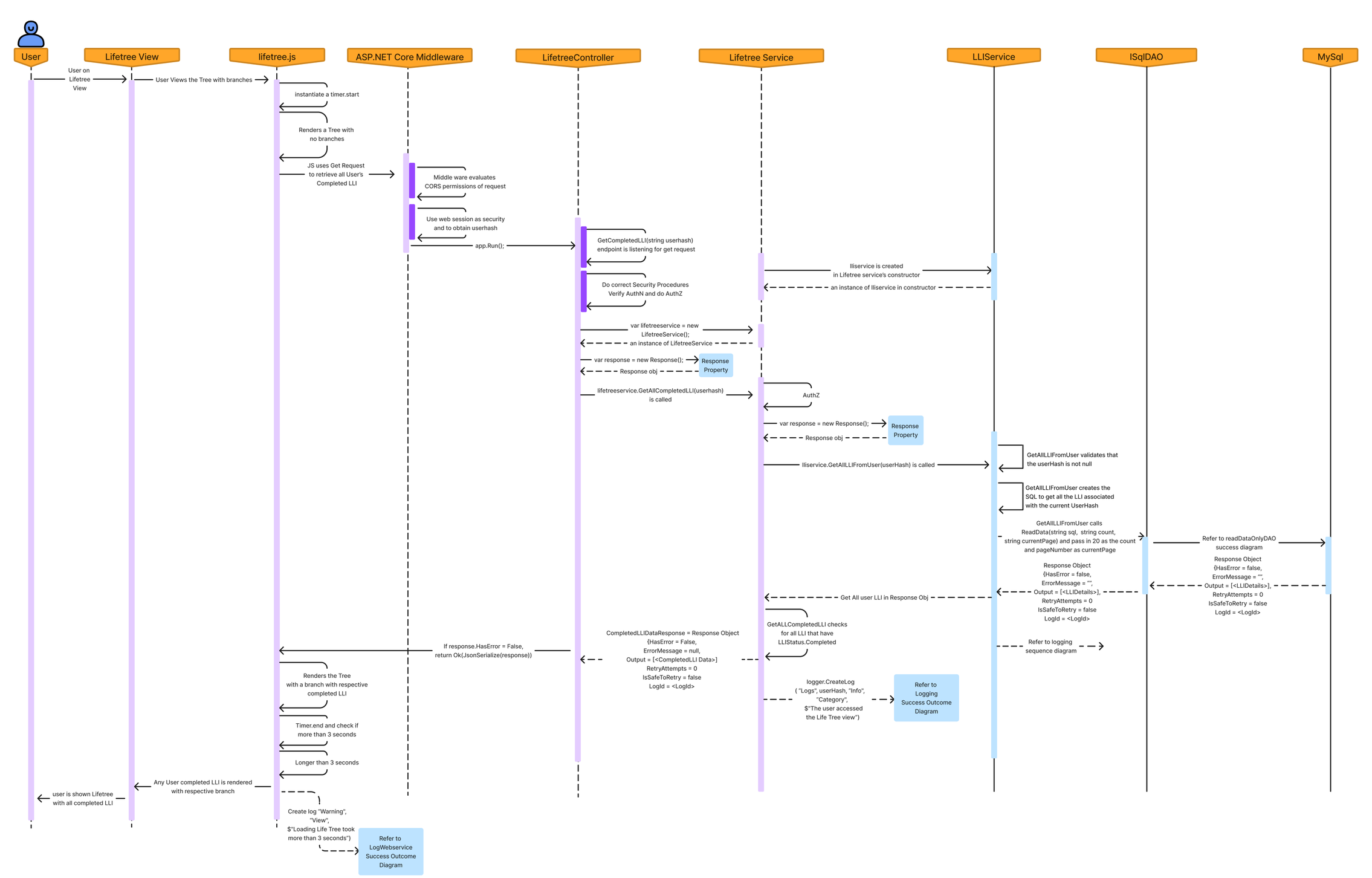
View Tree Failure Outcome: Tree Doesn't Grow


User
Lifetree View
lifetree.js
ASP.NET Core Middleware
LifetreeController
Lifetree Service
LLIService
ISqlDAO
MySql
User on Lifetree View
User Views the Tree with branches
Renders a Tree with no branches
JS uses Get Request to retrieve all User's Completed LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
GetCompletedLLI(string userhash) endpoint is listening for get request
Do correct Security Procedures Verify AuthN and do AuthZ
llisservice is created in Lifetree service's constructor
an instance of lliservice in constructor
var lifetreeservice = new LifetreeService();
an instance of LifetreeService
var response = new Response();
Response obj
Response Property
lifetreeservice.GetAllCompletedLLI(userhash) is called
AuthZ
lliservice.GetAllLLIFromUser(userHash) is called
GetAllLLIFromUser validates that the userHash is not null
GetAllLLIFromUser creates the SQL to get all the LLI associated with the current UserHash
GetAllLLIFromUser calls ReadData(string sql, string count, string currentPage) and pass in 20 as the count and pageNumber as currentPage
Refer to readDataOnlyDAO success diagram
Response Object (HasError = false, ErrorMessage = "", Output = [<LLIDetails>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
Get All user LLI in Response Obj
GetALLCompletedLLI checks for all LLI that have LLIStatus.Completed
Refer to logging sequence diagram
CompletedLLIDataResponse = Response Object (HasError = False, ErrorMessage = null, Output = [<CompletedLLI Data>] RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
If response.HasError = False, return Ok(JsonSerialize(response))
logger.CreateLog ("Logs", userHash, "Info", "Category", $"The user accessed the Life Tree view")
Refer to Logging Success Outcome Diagram
Tree does not render with correct amount of branches relative to completed LLI
displays "Tree height and branches aren't accurately represented" message
user is shown error message

