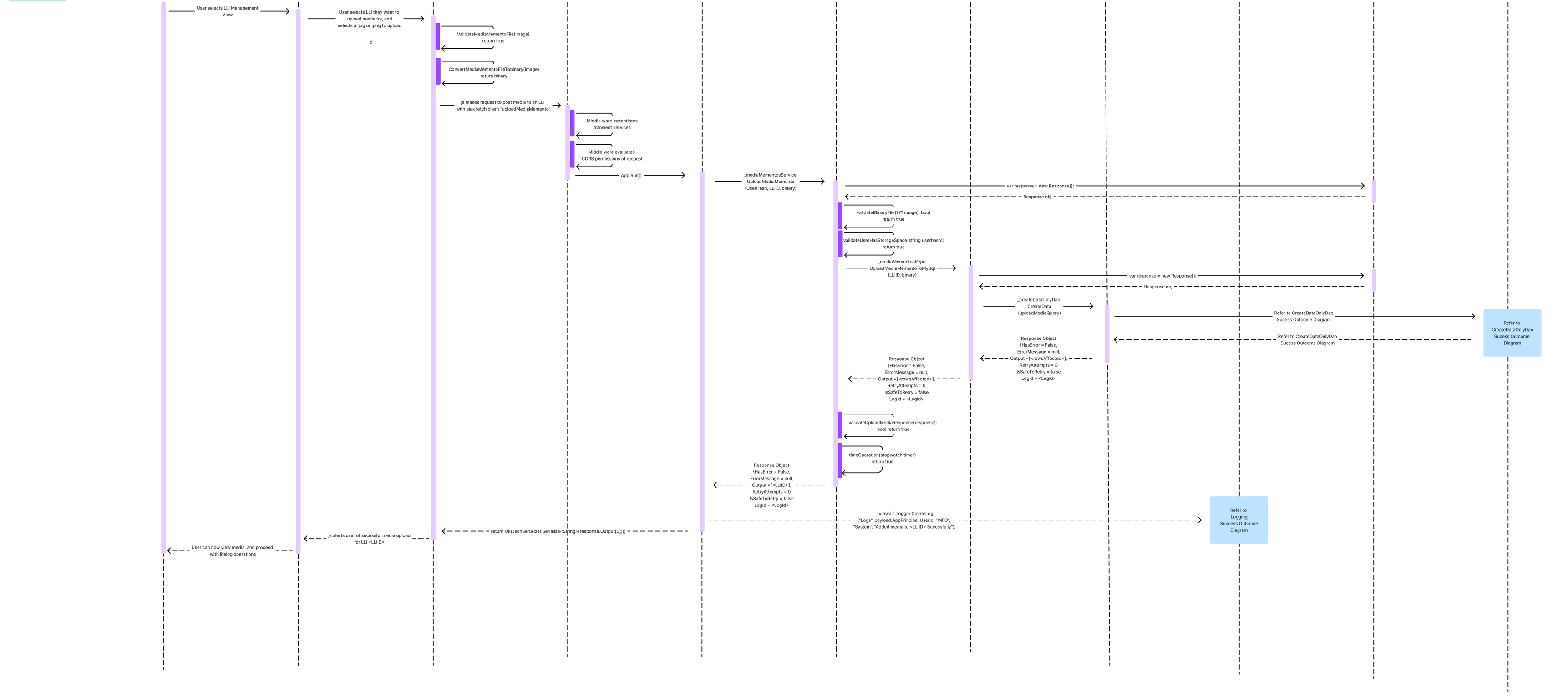
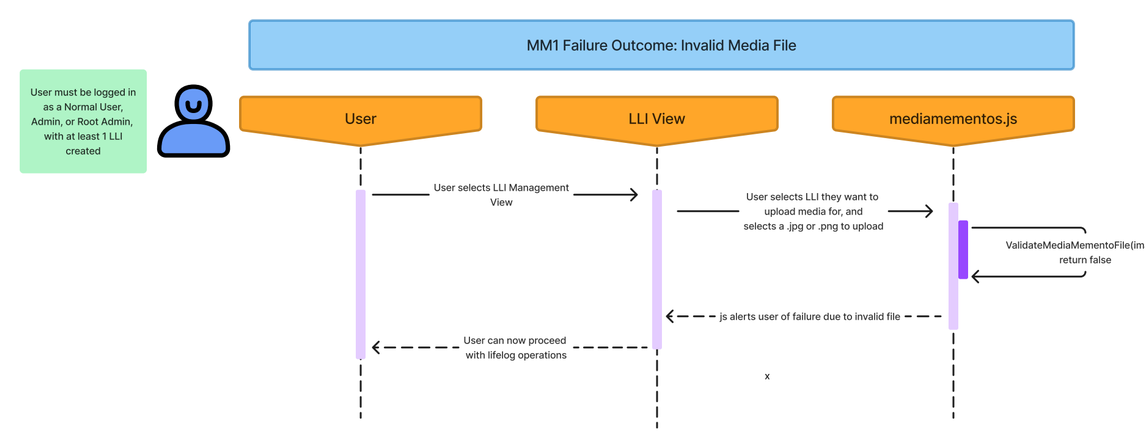
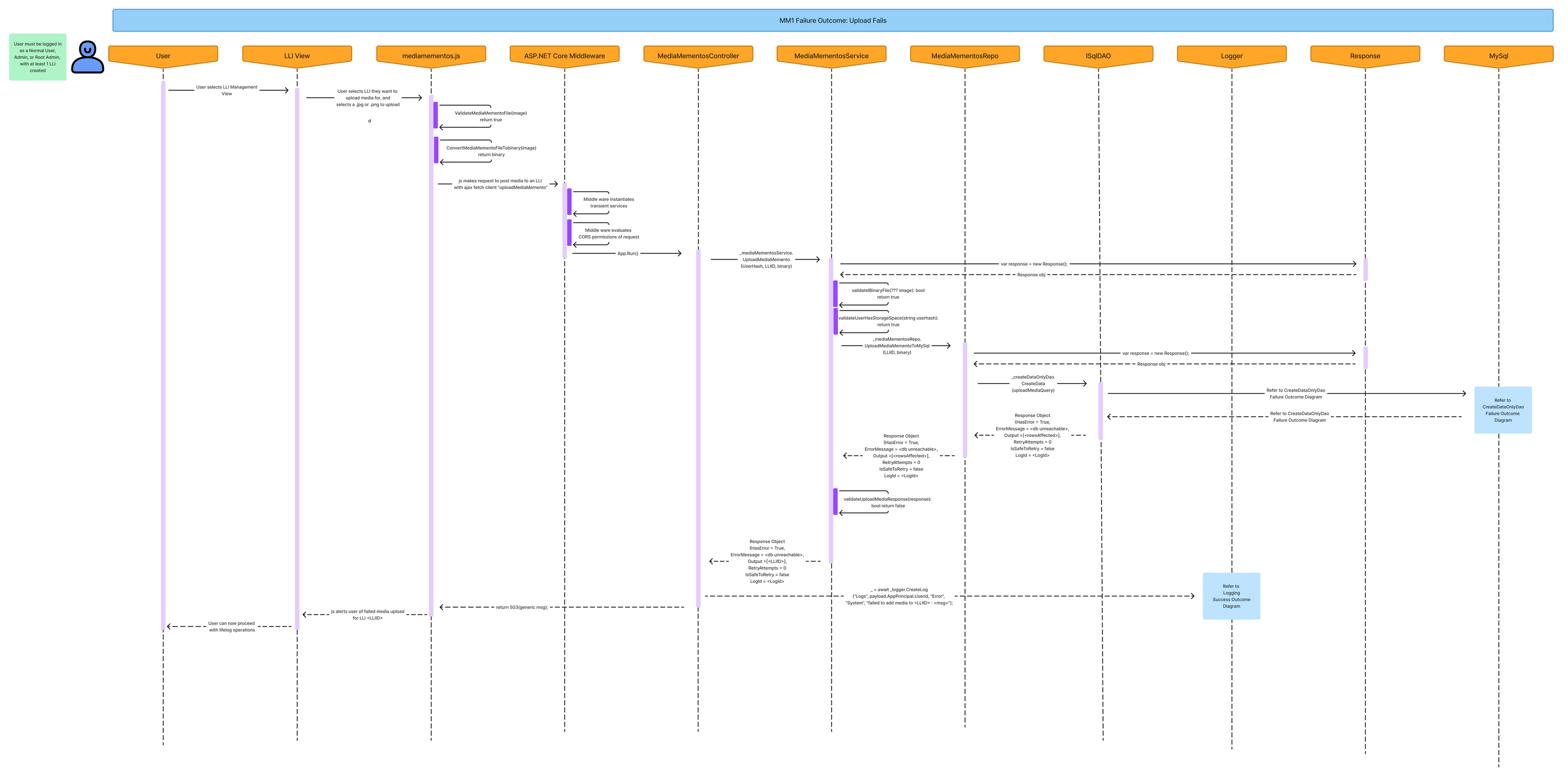
Media Mementos

