Self Service

Functional Requirements

Operator notes:

All instructional text is displayed in blue font color in the template. These should be deleted prior to the document being distributed for review and approval.

Update text within square brackets which is placeholder text with relevant text.

E.g. [yyyymmdd] should be changed to latest date 20190305

Use this basic structure and type sizing for all your covers. Do not manipulate or rescale any of the graphic components. Use initial caps in your headings with capital letters for proper nouns only.

# Document Control

Document location

| Location |
| --- |
| Sharepoint: [link](https://adminliveunc.sharepoint.com/sites/its/itio/idm/Shared%20Documents/Course%20Groups/Initial%20Implementation%20for%20Teaching%20%26%20Learning/Requirements%20and%20Design/Course%20Groups%20for%20Office%20365%20-%20FRD.docx?d=w33506deafa05485ab0a14af28599e846&csf=1) to Self Service.FRD.docx from Sharepoint |

Author

| Name | Department | Position |
| --- | --- | --- |
| Brian Darley | Identity Management | Systems Programmer/Specialist |

Revision history

| Version | Issue date | Author/editor | Description/Summary of changes |
| --- | --- | --- | --- |
| draft 1 | 3/21/2019 | Brian Darley | Initial draft |
|  |  |  |  |

Reviewed by

| Version | Issue date | Name | Position | Review date |
| --- | --- | --- | --- | --- |
| 1.0 | mm/dd/yyyy | Reviewer Name | Reviewer Position | mm/dd/yyyy |
|  |  |  |  |  |

Approvals

Approval refers to the approver’s acceptance of the content and overall intention of this document, including acceptance of any commitments described in order to successfully deliver the initiative. The approver, where relevant, also confirms that this document complies with relevant strategies, policies and regulatory requirements.

| Version | Issue date | Name | Position | Approval date |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |

Related documents

| Document | Location |
| --- | --- |
|  |  |

# Table of Contents

[1 INTRODUCTION 6](#_Toc494184294)

[1.1. Purpose 6](#_Toc494184295)

[1.2. Scope of Work 6](#_Toc494184296)

[1.3. Out of Scope 6](#_Toc494184297)

[1.4. As-Is State 6](#_Toc494184298)

[1.5. Contributing Stakeholders 6](#_Toc494184299)

[1.6. References 7](#_Toc494184300)

[1.7. Assumptions and Constraints 7](#_Toc494184301)

[1.7.1. Assumptions 7](#_Toc494184302)

[1.7.2. Constraints 7](#_Toc494184303)

[1.7.3. Dependencies 8](#_Toc494184304)

[1.7.4. Business Rules 8](#_Toc494184305)

[2 Use Cases 8](#_Toc494184306)

[2.1. Authentication/Authorization 8](#_Toc494184307)

[2.2. Client/API Relationship 9](#_Toc494184308)

[2.3. Api to Api Relationship 10](#_Toc494184309)

[3 FUNCTIONAL REQUIREMENTS 11](#_Toc494184311)

[3.1. Context 11](#_Toc494184312)

[3.2. User Requirements 12](#_Toc494184313)

[3.3. Data Flow Diagrams 12](#_Toc494184314)

[3.4. Logical Data Model/Data Dictionary 13](#_Toc494184315)

[3.4.1. Data import from PeopleSoft into Grouper 13](#_Toc494184316)

[3.4.2. Role Mapping from PeopleSoft into Grouper/Office 365 Roles 14](#_Toc494184317)

[3.4.3. Tentative Course Group Structure in Grouper 14](#_Toc494184318)

[3.5. Interface Requirements 14](#_Toc494184319)

[3.5.1. Group 1 - User Interface Authentication/Authorization 15](#_Toc494184320)

[3.5.2. Group 2 - Create New Group 15](#_Toc494184321)

[3.5.3. Group 3 - Manage Existing Group 16](#_Toc494184322)

[3.6. Data Requirements 16](#_Toc494184325)

[3.6.1. Data Synchronization between enrollment system and Grouper 16](#_Toc494184387)

[3.6.2. Data Synchronization between Grouper and Office 365 17](#_Toc494184388)

[4 OTHER REQUIREMENTS 18](#_Toc494184389)

[4.1. Operational Requirements 18](#_Toc494184390)

[4.1.1. Data Requirements 18](#_Toc494184391)

[4.1.2. Security and Privacy 18](#_Toc494184392)

[4.1.3. Audit Trail 18](#_Toc494184393)

[4.1.4. Non- 18](#_Toc494184394)

[4.1.5. Reliability 18](#_Toc494184395)

[4.1.6. Recoverability 19](#_Toc494184396)

[4.1.7. System Availability 19](#_Toc494184397)

[4.1.8. General Performance 19](#_Toc494184398)

[4.1.9. Capacity 20](#_Toc494184399)

[4.1.10. Data Retention 20](#_Toc494184400)

[4.1.11. Error Handling 20](#_Toc494184401)

[4.1.12. Validation Rules 20](#_Toc494184402)

