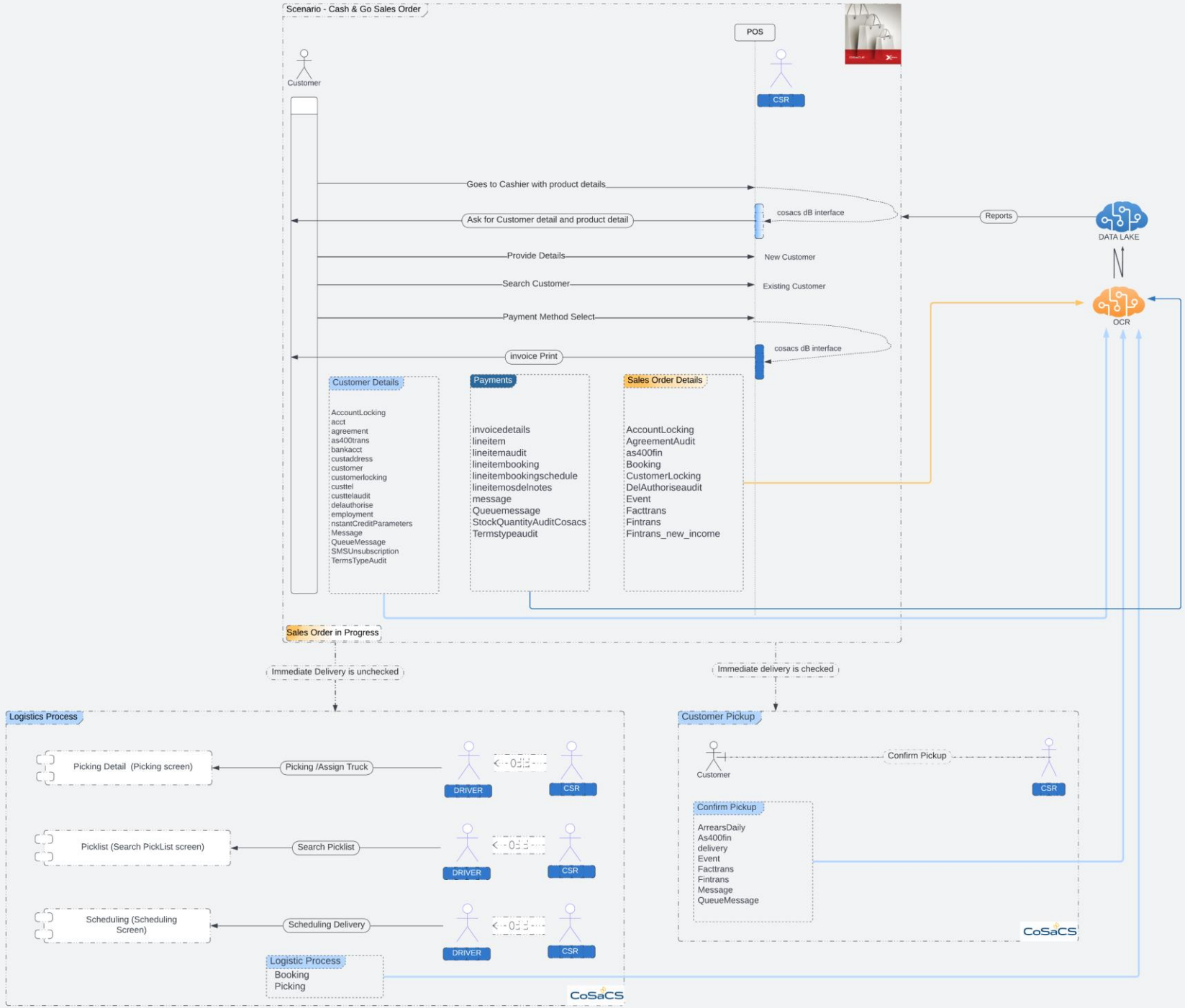
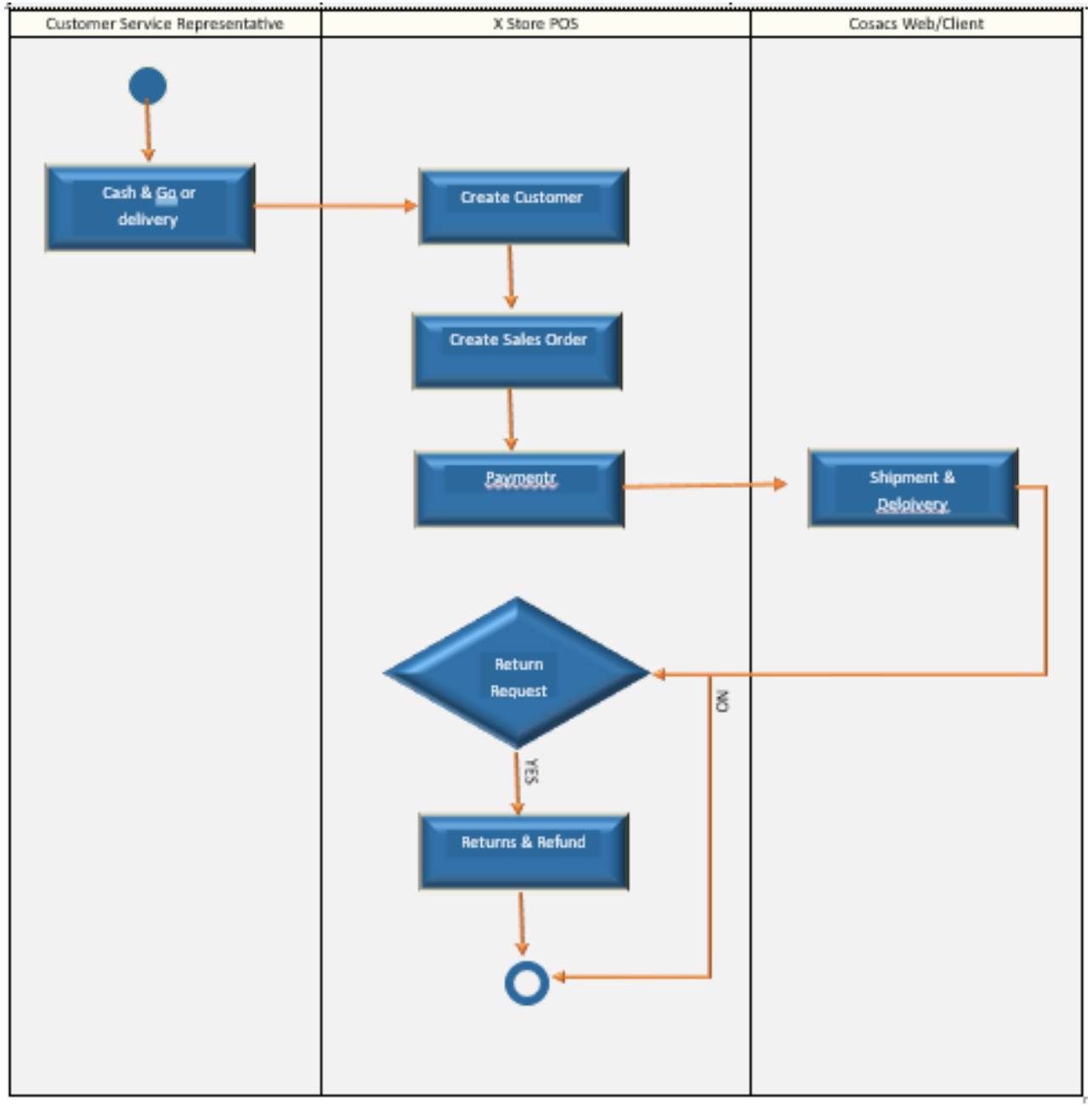
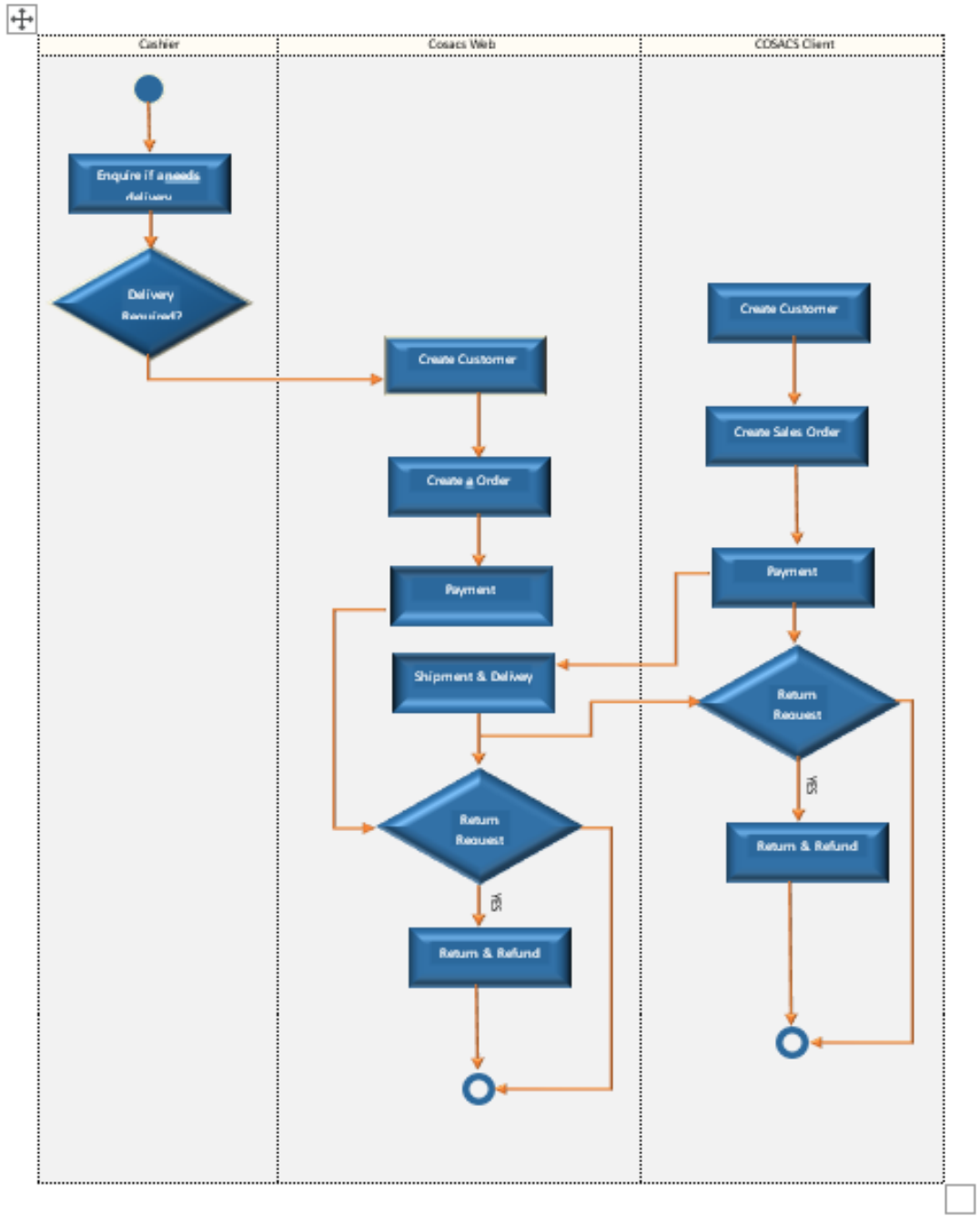
Low level System Design Diagram



To be diagram



As is diagram



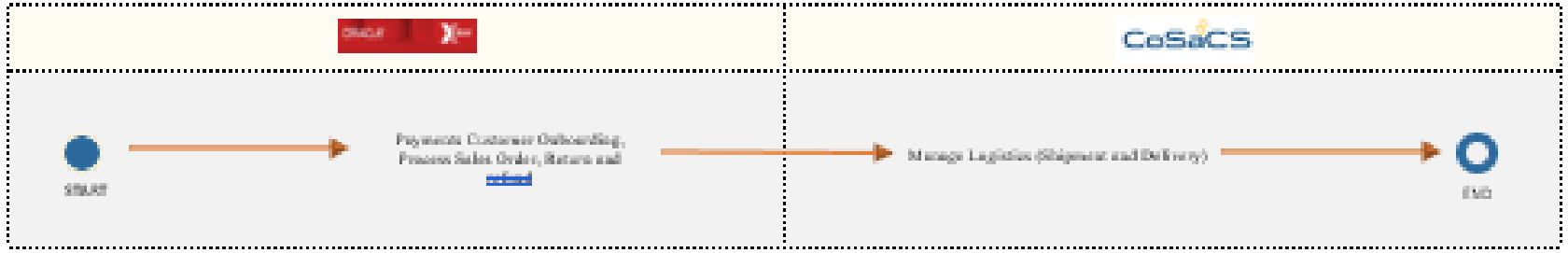
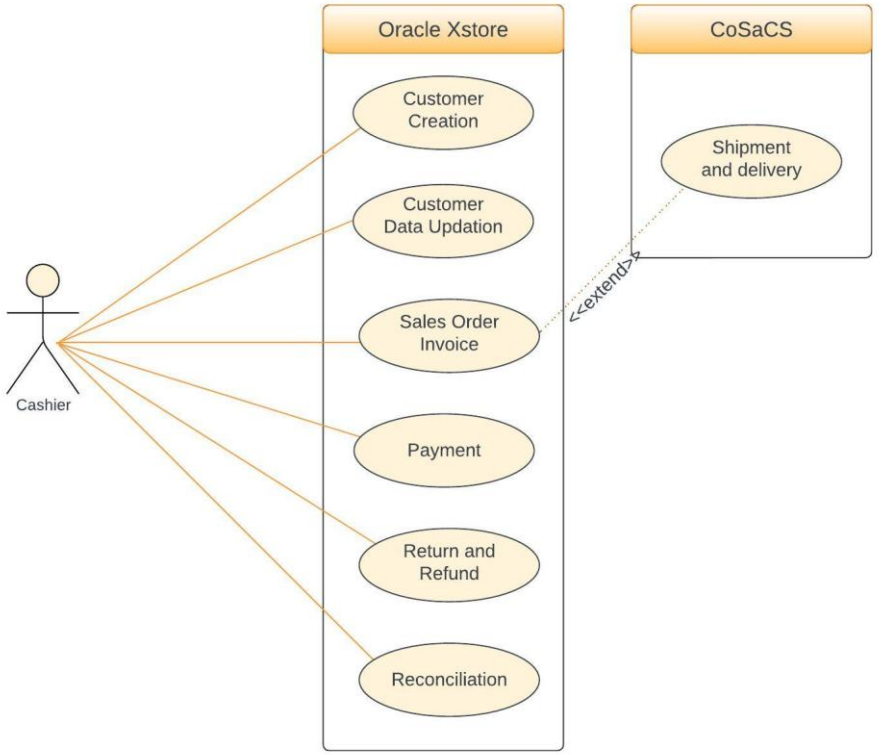
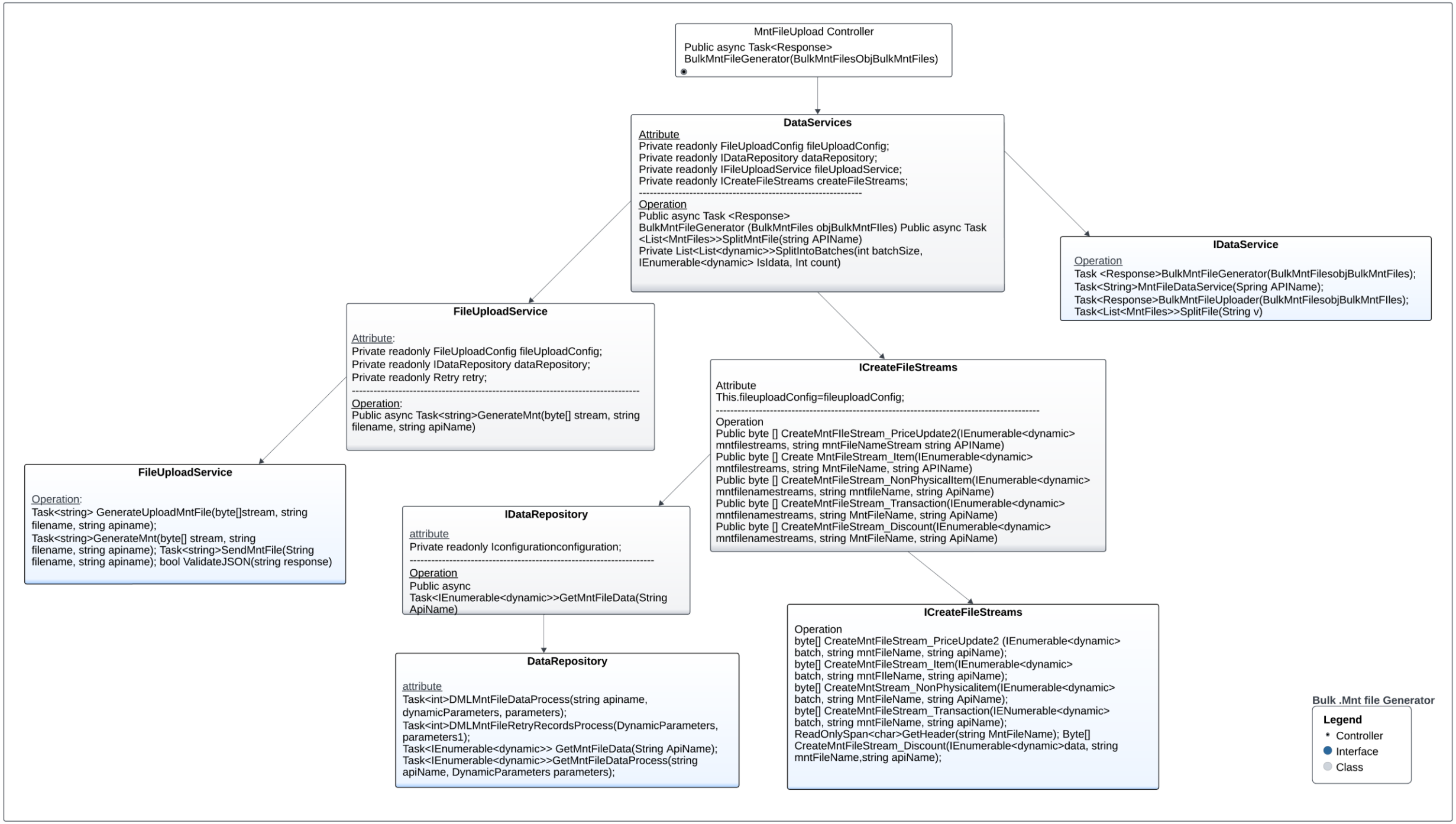
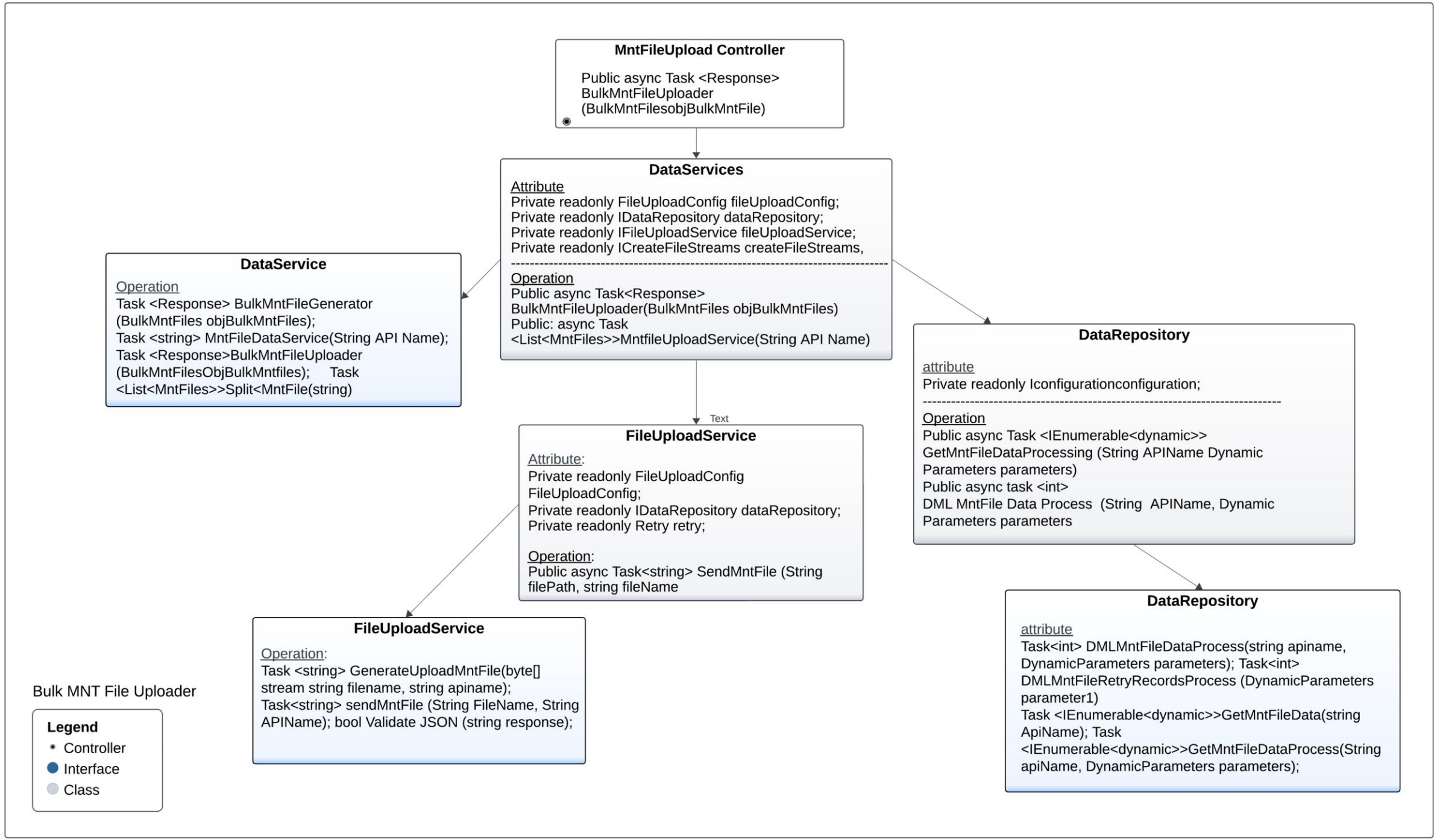


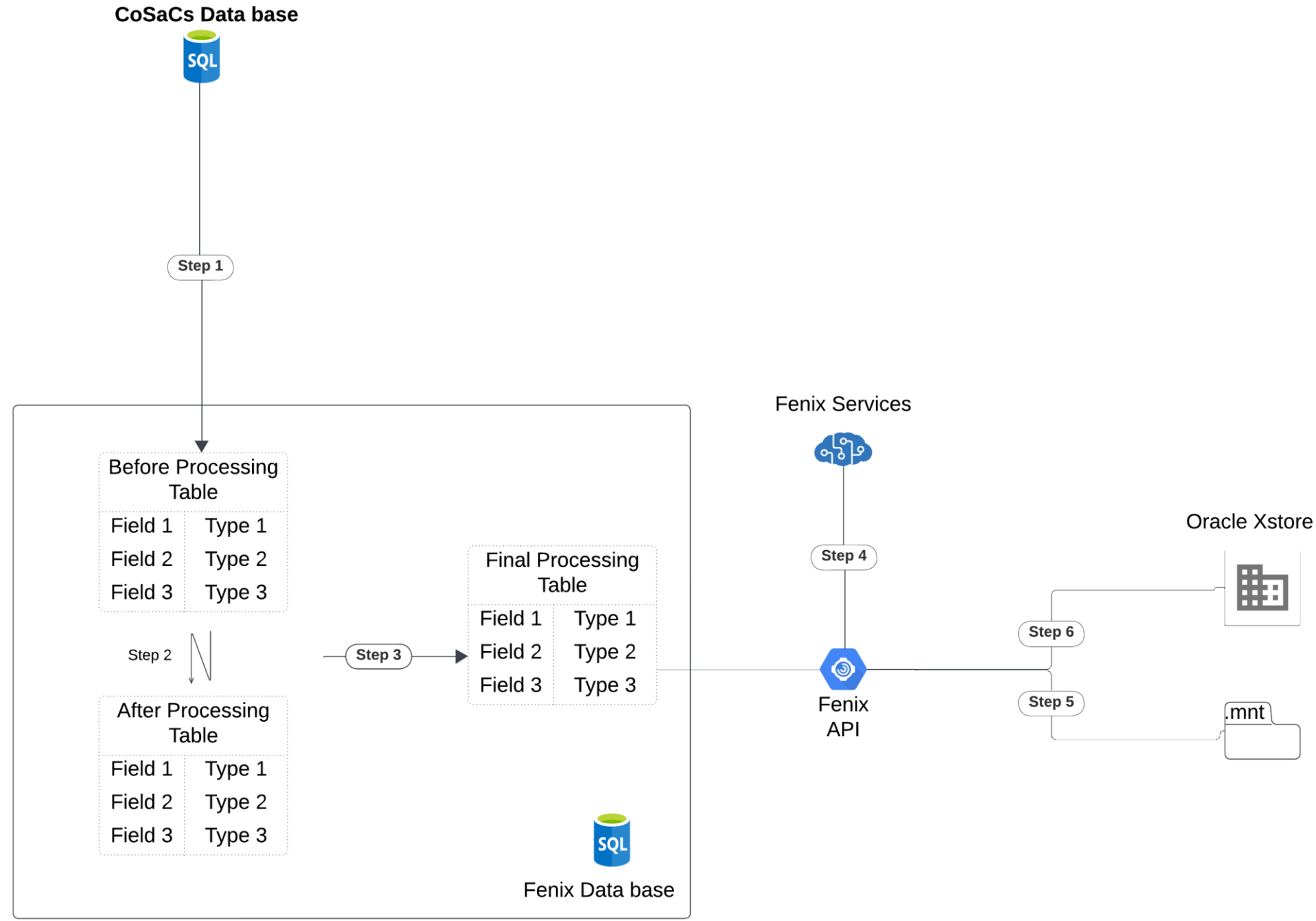
Figure 1: Proposed Business Process

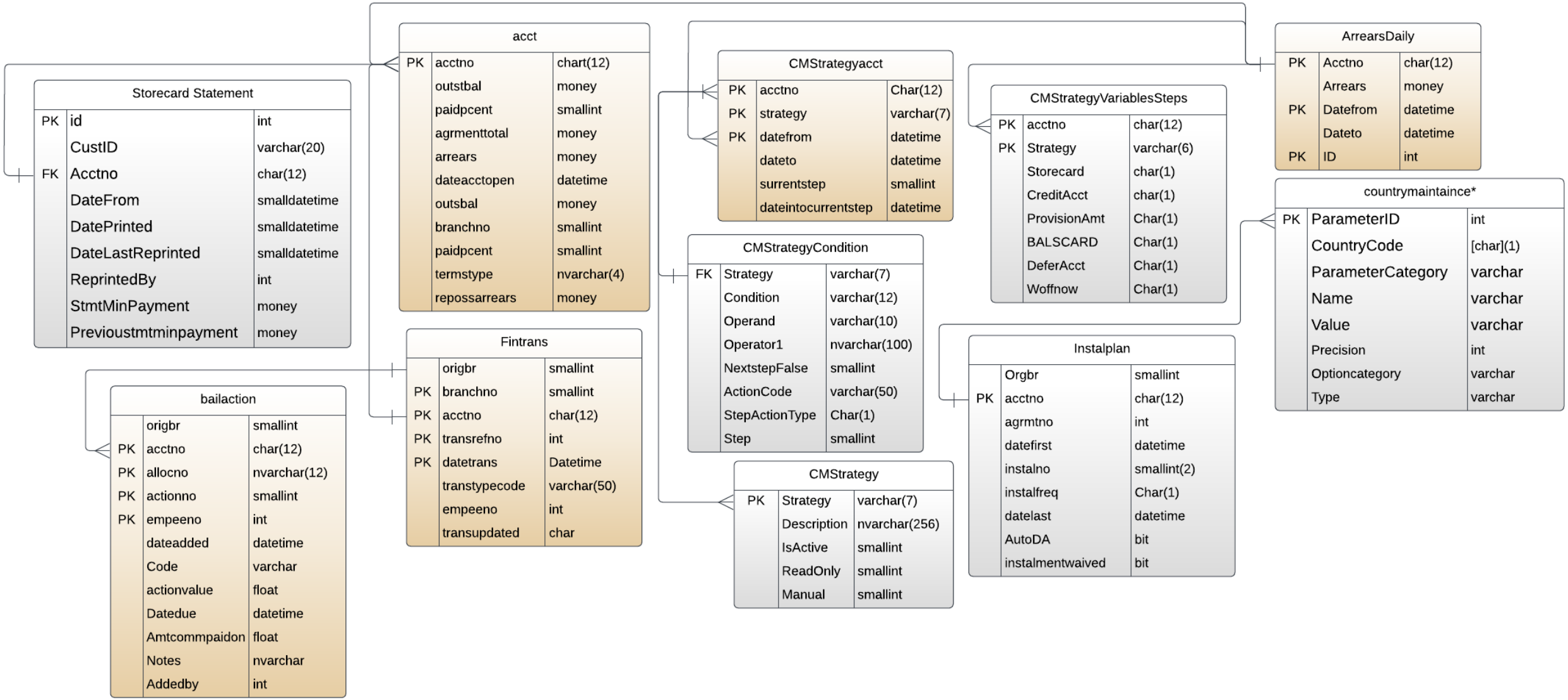
Use case diagram




MntFileUpload Controller
Public async Task<Response>
BulkMntFileGenerator(BulkMntFilesObjBulkMntFiles)
DataServices
Attribute
Private readonly FileUploadConfig fileUploadConfig;
Private readonly IDataRepository dataRepository;