View Tree Failure Outcome: Branches are not accurate

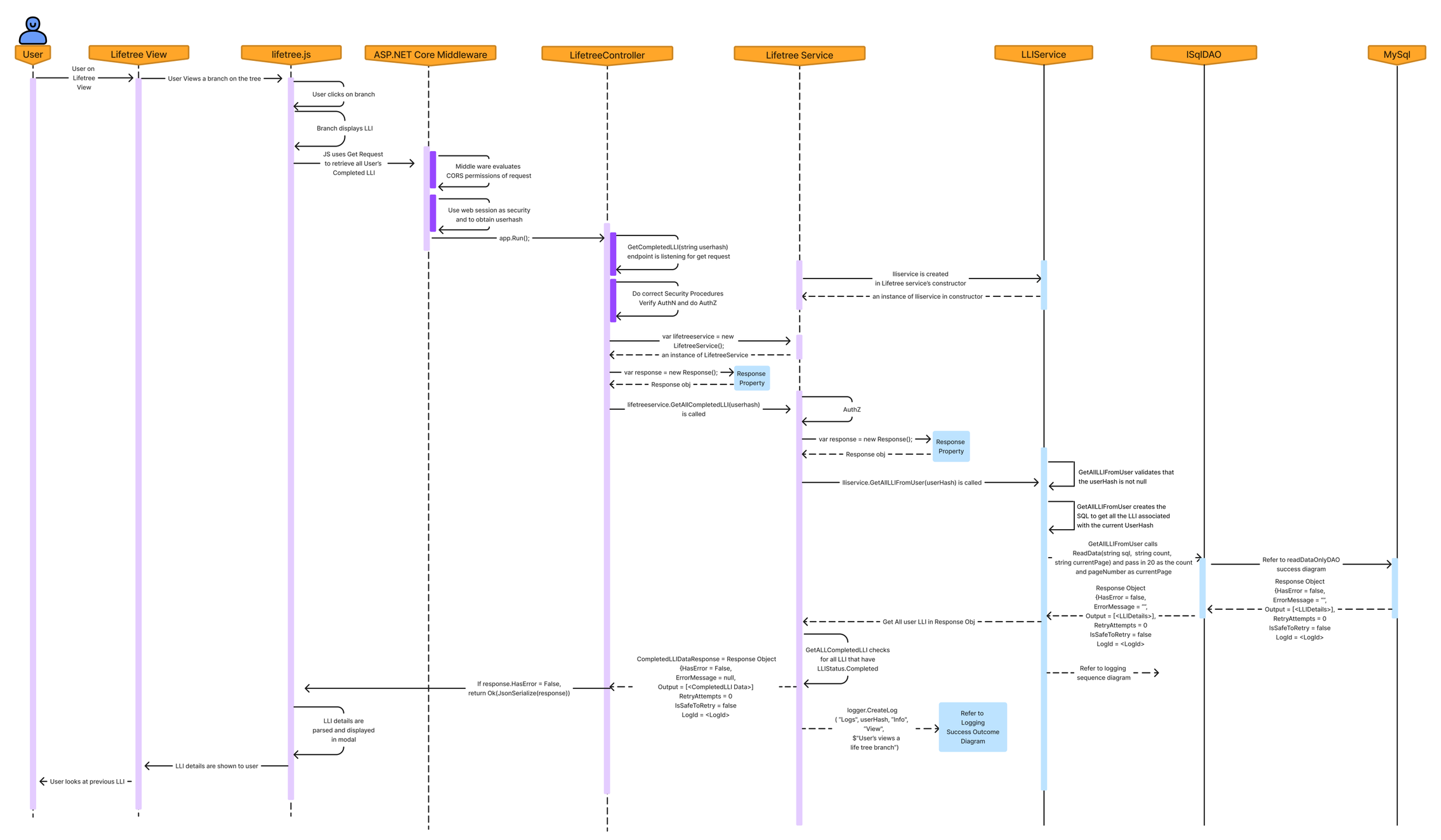

User
Lifetree View
lifetree.js
ASP.NET Core Middleware
LifetreeController
Lifetree Service
LLIService
ISqlDAO
MySql
User on Lifetree View
User Views the Tree with branches
Renders a Tree with no branches
JS uses Get Request to retrieve all User's Completed LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
GetCompletedLLI(string userhash) endpoint is listening for get request
Do correct Security Procedures Verify AuthN and do AuthZ
llisservice is created in Lifetree service's constructor
an instance of lliservice in constructor
var lifetreeservice = new LifetreeService();
an instance of LifetreeService
var response = new Response();
Response obj
Response Property
lifetreeservice.GetAllCompletedLLI(userhash) is called
AuthZ
lliservice.GetAllLLIFromUser(userHash) is called
GetAllLLIFromUser validates that the userHash is not null
GetAllLLIFromUser creates the SQL to get all the LLI associated with the current UserHash
GetAllLLIFromUser calls ReadData(string sql, string count, string currentPage) and pass in 20 as the count and pageNumber as currentPage
Refer to readDataOnlyDAO success diagram
Response Object (HasError = false, ErrorMessage = "", Output = [<LLIDetails>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
Get All user LLI in Response Obj
GetALLCompletedLLI checks for all LLI that have LLIStatus.Completed
GetAllCompletedLLI doesn't return correct data
Refer to logging sequence diagram
CompletedLLIDataResponse = Response Object (HasError = True, ErrorMessage = "Life Tree Information is inaccurate", Output = null, RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
If response.HasError = True, return error code 500
logger.CreateLog ("Logs", userHash, "ERROR", "View", $"Life Tree Information is inaccurate")
Refer to Logging Success Outcome Diagram
displays "Tree height and branches aren't accurately represented" message
user is shown error message

View Tree Failure Outcome: Shown Tree not in 3 seconds

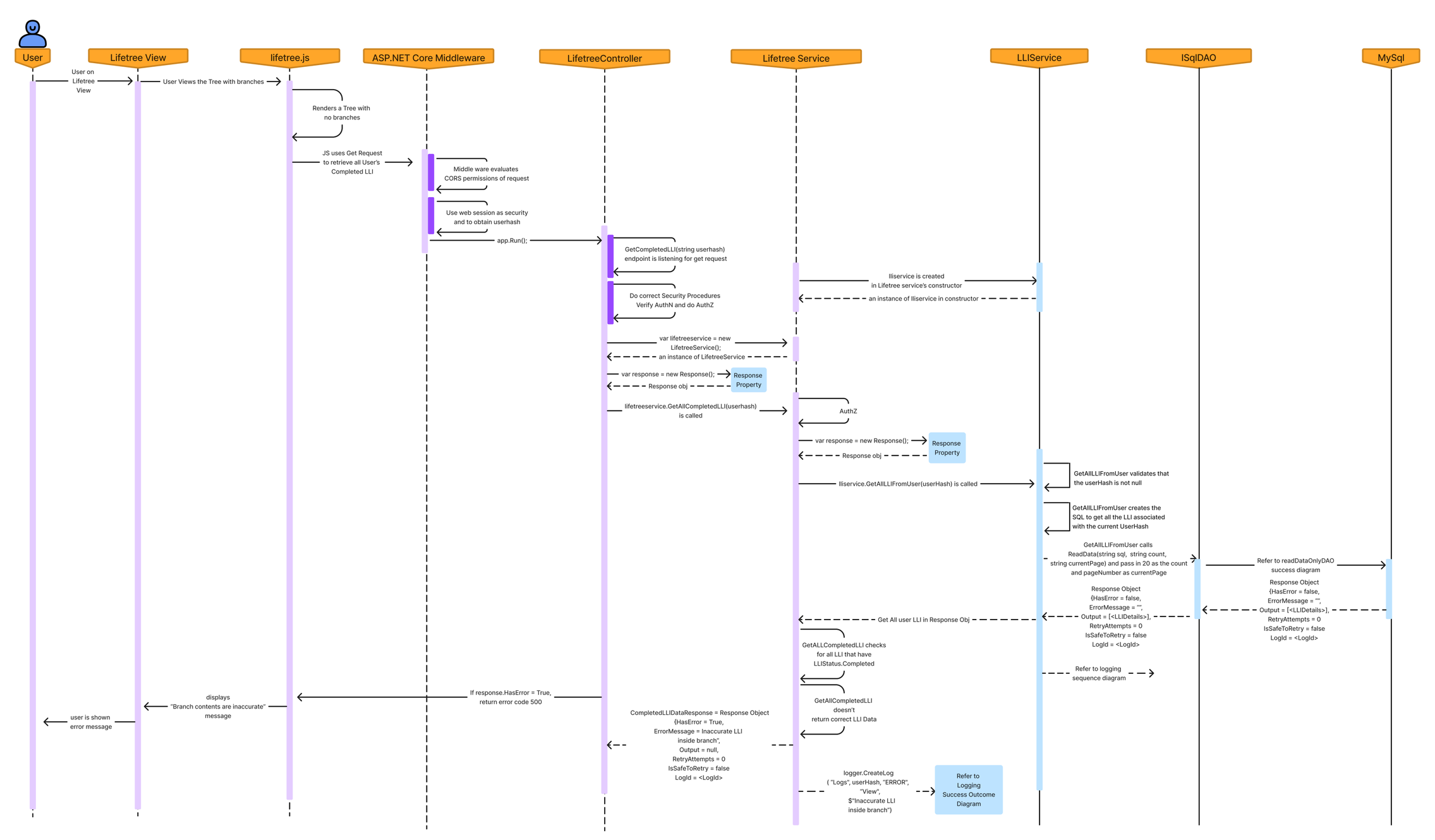

User
Lifetree View
lifetree.js
ASP.NET Core Middleware
LifetreeController
Lifetree Service
LLIService
ISqlDAO
MySql
User on Lifetree View
User Views the Tree with branches
Instantiate a timer.start
Renders a Tree with no branches
JS uses Get Request to retrieve all User's Completed LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
GetCompletedLLI(string userhash) endpoint is listening for get request
Do correct Security Procedures Verify AuthN and do AuthZ
llisservice is created in Lifetree service's constructor
an instance of lliservice in constructor
var lifetreeservice = new LifetreeService();
an instance of LifetreeService
var response = new Response();
Response obj
Response Property
lifetreeservice.GetAllCompletedLLI(userhash) is called
AuthZ
lliservice.GetAllLLIFromUser(userHash) is called
GetAllLLIFromUser validates that the userHash is not null
GetAllLLIFromUser creates the SQL to get all the LLI associated with the current UserHash
GetAllLLIFromUser calls ReadData(string sql, string count, string currentPage) and pass in 20 as the count and pageNumber as currentPage
Refer to readDataOnlyDAO success diagram
Response Object (HasError = false, ErrorMessage = "", Output = [<LLIDetails>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
Get All user LLI in Response Obj
GetALLCompletedLLI checks for all LLI that have LLIStatus.Completed
Refer to logging sequence diagram
CompletedLLIDataResponse = Response Object (HasError = False, ErrorMessage = null, Output = [<CompletedLLI Data>] RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
If response.HasError = False, return Ok(JsonSerialize(response))
logger.CreateLog ("Logs", userHash, "Info", "Category", $"The user accessed the Life Tree view")
Refer to Logging Success Outcome Diagram
Renders the Tree with a branch with respective completed LLI
Timer.end and check if more than 3 seconds
Longer than 3 seconds
Any User completed LLI is rendered with respective branch
user is shown Lifetree with all completed LLI
Create log "Warning", "View", $"Loading Life Tree took more than 3 seconds")
Refer to LogWebservice Success Outcome Diagram

