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

Contents

[1 INTRODUCTION 4](#_Toc7440275)

[1.1. Purpose 4](#_Toc7440276)

[1.2. Scope of Work 4](#_Toc7440277)

[1.3. Out of Scope 4](#_Toc7440278)

[1.4. Contributing Stakeholders 4](#_Toc7440279)

[1.5. References 5](#_Toc7440280)

[1.6. Assumptions and Constraints 5](#_Toc7440281)

[1.6.2. Assumptions 5](#_Toc7440282)

[1.6.3. Constraints 5](#_Toc7440283)

[1.6.4. Dependencies 6](#_Toc7440284)

[1.6.5. Business Rules 6](#_Toc7440285)

[2 Use Cases 6](#_Toc7440286)

[2.4. Authentication/Authorization 6](#_Toc7440287)

[2.5. Client/API Relationship 7](#_Toc7440288)

[2.6. API to API Relationship 7](#_Toc7440289)

[3 FUNCTIONAL REQUIREMENTS 8](#_Toc7440290)

[3.4. Context 8](#_Toc7440291)

[3.5. User Requirements 8](#_Toc7440292)

[3.6. Data Flow Diagrams 9](#_Toc7440293)

[3.7. Logical Data Model/Data Dictionary 10](#_Toc7440294)

[3.7.1. DAL.Data.API – LiveDu 10](#_Toc7440295)

[3.7.2. DAL.Data.API – Lyris 11](#_Toc7440296)

[3.7.3. DAL.Data.API – Office365 12](#_Toc7440297)

[3.7.4. Role Mapping / Access Restrictions 12](#_Toc7440298)

[3.7.5. Self Service Menu Structure 13](#_Toc7440299)

[3.8. Interface Requirements 14](#_Toc7440300)

[3.8.2. User Interface Authentication/Authorization 14](#_Toc7440301)

[3.8.3. Dashboard Email Sign-Up (/dashboard) 14](#_Toc7440302)

[3.8.4. List Manager 15](#_Toc7440303)

[3.8.5. Mass Mail 15](#_Toc7440304)

[3.8.6. Win Tools 15](#_Toc7440305)

[3.9. Caching Requirements 16](#_Toc7440306)

[3.9.2. Caching connections and results for resources 16](#_Toc7440307)

[3.9.3. Logging Requirements 16](#_Toc7440308)

[4 APPENDIX A - GLOSSARY 17](#_Toc7440309)

[5 APPENDIX B - UI Wireframes 17](#_Toc7440310)

[5.4. Self Service Landing Page 17](#_Toc7440311)

[5.5. Dashboard 18](#_Toc7440312)

[5.5.1 Email Account Request 18](#_Toc7440313)

[5.5.2 My Aliases 19](#_Toc7440314)

[5.5.3 Lyris Subscriptions 19](#_Toc7440315)

[5.5.4 My AD Groups 20](#_Toc7440316)

[5.5.5 List Manager 20](#_Toc7440317)

[5.5.6 Lyris List Login 21](#_Toc7440318)

[5.5.7 Lyris Information 21](#_Toc7440319)

[5.5.8 Lyris User Tools 22](#_Toc7440320)

[5.5.9 Lyris Change List Owner 22](#_Toc7440321)

[5.5.10 Lyris Create List 23](#_Toc7440322)

[5.5.11 Lyris Delete List 23](#_Toc7440323)

[5.5.12 Lyris Postmaster Tools 24](#_Toc7440324)

[5.5.13 Lyris List Deletions 25](#_Toc7440325)

[5.5.14 Win-Tools Exchange Tools 26](#_Toc7440326)

[5.5.15 Win-Tools Provisioning 27](#_Toc7440327)

[5.5.16 Win-Tools Aliases 27](#_Toc7440328)

[5.5.17 Win-Tools Resources 28](#_Toc7440329)

[5.5.18 Win-Tools Compromised Accounts 28](#_Toc7440330)

[5.5.19 Win-Tools MFA 28](#_Toc7440331)

