

Collection Web-Site Design Document

Last updated 12 June 2020.

Overview

This web-site is to be based on the function, layout, styling and animation of the current Word-Press web-site. Allowing for greater control and ease of maintenance of the system, as well as providing the base for expansion of provision of Cyber-Art.

Web-Site Pages

These are as defined in the Google-Drive Website documents

Navigation

This is defined in the Google-Drive Website documents

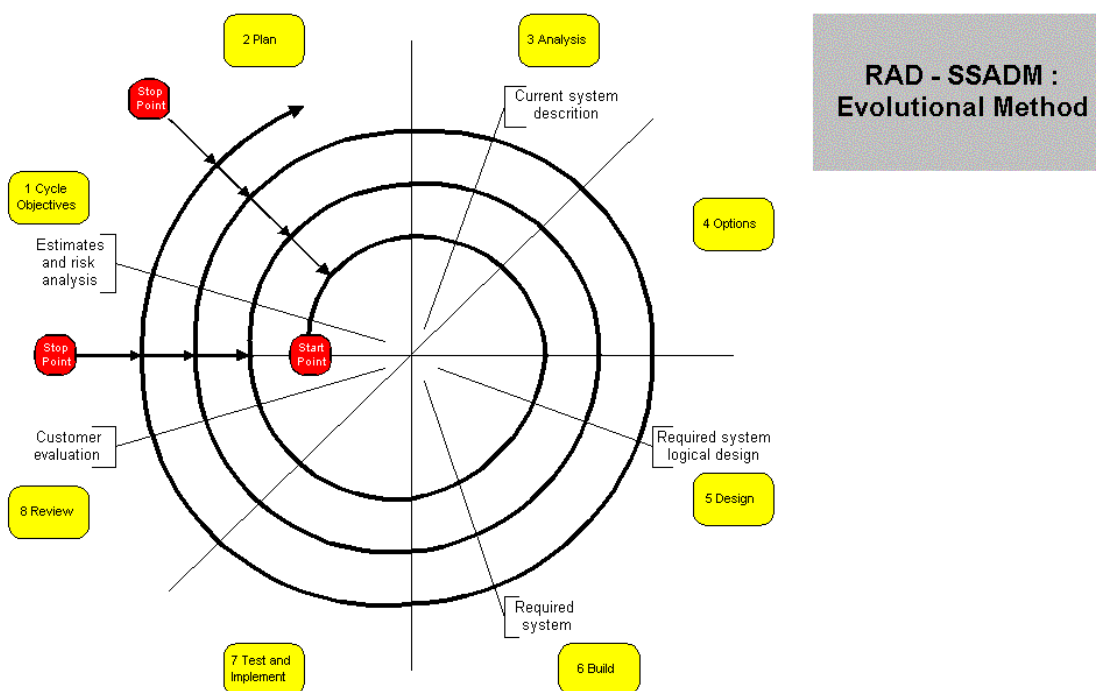
Collection Web-site Paradigm, Praxis and Anabis

Set of rules, learning theory and paths travelled.

Proposed interaction with the web-site is by a Micro-Service Oriented Architecture, using an Entity Object data store. Implementation by use of Microsoft Razor-Pages and Blazor server application, combined with Blazor Web-Assembly (WASM) client front-end, when it becomes available in May this year.

The design process is based on 'SOLID' object-oriented principles, and 'ACID' database transactions; using micro-services interfacing to a data access layer referencing the object data store; sitting on top of a transaction-based e-commerce sub-system, allowing for long-term disconnected asset purchases, prevalent with mobile smart devices.

The development will use RAD techniques to promote a quick release of the web-site tailored to the company's exact needs, in an iterative manner. The initial prototype will not use the Entity Object data store, but a simple relational table database, to test the design; before embarking on the full object-oriented design implementation.



Do not worry, I do not intend using the full SSADM, just the parts we need.

Entities

Entities, Objects and Constructs: Entities are the collection of data-structures, function methods and temporal paradigms that encapsulate a virtualisation. Objects are the instantiation of Entities into the real world of data. Constructs are the loose aggregation of data-structures and functions, interlinked by data and position-based dependencies, without due regard to isolation or timing. Entities have Life Histories.

ELH: Entity life history, can be defined as a diagram or a temporal list; often hierarchical.

Users (Artists, Owners and Staff)

The clients, customers and staff; who use the web-site. These users are generally supported using the OWIN sub-system for their ELHs.

Artwork

Contains a number of Editions.

ELH: Creation, Release, Retraction, Archive, Dissolution.

Edition

Can contain a number of Bids.

ELH: Creation, Purchase*, Resale*, Buy-back, Deletion.

Bid

ELH: Creation, Conclusion, Deletion.

