System Requirements

System requirements express characteristics and functions of a system designed to meet the needs of YUL as expressed in interviews and existing documentation. Where applicable, requirements were derived directly from expressed stakeholder requirements in an effort to granularly define the needs of the users as it relates to a system and its functions. Priorities have been applied with a phased approach to implementation in mind. Short term priorities are assumed to be urgent and should be addressed in 1-2 years, medium term in 3-4 years, and long term in 5+ years.

Pre-ingest

#	Requirement	Dependencies	Priority
SYS-001	Provide mechanisms for configuration of metadata exchange between preservation system and pre-ingest curation tools		Short
SYS-002	Extract data from external systems (e.g. Ladybird, Archivists Toolkit) for submission to preservation environment	SYS-001	Immediate
SYS-003	Identify relevant metadata in external systems for addition to submission package via universal content identifier of content items	SYS-001	Short
SYS-004	Write external and user-supplied metadata to flat file for storage in preservation environment	SYS-001, SYS007, SYS-008	Short
SYS-005	Support standard protocols for packaging content items and metadata (e.g. BagIt)		Immediate
SYS-006	Provide mechanisms for configuration of required metadata fields for submission of files to preservation system		Short
SYS-008	Restrict submission of packages if required metadata fields incomplete	SYS-006	Medium
SYS-009	Prompt users to complete required metadata fields when incomplete	SYS-008	Medium
SYS-010	Provide mechanisms for navigating local file system to select content for submission	SYS-014	Short
SYS-011	Generate checksums for package contents and package if none exist		Short
SYS-014	Provide desktop or web client for managing package creation and creation/collection of metadata		Medium
SYS-015	Provide user interface to examine status of packages in preparation for submission		Medium

Ingest

#	Requirement	Dependencies	Priority
SYS-012	Actively monitor drop folder(s) for submission of individual files or content packages		Short
SYS-017	Verify submitted packages are transmitted from trusted submission sites/networks (e.g. via IP whitelisting or similar mechanism)	SYS-112	Short
SYS-018	Queue submitted packages for ingest processing ordered by date/time of submission or other configured specification	SYS-071	Short
SYS-022	Validate presence of required package contents against configured package specification	SYS-021	Short
SYS-023	Parse metadata from submission package and index in system database		Short
SYS-024	Validate package against submitted checksum values	SYS-023	Immediate
SYS-025	Generate checksums for data objects if absent		Immediate
SYS-031	Reject package if one of following is true: - incomplete package contents - failure of checksum validation	SYS-022, SYS024, SYS-030	Short
SYS-032	Retain failed submission packages in staging storage area until submission of valid package	SYS-031	Medium
SYS-033	Delete failed submission packages from staging storage area upon successful ingest of valid package	SYS-032	Short
SYS-028	Assign unique identifiers to all digital objects		Immediate
SYS-036	Assign archival package class to content object	SYS-021	Short
SYS-038	Generate archival package from submitted data elements and objects created during ingest	SYS-021	Immediate
SYS-039	Retain original file hierarchies as submitted		Short
SYS-040	Retain original file names of data objects as submitted		Short

SYS-041	Transfer archival package to preservation storage location upon completion of ingest processes		Immediate

Data management

#	Requirement	Dependencies	Priority
SYS-044	Support automatic or manual generation of archival package relationships based on submitted metadata		Short
SYS-029	Support for an in-system technical registry or integration with third-party registries and tools (e.g. PRONOM, DROID, JHOVE, SCOUT, etc.)		Medium
	Provide mechanisms for manual and automated update of format characteristics and documentation (e.g. bulk update of dependent computing environment for file **SYS-131** format)	SYS-029	Medium
SYS-030	Support ad-hoc and/or regularly scheduled format validation and characterization processes on objects managed by system	SYS-029	Medium
SYS-127	Generate software and hardware relationships between objects and technology records based on results of file characterization and validation	SYS-130	Medium
SYS-130	Support creation of non-functional metaobject records for documentation of software/hardware, as well as related object dependencies		Medium
SYS-064	Support manual and/or automated update of obsolescence risk register through updates to technical registry and integrated technology watch tools	SYS-029	Medium
SYS-048	Support ad-hoc, manual deletion of data objects from system according to configured deletion processes	SYS-046	Medium
SYS-049	Retain soft-deleted objects for configured retention period before complete removal from storage	SYS-046	Medium
SYS-050	Provide mechanism for restoration of softdeleted objects by admin users		Medium

SYS-051	Represent preservation storage as navigable file system in sytem interface		Long
SYS-052	Retrieve updated metadata from external systems and append to or update archival packages	SYS-001	Medium
SYS-054	Remove old versions of static metadata objects in archival packages upon receipt of updated metadata	SYS-052	Long
SYS-056	Perform ongoing automatic checksum validation on files in preservation environment according to configured schedule or ad-hoc manual validation		Short
SYS-062	Perform checksum validation following any transfer of data out of preservation system environment		Medium
SYS-060	Remove data object with checksum failure from primary storage upon review and approval by admin	SYS-046	Immediate
SYS-063	Support bulk export of data to external media in the event of a withdrawal of data from the preservation system		Immediate
SYS-070	Built-in transcoding capabilities or integration with transcode farm service		Long
SYS-072	Provide mechanisms for ad-hoc configuration of transcoding targets via system or independent tool/service interface	SYS-070	Long
SYS-099	Support ad-hoc configuration and embedding of metadata into files during transcode (e.g., IPTC metadata, custom metadata in file headers)		Long
SYS-100	Allow users to define, save, and apply embedded metadata specifications (e.g. templates)	SYS-099	Long