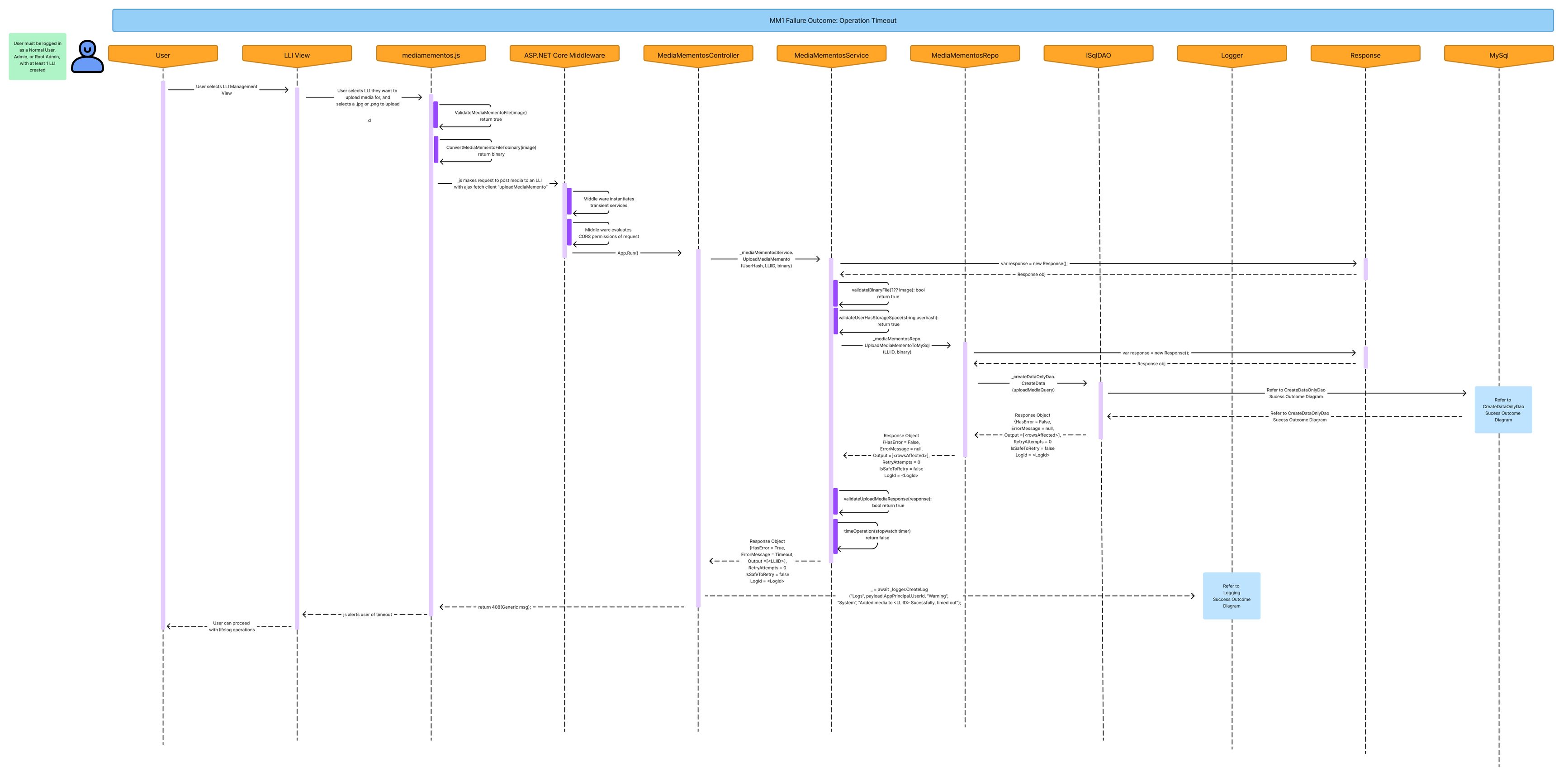
Goals : Performance, Security, Extensibility