[4.1.13. Conventions/Standards 20](#_Toc494184403)

[5 APPENDIX A - GLOSSARY 22](#_Toc494184404)

[6 APPENDIX B - UI Wireframes 22](#_Toc494184405)

[6.1. Office 365 Groups main screen 22](#_Toc494184406)

[6.2. Course Groups main screen 23](#_Toc494184407)

[6.3. Create new class group screen 24](#_Toc494184408)

[6.4. Create New Class Group - Confirmation 25](#_Toc494184409)

[6.5. Manage existing group 25](#_Toc494184410)

[7 Appendix C: Data Metrics from PeopleSoft 27](#_Toc494184411)

[8 Recent Decisions/Answers Based on Questions or Discussion 27](#_Toc494184412)

[8.1. When a course group creator is no longer an instructor for a group/no longer at UNC? 27](#_Toc494184413)

[8.2. Can groups be deleted in O365? What happens to them when the next sync happens? 28](#_Toc494184414)

[8.3. Who are updaters of Grouper includes/exclude groups, and readers of composite groups? 28](#_Toc494184415)

[8.4. See FR3.3 - When listing all the sections available to add/remove for an existing group, what if an instructor was originally assigned to a section but no longer is? 28](#_Toc494184416)

[8.5. FERPA Issues 28](#_Toc494184417)

[8.6. Where should the application be hosted -- Self Service or Improv? 28](#_Toc494184418)

[8.7. Create groups for future terms? 28](#_Toc494184419)

[9 Appendix D: Decisions/Questions in Progress 28](#_Toc494184420)

[9.1. Can any group owner manage an existing group, or just the original person who created the group? 28](#_Toc494184421)

[9.2. How often will courses and sections be removed from PeopleSoft? Should they be removed in Grouper when that happens, or left for historical purposes? 28](#_Toc494184422)

[9.3. Are include/exclude Grouper groups important for courses in general 29](#_Toc494184423)

[9.4. What changes should be instantaneous on a term cutover (based on PS data), verses a more controlled manual cutover 29](#_Toc494184424)

[9.5. Lifecycle of course groups in Office 365 29](#_Toc494184425)

[9.6. Is PS data available 2-3 semesters in advance? We would want data for maybe 3 terms to manage fall courses. 29](#_Toc494184426)

[9.7. Should the member count summary on the Grouper Listing page be taken out? 29](#_Toc494184427)

# 

# INTRODUCTION

The functional requirements specification/document (FRD) is a formal statement of the application functional and operational requirements. It outlines the functional needs, use cases, and visual appearance of a three tiered system to facilitate management of email, email aliases, Active Directory, Listserv and MassMail management, as well as management of other Microsoft services and products.

## Purpose

*(from the Project Charter)* The selfservice.unc.edu site is a combination of applications that are using PowerShell on the back end to interact with AD and Office 365.  We would like to re-architect this solution to call the Graph API when possible, and to separate out the PowerShell layer from the UI layer in a client/server manner. Using this service layer should improve the behavior and performance of the UI. At the same time, we would like to include changes to logging and monitoring, as well as many requested fixes for the UI.

## Scope of Work

Areas covered by this project will include:

* Provide mock views of the proposed site layout
* Implement Shibboleth service
* Implementing Microsoft Identity Server
* Implement ASP.NET WebApi
* An interface to enable users to interact with the functions that Self Service currently exposes.

## Out of Scope

* Mass Mail

## Contributing Stakeholders

Stakeholders are the individuals or groups who have a vested interest in this project. The following internal and external stakeholders were contacted to contribute to the requirements in this document:

| Name | Department | Position |
| --- | --- | --- |
| Dennis Schmidt | ITS - IT Infrastructure | AVC for Institutional Privacy and CISO, ITS - VC - CIO |
| Celeste Copeland | Identity Management | Identity Management Manager |
| Brian Darley | Identity Management | Systems Programmer/Specialist |
| Lyndon Joyce | User Support and Engagement | Service Desk Tier 2 Lead |
|  |  |  |

## References

Project Charter: [Redesign and Upgrade selfservice.unc.edu - Project Charter](https://adminliveunc.sharepoint.com/:w:/r/sites/its/itio/idm/_layouts/15/WopiFrame.aspx?sourcedoc=%7B5080AD81-F2F2-4944-ABB5-D8D59749827F%7D&file=Redesign%20and%20Upgrade%20selfservice.unc.edu%20-%20Project%20Charter.docx&action=default) (restricted link)

Key Meetings:

00/00/2019: Initial meeting

00/00 to 00/00/2019: Requirements, use cases, wireframes

## Assumptions and Constraints

### Assumptions

Assumptions are factors that are believed to be true, but have not been verified. Assumptions can significantly affect requirement and solution design and thus should be clearly defined at the start of the project. The following assumptions were identified during the analysis phase of this project:

| ID# | Date Identified | Assumptions | Impact (Scope/Resource/Time) |
| --- | --- | --- | --- |
| 1. | 00/00/2019 | Implementing Shibboleth on IIS to facilitate Single Sign on | Research and testing will be performed to confirm proper implementation |
| 2. | 00/00/2019 | Authorization will be outsourced to ASP.NET Identity Server | Roles/Claims are not provided by Shibboleth. Therefore, users groups, rights, and other user specific details should be managed by a service dedicated for this purpose. In addition, the service layer will be called by third party applications not authenticated by Shibboleth. All service calls to the API will require JWT token. |