LT2 User Story: View Branch

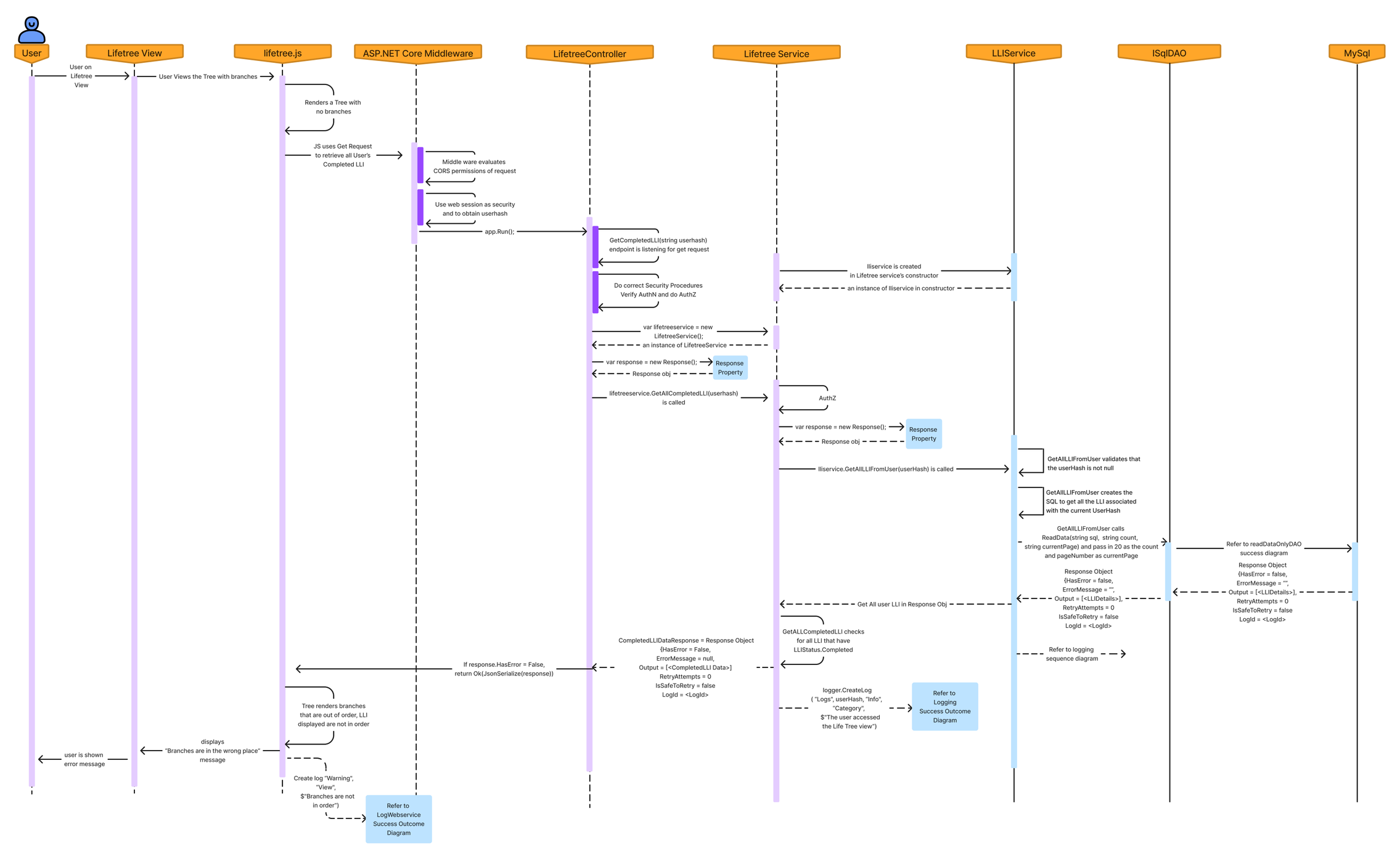
View Branch Success Outcome


User
Lifetree View
lifetree.js
ASP.NET Core Middleware
LifetreeController
Lifetree Service
LLIService
ISqlDAO
MySql
User on Lifetree View
User Views a branch on the tree
User clicks on branch
Branch displays LLI
JS uses Get Request to retrieve all User's Completed LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
GetCompletedLLI(string userhash) endpoint is listening for get request
Do correct Security Procedures Verify AuthN and do AuthZ
lliservice is created in Lifetree service's constructor
an instance of lliservice in constructor
var lifetreeservice = new LifetreeService();
an instance of LifetreeService
var response = new Response();
Response obj
Response Property
lifetreeservice.GetAllCompletedLLI(userhash) is called
AuthZ
lliservice.GetAllLLIFromUser(userHash) is called
GetAllLLIFromUser validates that the userHash is not null
GetAllLLIFromUser creates the SQL to get all the LLI associated with the current UserHash
GetAllLLIFromUser calls ReadData(string sql, string count, string currentPage) and pass in 20 as the count and pageNumber as currentPage
Refer to readDataOnlyDAO success diagram
Response Object (HasError = false, ErrorMessage = "", Output = [<LLIDetails>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
Get All user LLI in Response Obj
GetALLCompletedLLI checks for all LLI that have LLIStatus.Completed
Refer to logging sequence diagram
CompletedLLIDataResponse = Response Object (HasError = False, ErrorMessage = null, Output = [<CompletedLLI Data>] RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
If response.HasError = False, return Ok(JsonSerialize(response))
logger.CreateLog ("Logs", userHash, "Info", "View", $"User's views a life tree branch")
Refer to Logging Success Outcome Diagram
LLI details are parsed and displayed in modal
LLI details are shown to user
User looks at previous LLI

View Branch Failure Outcome: Branch contents are inaccurate


User
Lifetree View
lifetree.js
ASP.NET Core Middleware
LifetreeController
Lifetree Service
LLIService
ISqlDAO
MySql
User on Lifetree View
User Views the Tree with branches
Renders a Tree with no branches
JS uses Get Request to retrieve all User's Completed LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
GetCompletedLLI(string userhash) endpoint is listening for get request
Do correct Security Procedures Verify AuthN and do AuthZ
lliservice is created in Lifetree service's constructor
an instance of lliservice in constructor
var lifetreeservice = new LifetreeService();
an instance of LifetreeService
var response = new Response();
Response obj
Response Property
lifetreeservice.GetAllCompletedLLI(userhash) is called
AuthZ
lliservice.GetAllLLIFromUser(userHash) is called
GetAllLLIFromUser validates that the userHash is not null
GetAllLLIFromUser creates the SQL to get all the LLI associated with the current UserHash
GetAllLLIFromUser calls ReadData(string sql, string count, string currentPage) and pass in 20 as the count and pageNumber as currentPage
Refer to readDataOnlyDAO success diagram
Response Object (HasError = false, ErrorMessage = "", Output = [<LLIDetails>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
Get All user LLI in Response Obj
GetALLCompletedLLI checks for all LLI that have LLIStatus.Completed
GetAllCompletedLLI doesn't return correct LLI Data
Refer to logging sequence diagram
CompletedLLIDataResponse = Response Object (HasError = True, ErrorMessage = Inaccurate LLI inside branch", Output = null, RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
If response.HasError = True, return error code 500
logger.CreateLog ("Logs", userHash, "ERROR", "View", $"Inaccurate LLI inside branch")
Refer to Logging Success Outcome Diagram
displays "Branch contents are inaccurate" message
user is shown error message