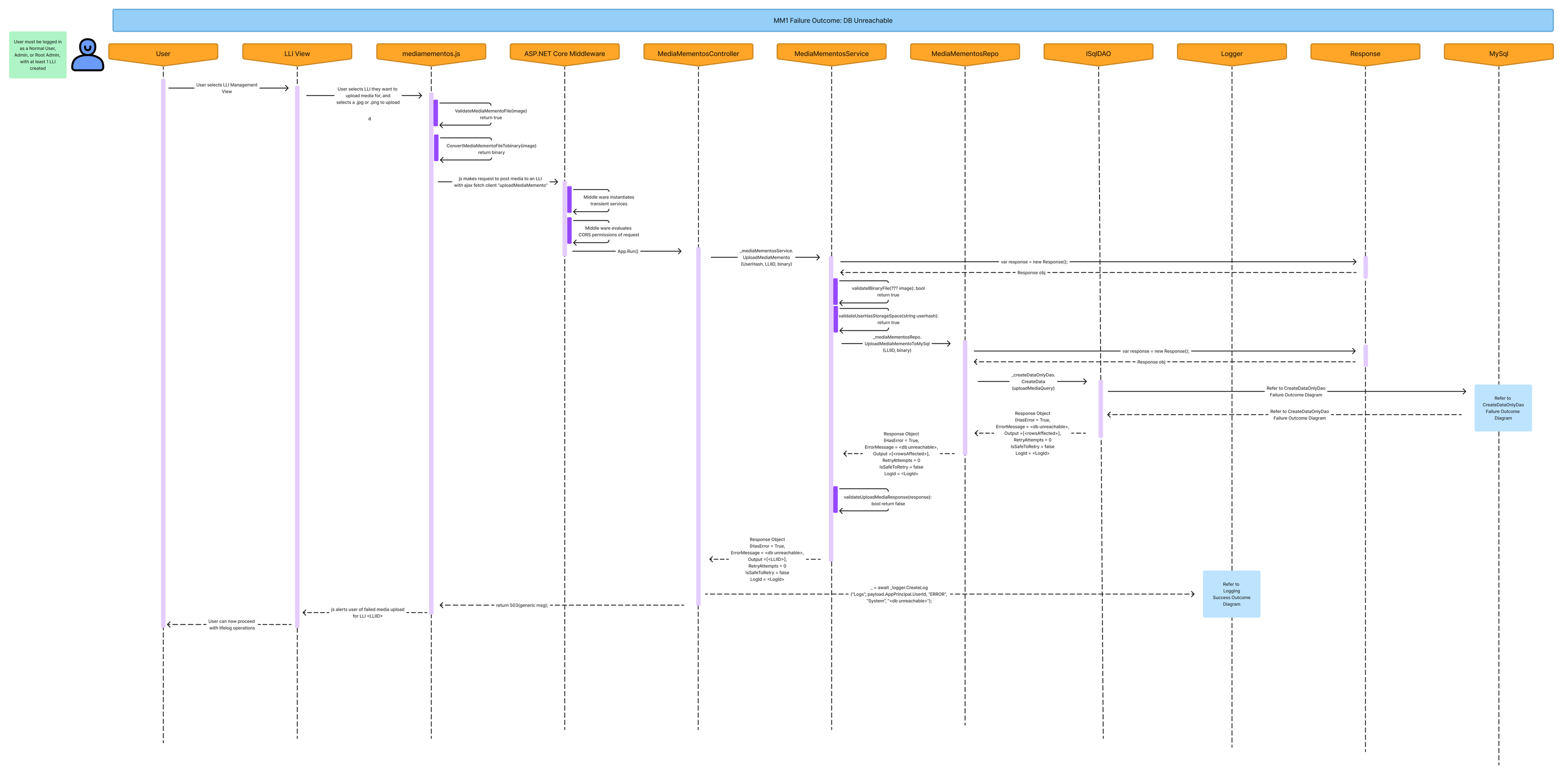
MM1 : Upload Media Mementos


MM1 Success Outcome: Upload Media
User must be logged in as a Normal User, Admin, or Root Admin, with at least 1 LLI created
User
LLI View
mediamementos.js
ASP.NET Core Middleware
MediaMementosController
MediaMementosService
MediaMementosRepo
ISqlDAO
Logger
Response
MySql
User selects LLI Management View
User selects LLI they want to upload media for, and selects a .jpg or .png to upload
d
ValidateMediaMementoFile(image) return true
ConvertMediaMementoFileTobinary(image) return binary
.js makes request to post media to an LLI with ajax fetch client "uploadMediaMemento"
Middle ware instantiates transient services
Middle ware evaluates CORS permissions of request
App.Run()
_mediaMementosService. UploadMediaMemento (UserHash, LLIID, binary)
var response = new Response();
Response obj
validateBinaryFile(??? image): bool return true
validateUserHasStorageSpace(string userhash): return true
_mediaMementosRepo. UploadMediaMementoToMySql (LLIID, binary)
var response = new Response();
Response obj
_createDataOnlyDao. CreateData (uploadMediaQuery)
Refer to CreateDataOnlyDao Sucess Outcome Diagram
Refer to CreateDataOnlyDao Sucess Outcome Diagram
Refer to CreateDataOnlyDao Sucess Outcome Diagram
Response Object {HasError = False, ErrorMessage = null, Output =[<rowsAffected>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>}
Response Object {HasError = False, ErrorMessage = null, Output =[<rowsAffected>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>}
validateUploadMediaResponse(response): bool return true
timeOperation(stopwatch timer) return true
Response Object {HasError = False, ErrorMessage = null, Output =[<LLIID>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>}
_ = await _logger.CreateLog ("Logs", payload.AppPrincipal.UserId, "INFO", "System", "Added media to <LLIID> Sucessfully");
Refer to Logging Success Outcome Diagram
return Ok(JsonSerializer.Serialize<String>(response.Output[0]));
.js alerts user of sucessful media upload for LLI <LLIID>
User can now view media, and proceed with lifelog operations

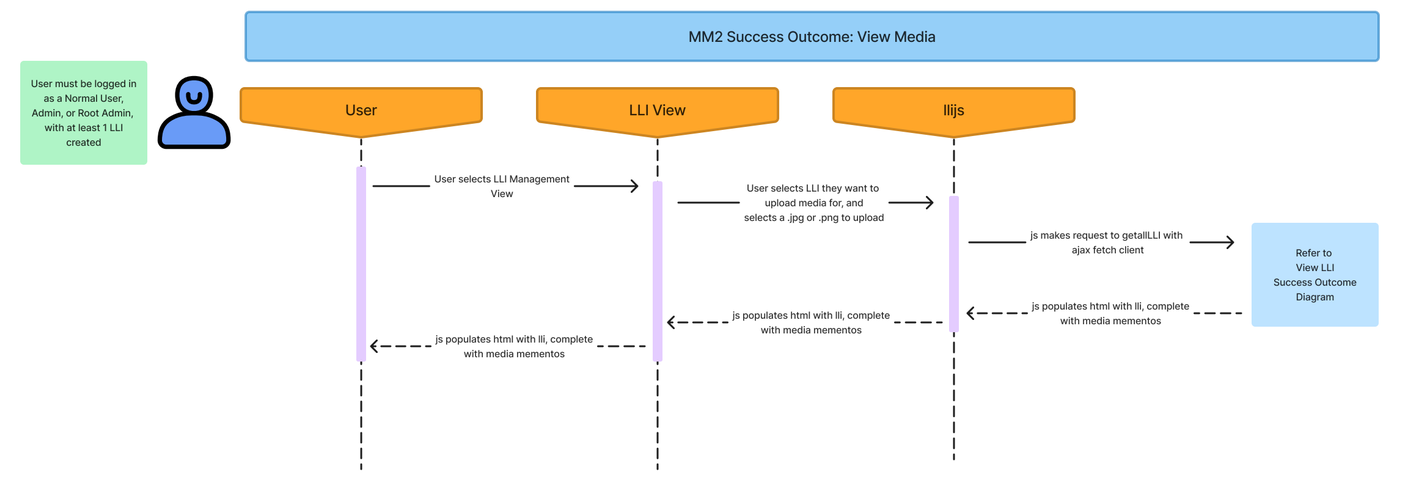

MM1 Failure Outcome: Invalid Media File
User must be logged in as a Normal User, Admin, or Root Admin, with at least 1 LLI created