### Constraints

Constraints are organizational or technical boundary conditions that restrict how the solution must be designed and constructed. The primary ones are Time, Resources (people, equipment and budget) and Performance Criteria. The following constraints were identified during the analysis phase of this project:

| ID# | Date Identified | Constraint | Consequence (Scope/Resource/Time) |
| --- | --- | --- | --- |
| 1. | 03/21/2019 | Not all required API end points are present | Missing API methods will need to be identified and implemented |
| 2. | 03/21/2019 | Service Accounts In test have limited rights | Some resources have limited rights. Rights will need to be elevated. Example: AD Dynamic Group Creation |
| 3. | 03/21/2019 | Identity Server Tech Debt | Implementation details related to Identity Server will be addressed. UI cleaned up. |

### Dependencies

| ID# | Date Identified | Dependency | Consequence (Scope/Resource/Time) |
| --- | --- | --- | --- |
|  |  |  |  |

### Business Rules

Business rules depict and enforce the policy of an organization.

| ID# | Date Identified | Business Rule Description | Business Rule Source |
| --- | --- | --- | --- |
| 1. | 3/26/2019 | API architecture follows domain driven architecture. Each endpoint facilitates a business layer checking rules and perform object mappings | Business layer, existing SelfService implementation. |

# Use Cases

The Use Case Models help to define the scope of a solution. The Use Case describes the desired result that a “user” (or system) needs to achieve through interaction with a system. The primary purpose of the Use Case is to capture the required system behavior from the perspective of the end-user in achieving one or more desired goals. A Use Case narrative contains a description of associations and interaction between actors and the system. The use case model may also be represented visually in UML in order to show relationships with other use cases and actors. Essentially, the Use Case diagrams the “optimum” path to achieve the goal as well as the known exceptions, alterations and extensions.

## Authentication/Authorization

|  |  |
| --- | --- |
| Use Case A | Definition |
| ID | SelfService Authentication |
| Title | First time visitor to SelfService |
| Actor | Shibboleth/Identity Server |
| Pre-Conditions |  |
| Post Conditions | Auth token is created and provided to consumer |
| Goal | User obtains an auth token, token contains basic information about the user including assigned roles. |
| Main Scenario | 1. User navigates to SelfService 2. Site has been registered and protected under Shibboleth. 3. User enters credentials, if successful user is redirected back to SelfService. 4. SelfService confirms the existence of an identity token. 5. If token is not present, SelfService redirects to Identity Server 6. Identity Server checks the existence of header variable ‘UID’ provided by Shibboleth. 7. Identity Server retrieves basic user info and assigned AD groups for authorization. 8. Identity Server redirects back to SelfService with newly generated token. 9. Client application (SelfService) stores token. 10. Each API request from the client application passes the identity token in the header. 11. API service check for the existence of token, validates token, and verifies authorization. |

## Client/API Relationship

|  |  |
| --- | --- |
| Use Case B | Definition |
| ID | Client/API Relationship |
| Title | Client communications for requests |
| Actor | Vue.Js Client, Web API |
| Pre-Conditions | User has valid identity token |
| Post Conditions | Request for resources is successfully received and returned |
| Goal | Each exposed endpoint requires a valid identity token, upon request the service will validate token and return an appropriate response. |
| Main Scenario | 1. Request is made from SelfService 2. Token is passed in headers upon each request to SelfService API 3. SelfService API validates request and either rejects request or returns a response based on the endpoint and payload submitted to the endpoint. 4. If token is invalid, not present, or user is not authorized access to a particular endpoint, a 401 response is returned. 5. If request is valid but an error ensues a 400 error will be returned or in rare cases a 500 error. 6. Each request made to SelfService API will record user making the request |

## API to API Relationship

|  |  |
| --- | --- |
| Use Case C | Definition |
| ID | API to API Relationship |
| Title | SelfService API communicates to other API resources |
| Actor | SelfService API (Gateway API), Micro Service API’s |
| Pre-Conditions | User has valid identity token, SelfService API is communicating from known IP that has been previously whitelisted by receiving API’s |
| Post Conditions | Successful request/response from Micro Service APIs from API Gateway |
| Goal | Simplify architecture and secure resources |
| Main Scenario | 1. SelfService API (Gateway API) will be the central hub in the request/response pipeline. 2. Each endpoint in the Gateway API will be segregated by various islands offered in the existing SelfService. (Dashboard, List Manager, Mass Mail, Win Tools, and Office 365 Groups 3. Each endpoint will require a valid authenticated token in order to communicate with an endpoint offered in the Gateway API. 4. Gateway API will be accessible publicly, but API resources that are sensitive will be protected and not accessible directly. 5. API resources will have a whitelist of allowed IP’s. At a future date, the API’s will be protected using Authorization Code Flow using client id and secret. |

# FUNCTIONAL REQUIREMENTS

## Context

[Provide a context diagram of the system, with explanations as applicable. The context of a system refers to the connections and relationships between the system and its environment.]