Private readonly IFileUploadService fileUploadService;
Private readonly ICreateFileStreams createFileStreams;
Operation
Public async Task <Response>
BulkMntFileGenerator (BulkMntFiles objBulkMntFIles) Public async Task
<List<MntFiles>>SplitMntFile(string APIName)
Private List<List<dynamic>>SplitIntoBatches(int batchSize,
IEnumerable<dynamic> IsIdata, Int count)
IDataService
Operation
Task <Response>BulkMntFileGenerator(BulkMntFilesobjBulkMntFiles);
Task<String>MntFileDataService(Spring APIName);
Task<Response>BulkMntFileUploader(BulkMntFilesobjBulkMntFIles);
Task<List<MntFiles>>SplitFile(String v)
FileUploadService
Attribute:
Private readonly FileUploadConfig fileUploadConfig;
Private readonly IDataRepository dataRepository;
Private readonly Retry retry;
Operation:
Public async Task<string>GenerateMnt(byte[] stream, string
filename, string apiName)
FileUploadService
Operation:
Task<string> GenerateUploadMntFile(byte[]stream, string
filename, string apiname);
Task<string>GenerateMnt(byte[] stream, string
filename, string apiname); Task<string>SendMntFile(String
filename, string apiname); bool ValidateJSON(string response)
ICreateFileStreams
Attribute
This.fileuploadConfig=fileuploadConfig;
Operation
Public byte [] CreateMntFIleStream_PriceUpdate2(IEnumerable<dynamic>
mntfilestreams, string mntFileNameStream string APIName)
Public byte [] Create MntFileStream_Item(IEnumerable<dynamic>
mntfilestreams, string MntFileName, string APIName)
Public byte [] CreateMntFileStream_NonPhysicalItem(IEnumerable<dynamic>