Event

Contains a number of Tickets

ELH: Creation, Conclusion, Archive, Deletion.

Ticket

ELH: Creation, Purchase, Deletion.

Purchase Entity

This is a transient entity that is used to control purchase payment transactions in conjunction with the 'Entity State Management' data-set.

ELH: Creation, Locking, Payment, Unlocking, Deletion.

Process Paths

Workflows, Paths and State-Machines: A Workflow is the possible Paths through a State-Machine. To this end there are 'Main Paths', 'Side Paths' and 'Deviant Paths'. A Main-Path is the normal method of working in a system, whereas a Side-Path is a little used but necessary change to the normal working (e.g. Artwork Edition unavailable); but a Deviant-Path is a stop to the working of a system, requiring unusual steps to be taken (e.g. purchase cancelation). A State-Machine is the accumulation of the process steps available in a system, with step precursors, prerequisites and possible future destination steps; often described by use of workflow diagrams or flow-charts. There can be many state-machines, each associated with a particular sub-system.

Base Processes

Payment Process Paths

Main Path

- 1 Lock 'referenced record'
- 2 Request payment from 3rd party with User credentials.
- 3 Wait for response from 3rd party. (side path 1 or 2)
- 4 Response 'OK'
- 5 Update 'referenced record'.
- 6 Unlock 'referenced record'.
- 7 Exit with success message 'done'.

Side Path 1

- 1 Response 'Card Refused'
- 2 Unlock 'referenced record'.
- 3 Exit with failed message 'sorry, card refused'.

Side Path 2

- 1 Response 'system down'
- 2 Unlock 'referenced record'.
- 3 Exit with failed message 'sorry, problem with card system, please try again later'.

Deviant Path

User Cancelled transaction.

- 1 Attempt Abort card payment (idempotent request)
- 2 If Abort failed, continue parent process
- 2 Unlock 'referenced record' (idempotent function).
- 3 Exit with failed message 'user cancelled'.

Refund Process Path

Main Path

- 1 Request refund from 3rd party with User credentials.
- 2 Wait for response from 3rd party. (side path)
- 3 Response 'OK'
- 4 Update 'referenced record'.
- 5 Exit with success message 'done'.

Side Path

- 1 Response 'system down'
- 2 Exit with failed message 'sorry, problem with card system, please try again later'.

Artwork Edition Processes

Edition Purchase Workflow

Main Path

1. Receive request to purchase

2. Check for availability (side path 1)
3. OK, edition is available
4. Start transaction – lock records
5. Re-check if edition is still available (side path 1)
6. Assemble user credentials and financial details
7. Enact 'Payment Process' workflow (side path 2)
8. Update edition records
9. End transaction 'Commit' – unlock records
10. Exit with message to User 'success'

Side Path 1

1. Edition not available
2. If in-transaction, end Transaction 'Roll-back'
3. Exit with message to User 'sorry no editions available'

Side Path 2

1. Problem with payment
2. End Transaction 'Roll-back'
3. Exit with Reflected message to User

Deviant Path

User Cancelled transaction.

1. If payment in-process signal abort-payment
2. If abort failed, continue parent process
3. If in-transaction, end Transaction 'Roll-back'
4. Exit with Reflected message to User or failed message 'user cancelled'.

Edition Resale Workflow

Main Path

1. Receive request to purchase
2. Start transaction – lock records
3. Assemble user credentials and financial details
4. Enact 'Payment Process' workflow (side path)
5. Update edition records
6. End transaction 'Commit' – unlock records
7. e-mail previous owner with message 'edition sold'
8. Exit with message to User 'success'

Side Path

1. Problem with payment
2. End Transaction 'Roll-back'
3. Exit with Reflected message to User

Deviant Path

User Cancelled transaction.

1. If payment in-process signal abort-payment
2. If abort failed, continue parent process
3. If in-transaction, end Transaction 'Roll-back'
4. Exit with Reflected message to User or failed message 'user cancelled'.

Edition Refund Workflow

Main Path

1. Receive request to refund
2. Start transaction – lock records
3. Assemble user credentials and financial details
4. Enact 'Refund Process' workflow (side path)
5. Update edition records
6. End transaction 'Commit' – unlock records
7. Exit with message to User 'success'

Side Path

1. Problem with refund
2. End Transaction 'Roll-back'
3. Exit with Reflected message to User

Event Ticket Processes

Ticket Purchase Workflow

Main Path

Ticket Refund Workflow

Main Path

Artwork Asset Processes V2

Asset Purchase Workflow, Separate Authorise and Charge

Main Path

1. Receive request to purchase
2. Check for availability (side path 1)
3. OK, edition is available
4. Assemble user and financial details, create Checkout Session
5. Enact 'Payment Authorisation' (side path 2)
6. Re-check if edition is still available (side path 1)
7. Start transaction – lock records
8. Update edition records
9. Collect 'Payment Charge'
10. End transaction 'Commit' – unlock records
11. Exit with message to User 'success'

Side Path 1

1. Edition not available
2. Cancel Payment
3. Exit with message to User 'sorry no editions available'

Side Path 2

1. Problem with payment or User cancels
2. Exit with Reflected message to User

Deviant Path

Problem with Stripe system.