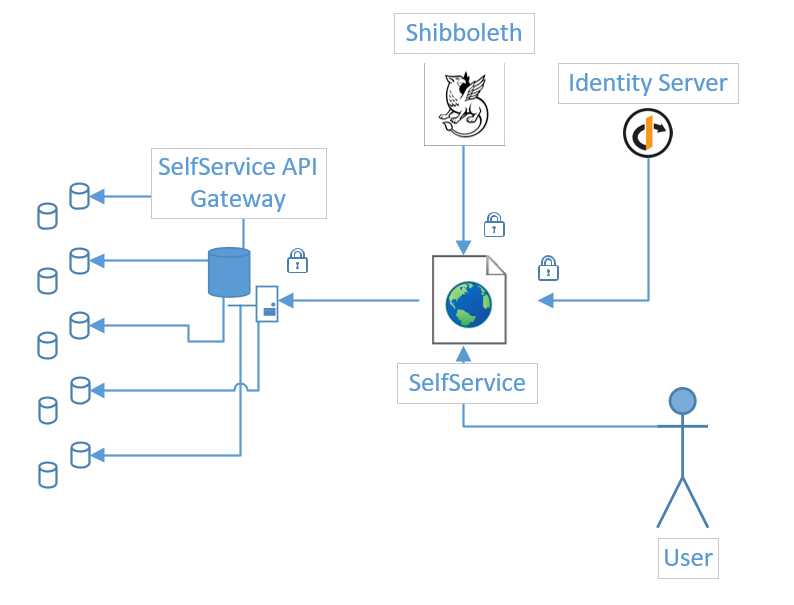


Exhibit 1 - System interactions

## User Requirements

[Provide requirements of the system, user or business, taking into account all major classes/categories of users. Provide the type of security or other distinguishing characteristics of each set of users. List the functional requirements that compose each user requirement. As the functional requirements are decomposed, the highest level functional requirements are traced to the user requirements. Inclusion of lower level functional requirements is not mandatory in the traceability to user requirements if the parent requirements are already traced to them.

User requirement information can be in text or process flow format for each major user class that shows what inputs will initiate the system functions, system interactions, and what outputs are expected to be generated by the system. The scenarios should be comprehensive, to the extent that all user types and all major functions are covered. Give each user requirement a unique number. Typically, user requirements have a numbering system that is separate from the functional requirements. Requirements may be labeled with a leading “U” or other label indicating user requirements.]

| **Req. #** | **Requirement Definition** | **Priority** |
| --- | --- | --- |
| U1.0 | User can access the user interface using a web browser | Must |
| U1.1 | In the UI, user will authenticate with enterprise userid/password before accessing application | Must |
| U1.1.1 | If authentication fails, user will not be able to continue into the application | Must |
| U1.1.2 | Application must be accessible using phone, tablet, or PC | Must |

## Data Flow Diagrams

[Decompose the context level diagrams to determine the functional requirements. Data flow diagrams should be decomposed down to the functional primitive level. These diagrams are further decomposed during design.]