View Branch Failure Outcome: Branches out of order

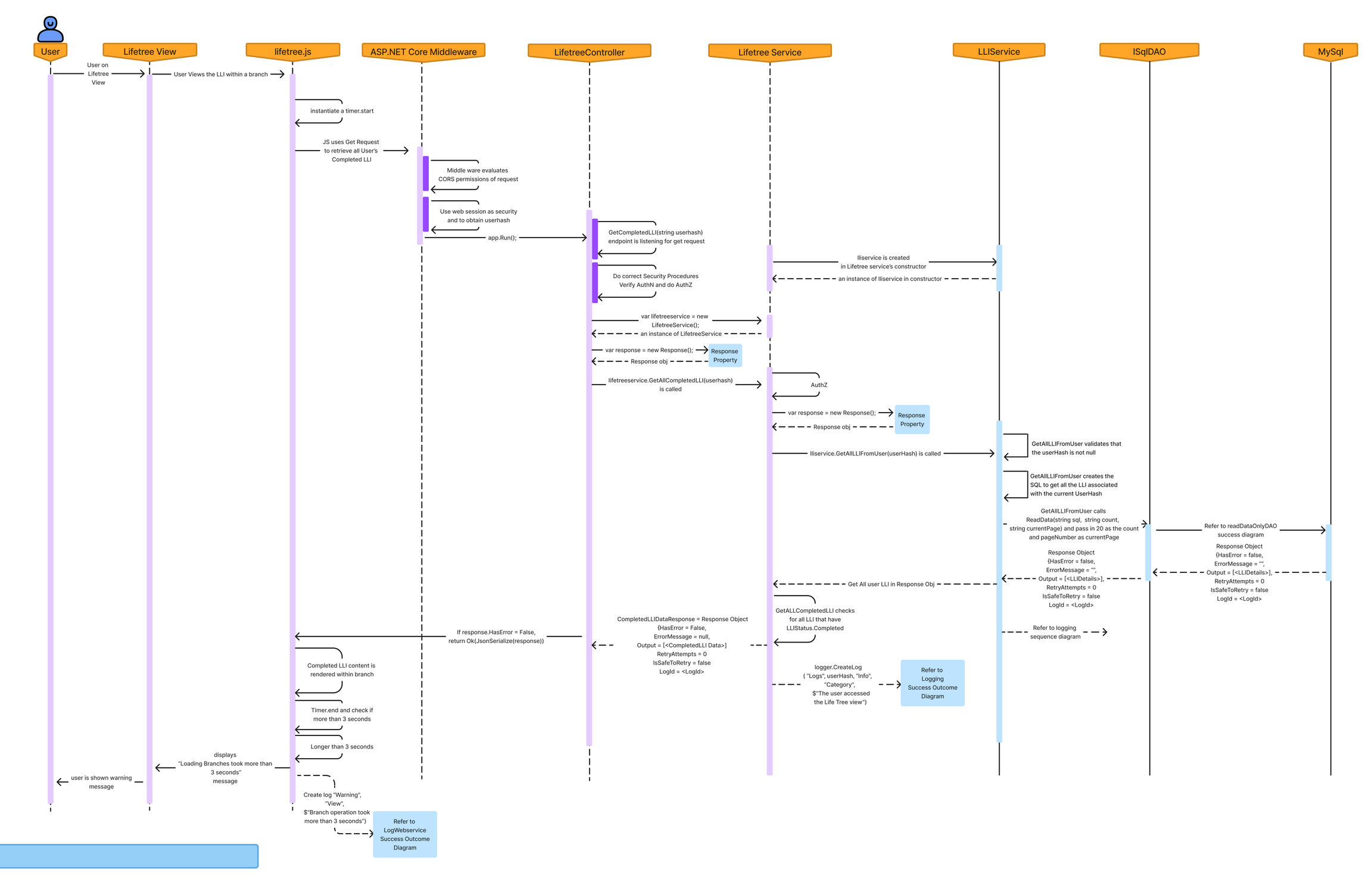

User
Lifetree View
lifetree.js
ASP.NET Core Middleware
LifetreeController
Lifetree Service
LLIService
ISqlDAO
MySql
User on Lifetree View
User Views the Tree with branches
Renders a Tree with no branches
JS uses Get Request to retrieve all User's Completed LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
GetCompletedLLI(string userhash) endpoint is listening for get request
Do correct Security Procedures Verify AuthN and do AuthZ
lliservice is created in Lifetree service's constructor
an instance of lliservice in constructor
var lifetreeservice = new LifetreeService();
an instance of LifetreeService
var response = new Response();
Response obj
Response Property
lifetreeservice.GetAllCompletedLLI(userhash) is called
AuthZ
lliservice.GetAllLLIFromUser(userHash) is called
GetAllLLIFromUser validates that the userHash is not null
GetAllLLIFromUser creates the SQL to get all the LLI associated with the current UserHash
GetAllLLIFromUser calls ReadData(string sql, string count, string currentPage) and pass in 20 as the count and pageNumber as currentPage
Refer to readDataOnlyDAO success diagram
Response Object (HasError = false, ErrorMessage = "", Output = [<LLIDetails>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
Get All user LLI in Response Obj
GetALLCompletedLLI checks for all LLI that have LLIStatus.Completed
Refer to logging sequence diagram
CompletedLLIDataResponse = Response Object (HasError = False, ErrorMessage = null, Output = [<CompletedLLI Data>] RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
If response.HasError = False, return Ok(JsonSerialize(response))
logger.CreateLog ("Logs", userHash, "Info", "Category", $"The user accessed the Life Tree view")
Refer to Logging Success Outcome Diagram
Tree renders branches that are out of order, LLI displayed are not in order
displays "Branches are in the wrong place" message
user is shown error message
Create log "Warning", "View", $"Branches are not in order")
Refer to LogWebservice Success Outcome Diagram

View Branch Failure Outcome: LLI not Successfully Fetched in Database

Fetching LLI in Life Tree uses LLI Service's GetAllLLIFromUser and not interacting with the database itself. Because of this, the testing for this failure outcome is being done within the LLI Service's GetAllLLIFromUser already, thus this cannot be tested for within LifeTree Service

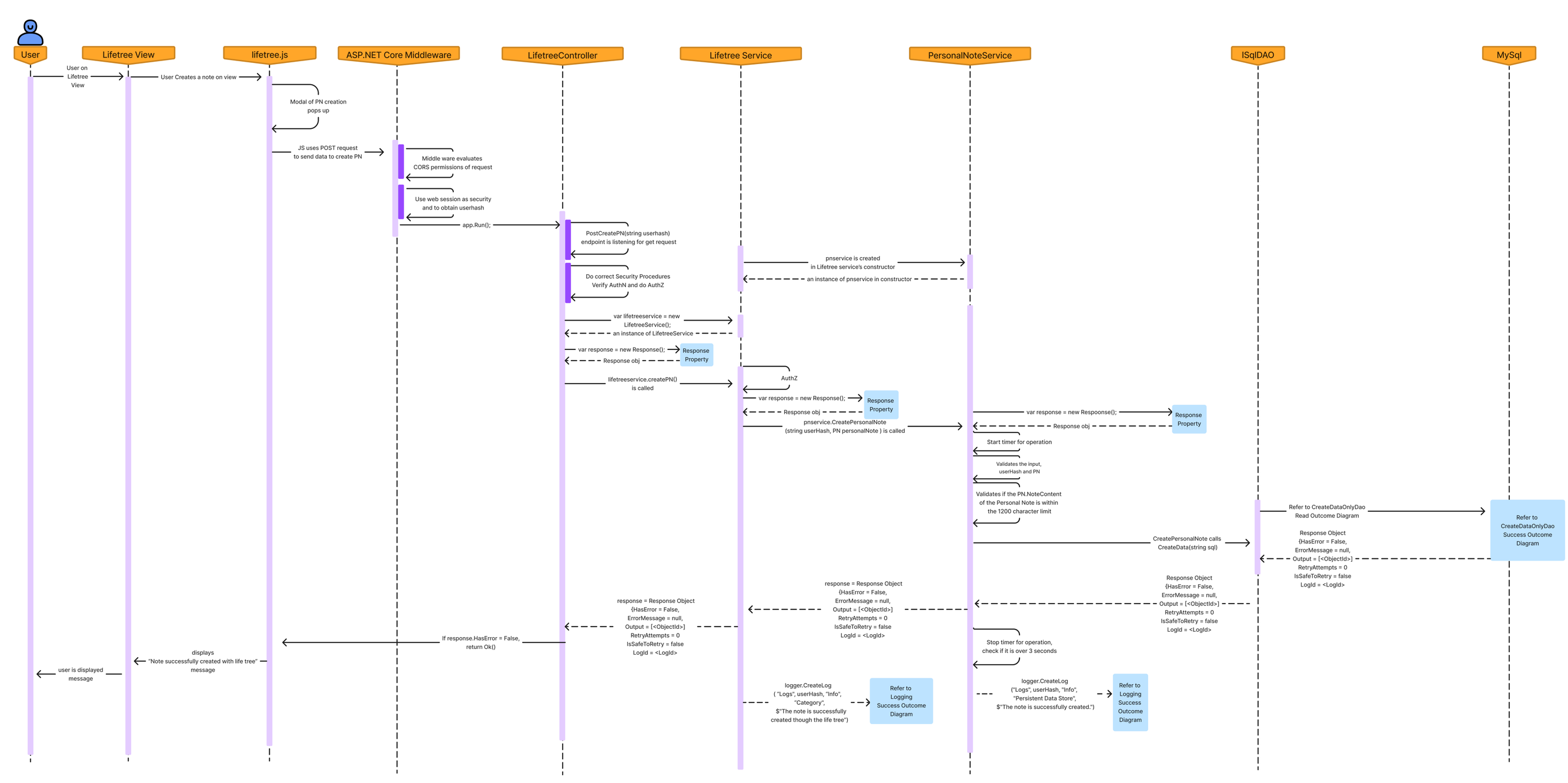
Refer to View LLI success Outcome. Starting at the beginning of LLI Service up until the end of LLI service