mntfilenamestreams, string mntfileName, string ApiName)
Public byte [] CreateMntFileStream_Transaction(IEnumerable<dynamic>
mntfilenamestreams, string MntFileName, string ApiName)
Public byte [] CreateMntFileStream_Discount(IEnumerable<dynamic>
mntfilenamestreams, string MntFileName, string ApiName)
IDataRepository
attribute
Private readonly Iconfigurationconfiguration;
Operation
Public async
Task<IEnumerable<dynamic>>GetMntFileData(String
ApiName)
DataRepository
attribute
Task<int>DMLMntFileDataProcess(string apiname,
dynamicParameters, parameters);
Task<int>DMLMntFileRetryRecordsProcess(DynamicParameters,
parameters1);
Task<IEnumerable<dynamic>> GetMntFileData(String ApiName);
Task<IEnumerable<dynamic>>GetMntFileDataProcess(string
apiName, DynamicParameters parameters);
ICreateFileStreams
Operation
byte[] CreateMntFileStream_PriceUpdate2 (IEnumerable<dynamic>
batch, string mntFileName, string apiName);
byte[] CreateMntFileStream_Item(IEnumerable<dynamic>
batch, string mntFIleName, string apiName);
byte[] CreateMntStream_NonPhysicalitem(IEnumerable<dynamic>
batch, string MntFileName, string ApiName);
byte[] CreateMntFileStream_Transaction(IENumerable<dynamic>
batch, string mntFileName, string apiName);
ReadOnlySpan<char>GetHeader(string MntFileName); Byte[]
CreateMntFileStream_Discount(IEnumerable<dynamic>data, string
mntFileName,string apiName);
Bulk .Mnt file Generator
Legend
Controller
Interface
Class


MntFileUpload Controller