Migration

#	Requirement	Dependencies	Priority
SYS-069	Support bulk export of data to a temporary network share for content migration purposes		Medium

SYS-074	Automatic packaging and submission of migrated data objects and metadata for ingest		Medium
SYS-075	Retain original objects in preservation environment as "parent" version of migrated content	SYS-074	Short
SYS-076	Remove obsolete object versions (not original version, see SYS-075) upon successful ingest of migrated content	SYS-074	Long

Emulation

#	Requirement	Dependencies	Priority
SYS-078	Support integration with emulation services or applications		Long
SYS-079	Able to launch emulated computing environment directly from preservation system interface	SYS-078	Long
SYS-080	Seamless transition from preservation environment into emulation of software or content display in original computing environment	SYS-079	Long
SYS-082	Automatic ingest of created or altered objects as new versions or new packages into preservation system	SYS-081	Long

Metadata

#	Requirement	Dependencies	Priority
SYS-083	Support the implementation and management of custom data models in system database		Short
SYS-084	Out-of-box support for standard metadata schema in system database (e.g. PREMIS, DublinCore, METS, MODS, etc.)		Short
SYS-086	Provide mechanism to append/replace metadata into existing flat files in storage upon addition or edit of data in system database	SYS-128	Immediate
SYS-087	Able to export metadata from system database in multiple formats (e.g. XML, CSV, etc.)		Immediate

SYS-088	Provide cataloging interface and mechanisms for manual input of metadata into system database		Medium
SYS-132	Support comprehensive logging of all activities and actions within the system as preservation metadata, including ingest, deletion of objects and metadata, checksum validation, backup and replacement of digital objects, transcoding, downloads, and all other configurable activities		Immediate
SYS-089	Log identity of users with associated actions as preservation metadata in system database		Short
SYS-123	Out-of-box support for ad-hoc and automated metadata extraction and generation or support for integration with comparable tools		Short
SYS-128	Automatically index and store extracted metadata in system database	SYS-123	Short
SYS-045	Provide visual representation of package relationships in system interface		Long
SYS-077	Document version deltas in system database upon replacement or submission of new version of any replaced objects or elements	SYS-076	Medium
SYS-129	Support bulk update of metadata fields for multiple data object records at once		Immediate
SYS-134	Support navigation of preservation storage in system interface through browsing of faceted metadata fields		Short

Reporting

#	Requirement	Dependencies	Priority
SYS-090	Support querying of all fields in system database and/or search interface	SYS-084	Short
SYS-092	Allow users to save queries for future use	SYS-090	Long
SYS-093	Allow users to designate queries as available for all users or private to user who created	SYS-092	Long
SYS-094	Display results of queries in system interface	SYS-090	Short

SYS-095	Export query results as reports in various file formats (e.g. PDF, csv, XML, etc.)	SYS-090	Short
SYS-125	Support configuration of report templates with customizable layouts, reporting metrics, and creation schedules (e.g. monthly reports on format distribution, obsolescence monitoring, and other preservation activities)		Medium
SYS-096	Provide user-configured visualizations of system data (e.g. growth of storage size over six month period, # of uploads by week/month/year) in system interface (e.g. dashboard display)		Long
SYS-097	Alow customization of individual users' interface display of data visualizations	SYS-096	Long
SYS-122	Provide mechanisms for configuration of automated notifications and notification language at all points of data management workflows, including failed submission packages, requests to delete, obsolescence warnings, checksum failures, etc.	SYS-031, SYS-046	Immediate

Access

#	Requirement	Dependencies	Priority
SYS-098	Support download of preservation objects and/or newly transcoded derivatives upon request	SYS-070	Immediate
SYS-101	Provide mechanisms for automated, verified distribution of access copies from preservation system to access platforms		Medium
SYS-119	Automatically retrieve and cache newly submitted and transcoded access copies in low-latency storage upon request from external access system or user		Medium
SYS-098	Support download of preservation objects and/or newly transcoded derivatives upon request	SYS-070	Immediate

Security

#	Requirement	Dependencies	Priority

SYS-104	Enforce restrictions on actions and system functions based on assigned user role/group and permissions	SYS-103	Immediate
SYS-106	Provide mechanism for user registration and account generation or integrate with existing authentication service		Immediate
SYS-107	Apply configured baseline permissions specification upon creation of new user account	SYS-103	Immediate
SYS-108	Locally encrypt data objects designated for hightened security using standard encryption protocols		Short
SYS-109	Provide mechanisms for automated or manual management of local encryption keys		Short
SYS-110	De-encrypt files upon retrieval from secure storage locations for delivery or transcoding		Short
SYS-111	Support secure transfer protocols for submission of packages across network connections (e.g. SSL, SSH)		Short

Bit Preservation

#	Requirement	Dependencies	Priority
SYS-113	Automatically replicate ingested content to additional storage media as specified by associated content class preservation policy (e.g. single or multiple replicated copies, geographic dispersal, multiple storage media types, etc.)	SYS-133	Short
SYS-115	Replicate all operating software and databases to secondary disk and offsite data tape storage according to configured timetable		Short
SYS-116	Provide seamless failover to secondary storage in case of primary system failure	SYS-113	Short
SYS-135	Communicate between system and storage layers to exchange metadata (e.g. storage location of data objects and redundant copies, results of executed bit preservation processes)		Short