User
LLI View
mediamementos.js
User selects LLI Management View
User selects LLI they want to upload media for, and selects a .jpg or .png to upload
ValidateMediaMementoFile(image) return false
.js alerts user of failure due to invalid file
User can now proceed with lifelog operations
x


MM1 Failure Outcome: Upload Fails
User must be logged in as a Normal User, Admin, or Root Admin, with at least 1 LLI created
User
LLI View
mediamementos.js
ASP.NET Core Middleware
MediaMementosController
MediaMementosService
MediaMementosRepo
ISqlDAO
Logger
Response
MySql
User selects LLI Management View
User selects LLI they want to upload media for, and selects a .jpg or .png to upload
d
ValidateMediaMementoFile(image) return true
ConvertMediaMementoFileTobinary(image) return binary
.js makes request to post media to an LLI with ajax fetch client "uploadMediaMemento"
Middle ware instantiates transient services
Middle ware evaluates CORS permissions of request
App.Run()
_mediaMementosService. UploadMediaMemento (UserHash, LLIID, binary)
var response = new Response();
Response obj
validateBinaryFile(??? image): bool return true
validateUserHasStorageSpace(string userhash): return true
_mediaMementosRepo. UploadMediaMementoToMySql (LLIID, binary)
var response = new Response();
Response obj
_createDataOnlyDao. CreateData (uploadMediaQuery)
Refer to CreateDataOnlyDao Failure Outcome Diagram
Refer to CreateDataOnlyDao Failure Outcome Diagram
Refer to CreateDataOnlyDao Failure Outcome Diagram
Response Object {HasError = True, ErrorMessage = <db unreachable>, Output =[<rowsAffected>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>}
Response Object {HasError = True, ErrorMessage = <db unreachable>, Output =[<rowsAffected>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>}
validateUploadMediaResponse(response): bool return false
Response Object {HasError = True, ErrorMessage = <db unreachable>, Output =[<LLIID>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>}
_ = await _logger.CreateLog ("Logs", payload.AppPrincipal.UserId, "Error", "System", "failed to add media to <LLIID> : <msg>");
Refer to Logging Success Outcome Diagram
return 503(generic msg);
.js alerts user of failed media upload for LLI <LLIID>
User can now proceed with lifelog operations