1. Cancel Payment Intent
2. Rollback Edition transfer
3. End transaction 'Rollback' – unlock records
4. Exit with message to User 'sorry problem with card processor, please try again later'

Asset Refund Workflow

Main Path

Bid Workflow

Main Path

Data Models

Objects are instantiated Models.

Entities are Objects with a Timeline.

Objects(data) are comprised of data-records in data tables.

Records(data) are both primary (fields and keys) and secondary (attributes and metadata).

User Identity and Control (OWIN)

Standard Microsoft implementation of identity, using both Roles and Claims; with external Meta-Data and Attributes. **OWIN** = Open Web Interface for dot Net.

User Roles

- **Administrators:** generally AA staff able to access all Admin pages, except financial.
- **Controllers:** AA staff able to access most Admin pages.
- **Staff:** AA staff able to access some Admin pages.
- **Finance:** AA staff able to access financial Admin pages.
- **Owners:** users, also known as Customers.
- **Artists:** users, also known as Clients.
- **Special:** Users, also known as Star Customers, allowed to purchase 'Star' editions when released tranches of editions are sold-out.

User Claims

Generally used for User-Details that are under GDPR scope, using predefined identifiers.

User Tokens

Single Sign-on, for a 3rd party; we authenticate for them.

User Logins

Single Sign-on, via a 3rd party; they authenticate for us.

User Meta-Data (outside OWIN)

Singular non-control open format data, often by blob-links; normally for Artists details and work biographies.

User Attributes (outside OWIN)

Predefined user control data, not in GDPR scope.

Artworks

Artistic creations with a cyber-media theme. Has Attributes.

Assets and Editions

Available licenses to own impressions of Artworks, either limited editions or normal assets. Has Attributes.

Asset Log

Log of Asset history, keyed on Date-Time-Stamp.

Bids

Collection of Bid events for Editions or Assets, keyed on Date-Time-Stamp.

Events

Situations where tickets are available to be purchased; via third party. Third party yet to be determined. Has Attributes.

Tickets

An e-Ticket for an Event.

Transaction Log

Precis of purchases and asset life events, keyed on Date-Time-Stamp, has meta-data.

Entity State Management

Single point of reference for Entity State, linked to transactions and e-commerce sub-system. Keyed on Date-Time-Stamp, has attributes.

User Log of Interactions

Useful during fault finding and customer support calls, and can be used by marketing for non-sales. Keyed on Date-Time-Stamp, has attributes.

e-Commerce System

Base on 'Stripe' payments and Purchase orders. Interacts with other entities to perform purchases reliably.

Asset Ownership Validation

External API to authenticate ownership claims, possibly based on asset block-chain derivation and validation.

External (Blob) Data Store

In the cloud simple data store for binary-large-objects; images, VR/AR data packages, etc..

System Metadata

This construct allows for the system operating object data to be decoupled from the constraints of the underlying database system and website configuration infrastructure.

Hierarchical, System data store; used to store key data and 'look-up' tables.

Model-Object Attribute sub-system

This construct allows for the object attribute data to be decoupled from the constraints of the underlying database system infrastructure.

This a 5th Normal implementation of extra data fields required by Model Objects. It comprises two data tables directly linked to the parent Object, plus a system-wide data table defining field data types. The first is for the pre-defined attribute field names and field data-type with optional data defaults and limits. The second is the place for the field attribute data, supporting variant data stores. The system-wide data table of data types is used to both limit the data-types used but also to allow for automated format conversion and type checking.