View Branch Failure Outcome: Branch Content more than 3 seconds

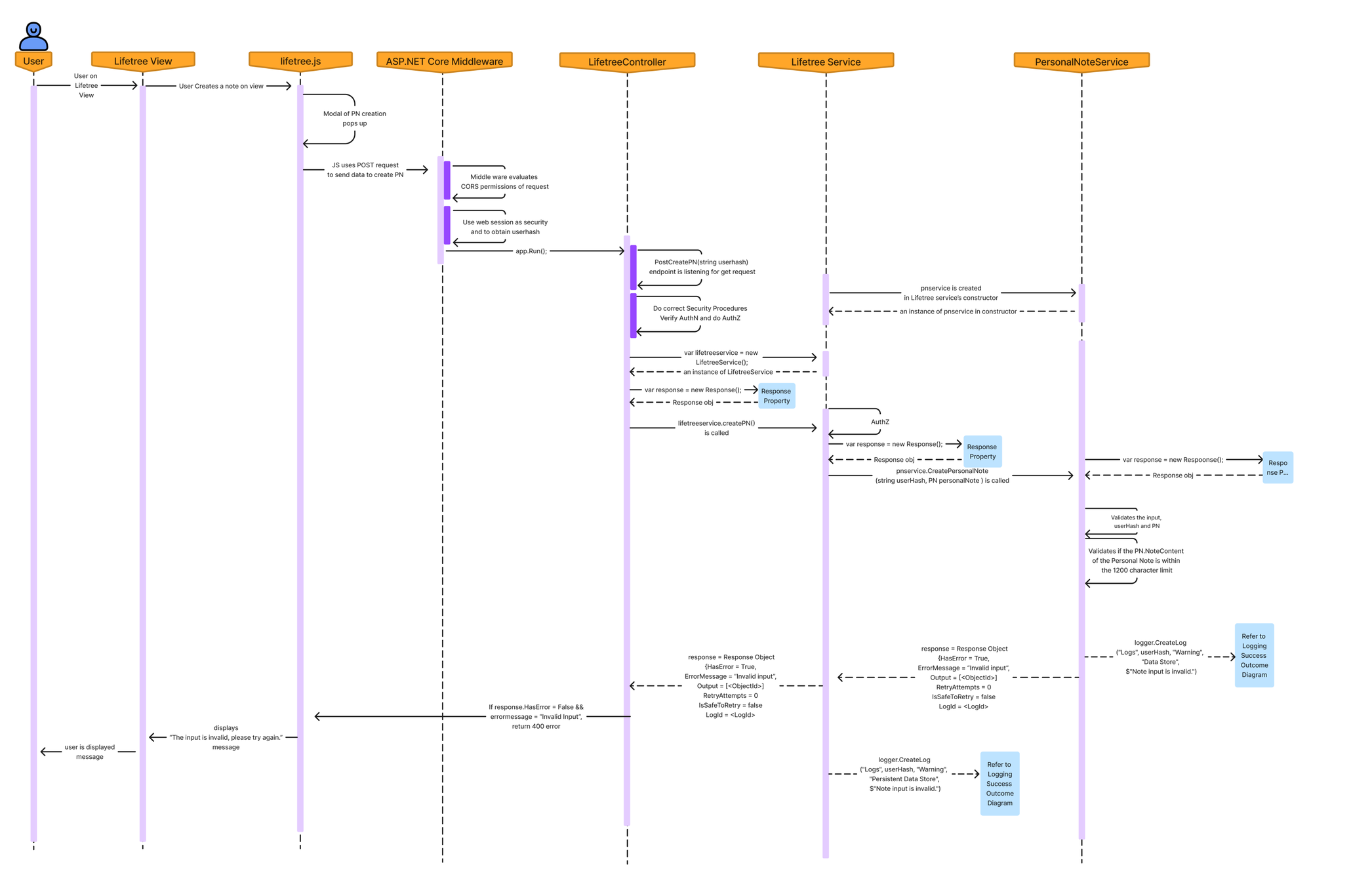

User
Lifetree View
lifetree.js
ASP.NET Core Middleware
LifetreeController
Lifetree Service
LLIService
ISqlDAO
MySql
User on Lifetree View
User Views the LLI within a branch
Instantiate a timer.start
JS uses Get Request to retrieve all User's Completed LLI
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
GetCompletedLLI(string userhash) endpoint is listening for get request
Do correct Security Procedures Verify AuthN and do AuthZ
lliservice is created in Lifetree service's constructor
an instance of lliservice in constructor
var lifetreeservice = new LifetreeService();
an instance of LifetreeService
var response = new Response();
Response obj
Response Property
lifetreeservice.GetAllCompletedLLI(userhash) is called
AuthZ
lliservice.GetAllLLIFromUser(userHash) is called
GetAllLLIFromUser validates that the userHash is not null
GetAllLLIFromUser creates the SQL to get all the LLI associated with the current UserHash
GetAllLLIFromUser calls ReadData(string sql, string count, string currentPage) and pass in 20 as the count and pageNumber as currentPage
Refer to readDataOnlyDAO success diagram
Response Object (HasError = false, ErrorMessage = "", Output = [<LLIDetails>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
Get All user LLI in Response Obj
GetALLCompletedLLI checks for all LLI that have LLIStatus.Completed
Refer to logging sequence diagram
CompletedLLIDataResponse = Response Object (HasError = False, ErrorMessage = null, Output = [<CompletedLLI Data>] RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
If response.HasError = False, return Ok(JsonSerialize(response))
logger.CreateLog ("Logs", userHash, "Info", "Category", $"The user accessed the Life Tree view")
Refer to Logging Success Outcome Diagram
Completed LLI content is rendered within branch
Timer.end and check if more than 3 seconds
Longer than 3 seconds
displays "Loading Branches took more than 3 seconds" message
user is shown warning message
Create log "Warning", "View", $"Branch operation took more than 3 seconds")
Refer to LogWebservice Success Outcome Diagram

LT3 User Story: Create PN

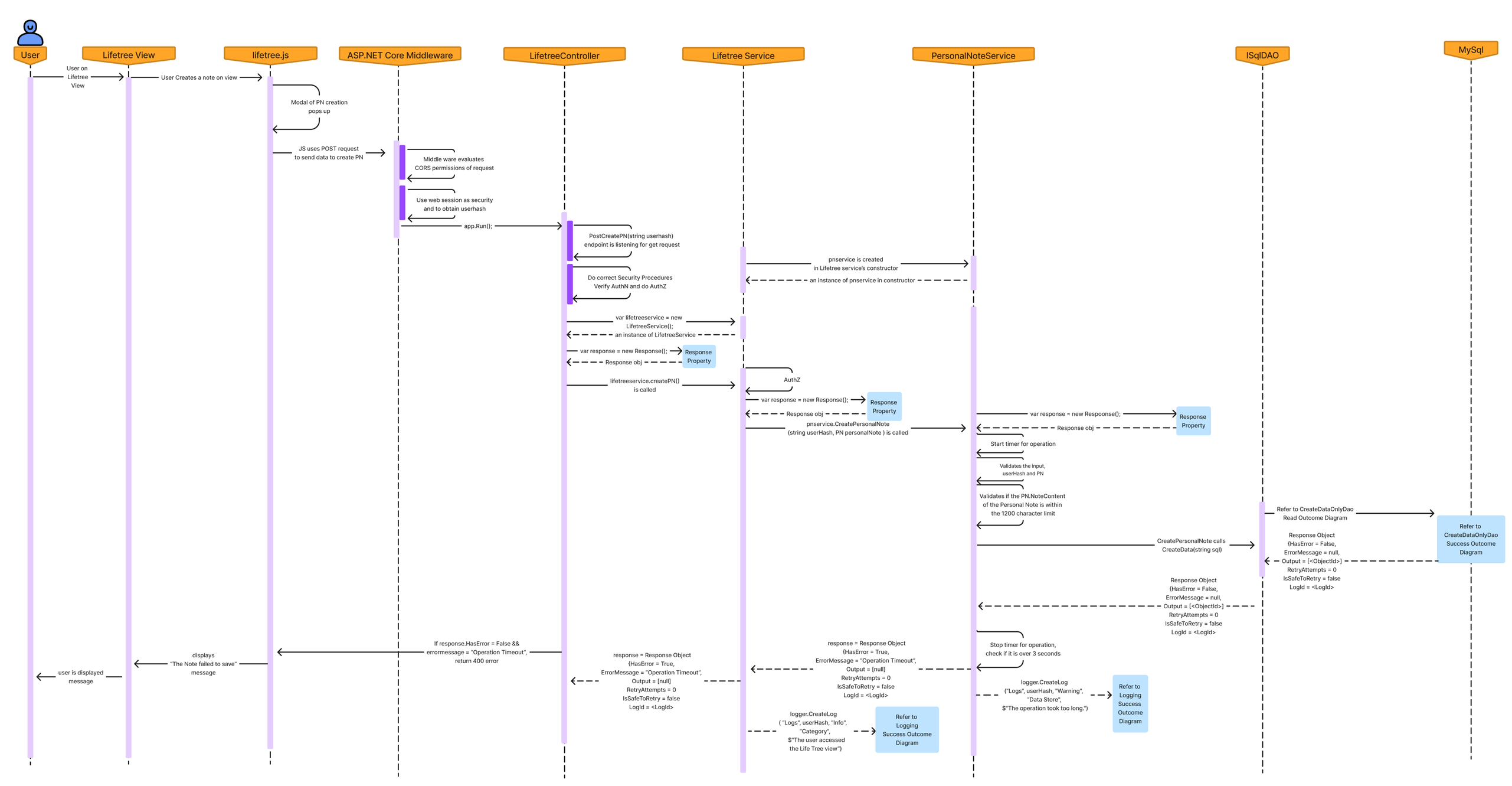
Create PN Success Outcome


User
Lifetree View
lifetree.js
ASP.NET Core Middleware
LifetreeController
Lifetree Service
PersonalNoteService
ISqlDAO
MySql
User on Lifetree View
User Creates a note on view
Modal of PN creation pops up
JS uses POST request to send data to create PN
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
PostCreatePN(string userhash) endpoint is listening for get request
Do correct Security Procedures Verify AuthN and do AuthZ
pnservice is created in Lifetree service's constructor
an instance of pnservice in constructor
var lifetreeservice = new LifetreeService();
an instance of LifetreeService
var response = new Response();
Response obj
Response Property
lifetreeservice.createPN() is called
AuthZ
pnservice.CreatePersonalNote (string userHash, PN personalNote) is called
var response = new Respoonse();
Start timer for operation
Validates the input, userHash and PN
Validates if the PN.NoteContent of the Personal Note is within the 1200 character limit
CreatePersonalNote calls CreateData(string sql)
Refer to CreateDataOnlyDao Read Outcome Diagram
Refer to CreateDataOnlyDao Success Outcome Diagram
Response Object (HasError = False, ErrorMessage = null, Output = [<ObjectId>] RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
response = Response Object (HasError = False, ErrorMessage = null, Output = [<ObjectId>] RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
Stop timer for operation, check if it is over 3 seconds
If response.HasError = False, return Ok()
displays "Note successfully created with life tree" message
user is displayed message
logger.CreateLog ("Logs", userHash, "Info", "Category", $"The note is successfully created though the life tree")
logger.CreateLog ("Logs", userHash, "Info", "Persistent Data Store", $"The note is successfully created.")
Refer to Logging Success Outcome Diagram