MM1 Failure Outcome: Operation Timeout
User must be logged in as a Normal User, Admin, or Root Admin, with at least 1 LLI created
User
LLI View
mediamementos.js
ASP.NET Core Middleware
MediaMementosController
MediaMementosService
MediaMementosRepo
ISqlDAO
Logger
Response
MySql
User selects LLI Management View
User selects LLI they want to upload media for, and selects a .jpg or .png to upload
d
ValidateMediaMementoFile(image) return true
ConvertMediaMementoFileTobinary(image) return binary
.js makes request to post media to an LLI with ajax fetch client "uploadMediaMemento"
Middle ware instantiates transient services
Middle ware evaluates CORS permissions of request
App.Run()
_mediaMementosService. UploadMediaMemento (UserHash, LLIID, binary)
var response = new Response();
Response obj
validateBinaryFile(??? image): bool return true
validateUserHasStorageSpace(string userhash): return true
_mediaMementosRepo. UploadMediaMementoToMySql (LLIID, binary)
var response = new Response();
Response obj
_createDataOnlyDao. CreateData (uploadMediaQuery)
Refer to CreateDataOnlyDao Sucess Outcome Diagram
Refer to CreateDataOnlyDao Sucess Outcome Diagram
Refer to CreateDataOnlyDao Sucess Outcome Diagram
Response Object {HasError = False, ErrorMessage = null, Output =[<rowsAffected>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>}
Response Object {HasError = False, ErrorMessage = null, Output =[<rowsAffected>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>}
validateUploadMediaResponse(response): bool return true
timeOperation(stopwatch timer) return false
Response Object {HasError = True, ErrorMessage = Timeout, Output =[<LLIID>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>}
_ = await _logger.CreateLog ("Logs", payload.AppPrincipal.UserId, "Warning", "System", "Added media to <LLIID> Sucessfully, timed out");
Refer to Logging Success Outcome Diagram
return 408(Generic msg);
.js alerts user of timeout
User can proceed with lifelog operations