Data Tables

From Rodrigo

Artwork properties

- Title
- Subtitle
- Description
- Artist ID
- Size Description
- Year
- Edition Size
- AP Size (Artist Prove)
- Colour
- Artwork Type (AR, VR, MR, Image, Video, Other)
- Thumb Image URL
- Full Image URL // Also could be a reference for the
- Trademark URL

Artist

- First Name
- Last Name
- Known Name
- Description
- Image URL
- Signature URL

Edition

- Edition Number
- Artwork ID

Prototype Database (Relational only)

AspNetUsers as per OWIN

AspNetRoles as per OWIN

AspNetUserRoles as per OWIN

AspNetUserClaims as per OWIN

Artists

Field	Type	Key	Details
UserId	uuid	primary	foreign key
FirstName	a50		GDPR problem
LastName	a50		GDPR problem
KnownName	a80		possible GDPR problem
Description	a400		possible GDPR problem
ImageURL	a250		link to external blob
SignatureURL	a250		link to external blob
Artist_Notes	aMax		

Owners

Field	Type	Key	Details
UserId	uuid	primary	foreign key
Owner_AKA	a80		Display name for owner, nickname
Owner_Notes	aMax		

User Claim Types

Field	Type	Key	Details
ClaimType	a80	primary	used in OWIN User-Claims
ClaimType_Desc	a400		description of type

Artworks

Field	Type	Key	Details
Id	uuid	primary	aka ArtworkId
Title	a80		
Subtitle	a250		
a-Description	aMax		
ArtistId	uuid	secondary	Artists User-Id
SizeDescription	a400		
Year	a10		
EditionSize	a80		
APSize	a80		Artist Prove(?)
Colour	a80		
ArtworkType	a40		AR, VR, MR, Image, Video, Other
ThumbImageURL	a250		link to external blob
FullImageURL	a250		link to external blob; also could be a reference for the (?)
TrademarkURL	a250		link to external blob
Artwork_Notes	aMax		

Assets

Field	Type	Key	Details
Id	uuid	primary	aka AssetId
ArtworkId	uuid	foreign	Unique constraint with EditionNumber
OwnerId	uuid	foreign	can be null
EditionNumber	n4	secondary	unique constraint with ArtworkId
Asset_Notes	aMax		

Direct link to Asset-Log table, indirect link to Transaction-Log table.

Candidate for data-object tables.

AssetLog

Field	Type	Key	Details
Id	uuid	primary	aka AssetLogId
AssetLog_DTS	datetime	secondary	
AssetId	uuid	foreign	
AssetLog_Desc	aMax		

TransactionLog

Field	Type	Key	Details
Id	uuid	primary	aka TransLogId
TransLog_DTS	datetime	secondary	
UserId	uuid	foreign	
AssetId	uuid	foreign	
TransLog_Desc	aMax		

Bids

Field	Type	Key	Details
Id	uuid	primary	aka BidId
Bid_DTS	datetime	secondary	
UserId	uuid	foreign	person making the bid
AssetId	uuid	foreign	
Bid_Amount	n4	secondary	Bid amount in US-Dollars
Bid_Notes	aMax		mini-blog on bid

Entity State Management

Sub-system to handle Entity States, also known as 'Event Sourcing'; a single point of entity state control, independent of web-page internal states, mobile-app internal states or session states. To enable the implementation of Entity-Life-Histories using process workflows.

EntityState (object)

Field	Type	Key	Details
Id	uuid	primary	aka ESId
ES_DTS	datetime	secondary	the date-time-stamp of the state change
EntityId	uuid	foreign	the entity whose state we are controlling
ESTId	n4	foreign	type or name of entity
ESEId	n4	foreign	entity state (enum?)

EntityStateMachines (model)

Field	Type	Key	Details
Id	uuid	primary	aka ESMId
EntityId	uuid	foreign	the entity whose state we are controlling
ESM_Workflow	aMax		Workflow as XML construct
ESM_Name	a50		human identifier
ESM_Notes	aMax		mini-blog

EntityStateTypes (model look-up)