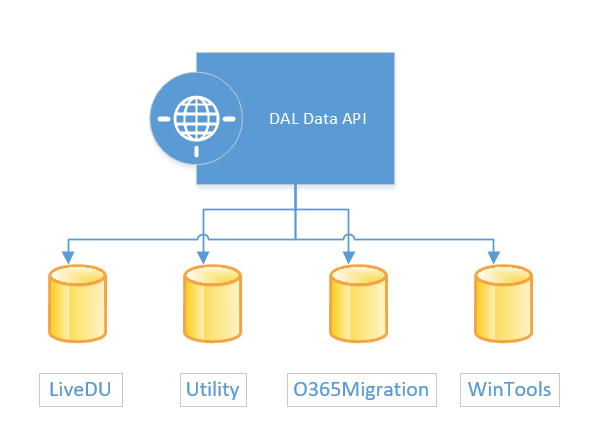


Exhibit 2 - Data flow from DAL Data API to Various Dependent Databases

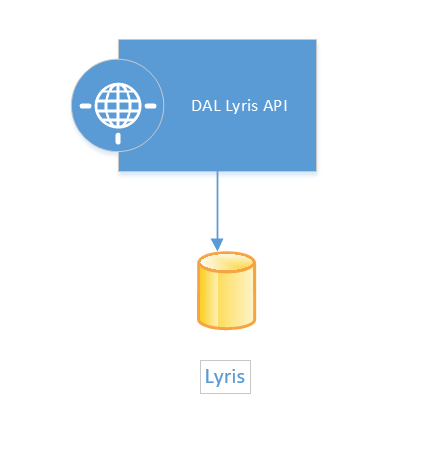


Exhibit 3 - Data flow from DAL Lyris API to Lyris Database

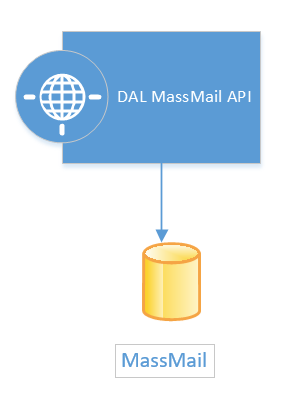


Exhibit 4 - Data flow from DAL MassMail API to MassMail Database

## Logical Data Model/Data Dictionary

### DAL.Data.API – LiveDu

Entity: Alias Domain Setting

AD Domain Address

RemoteRoutingAddress

EmailAddress

LiveAddress

Entity: AllowedEmailLdapAlias

ID

Affiliate

Entity: AvailableOfficeLicenses

SkuPartNumber

SkuId

Populations (Student, Employee, Affiliate)

Entity: SupplementalMailAlias

Id

Population (Student, Employee, Affiliate)

Domain

Entity: DistributionGroupMember

Id

GroupName

Onyen

UserPrincipalName

Entity: MFAStatus

Id

Onyen

UserPrincipalName

DisplayName

Status

NewStatus

Entity: Provisioning

JobType

Onyen

SubmittedDate

MailboxType

Status

StatusDetail

ScheduledDate

CreateDate

SubmittedBy

NotifiedDate

### DAL.Data.API – Lyris

Entity: DormantList

ID

ListName

IsDormant

IsRenewed

IsNotified

IsDeleted

FromNotifiedDate

SubscriberEmail

IsAdmin

Entity: Deletetion

ListName

DeleteDate

CreateDate

LastLogged

Entity: SubscriberDump

ListName

SubscriberEmail

IsListAdmin

ModifiedDate

### DAL.Data.API – Office365

Entity: MfaUser

ID

UID

PID

DisplayName

MFAEnabled

MFAExemptBeginDate

MFAExemptEndDate

Reason

IncidentNumber

### Role Mapping / Access Restrictions

| App Section | Role | Restrictions |
| --- | --- | --- |
| Home | N/A |  |
| Dashboard | N/A |  |
| List Manager | ITS\_WSP-Tools-ListManager-Postmaster | Postmaster Tools, List Deletions |
| Mass Mail | N/A |  |
| Win Tools | ITS\_WSP-Tools-AdminTools-Home  ITS\_WSP-Tools-ADTools-Home  ITS\_WSP-Tools-ADTools-AccountInfo  ITS\_WSP-Tools-ADTools-OUAdminDirectory  ITS\_WSP-Tools-ExchangeTools-Home  ITS\_WSP-Access-CompromisedAccount-Managers  ITS\_WSP-Tools-ExchangeTools-MailboxCreation  ITS\_WSP-Tools-ExchangeTools-Aliases  ITS\_WSP-Tools-ExchangeTools-CompromisedAccount  ITS\_WSP-Tools-Systems-Home  ITS\_WSP-Tools-Systems-Utility  ITS\_WSP-Tools-Systems-ADManagement | Win Tools  AD Tools  AD Tools Account Info  AD Tools OU Admin Directory  Exchange Tools  MFA  Mailbox Creation  Alias Management  Compromised Accounts  Systems Tools  Utility |
| Office365 Groups | N/A |  |

### Self Service Menu Structure

Selfservice.unc.edu (Landing Page)

|

* Dashboard
* List Manager
* |
* Landing Page
* Login for Lyris
* Information
* User Tools
* |
* Landing Page
* Check Subscriptions
* Search Lists
* Admin Tools
* |
* Landing Page
* Change List Owner
* Create List
* Delete List
* Postmaster Tools\*
* List Deletions\*
* Mass Mail \*\*
* Win Tools\*
* |
* Exchange Tools
* |
* Landing Page
* Provisioning
* Aliases
* Resources
* Compromised Accounts\*
* MFA\*
* AD Tools\*
* |
* Landing Page
* Organization Units
* Account Info
* Account Lockouts
* Groups
* Systems\*
* |
* Landing Page
* Account Management
* User Lockouts
* Shared Mailbox
* Resource Mailbox
* Alias Authority
* Forwarding Authority
* Group Management
* Office365 Groups

## Interface Requirements

### User Interface Authentication/Authorization

| **Section/ Requirement ID** | **Requirement Definition** | **Priority** |
| --- | --- | --- |
| UI1.1 | Authentication via Shibboleth | Must |
| UI 1.2 | After authentication, redirect to SelfService | Must |
| UI 1.3 | SelfService redirects to Identity Server for authorization and user identification | Must |
| UI 1.4 | Returning to SelfService, an identity token will be provided | Must |
| UI 1.5 | SelfService sets default Bearer token for all subsequent HTTP calls | Must |
| UI 1.6 | Call is made to SelfService Business API for available routes given available roles | Must |