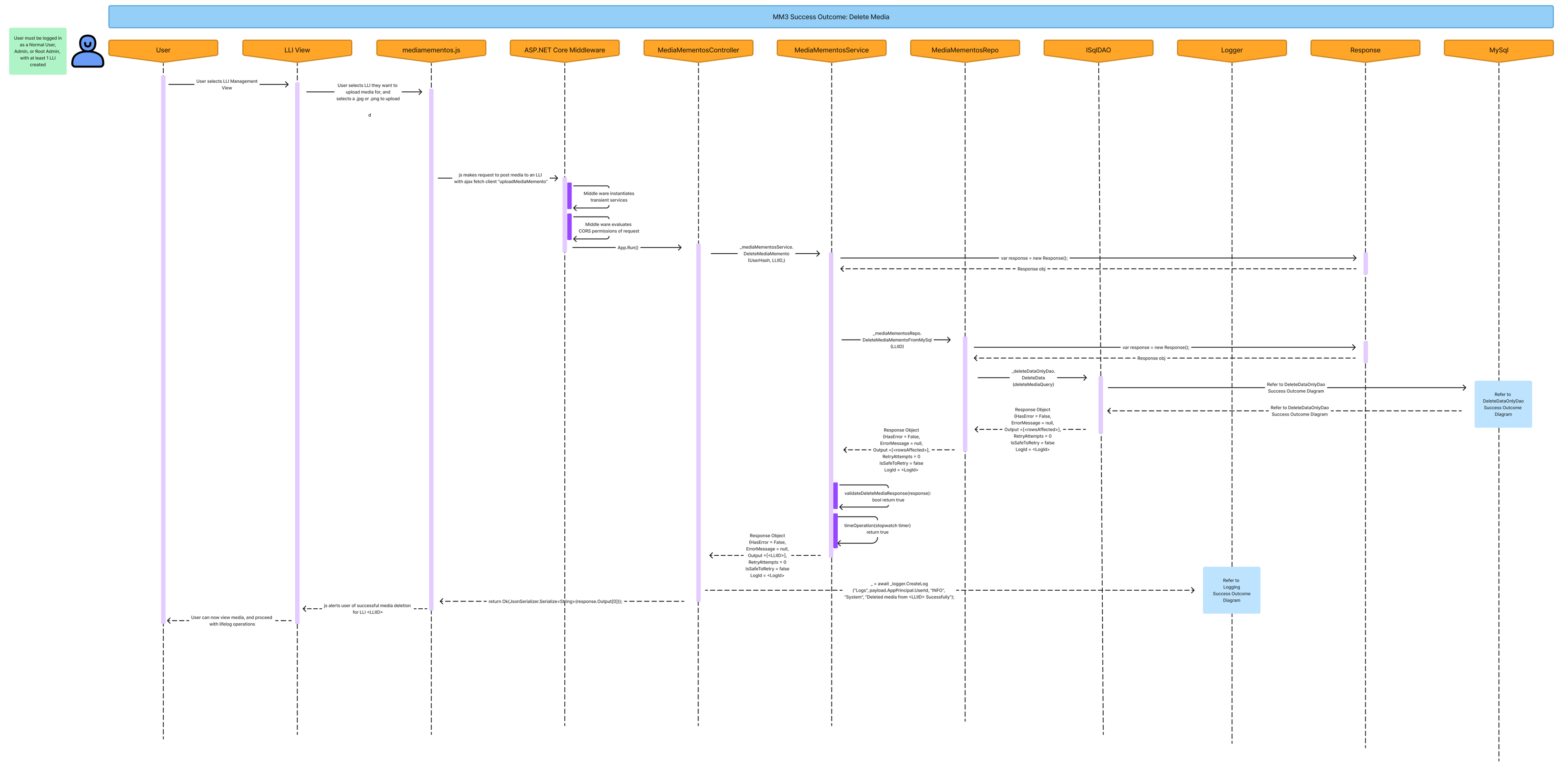

MM1 Failure Outcome: DB Unreachable
User must be logged in as a Normal User, Admin, or Root Admin, with at least 1 LLI created
User
LLI View
mediamementos.js
ASP.NET Core Middleware
MediaMementosController
MediaMementosService
MediaMementosRepo
ISqlDAO
Logger
Response
MySql
User selects LLI Management View
User selects LLI they want to upload media for, and selects a .jpg or .png to upload
d
ValidateMediaMementoFile(image) return true
ConvertMediaMementoFileTobinary(image) return binary
.js makes request to post media to an LLI with ajax fetch client "uploadMediaMemento"
Middle ware instantiates transient services
Middle ware evaluates CORS permissions of request
App.Run()
_mediaMementosService. UploadMediaMemento (UserHash, LLIID, binary)
var response = new Response();
Response obj
validateBinaryFile(??? image): bool return true
validateUserHasStorageSpace(string userhash): return true
_mediaMementosRepo. UploadMediaMementoToMySql (LLIID, binary)
var response = new Response();
Response obj
_createDataOnlyDao. CreateData (uploadMediaQuery)
Refer to CreateDataOnlyDao Failure Outcome Diagram
Refer to CreateDataOnlyDao Failure Outcome Diagram
Refer to CreateDataOnlyDao Failure Outcome Diagram
Response Object {HasError = True, ErrorMessage = <db unreachable>, Output =[<rowsAffected>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>}
Response Object {HasError = True, ErrorMessage = <db unreachable>, Output =[<rowsAffected>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>}
validateUploadMediaResponse(response): bool return false
Response Object {HasError = True, ErrorMessage = <db unreachable>, Output =[<LLIID>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>}
_ = await _logger.CreateLog ("Logs", payload.AppPrincipal.UserId, "ERROR", "System", "<db unreachable>");
Refer to Logging Success Outcome Diagram
return 503(generic msg);
.js alerts user of failed media upload for LLI <LLIID>
User can now proceed with lifelog operations