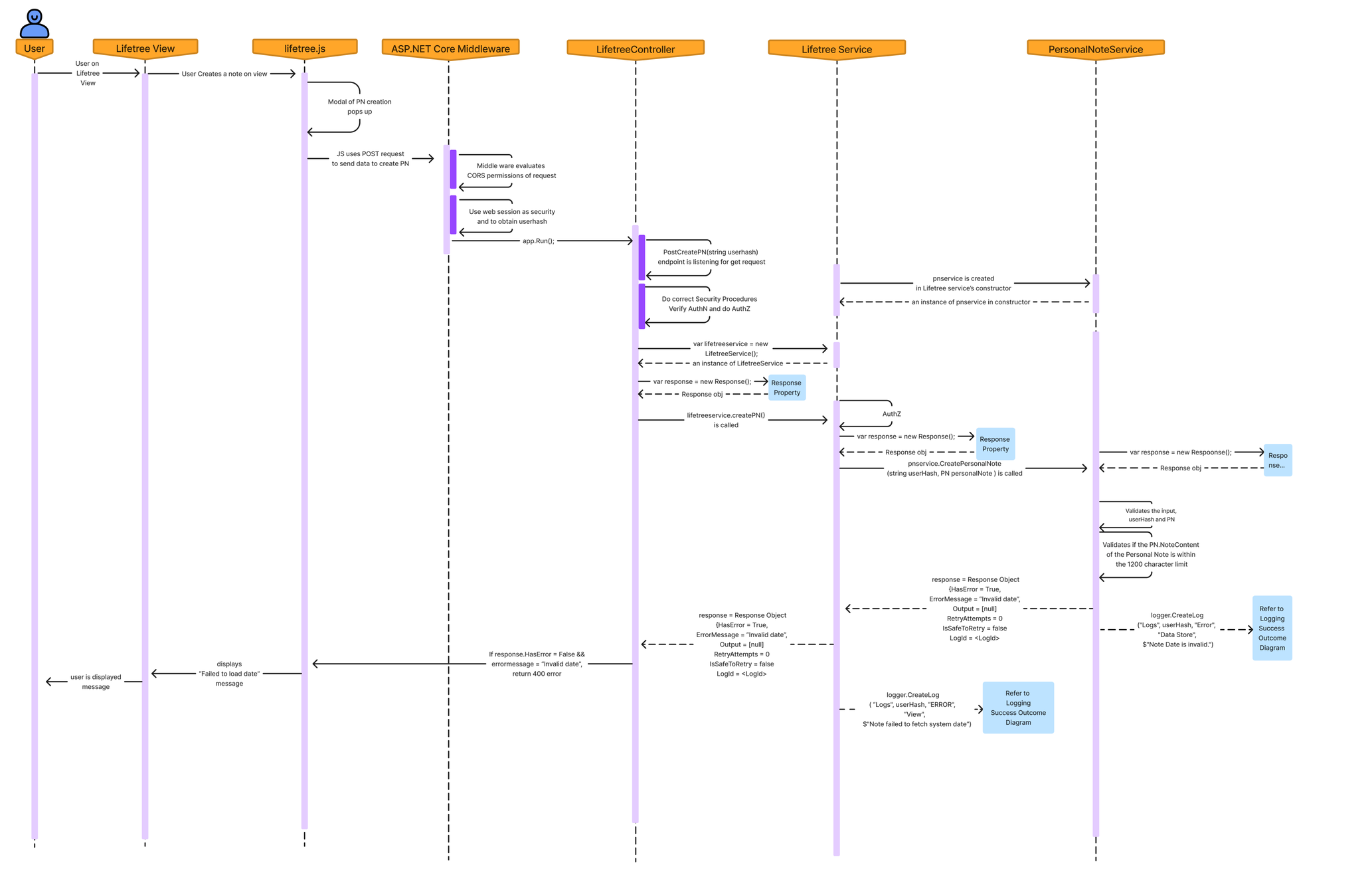
Create PN Failure Outcome: input is invalid


User
Lifetree View
lifetree.js
ASP.NET Core Middleware
LifetreeController
Lifetree Service
PersonalNoteService
User on Lifetree View
User Creates a note on view
Modal of PN creation pops up
JS uses POST request to send data to create PN
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
PostCreatePN(string userhash) endpoint is listening for get request
Do correct Security Procedures Verify AuthN and do AuthZ
pnservice is created in Lifetree service's constructor
an instance of pnservice in constructor
var lifetreeservice = new LifetreeService();
an instance of LifetreeService
var response = new Response();
Response obj
Response Property
lifetreeservice.createPN() is called
AuthZ
pnservice.CreatePersonalNote (string userHash, PN personalNote) is called
var response = new Respoonse();
Validates the input, userHash and PN
Validates if the PN.NoteContent of the Personal Note is within the 1200 character limit
logger.CreateLog ("Logs", userHash, "Warning", "Data Store", $"Note input is invalid.")
Refer to Logging Success Outcome Diagram
response = Response Object (HasError = True, ErrorMessage = "Invalid input", Output = [<ObjectId>] RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
If response.HasError = False && errormessage = "Invalid Input", return 400 error
displays "The input is invalid, please try again." message
user is displayed message
logger.CreateLog ("Logs", userHash, "Warning", "Persistent Data Store", $"Note input is invalid.")