### Dashboard Email Sign-Up (/dashboard)

| **Section/ Requirement ID** | **Requirement Definition** | **Priority** |
| --- | --- | --- |
| UI 2.1 | Next GAL Sync Notification Widget | Must |
| UI 2.1.1 | High level metrics displayed (e-mail provisions, AD lock counts) | May |
| UI 2.2 | E-mail sign-up  Intended for individual users to initiate their own mail provisioning. Also provides details for dual mailbox handling with UNCH and UNC | Must |
| UI 2.3 | Lyris List Subscriptions  Lists current Lyris Subscriptions | Must |
| UI 2.4 | Self-manage AD aliases  Allows for individuals to manage e-mail aliases | Must |
| UI 2.5 | List resources  Lists Exchange resources | Must |

### List Manager

| **Section/ Requirement ID** | **Requirement Definition** | **Priority** |
| --- | --- | --- |
| UI 3.1 | User can access landing page for List Manage | Must |
| UI 3.2.1 | Log In List Members (Link should reflect proper domain, i.e. test or production) | Must |
| UI.3.2.2 | Log In Administrator (Link should reflect proper domain, i.e. test or production) | Must |
| UI.3.3 | Information Links | Must |
| UI.3.4 | User Tools Browse Lists, Reset Password (Links should reflect proper domain) | Must |
| UI.3.4.1 | List subscribed subscriptions | Must |
| UI.3.4.2 | Search Lists (Use like expression, Visit and Subscribe reflect proper domain links) | Must |
| UI.3.5 | Admin Tools, links to available resources under Admin Tools | Must |
| UI.3.5.1 | Change List Owner | Must |
| UI.3.5.2 | Create List | Must |
| UI.3.5.3 | Delete List | Must |
| UI.3.6 | If user has appropriate roles for access, show Postmaster Tools | Must |
| UI.3.7 | If user has appropriate roles for access, show List Deletions | Must |

### Mass Mail

| **Section/ Requirement ID** | **Requirement Definition** | **Priority** |
| --- | --- | --- |
| UI 4.1 | User can access landing page for Mass Mail | Must |
| UI 4.2 | Create Request Wizard | Must |
| UI.4.3 | View Request | Must |
| UI.4.4 | Archives | Must |

### Win Tools

| **Section/ Requirement ID** | **Requirement Definition** | **Priority** |
| --- | --- | --- |
| UI 5.1 | User can access landing page for Win Tools (User must be assigned to appropriate group) | Must |
| UI 5.2 | Exchange Tools (User must be assigned to appropriate group) | Must |
| UI.5.2.1 | Provisioning | Must |
| UI.5.2.2 | Aliases | Must |
| UI.5.2.3 | Resources | Must |
| UI.5.2.4 | Compromised Accounts | Must |
| UI.5.2.5 | MFA (User must be assigned to appropriate group) | Must |
| UI.5.3 | AD Tools (User must be assigned to appropriate group) | Must |
| UI.5.3.1 | Organizational Units | Must |
| UI.5.3.2 | Account Info | Must |
| UI.5.3.3 | Account Lockouts | Must |
| UI.5.3.4 | Groups | Must |
| UI.5.4 | Systems (User must be assigned to appropriate group) | Must |
| UI.5.4.1 | Account Management | Must |
| UI.5.4.2 | Lockouts | Must |
| UI.5.4.3 | Dashboard | Must |
| UI.5.4.4 | Shared Mailbox | Must |
| UI.5.4.5 | Resource Mailbox | Must |
| UI.5.4.6 | Alias Authority | Must |
| UI.5.4.7 | Forwarding Authority | Must |
| UI.5.4.8 | Group Management | Must |

## Data Requirements

### Data Synchronization between enrollment system and Grouper

| **Section/ Requirement ID** | **Requirement Definition** | **Priority** |
| --- | --- | --- |
| D1.1 | Grouper will fully update its course and section groups and enrollment from the enrollment system (e.g. PeopleSoft) at least three times per day at regular intervals | Must |
| D1.1.1 | New course sections appearing in the enrollment data will cause new section groups to be created in Grouper; course groups will also be create if not already existing | Must |
| D1.1.2 | Course sections no longer appearing in the enrollment data will cause the section groups to be removed from Grouper | Should?  Must? |
| D1.1.3 | Courses no longer appearing in the enrollment data, or in which no sections exist, will cause the course tree to be removed in Grouper, including all its sections | Should?  Must? |