Admin

#	Requirement	Dependencies	Priority
SYS-021	Support configuration of (submission and archival) package profiles/ classes and derivation relationships between them		Short
SYS-046	Support configuration of workflows for all system processes with defined event triggers, review and approval requirements, and notification distribution		Short
SYS-053	Provide mechanism for selection of target metadata fields in external systems for extraction	SYS-001	Long
SYS-071	Provide queue in system interface for managing and scheduling processing jobs, including ingest of packages, transcoding, migration, etc.		Short
SYS-085	Support configuration of data capture at multiple points in system workflows	SYS-083, SYS-084	Medium
SYS-103	Support configuration of user permissions for system interaction and capabilities based on roles/groups (e.g. read only access, read/write, etc.)		Immediate
SYS-105	Provide admin interface for configuration of user accounts and assignation of permission levels to individual user accounts or groups	SYS-103	Short
SYS-112	Support configuration and management of whitelisted submission sites and systems		Medium
SYS-117	Support configuration of schedules for verification of all copies of data against stored checksums	SYS-113	Short
SYS-118	Support configuration of retention schedules for deletion of data objects, soft-deleted objects, and cached access copies	SYS-046	Medium
SYS-121	Provide mechanisms for managing scalable computing resources for all processes performed by system		Medium
SYS-133	Support configuration of hierarchical storage policies and procedures for various content types and preservation levels		Short

Use Cases

UC-1 Ingest

Scenario: Packages of digital assets are uploaded to the System for ingest into the preservation environment. System validates the package contents and completes the multi-step ingest process. The components of the package are used to generate a new archival package and the package contents are placed in their designated storage environment locations according to policies set for each content class and file type.

Actors: User, Admin

Pre-conditions: Content class package requirements defined and configured within system; ingest workflow configured; checksums generated for each asset and stored in package metadata

Outcome: Objects and metadata placed within designated storage environment and indexed in system database

Steps:

#	User	System
1	User completes submission of package via pre-ingest curation tool	Receives notice of incoming package submission
2		Confirms approved submission location from whitelist
3		Initiates secure transfer of package to ingest staging area
4		Adds package to ingest processing queue
5	Admin places package at top of processing queue	Validates package contents against content class package specification
6		Parses package checksums from submitted metadata files
7		Verifies checksum of package and contents
8		Parses metadata from package
9		Indexes and adds metadata to system database
11		Generates archival package of submitted contents according to specified package class

12		Replicates two copies of package to two offsite disk storage centers according to assigned storage policy
13		Validates checksums of package copies upon transfer to offsite storage
14		Logs all steps of ingest process as events with event ID, event type, event date/time, event outcome, and agent responsible in system database
15		Writes ingest and archival packaging process metadata to flat file for storage

Exception Flow:

#	User	System
1		Validates package contents against content class package specification
2		Recognizes that package is missing required metadata object
3		Cancels ingest
4		Marks package as pending review in ingest queue
5		Transfers failed package to temporary storage environment
6		Sends notification of failed submission to Admin and User
7		Logs failed ingest event and details in database
8	User repackages object with all required elements	See steps 1-3 above
9		Recognizes new version of package via content identifier submitted with package
10		See steps 4-13 above
11		Removes failed package from temporary storage and permanently deletes

Associated Requirements

- SYS-012. Actively monitor drop folder(s) for submission of individual files or content packages

- SYS-017. Verify submitted packages are transmitted from trusted submission sites/networks
- SYS-018. Queue submitted packages for ingest processing ordered by date/time of submission
- SYS-022. Validate presence of required package contents against configured package specification
- SYS-023. Parse metadata from submission package and index in system database
- SYS-024. Validate package against submitted checksum values
- SYS-025. Generate checksums for data objects if absent
- SYS-028. Assign unique identifiers to all digital objects
- SYS-031. Reject package if one of following is true:
    - o incomplete package contents
    - o failure of checksum validation
- SYS-032. Retain failed submission packages in staging storage area until submission of valid package
- SYS-033. Delete failed submission packages from staging storage area upon successful ingest of valid package
- SYS-035. Log all ingest activity as preservation metadata in system database and static file in standard schema (e.g. PREMIS)
- SYS-036. Assign archival package class to content object
- SYS-038. Generate archival package from submitted data elements and objects created during ingest
- SYS-039. Retain original file hierarchies as submitted
- SYS-040. Retain original file names of submitted files
- SYS-041. Transfer archival package to preservation storage location upon completion of ingest processes
- SYS-042. Discard elements from submission package not needed in archival package
- SYS-046. Support configuration of workflows for all system processes with defined event triggers, review and approval requirements, and notification distribution
- SYS-062. Perform checksum validation following any transfer of data out of preservation system environment
- SYS-089. Log actions of users as preservation metadata in system database
- SYS-113. Automatically replicate ingested content to additional storage media as specified by associated content class preservation policy (e.g. single or multiple replicated copies, geographic dispersal, multiple storage media types, etc.)
- SYS-132. Support comprehensive logging of all activities and actions within the system as preservation metadata, including ingest, deletion of objects and metadata, checksum validation, backup and replacement of digital objects, transcoding, downloads, and all other configurable activities

- SYS-135. Communicate between system and storage layers to exchange metadata (e.g. storage location of data objects and redundant copies, results of executed bit preservation processes)

UC-2 Deletion

Scenario: User selects data object from system interface and chooses to delete. System notifies Admin of User request to delete. Admin approves delete request. System restricts access to and display of object in interface. Soft-deleted object is retained in storage for 30 days and then removed permanently.