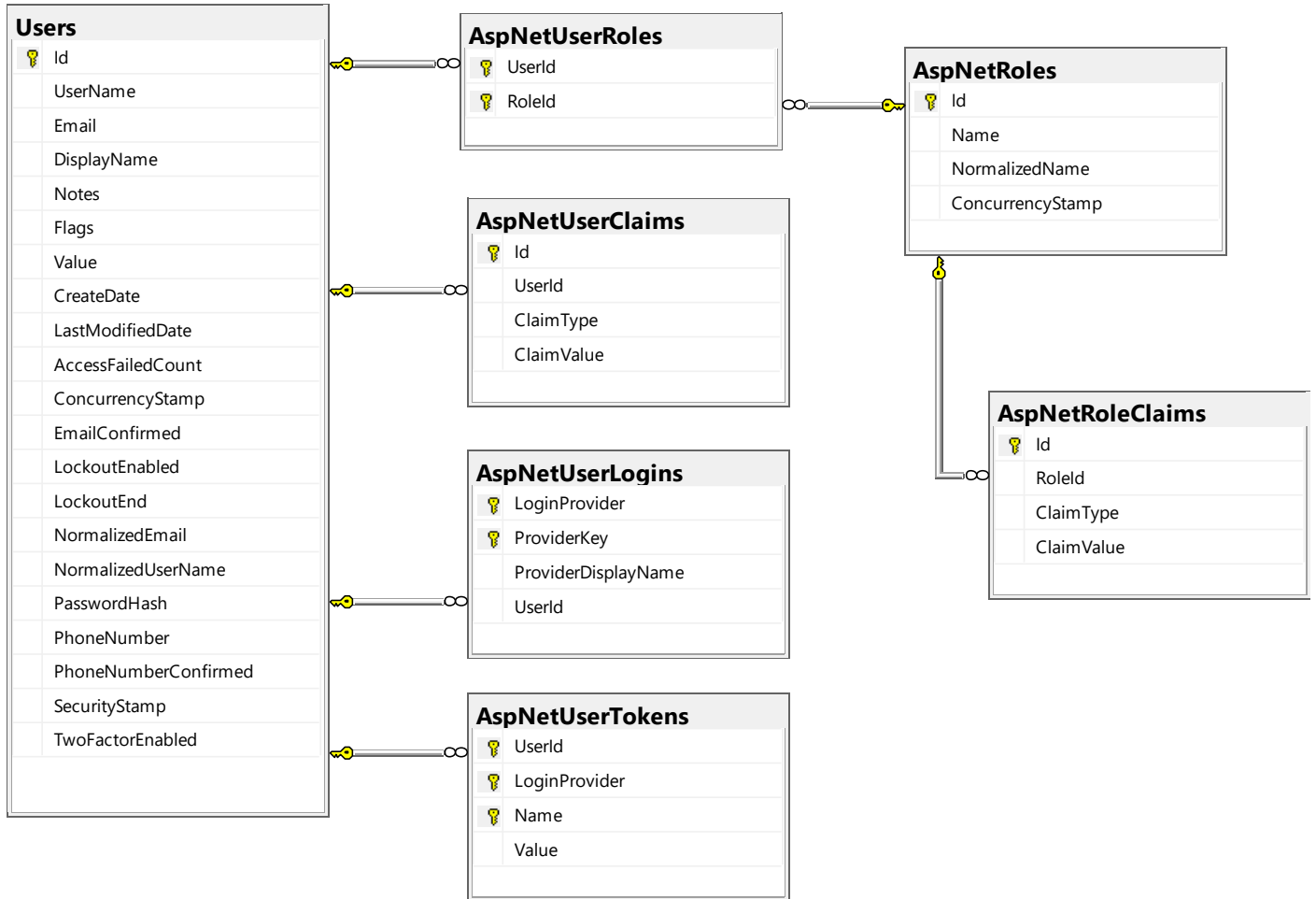
Field	Type	Key	Details
ESTId	n4	primary	auto-number
EST_Name	a50		human identifier
EST_Notes	aMax		mini-blog

EntityStateEnums (model look-up)

Field	Type	Key	Details
ESEId	n4	primary	auto-number
ESTId	n4	foreign	entity type
ESE_Name	a50		human identifier
ESE_Notes	aMax		mini-blog