Create PN Failure Outcome: Note not saved within 3 seconds

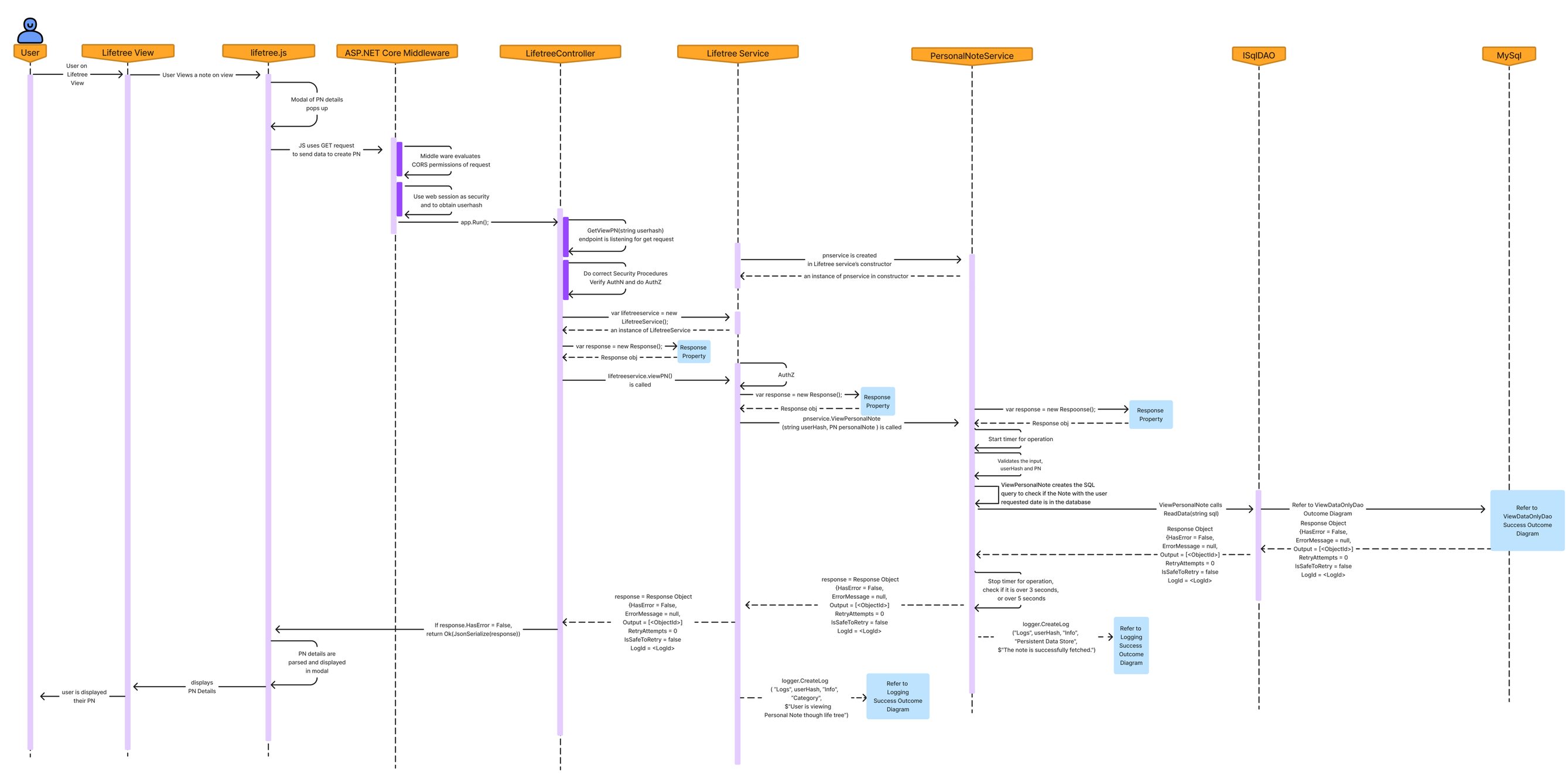

User
Lifetree View
lifetree.js
ASP.NET Core Middleware
LifetreeController
Lifetree Service
PersonalNoteService
ISqlDAO
MySql
User on Lifetree View
User Creates a note on view
Modal of PN creation pops up
JS uses POST request to send data to create PN
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
PostCreatePN(string userhash) endpoint is listening for get request
Do correct Security Procedures Verify AuthN and do AuthZ
pnservice is created in Lifetree service's constructor
an instance of pnservice in constructor
var lifetreeservice = new LifetreeService();
an instance of LifetreeService
var response = new Response();
Response obj
Response Property
lifetreeservice.createPN() is called
AuthZ
pnservice.CreatePersonalNote (string userHash, PN personalNote) is called
var response = new Respoonse();
Start timer for operation
Validates the input, userHash and PN
Validates if the PN.NoteContent of the Personal Note is within the 1200 character limit
CreatePersonalNote calls CreateData(string sql)
Refer to CreateDataOnlyDao Read Outcome Diagram
Refer to CreateDataOnlyDao Success Outcome Diagram
Response Object (HasError = False, ErrorMessage = null, Output = [<ObjectId>] RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
Stop timer for operation, check if it is over 3 seconds
response = Response Object (HasError = True, ErrorMessage = "Operation Timeout", Output = [null] RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
If response.HasError = False && errormessage = "Operation Timeout", return 400 error
displays "The Note failed to save" message
user is displayed message
logger.CreateLog ("Logs", userHash, "Warning", "Data Store", $"The operation took too long.")
logger.CreateLog ("Logs", userHash, "Info", "Category", $"The user accessed the Life Tree view")
Refer to Logging Success Outcome Diagram

Create PN Failure Outcome: Note Date is incorrect

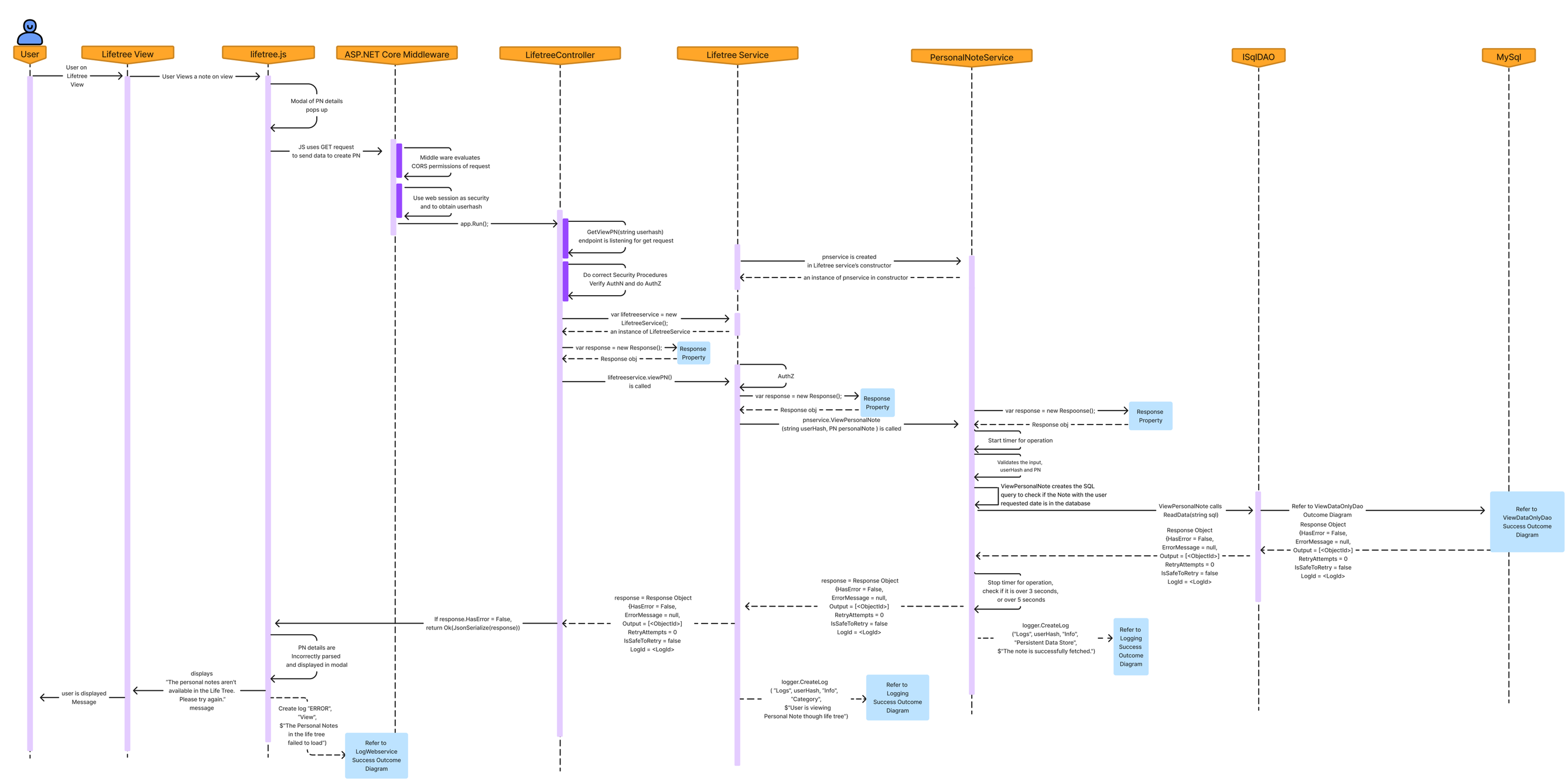

User
Lifetree View
lifetree.js
ASP.NET Core Middleware
LifetreeController
Lifetree Service
PersonalNoteService
User on Lifetree View
User Creates a note on view
Modal of PN creation pops up
JS uses POST request to send data to create PN
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
PostCreatePN(string userhash) endpoint is listening for get request
Do correct Security Procedures Verify AuthN and do AuthZ
pnservice is created in Lifetree service's constructor
an instance of pnservice in constructor
var lifetreeservice = new LifetreeService();
an instance of LifetreeService
var response = new Response();
Response obj
Response Property
lifetreeservice.createPN() is called
AuthZ
pnservice.CreatePersonalNote (string userHash, PN personalNote) is called
var response = new Respoonse();
Validates the input, userHash and PN
Validates if the PN.NoteContent of the Personal Note is within the 1200 character limit
response = Response Object (HasError = True, ErrorMessage = "Invalid date", Output = [null] RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
logger.CreateLog ("Logs", userHash, "Error", "Data Store", $"Note Date is invalid.")
If response.HasError = False && errormessage = "Invalid date", return 400 error
displays "Failed to load date" message
user is displayed message
logger.CreateLog ("Logs", userHash, "ERROR", "View", $"Note failed to fetch system date")
Refer to Logging Success Outcome Diagram