Actors: User, Admin

Pre-conditions: Workflow for notifications and processing of deletion requests configured; retention schedule for deleted files configured; protocols for secure removal of data from storage media

Outcome: Digital object removed from preservation environment

Steps:

#	User	System
1	User selects data object in system interface	
2	User chooses to delete object using button in interface or pressing delete on keyboard	Visual representation of object is faded in interface to notate pending deletion
3		Notifies Admin of deletion request
4		Prompts Admin to make decision regarding retention or permanent deletion of data object
5	Admin approves deletion request	Access to object record is restricted in system interface
6		Retains object in storage for 30 days
7		Object is permanently removed from storage
8		Logs all steps of deletion process as events with event ID, event type, event date/time, event outcome, and agent responsible in system database

Alternate Flow:

#	User	System

1		See steps 1-5 above
2	User requests object be retrieved from “soft-delete” status	
3	Admin prompts system to retrieve object from temporary “soft-delete” storage	Returns object to permanent storage
4		Object record is made accessible in system interface
5		Logs all steps of retrieval process as events with event ID, event type, event date/time, event outcome, and agent responsible in system database

Associated Requirements

- SYS-046. Support configuration of workflows for all system processes with defined event triggers, review and approval requirements, and notification distribution
- SYS-048. Support ad-hoc, manual deletion of data objects from system according to configured deletion processes
- SYS-049. Retain soft-deleted objects for configured retention period before complete removal from storage
- SYS-050. Provide mechanism for restoration of soft-deleted objects by admin users
- SYS-089. Log actions of users as preservation metadata in system database
- SYS-118. Support configuration of retention schedules for deletion of data objects, soft-deleted objects, and cached access copies
- SYS-132. Support comprehensive logging of all activities and actions within the system as preservation metadata, including ingest, deletion of objects and metadata, checksum validation, backup and replacement of digital objects, transcoding, downloads, and all other configurable activities

UC-3 Successful Checksum Validation

Scenario: System completes validation of group of archival packages belonging to User. System sends notification of successful checksum validations to User.

Actors: User

Pre-conditions: Checksums generated and stored for each asset; Frequency and rate of integrity monitoring defined and monitoring programmed into System; System configured to send validation reports/notifications

Outcome: User receives notice or report of checksum validation from System

Steps:

#	User	System
1		Generates new checksum for files according to schedule
2		Parses checksums from stored file metadata and runs diff check
3		Logs validation event, result, and date
4		Generates report indicating successful validation and sends it to User

Associated Requirements

- SYS-046. Support configuration of workflows for all system processes with defined event triggers, review and approval requirements, and notification distribution
- SYS-056. Perform checksum validation on all files in preservation environment according to configured schedule
- SYS-089. Log actions of users as preservation metadata in system database
- SYS-122. Provide mechanisms for configuration of automated notifications and notification language at all points of data management workflows, including failed submission packages, requests to delete, obsolescence warnings, checksum failures, etc.
- SYS-132. Support comprehensive logging of all activities and actions within the system as preservation metadata, including ingest, deletion of objects and metadata, checksum validation, backup and replacement of digital objects, transcoding, downloads, and all other configurable activities

UC-4 Checksum Failure

Scenario: System performs automated file fixity check via checksum verification. Verification of file fails. Admin approves replacement of corrupted file from backup. System completes replacement of corrupt file and notifies Admin.

Actors: User, Admin

Pre-conditions: Periodic checksum verification scheduled and configured; automated reports configured and scheduled; checksums generated for each asset and stored in package

Outcome: Valid copy of asset in primary storage; verification and replacement event logged by system

Steps:

#	User	System
1		Checksum validation of file fails
2		Alerts Admin and User to fixity check mismatch
3	Admin receives fixity check mismatch notification	System prompts Admin to replace corrupt file with back-up copy of file from secondary storage
4	Admin approves System request to replace corrupt file with back-up copy	Pulls back-up copy from secondary storage
5		Successfully verifies checksum of backup
8		Permanently deletes corrupt file from primary storage
6		Writes back-up copy to archival package on primary storage
9		Sends report to Admin and User indicating corrupt file has been replaced with verified copy
		Logs all steps of validation and replacement process as events with
		event ID, event type, event date/time, event outcome, and agent responsible in system database

Associated Requirements

- SYS-046. Support configuration of workflows for all system processes with defined event triggers, review and approval requirements, and notification distribution
- SYS-056. Perform checksum validation on all files in preservation environment according to configured schedule
- SYS-060. Remove data object with checksum failure from primary storage upon review and approval by admin
- SYS-062. Perform checksum validation following any transfer of data out of preservation system environment
- SYS-089. Log actions of users as preservation metadata in system database
- SYS-122. Provide mechanisms for configuration of automated notifications and notification language at all points of data management workflows, including failed submission packages, requests to delete, obsolescence warnings, checksum failures, etc.
- SYS-132. Support comprehensive logging of all activities and actions within the system as preservation metadata, including ingest, deletion of objects and

metadata, checksum validation, backup and replacement of digital objects, transcoding, downloads, and all other configurable activities

UC-5 Transcoding/File Caching

Scenario: User requires a new derivative of a file in the preservation system. User defines transcoding target and embedded metadata specifications. System transcodes and makes derivative available for download. System retains copy of derivative in low latency cache storage for 30 days. User requires a second copy of derivative and retrieves from system without transcoding new copy.