### Data Synchronization between Grouper and Office 365

| **Section/ Requirement ID** | **Requirement Definition** | **Priority** |
| --- | --- | --- |
| D2.1 | When membership of a section’s instructor, TA, or student group in Grouper changes, all affected O365 course groups using it (as defined below) will be updated within 5 minutes | Must |
| D2.1.1 | If technology allows, update should happen within 30 seconds | Should |
| D2.1.2 | If a course group has enabled the option to keep instructor membership in sync, a Grouper addition/removal from an instructor group will update all such course groups that have opted to add instructors as list owners or as members, adding or removing them as an owner or a member, as defined for each group | Must |
| D2.1.2.1 | The O365 course group’s creator cannot be removed as an owner from the course group, even if it is removed from the relevant Grouper group | Must |
| D2.1.3 | If a course group has enabled the option to keep TA membership in sync, a Grouper addition/removal from an TA group will update all such course groups that have opted to add TA’s as list owners or as members, adding or removing them as an owner or a member, as defined for each group | Must |
| D2.1.4 | If a course group has enabled the option to keep student membership in sync, a Grouper addition/removal from a student group will update all such course groups in the system | Must |
| D2.2 | If Grouper tries to update a course group that no longer exists in Office 365 (possibly from a manual deletion), Grouper will log the failure to update the non-existent group | May |
| D2.3 | If a member or owner has been manually added to the course group via the Office 465 interface, the Grouper sync would potentially remove it, if the group preference is set to sync with enrollment data | May |
| D3.1 | If enrollment data or group sections have changed so that the group maintainer is no longer an instructor of any section (and thus not a group owner), a report or notice will be sent to Teaching & Learning, noting the group affected | Must |

# OTHER REQUIREMENTS

## Operational Requirements

### Data Requirements

### Security and Privacy

| **Section/ Requirement ID** | **Requirement Definition** | **Consequence** |
| --- | --- | --- |
| SP1.1 | FERPA issues TBD | ??? |
| SP1.2 | Membership in Grouper groups will not be publicly viewable; access to it will be for Identity Management and other stakeholding groups such as Teaching and Learning or the Help Desk (TBD) | Loss of student privacy |
| SP1.3 | The ability to update include and exclude groups in Grouper will be limited to a defined set of roles; e.g., Identity Management and other stakeholding groups such as Teaching and Learning or the Help Desk (TBD) | Privilege escalation as this effectively adds users to the course group |
| SP1.4 | Instructors cannot manage course groups that have been created by other instructors (TBD how to transfer ownership?) |  |

### Audit Trail

| **Section/ Requirement ID** | **Requirement Definition** | **Subsystem** |
| --- | --- | --- |
| AU1.1 | Group creation and pertinent information | UI |
| AU1.2 | Changes to Grouper groups due to enrollment system of record changes | Grouper |
| AU1.3 | Grouper pushing a sync to Office 365 | Grouper |
| AU1.4 | Changing course group settings | UI |

### Non-

### Reliability

1. [State the following in this section:
   1. State the damage can result from failure of this system—indicate the criticality of the software, such as:
      1. Loss of human life
      2. Complete or partial loss of the ability to perform a mission-critical function
      3. Loss of revenue
      4. Loss of employee productivity
   2. What is the minimum acceptable level of reliability?
2. State required reliability:
   1. Mean-Time-Between-Failure is the number of time units the system is operable before the first failure occurs.
   2. Mean-Time-To-Failure is the number of time units before the system is operable divided by the number of failures during the time period.
   3. Mean-Time-To-Repair is the number of time units required to perform system repair divided by the number of repairs during the time period.]

*Reliability is the probability that the system processes work correctly and completely without being aborted.*

TBD

### Recoverability

[Answer the following questions in this section:

1. In the event the application is unavailable to users (down) because of a system failure, how soon after the failure is detected must function be restored?
2. In the event the database is corrupted, to what level of currency must it be restored? For example “The database must be capable of being restored to its condition of no more than 1 hour before the corruption occurred”.
3. If the processing site (hardware, data, and onsite backup) is destroyed, how soon must the application be able to be restored?]

*Recoverability is the ability to restore function and data in the event of a failure.*

TBD

### System Availability

[State the period during which the application must be available to users. For example, “The application must be available to users Monday through Friday between the hours of 6:30 a.m. and 5:30 p.m. EST.” If the application must be available to users in more than one time zone, state the earliest start time and the latest stop time. Consider daylight savings time, too.

Include use peak times. These are times when system unavailability is least acceptable.]

*System availability is the time when the application must be available for use. Required system availability is used in determining when maintenance may be performed.*

TBD

### General Performance

[Describe the requirements for the following:

1. Response time for queries and updates
2. Throughput
3. Expected rate of user activity (for example, number of transactions per hour, day, or month, or cyclical periods)

Specific performance requirements, related to a specific functional requirement, should be listed with that functional requirement.

Response time for the UI can be potentially slower than an average application, due to the number and types of queries that must be integrated in presenting the page. An upper limit of around 5 seconds per page generation is expected.

### Capacity

[List the required capacities and expected volumes of data in business terms. Do not state capacities in terms of system memory requirements or disk space—if growth trends or projections are available, provide them]

TBD

### Data Retention

[Describe the length of time various forms of data must be retained and the requirements for its destruction.

For example, “The system shall retain application information for 3 years”. Different forms of data include: system documentation, audit records, database records, access records.]