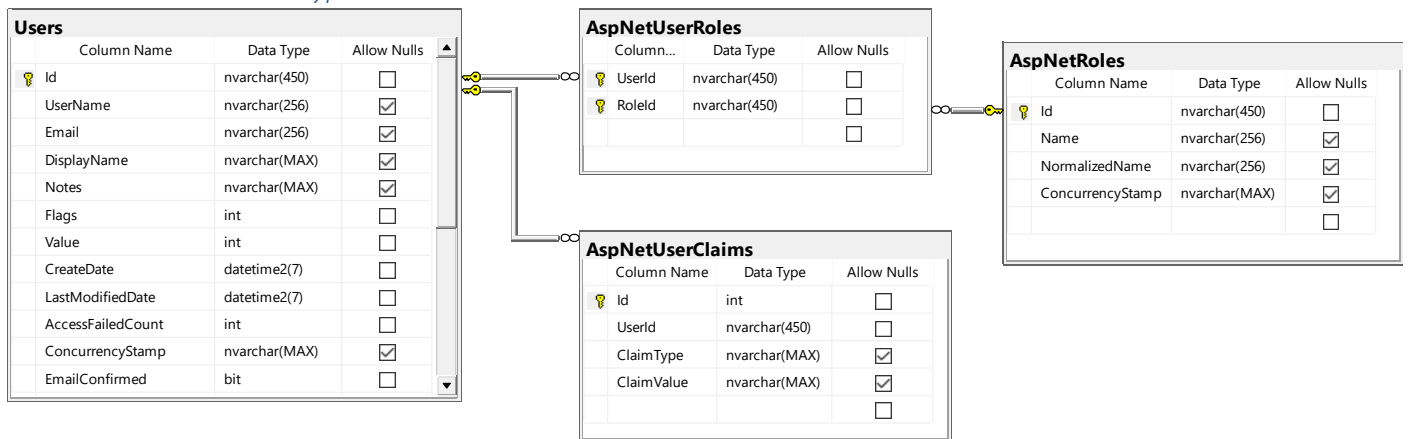
Database Diagrams

Full Asp.Net OWIN, all versions

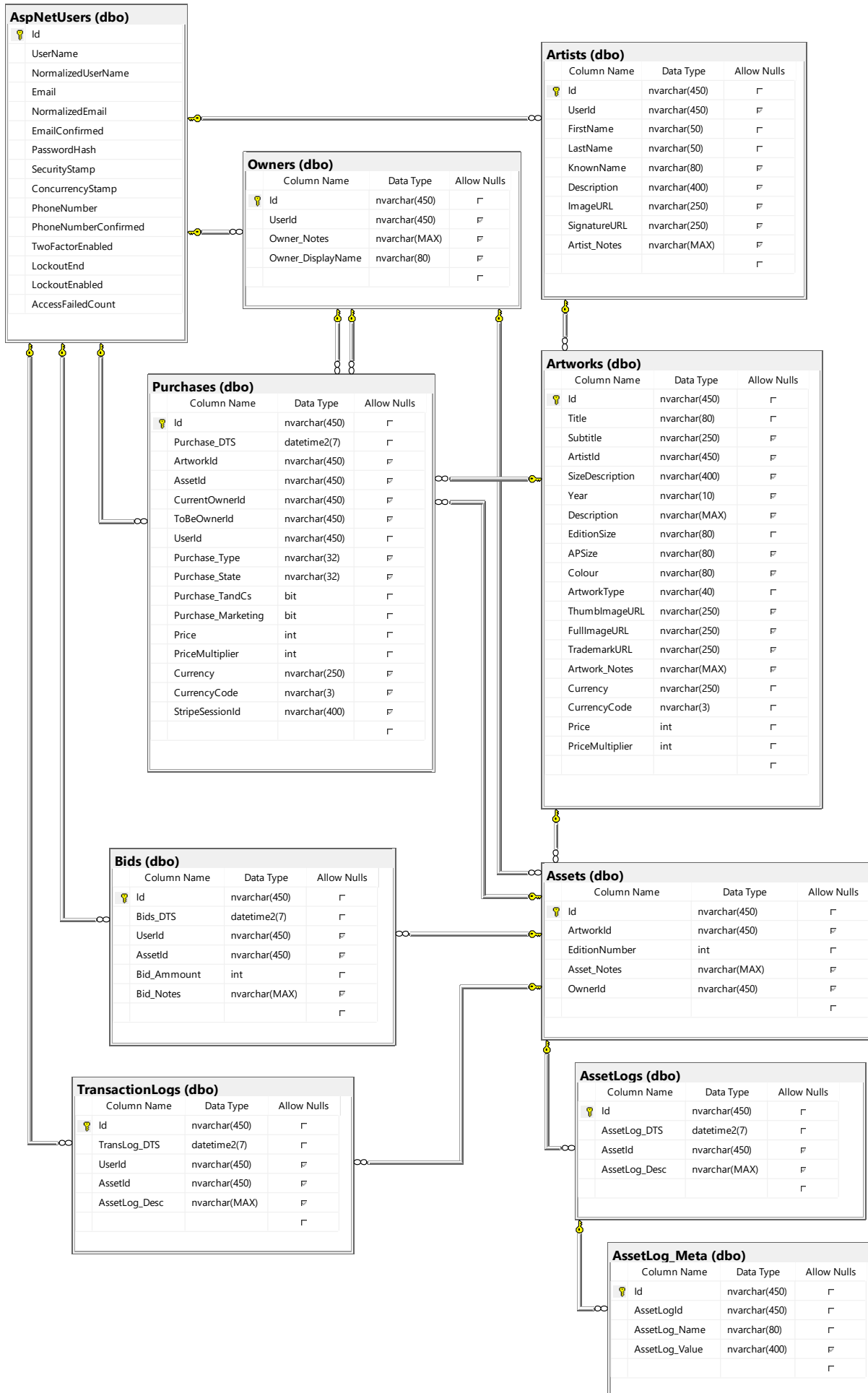


AspNetUserLogins and AspNetUserTokens tables are part of OWIN single sign-on, 3rd party support.
 AspNetRoleClaims is not used.