Actors: User

Pre-conditions: User permissions configured and assigned transcoding privileges; derivative/access cache retention schedule defined; protocols for secure removal of data from storage media

Outcome: User receives multiple copies of derivative file. File deleted after 30 days of no further requests for download

Steps:

#	User	System
1	User selects object for transcoding	Launches transcoding configuration module
2	User defines transcoding specifications in module	
3	User selects embedded metadata template from list	Transcode enters queue of processing jobs in system interface
4	Admin receives request from User for immediate processing	Transcode enters queue of processing jobs in system interface
5	Admin moves ingest job to top of queue	Retrieves master or derivative master from primary storage
6		Validates checksum of retrieved object
7		Generates derivative copy according to specifications defined by User
8		Stores derivative copy in low-latency cache storage
9		Sends notification of job completion to User
10	User downloads file to desktop	

15 days pass		
11	User requests another copy of derivative to download	Checks low-latency cache storage for derivative copy of file
12		Sends notification that file is available for download
13	User downloads file to desktop	
30 days pass		
14		Checks logs of cache for files over 30 days in storage
15		Permanently removes derivative file from storage
16		Logs all steps of process as events with event ID, event type, event date/time, event outcome, and agent responsible in system database

Associated Requirements

- SYS-070. Built-in transcoding capabilities or integration with transcode farm service
- SYS-071. Provide queue in system interface for managing and scheduling processing jobs, including ingest of packages, transcoding, migration, etc.
- SYS-072. Provide mechanisms for ad-hoc configuration of transcoding targets via system or independent tool/service interface
- SYS-073. Generate new checksums for any transcoded derivative to be retained in preservation storage
- SYS-089. Log actions of users as preservation metadata in system database
- SYS-099. Support ad-hoc configuration and embedding of metadata into files during transcode (e.g., IPTC metadata, custom metadata in file headers)
- SYS-100. Allow users to define, save, and apply embedded metadata templates
- SYS-118. Support configuration of retention schedules for deletion of data objects, soft-deleted objects, and cached access copies
- SYS-119. Automatically retrieve and cache newly submitted and transcoded access copies in low-latency storage upon request from external access system or user
- SYS-122. Provide mechanisms for configuration of automated notifications and notification language at all points of data management workflows, including failed submission packages, requests to delete, obsolescence warnings, checksum failures, etc.
- SYS-132. Support comprehensive logging of all activities and actions within the system as preservation metadata, including ingest, deletion of objects and

metadata, checksum validation, backup and replacement of digital objects, transcoding, downloads, and all other configurable activities

UC-6 Reporting

Scenario: User requires a manifest report of all items in storage, containing numerous fields related to the individual files. User defines report parameters within system. System outputs results.

Actors: User

Pre-conditions: Metadata indexed and able to be queried by system; standard reporting formats defined

Outcome: Report created and downloaded by User.

Steps:

#	User	System
1	User selects option to create new report	
2	User customizes report parameters to output the following related to collections owned by User: file name, file location, format, and file size [by SQL query or other report building mechanism]	
3	User saves report parameters for future use and designates as available to other users	
4	User selects option to generate report	Queries indexed metadata
8		Compiles results into desired output format
9		Prompts User to download report
10	User downloads report to local machine	

Associated Requirements

- SYS-090. Support querying of all fields in system database and/or search interface
- SYS-092. Allow users to save queries for future use
- SYS-093. Allow users to designate queries as available for all users or private to user who created
- SYS-094. Display results of queries in system interface

- SYS-095. Export query results as reports in various file formats (e.g. PDF, csv, XML, etc.)
- SYS-125. Support configuration of report templates with customizable layouts, reporting metrics, and creation schedules (e.g. monthly reports on format distribution, obsolescence monitoring, and other preservation activities)

UC-7 Content Migration

Scenario: A risk assessment indicates that preservation master files of a particular format are likely to be inaccessible soon. Admin identifies assets for migration to new format and defines transcoding parameters. System completes migration, ingests new version, retains old assets as parent version, and logs the migration event as preservation metadata.

Actors: Admin

Pre-conditions: File migration policies and workflows defined and configured in system; version control specifications defined within system; technical registry up-todate and actively monitoring for obsolescence triggers

Outcome: New asset versions ingested and placed in storage. Original versions of assets retained in storage

Steps:

#	User	System
1	Admin identifies target files for migration	
2	Admin defines file type, encoding, and file characteristics of migration	
3	Admin selects to begin migration of target assets	Locates target assets
4		Computes checksums for target assets
5		Parses checksums from stored file metadata and runs diff check against new value
6		Moves target assets to processing
		environment/service
7		Verifies checksum after delivery of the target assets into processing environment

8		Batch transcodes target assets to defined parameters
9		Generates checksum for new asset versions
10		Performs ingest for new asset versions as SIPs (See UC-2)
11		Versions resulting AIP, relating it to original AIP
12		Prompts Admin to approve retention or deletion of previous version
13	Admin selects to remove previous version from preservation environment	Soft deletes previous version (see UC-2) and records version deltas in system database
14		Defines new files preservation masters and applies settings from previous version to new version
15		Retains original preservation masters and marks them as inactive
16		Logs action as preservation event and includes event type, event ID, date/time, agent responsible, event outcome
17		Generates and sends migration report to Admin
18	Admin saves report to local machine	