TBD

### Error Handling

[Describe system error handling.]

TBD

### Validation Rules

[Describe System Validation Rules.]

TBD

### Conventions/Standards

[Describe system conventions and standards followed.

For example: Microsoft standards are followed for windows, Institute of Electrical and Electronics Engineers (IEEE) for data formats, etc.]

TBD

# APPENDIX A - GLOSSARY

[Define terms, acronyms, and abbreviations used in the FRD.]

UNC-CH UNC-Chapel Hill

IDM The Identity Management group at UNC-Chapel Hill

O365 Office 365

TA Teaching assistant

UI User interface (e.g., a web application)

T&L Teaching and Learning group

# APPENDIX B - UI Wireframes

## Office 365 Groups main screen



Exhibit 3 - Office 365 Groups main screen

## Course Groups main screen



Exhibit 4 - Course Groups main screen

## Create new class group screen



Exhibit 5 - Create new class group

## Create New Class Group - Confirmation



Exhibit 6 - Create new class group confirmation

## Manage existing group



Exhibit 7 - Manage existing group

# Appendix C: Data Metrics from PeopleSoft

Session Code FA17. Based on sample data used by Sakai supplied by Ethan Kromhout 6/2017

| Description | Count |
| --- | --- |
| Number of subjects (AAAD, AMST, ...) | 143 |
| Number of courses (AAAD201, AHSC902, ...) | 2779 |
| Number of instructors assigned | 2629 |
| Total number of instructor assignments | 10872 |
| Number of students enrolled | 23424 |
| Total number of student enrollments | 105442 |
| Max number of courses per subject | 97 (EDUC) |
| Number of course sections | 7662 |
| Max number of sections per course | 73 (ENGL 105) |
| Max number of courses for a single instructor | 77 (multiple courses, in general are support staff for Friday Center or Law School) |
| Max sections for a course for a single instructor | 42 (CHEM 101L, Lori Del Negro [ladelneg] |
| Max number of instructors for a single course | 90 (NURS 994) |
| Max number of instructors for a single section | 7 (LAW 266-001, LAW514-001, ...) |
| Max number of students for a single course | 1240 (ENGL 105, 73 sections);  884 (BIOL 202, 18 sections) |
| Max number of students for a single section | 417 (ECON 125-001) |

# Recent Decisions/Answers Based on Questions or Discussion

## When a course group creator is no longer an instructor for a group/no longer at UNC?

There will need to be a regular report sent to T&L listing such conflict. There will need to be a management interface on the Self Service site to allow them to change an owner.

Note that until the course owner is changed, the original owner will still be able to manage the group in Self Service and Office 365.

## Can groups be deleted in O365? What happens to them when the next sync happens?

Course groups can be deleted, where they go into a Recycle Bin for 30 days before being permanently removed. Grouper should handle such cases by quietly skipping updates for the group. Optionally there can be a report to T&L to remove the group or turn off the automatic update.

## Who are updaters of Grouper includes/exclude groups, and readers of composite groups?

Just T&L should be readers of the course groups in Grouper, for support purposes.

We may not be needing include/exclude groups, as this can be better done in O365

## See FR3.3 - When listing all the sections available to add/remove for an existing group, what if an instructor was originally assigned to a section but no longer is?

The Manage Existing Group wireframe has been modified. The user can opt to remove it or leave it alone, but can’t add it back once the removal has been submitted.

## FERPA Issues

Suzanne Cadwell discussed with University Counsel the concern that onyens will be visible for classmates in the same section. The response was that it is ok to proceed since this is a business use.

## Where should the application be hosted -- Self Service or Improv?

The technical decisions on actual hosting of the application will be made by IdM. If the site is hosted outside of Self Service, we can still add a menu link within Self Service.

## Create groups for future terms?

The UI design now allow the user to choose a future term to create a group for

# Appendix D: Decisions/Questions in Progress

7/13/2017

## Can any group owner manage an existing group, or just the original person who created the group?

## How often will courses and sections be removed from PeopleSoft? Should they be removed in Grouper when that happens, or left for historical purposes?

8/7/2017

## Are include/exclude Grouper groups important for courses in general

Do we need the ability to add/remove ad-hoc users not in the official course enrollment? While it is not important for this application (users can be added through Office 365), are there other applications that would need that functionality. This is a broader decision to be made by IdM.

## What changes should be instantaneous on a term cutover (based on PS data), verses a more controlled manual cutover

Now that the main menu has an option to choose which term to manage, this isn’t as relevant for the application itself. It may be tangentially important to Grouper, if a pseudo-folder like ref:course:current is mapped to the currently active term, when should it be updated to the new term, and is it automated or manual.

## Lifecycle of course groups in Office 365

## Is PS data available 2-3 semesters in advance? We would want data for maybe 3 terms to manage fall courses.

## Should the member count summary on the Grouper Listing page be taken out?

It may be confusing whether this number is from official enrollment or from O365. If from O365, it is likely a performance hit because it will need to query every group.