Reduced Users with data types

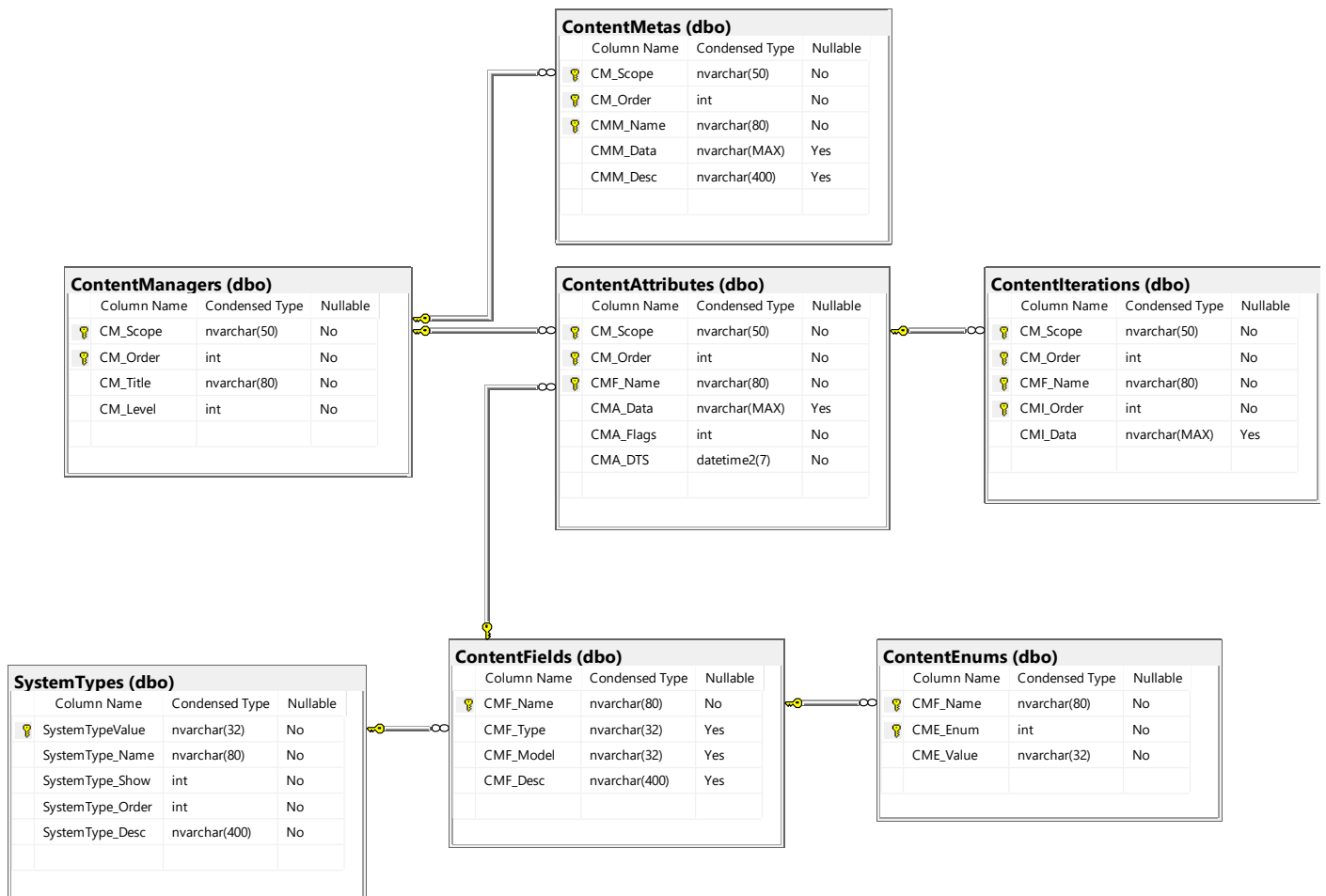


Artwork/Edition diagram, Prototype V1



Content Manager Entity diagram, Prototype V1

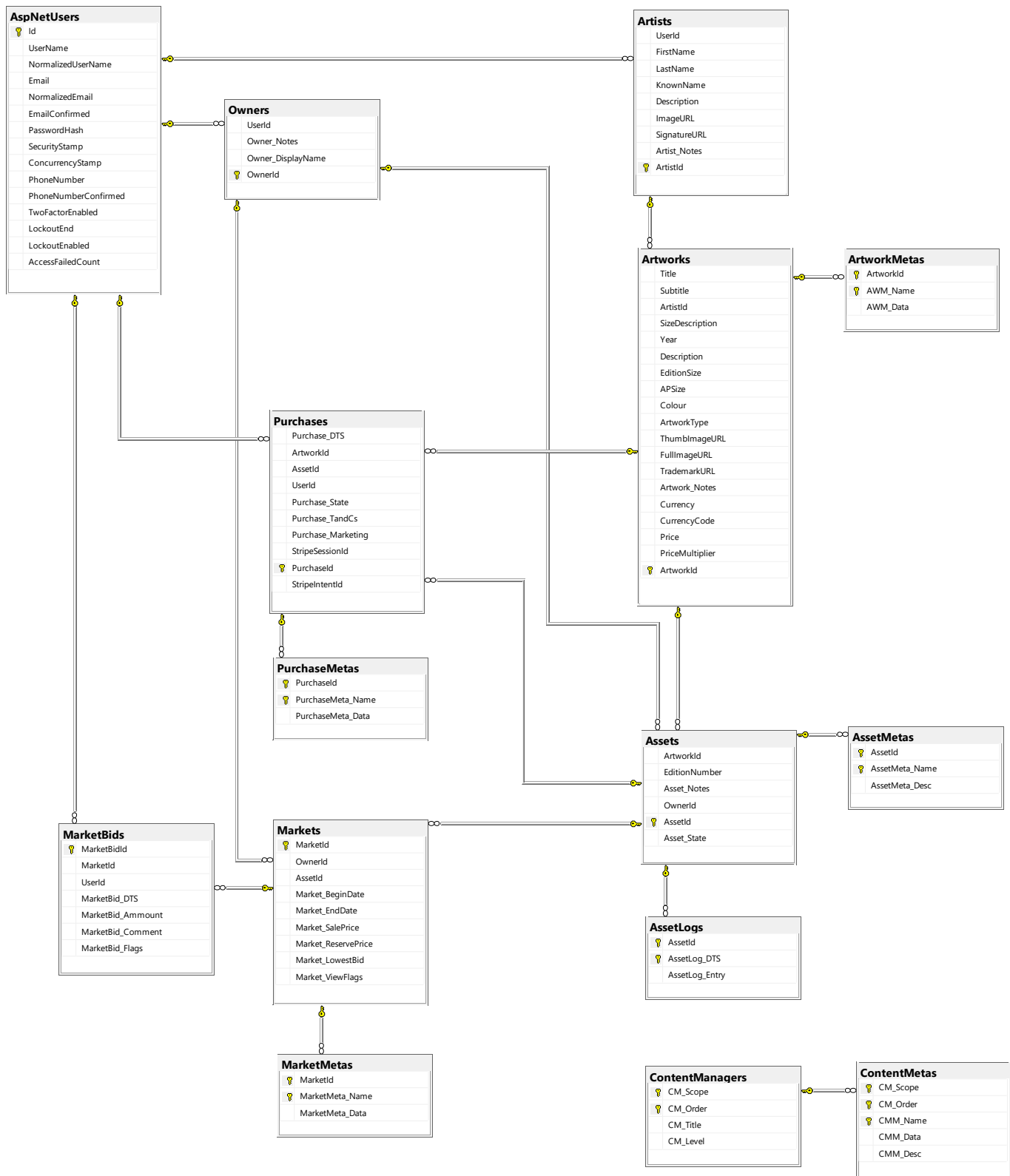
13 May 2020



System-Types table shown for completeness only.

Artwork/Edition diagram, Prototype V2

2 June 2020



Showing the Artwork, Asset, Market, Content and Purchase Entities.

New link not shown on diagram from Market-Bids table to Purchase table, one to many. Additional 'State' fields in Market and Market-Bids tables.

Field types in Entity Field Tables

System-Types Table Data Entries

Value	Name	Show	Description	Order
Text	Text	3	Normal text	1
MLText	Multiline Text	3	Normal text spread over a number of lines	2
HTMLText	HTML Text	2	HTML tagged text	3
XMLText	XML Text	1	XML encoded Text	4
JSON	JSON Text	2	JSON encoded text (JavaScript Object Notation)	5
Integer	Integer	3	Integer number (32bit)	6
Number	Decimal Number	3	Floating point decimal (32bit)	7
Bool	Boolean	1	true / false	8
DTS	Date-Time Stamp	2	Date-time format 'dd-mmm-yyyy hh:mm:ss' with month abbreviation in English	9
Date	Date (dd-mmm-yyyy)	3	Date-time format 'dd-mmm-yyyy' with English month abbreviation	10
Time	Time (hh:mm(:ss))	3	Date-time format 'hh:mm:ss' with or without seconds (date 01-01-0000)	11
URL	URL Address	3	URL or URI Address	12
Image	Image URL	2	URL address for an Image	13
UUID	UUID	1	Universal Unique Identifier or Global Unique Identifier (guid)	14
Static	Static Text	1	Static text, not editable except by Control Admins	15
BLOB	B-L-Ob	1	Binary Large Object; text, binary, compound or complex	16
Other	Other	0	Other, type specified elsewhere	0
Unknown	Unknown	0	Unknown or obsolete types	-1