Associated Requirements

- SYS-064. Support manual and/or automated update of obsolescence risk register through updates to technical registry and integrated technology watch tools
- SYS-069. Support bulk export of data to a temporary network share for content migration purposes
- SYS-070. Built-in transcoding capabilities or integration with transcode farm service
- SYS-071. Provide queue in system interface for managing and scheduling processing jobs, including ingest of packages, transcoding, migration, etc.
- SYS-072. Provide mechanisms for ad-hoc configuration of transcoding targets via system or independent tool/service interface
- SYS-073. Generate new checksums for any transcoded derivative to be retained in preservation storage
- SYS-074. Automatic packaging and submission of migrated data objects and metadata for ingest

- SYS-075. Retain original objects in preservation environment as "parent" version of migrated content
- SYS-076. Remove obsolete object versions (not original version, see SYS-075) upon successful ingest of migrated content
- SYS-077. Document version deltas in system database upon replacement or submission of new version of any replaced objects or elements
- SYS-089. Log actions of users as preservation metadata in system database
- SYS-122. Provide mechanisms for configuration of automated notifications and notification language at all points of data management workflows, including failed submission packages, requests to delete, obsolescence warnings, checksum failures, etc.
- SYS-132. Support comprehensive logging of all activities and actions within the system as preservation metadata, including ingest, deletion of objects and metadata, checksum validation, backup and replacement of digital objects, transcoding, downloads, and all other configurable activities

UC-8 Bulk Metadata Update

Scenario: Admin reviews new format profile indicating change is required in computing environment for specific type of file format. Admin conducts query for all files of this format and system returns results in interface. Admin selects all files returned by query and assigns new dependent computing environment.

Actors: Admin

Pre-conditions: System technical registry up-to-date; Data objects ingested and metadata indexed within database; Established computing environment dependencies applied to formats

Outcome: Files assigned up-to-date computing environment dependency via bulk metadata edit

Steps:

#	User	System
1	Admin updates technical registry according to recent changes in format profiles	
2	Admin generates query requesting all files of format .pdf version 1.7	Queries database for records of all data objects related to query
3		Returns display of all object records pertaining to .pdf version 1.7 files
4	Admin selects all results	

5	Admin chooses to edit records	
6	Admin selects computing environment field (or other editing mechanism)	
7	Admin changes field to reflect new dependency	Updates database to reflect change for all records selected from query results
8		Records edits as preservation metadata in system database

Associated Requirements

- SYS-029. Support for an in-system technical registry or integration with thirdparty registries and tools (e.g. PRONOM, DROID, JHOVE, SCOUT, etc.)
- SYS-088. Provide cataloging interface and mechanisms for manual input of metadata into system database
- SYS-089. Log actions of users as preservation metadata in system database
- SYS-090. Support querying of all fields in system database and/or search interface
- SYS-127. Generate software and hardware dependencies between objects and technology records based on results of file characterization and validation
- SYS-129. Support bulk update of metadata fields for multiple data object records at once
- SYS-130. Support creation of non-functional meta-object records for documentation of software/hardware, as well as related object dependencies
- SYS-131. Provide mechanisms for manual and automated update of format characteristics and documentation, as well as related object/technology dependencies (e.g. bulk update of dependent computing environment for file format)
- SYS-132. Support comprehensive logging of all activities and actions within the system as preservation metadata, including ingest, deletion of objects and metadata, checksum validation, backup and replacement of digital objects, transcoding, downloads, and all other configurable activities

UC-9 Processing Queue Management

Scenario: Admin queries system for data objects ingested within a recent time period. Query results are selected and scheduled for format characterization. A corresponding job is added to the processing. Admin moves characterization job to top of queue. System performs file characterization and updates metadata in system.

Actors: Admin

Pre-conditions: Data objects ingested within recent time period; format characterization processes and workflows configured in system

Outcome: Format characterization metadata added to system database

Steps:

#	User	System
1	Admin creates query for data objects	Queries database for records of all data
	ingested in system within past seven days	objects related to query
2		Returns display of all objects ingested within past seven days
3	Admin selects all results	
4	Admin schedules format characterization job for all query results	Adds characterization job to processing queue
5	Admin moves characterization job to the top of the queue	Completes current job before beginning characterization
6	Admin lowers processing resources used for job	Performs format characterization of packages to identify and validate how closely the files contained within packages match with prescribed formats and to identify probable formats used (where information was not provided in metadata)
7		Records format characterization metadata in system database
8		Begins next job in process queue

Associated Requirements

- SYS-030. Support ad-hoc and/or regularly scheduled format validation and characterization processes on objects managed by system
- SYS-071. Provide queue in system interface for managing and scheduling processing jobs, including ingest of packages, transcoding, migration, etc.
- SYS-085. Support configuration of data capture at multiple points in system workflows
- SYS-090. Support querying of all fields in system database and/or search interface
- SYS-094. Display results of queries in system interface
- SYS-121. Provide mechanisms for managing scalable computing resources for all processes performed by system
- SYS-127. Generate software and hardware dependencies between objects and technology records based on results of file characterization and validation
- SYS-131. Provide mechanisms for manual and automated update of format characteristics and documentation, as well as related object/technology

dependencies (e.g. bulk update of dependent computing environment for file format)

UC-10 Downloading objects

Scenario: Admin browses through objects in preservation system via faceted metadata fields. Admin selects subset of data objects from system interface for download. System transfers copies of preservation objects to cache storage and Admin downloads copies to local workstation.