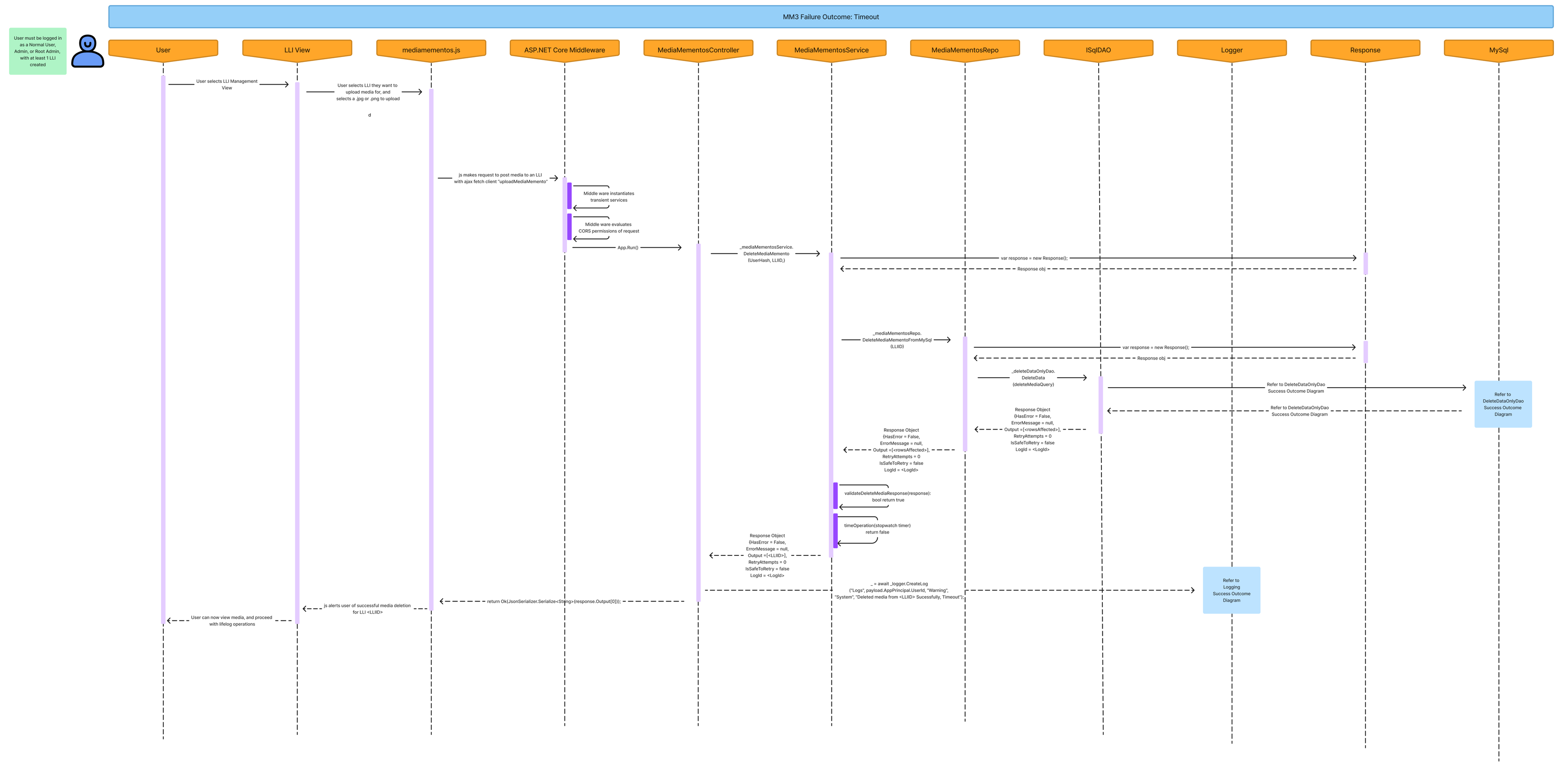
MM2 : View Media Mementos

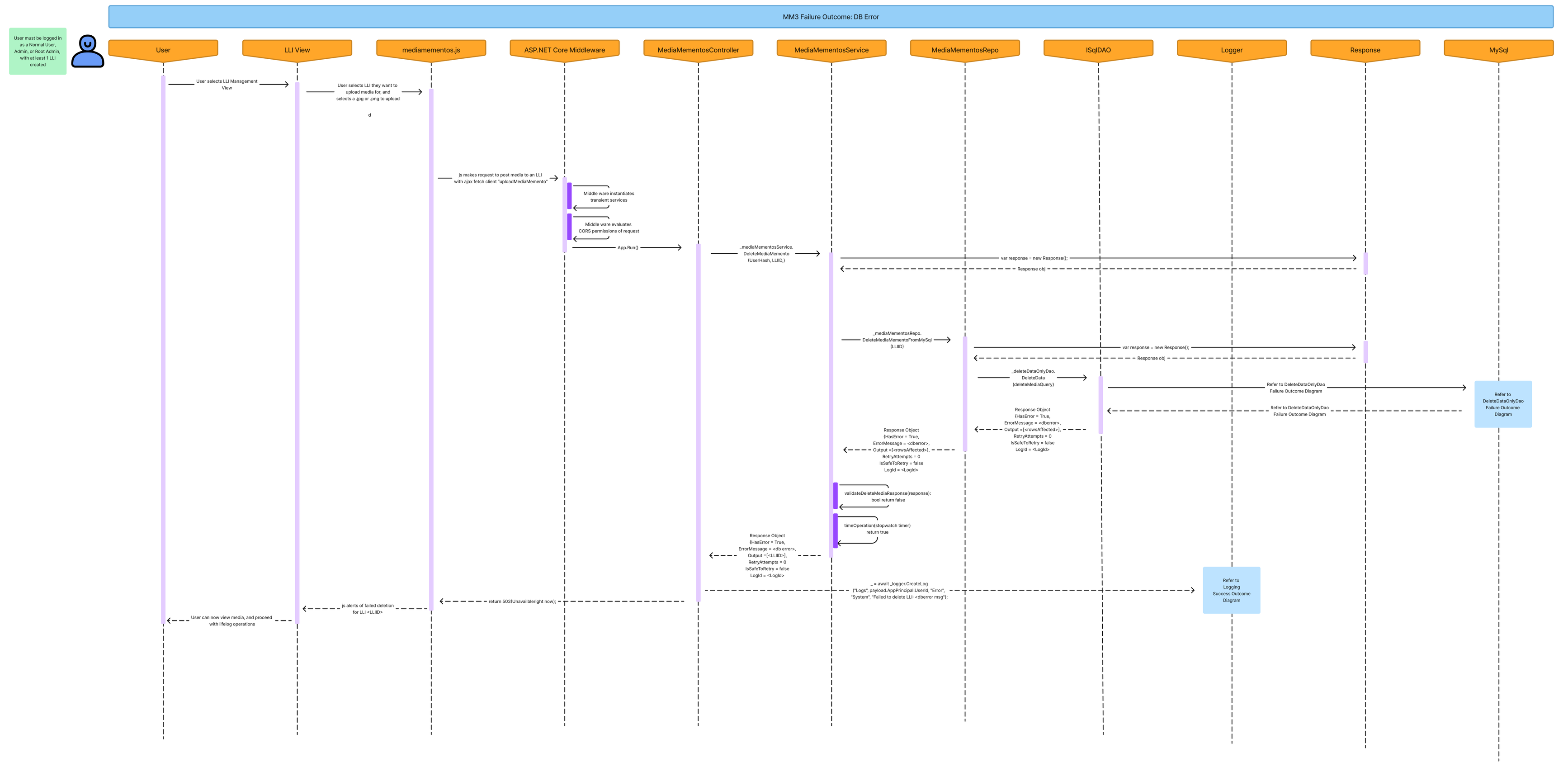

MM2 Success Outcome: View Media
User must be logged in as a Normal User, Admin, or Root Admin, with at least 1 LLI created
User
LLI View
llijs
User selects LLI Management View
User selects LLI they want to upload media for, and selects a .jpg or .png to upload
.js makes request to getAllLLI with ajax fetch client
Refer to View LLI Success Outcome Diagram
.js populates html with lli, complete with media mementos
.js populates html with lli, complete with media mementos
.js populates html with lli, complete with media mementos