LT4 User Story: View PN though Life Tree

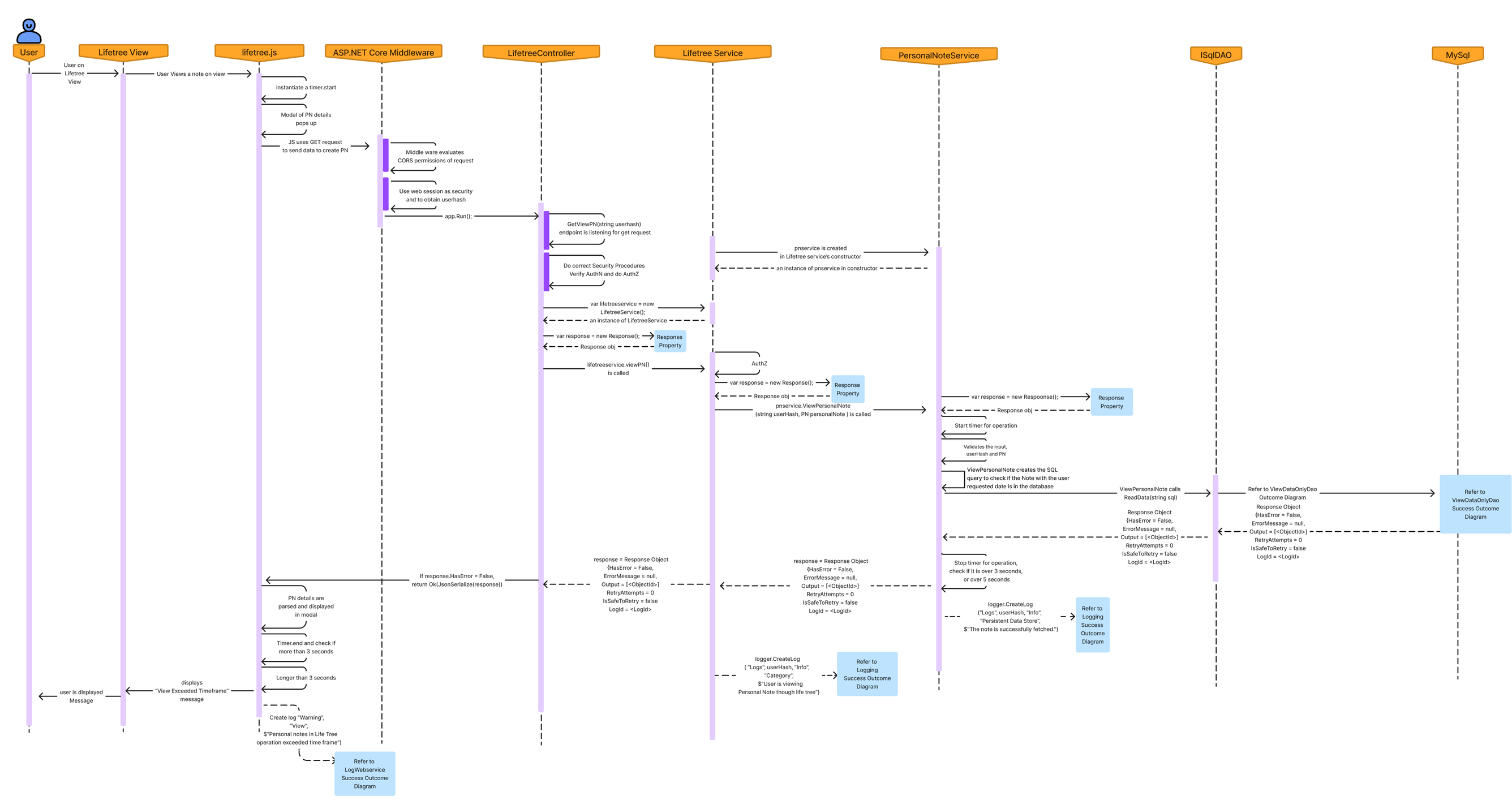
View PN Success Outcome


User
Lifetree View
lifetree.js
ASP.NET Core Middleware
LifetreeController
Lifetree Service
PersonalNoteService
ISqlDAO
MySql
User on Lifetree View
User Views a note on view
Modal of PN details pops up
JS uses GET request to send data to create PN
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
GetViewPN(string userhash) endpoint is listening for get request
Do correct Security Procedures Verify AuthN and do AuthZ
pnservice is created in Lifetree service's constructor
an instance of pnservice in constructor
var lifetreeservice = new LifetreeService();
an instance of LifetreeService
var response = new Response();
Response obj
Response Property
lifetreeservice.viewPN() is called
AuthZ
pnservice.ViewPersonalNote (string userHash, PN personalNote) is called
var response = new Respoonse();
Start timer for operation
Validates the input, userHash and PN
ViewPersonalNote creates the SQL query to check if the Note with the user requested date is in the database
ViewPersonalNote calls ReadData(string sql)
Refer to ViewDataOnlyDao Outcome Diagram
Refer to ViewDataOnlyDao Success Outcome Diagram
Response Object (HasError = False, ErrorMessage = null, Output = [<ObjectId>] RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
response = Response Object (HasError = False, ErrorMessage = null, Output = [<ObjectId>] RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
Stop timer for operation, check if it is over 3 seconds, or over 5 seconds
If response.HasError = False, return Ok(JsonSerialize(response))
PN details are parsed and displayed in modal
displays PN Details
user is displayed their PN
logger.CreateLog ("Logs", userHash, "Info", "Persistent Data Store", $"The note is successfully fetched.")
logger.CreateLog ("Logs", userHash, "Info", "Category", $"User is viewing Personal Note though life tree")
Refer to Logging Success Outcome Diagram

View PN Failure Outcome: Note not loaded in LifeTree View


User
Lifetree View
lifetree.js
ASP.NET Core Middleware
LifetreeController
Lifetree Service
PersonalNoteService
ISqlDAO
MySql
User on Lifetree View
User Views a note on view
Modal of PN details pops up
JS uses GET request to send data to create PN
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
GetViewPN(string userhash) endpoint is listening for get request
Do correct Security Procedures Verify AuthN and do AuthZ
pnservice is created in Lifetree service's constructor
an instance of pnservice in constructor
var lifetreeservice = new LifetreeService();
an instance of LifetreeService
var response = new Response();
Response obj
Response Property
lifetreeservice.viewPN() is called
AuthZ
pnservice.ViewPersonalNote (string userHash, PN personalNote) is called
var response = new Respoonse();
Start timer for operation
Validates the input, userHash and PN
ViewPersonalNote creates the SQL query to check if the Note with the user requested date is in the database
ViewPersonalNote calls ReadData(string sql)
Refer to ViewDataOnlyDao Outcome Diagram
Refer to ViewDataOnlyDao Success Outcome Diagram
Response Object (HasError = False, ErrorMessage = null, Output = [<ObjectId>] RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
response = Response Object (HasError = False, ErrorMessage = null, Output = [<ObjectId>] RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
Stop timer for operation, check if it is over 3 seconds, or over 5 seconds
If response.HasError = False, return Ok(JsonSerialize(response))
PN details are incorrectly parsed and displayed in modal
displays "The personal notes aren't available in the Life Tree. Please try again." message
user is displayed Message
Create log "ERROR", "View", $"The Personal Notes in the life tree failed to load")
Refer to LogWebservice Success Outcome Diagram
logger.CreateLog ("Logs", userHash, "Info", "Persistent Data Store", $"The note is successfully fetched.")
logger.CreateLog ("Logs", userHash, "Info", "Category", $"User is viewing Personal Note though life tree")
Refer to Logging Success Outcome Diagram

View PN Failure Outcome: Not Loaded within 3 seconds


User
Lifetree View
lifetree.js
ASP.NET Core Middleware
LifetreeController
Lifetree Service
PersonalNoteService
ISqlDAO
MySql
User on Lifetree View
User Views a note on view
Instantiate a timer.start
Modal of PN details pops up
JS uses GET request to send data to create PN
Middle ware evaluates CORS permissions of request
Use web session as security and to obtain userhash
app.Run();
GetViewPN(string userhash) endpoint is listening for get request
Do correct Security Procedures Verify AuthN and do AuthZ
pnservice is created in Lifetree service's constructor
an instance of pnservice in constructor
var lifetreeservice = new LifetreeService();
an instance of LifetreeService
var response = new Response();
Response obj
Response Property
lifetreeservice.viewPN() is called
AuthZ
pnservice.ViewPersonalNote (string userHash, PN personalNote) is called
var response = new Respoonse();
Start timer for operation
Validates the input, userHash and PN
ViewPersonalNote creates the SQL query to check if the Note with the user requested date is in the database
ViewPersonalNote calls ReadData(string sql)
Refer to ViewDataOnlyDao Outcome Diagram
Refer to ViewDataOnlyDao Success Outcome Diagram
Response Object (HasError = False, ErrorMessage = null, Output = [<ObjectId>] RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
response = Response Object (HasError = False, ErrorMessage = null, Output = [<ObjectId>] RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>
Stop timer for operation, check if it is over 3 seconds, or over 5 seconds
If response.HasError = False, return Ok(JsonSerialize(response))
PN details are parsed and displayed in modal
Timer.end and check if more than 3 seconds
Longer than 3 seconds
displays "View Exceeded Timeframe" message
user is displayed Message
Create log "Warning", "View", $"Personal notes in Life Tree operation exceeded time frame")
Refer to LogWebservice Success Outcome Diagram
logger.CreateLog ("Logs", userHash, "Info", "Persistent Data Store", $"The note is successfully fetched.")
logger.CreateLog ("Logs", userHash, "Info", "Category", $"User is viewing Personal Note though life tree")
Refer to Logging Success Outcome Diagram

View PN Failure Outcome: PN not Successfully Fetched in Database

Fetching PN in Life Tree uses PNService's ViewPersonalNote and not interacting with the database itself. Because of this, the testing for this failure outcome is being done within the PN Service's ViewPersonalNote already, thus this cannot be tested for within LifeTree Service

Refer to View Note success Outcome. Starting at the beginning of PN Service up until the end of PN service