Actors: Admin

Pre-conditions: Data objects ingested and characterized within system; metadata upto-date in system database; copies of preservation objects not already stored in cache storage

Outcome: Copies of preservation objects downloaded to local workstation

Steps:

#	User	System
1	Admin browses system contents via faceted metadata fields in interface	Narrows displayed contents based on Admin selections
2	Admin drills down to all objects of format type .pdf version 1.7 ingested within past 7 days	
3	Admin selects five objects for download	Locates objects in preservation storage
4		Computes checksums for objects
5		Parses checksums from stored file metadata and runs diff check against new value
6		Moves target assets to processing environment
7		Generates copies of preservation objects
8		Verifies checksums of new copies
9		Transfers copies to cache storage for delivery
10		Verifies checksums of new copies following transfer
11		Notifies Admin that copies are ready for download

12	Admin downloads copies to local computer for review	Logs all steps of process as events with event ID, event type, event date/time, event outcome, and agent responsible in system database

Alternate Scenario (API call for access copy delivery)

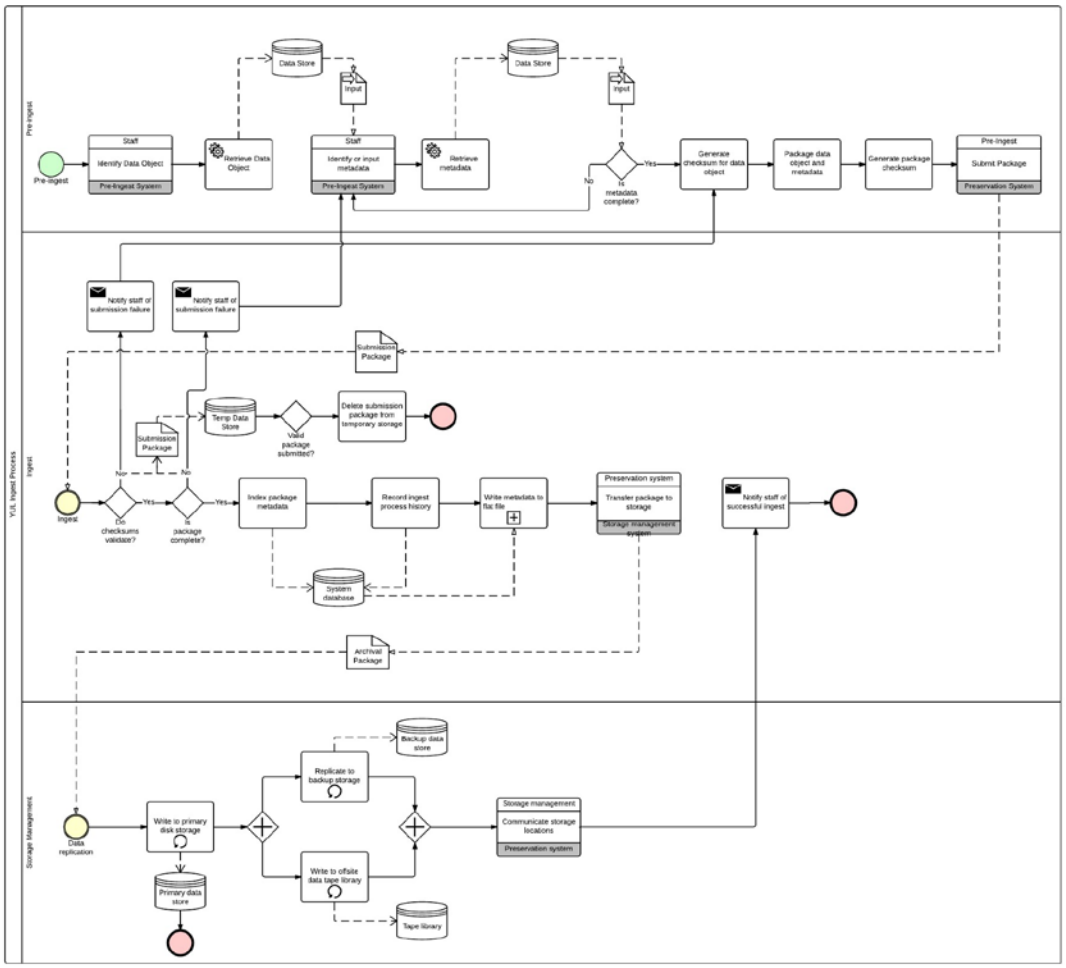
#	User	System
1		Receives API request for access copy from access platform
2		Checks low-latency cache storage for derivative copy of file
3		Does not find requested file in storage
4		Locates object in preservation storage
5		Computes checksum for object
6		Parses checksum from stored file metadata and runs diff check against new value
7		Moves target asset to processing environment
8		Generates derivative of access copy
9		Verifies checksum of new copies
10		Transfers copy to cache storage for delivery
11		Verifies checksums of new copy following transfer
12		Serves access copy to platform
13		Logs all steps of process as events with event ID, event type, event date/time, event outcome, and agent responsible in system database