Public async Task <Response>
BulkMntFileUploader
(BulkMntFilesobjBulkMntFile)
DataServices
Attribute
Private readonly FileUploadConfig fileUploadConfig;
Private readonly IDataRepository dataRepository;
Private readonly IFileUploadService fileUploadService;
Private readonly ICreateFileStreams createFileStreams,
Operation
Public async Task<Response>
BulkMntFileUploader(BulkMntFiles objBulkMntFiles)
Public: async Task
<List<MntFiles>>MntfileUploadService(String API Name)
DataService
Operation
Task <Response> BulkMntFileGenerator
(BulkMntFiles objBulkMntFiles);
Task <string> MntFileDataService(String API Name);
Task <Response>BulkMntFileUploader
(BulkMntFilesObjBulkMntfiles); Task
<List<MntFiles>>Split<MntFile(string)
Text
FileUploadService
Attribute:
Private readonly FileUploadConfig
FileUploadConfig;
Private readonly IDataRepository dataRepository;
Private readonly Retry retry;
Operation:
Public async Task<string> SendMntFile (String
filePath, string fileName
DataRepository
attribute
Private readonly Iconfigurationconfiguration;
Operation
Public async Task <IEnumerable<dynamic>>
GetMntFileDataProcessing (String APIName Dynamic
Parameters parameters)
Public async task <int>
DML MntFile Data Process (String APIName, Dynamic
Parameters parameters
FileUploadService
Operation:
Task <string> GenerateUploadMntFile(byte[]
stream string filename, string apiname);
Task<string> sendMntFile (String FileName, String
APIName); bool Validate JSON (string response);
DataRepository
attribute
Task<int> DMLMntFileDataProcess(string apiname,
DynamicParameters parameters); Task<int>
DMLMntFileRetryRecordsProcess (DynamicParameters
parameter1)
Task <IEnumerable<dynamic>>GetMntFileData(string
ApiName); Task
<IEnumerable<dynamic>>GetMntFileDataProcess(String
apiName, DynamicParameters parameters);
Bulk MNT File Uploader
Legend
Controller
Interface
Class


CoSaCs Data base
SQL
Step 1
Before Processing Table
Field 1 Type 1
Field 2 Type 2
Field 3 Type 3
Step 2
After Processing Table
Field 1 Type 1
Field 2 Type 2
Field 3 Type 3
Step 3
Final Processing Table
Field 1 Type 1
Field 2 Type 2
Field 3 Type 3
SQL
Fenix Data base
Fenix Services
Step 4
Fenix API
Step 6
Step 5
Oracle Xstore
.mnt



High Level ERD Credit Collection