MM2 Failure Outcome: Will be identical to view LLI Failures, Refer to view LLI failure diagrams

MM3 : Delete Media Mementos


MM3 Success Outcome: Delete Media
User must be logged in as a Normal User, Admin, or Root Admin, with at least 1 LLI created
User
LLI View
mediamementos.js
ASP.NET Core Middleware
MediaMementosController
MediaMementosService
MediaMementosRepo
ISqlDAO
Logger
Response
MySql
User selects LLI Management View
User selects LLI they want to upload media for, and selects a .jpg or .png to upload
d
.js makes request to post media to an LLI with ajax fetch client "uploadMediaMemento"
Middle ware instantiates transient services
Middle ware evaluates CORS permissions of request
App.Run()
_mediaMementosService. DeleteMediaMemento (UserHash, LLIID)
var response = new Response();
Response obj
_mediaMementosRepo. DeleteMediaMementoFromMySql (LLIID)
var response = new Response();
Response obj
_deleteDataOnlyDao. DeleteData (deleteMediaQuery)
Refer to DeleteDataOnlyDao Success Outcome Diagram
Refer to DeleteDataOnlyDao Success Outcome Diagram
Refer to DeleteDataOnlyDao Success Outcome Diagram
Response Object {HasError = False, ErrorMessage = null, Output =[<rowsAffected>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>}
Response Object {HasError = False, ErrorMessage = null, Output =[<rowsAffected>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>}
validateDeleteMediaResponse(response): bool return true
timeOperation(stopwatch timer) return true
Response Object {HasError = False, ErrorMessage = null, Output =[<LLIID>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>}
_ = await _logger.CreateLog ("Logs", payload.AppPrincipal.UserId, "INFO", "System", "Deleted media from <LLIID> Sucessfully");
Refer to Logging Success Outcome Diagram
return Ok(JsonSerializer.Serialize<String>(response.Output[0]));
.js alerts user of successful media deletion for LLI <LLIID>
User can now view media, and proceed with lifelog operations


MM3 Failure Outcome: Timeout
User must be logged in as a Normal User, Admin, or Root Admin, with at least 1 LLI created
User
LLI View
mediamementos.js
ASP.NET Core Middleware
MediaMementosController
MediaMementosService
MediaMementosRepo
ISqlDAO
Logger
Response
MySql
User selects LLI Management View
User selects LLI they want to upload media for, and selects a .jpg or .png to upload
d
.js makes request to post media to an LLI with ajax fetch client "uploadMediaMemento"
Middle ware instantiates transient services
Middle ware evaluates CORS permissions of request
App.Run()
_mediaMementosService. DeleteMediaMemento (UserHash, LLIID)
var response = new Response();
Response obj
_mediaMementosRepo. DeleteMediaMementoFromMySql (LLIID)
var response = new Response();
Response obj
_deleteDataOnlyDao. DeleteData (deleteMediaQuery)
Refer to DeleteDataOnlyDao Success Outcome Diagram
Refer to DeleteDataOnlyDao Success Outcome Diagram
Refer to DeleteDataOnlyDao Success Outcome Diagram
Response Object {HasError = False, ErrorMessage = null, Output =[<rowsAffected>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>}
Response Object {HasError = False, ErrorMessage = null, Output =[<rowsAffected>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>}
validateDeleteMediaResponse(response): bool return true
timeOperation(stopwatch timer) return false
Response Object {HasError = False, ErrorMessage = null, Output =[<LLIID>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>}
_ = await _logger.CreateLog ("Logs", payload.AppPrincipal.UserId, "Warning", "System", "Deleted media from <LLIID> Sucessfully, Timeout");
Refer to Logging Success Outcome Diagram
return Ok(JsonSerializer.Serialize<String>(response.Output[0]));
.js alerts user of successful media deletion for LLI <LLIID>
User can now view media, and proceed with lifelog operations


MM3 Failure Outcome: DB Error
User must be logged in as a Normal User, Admin, or Root Admin, with at least 1 LLI created
User
LLI View
mediamementos.js
ASP.NET Core Middleware
MediaMementosController
MediaMementosService
MediaMementosRepo
ISqlDAO
Logger
Response
MySql
User selects LLI Management View
User selects LLI they want to upload media for, and selects a .jpg or .png to upload
d
.js makes request to post media to an LLI with ajax fetch client "uploadMediaMemento"
Middle ware instantiates transient services
Middle ware evaluates CORS permissions of request
App.Run()
_mediaMementosService. DeleteMediaMemento (UserHash, LLIID)
var response = new Response();
Response obj
_mediaMementosRepo. DeleteMediaMementoFromMySql (LLIID)
var response = new Response();
Response obj
_deleteDataOnlyDao. DeleteData (deleteMediaQuery)
Refer to DeleteDataOnlyDao Failure Outcome Diagram
Refer to DeleteDataOnlyDao Failure Outcome Diagram
Refer to DeleteDataOnlyDao Failure Outcome Diagram
Response Object {HasError = True, ErrorMessage = <dberror>, Output =[<rowsAffected>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>}
Response Object {HasError = True, ErrorMessage = <dberror>, Output =[<rowsAffected>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>}
validateDeleteMediaResponse(response): bool return false
timeOperation(stopwatch timer) return true
Response Object {HasError = True, ErrorMessage = <db error>, Output =[<LLIID>], RetryAttempts = 0 IsSafeToRetry = false LogId = <LogId>}
_ = await _logger.CreateLog ("Logs", payload.AppPrincipal.UserId, "Error", "System", "Failed to delete LLI <dberror msg>");
Refer to Logging Success Outcome Diagram
return 503(Unavailible right now);
.js alerts of failed deletion for LLI <LLIID>
User can now view media, and proceed with lifelog operations