Associated Requirements

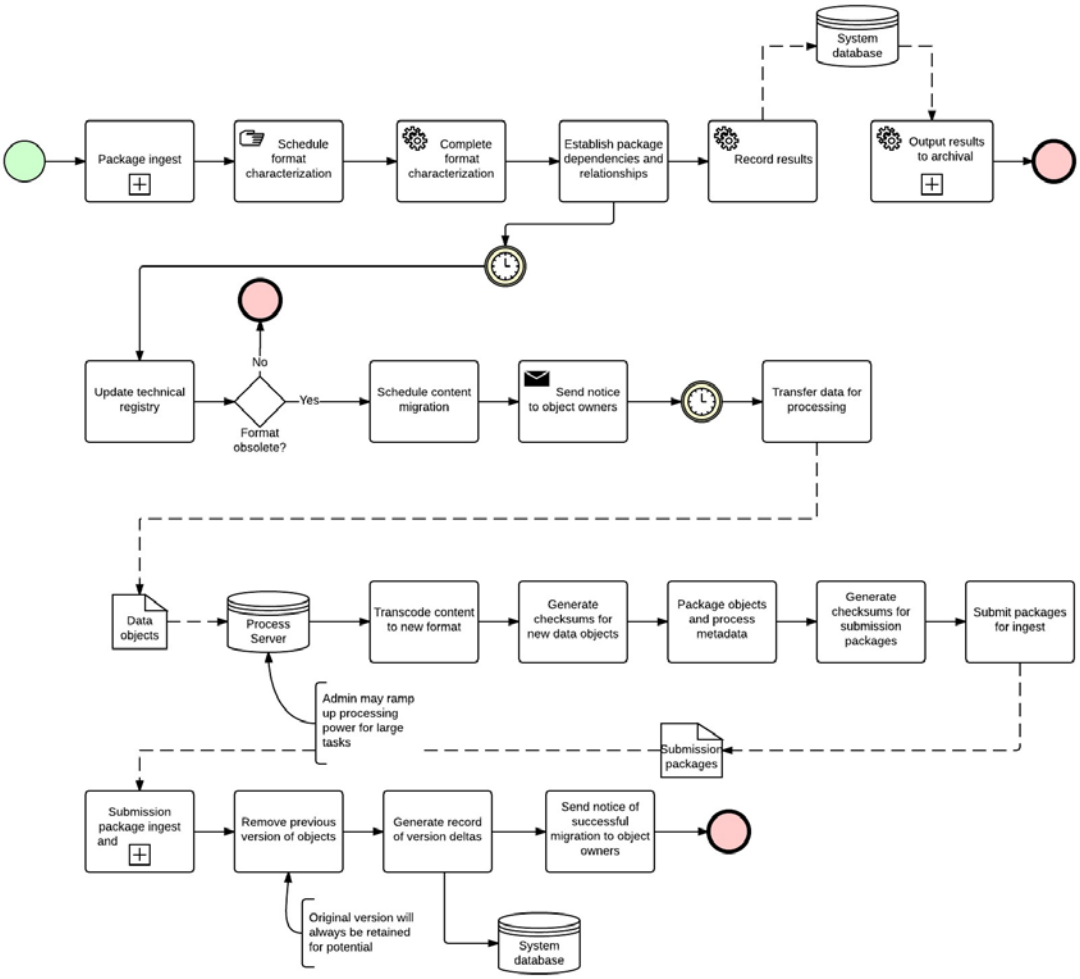
- SYS-046. Support configuration of workflows for all system processes with defined event triggers, review and approval requirements, and notification distribution
- SYS-062. Perform checksum validation following any transfer of data out of preservation system environment
- SYS-098. Support download of preservation objects and/or newly transcoded derivatives upon request
- SYS-101. Provide mechanisms for automated, verified distribution of access copies from preservation system to access platforms

- SYS-122. Provide mechanisms for configuration of automated notifications and notification language at all points of data management workflows, including failed submission packages, requests to delete, obsolescence warnings, checksum failures, etc.
- SYS-132. Support comprehensive logging of all activities and actions within the system as preservation metadata, including ingest, deletion of objects and metadata, checksum validation, backup and replacement of digital objects, transcoding, downloads, and all other configurable activities
- SYS-134. Support navigation of preservation storage in system interface through browsing of faceted metadata fields

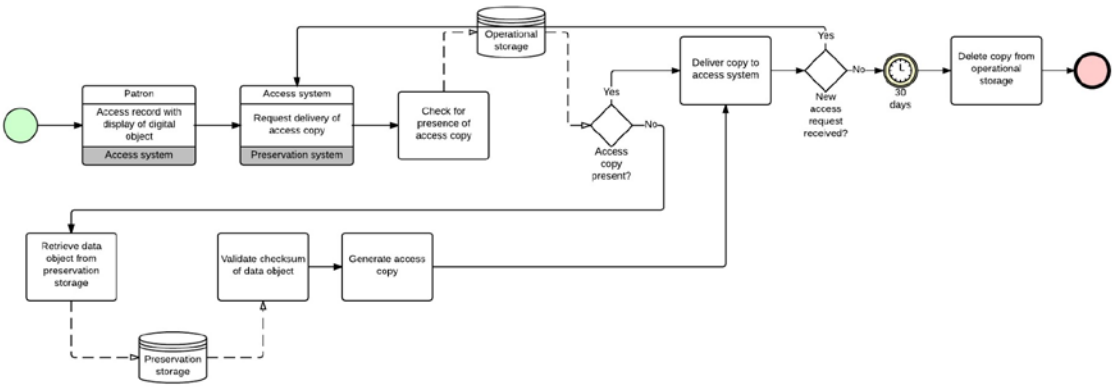
Appendix D – System Process Models



Ingest Process



Migration Process



Access process