# 

# 

# 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.]

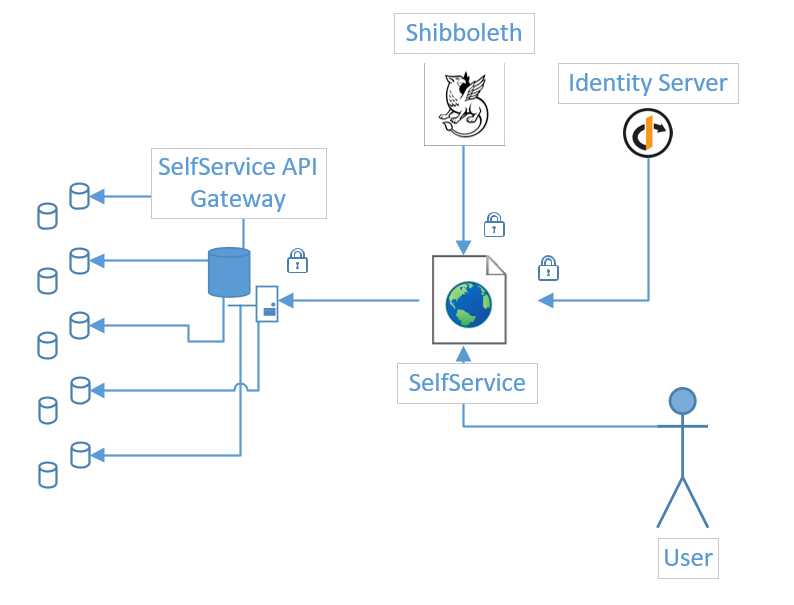


Exhibit 1 - System interactions

## 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.]

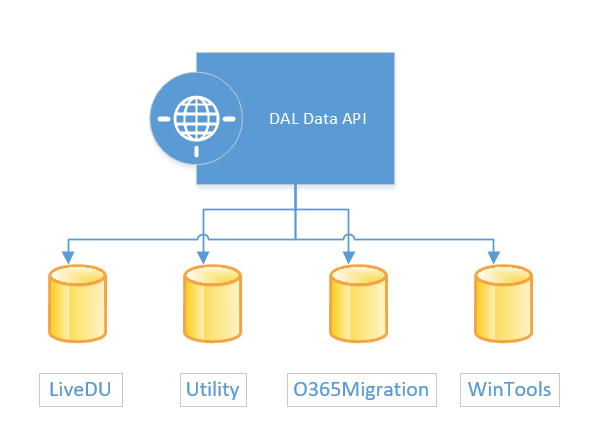


Exhibit 2 - Data flow from DAL Data API to Various Dependent Databases

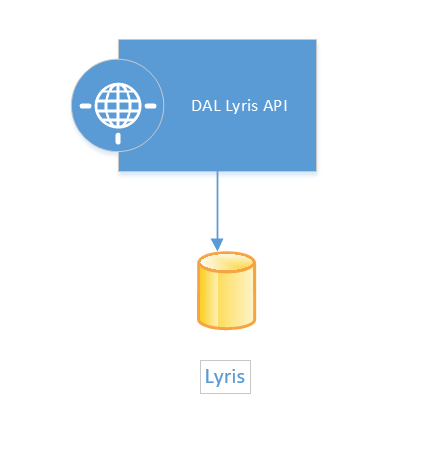


Exhibit 3 - Data flow from DAL Lyris API to Lyris Database

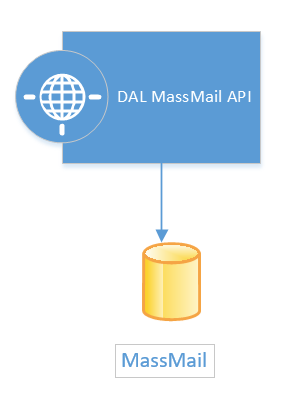


Exhibit 4 - Data flow from DAL MassMail API to MassMail Database

## 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 |

## Caching Requirements

### Caching connections and results for resources

| **Section/ Requirement ID** | **Requirement Definition** | **Priority** |
| --- | --- | --- |
| D1.1 | Connections to Office365 will provide caching to connections to improve performance | Must |
| D1.1.1 | DirSync status should be cached with a 30 minute cache policy to prevent repeated calls and overload connection / throttling policy with Microsoft | Must |
|  |  |  |
|  |  |  |

### Logging Requirements

| **Section/ Requirement ID** | **Requirement Definition** | **Priority** |
| --- | --- | --- |
| D2.1 | Logging will be flexible and allow for various methods of recording system health. | Must |
| D2.1.1 | For the interim all method calls will be logged and recorded including Info, Warning, and Error messages. As the product matures this can exclude info messages. | Should |
| D2.1.2 | Log events should include source of request as well as authenticated user id for each request when possible | Should |
| D2.1.3 | An API should be available to record and retrieve log messages to provide better responses to issues. | Must |
| D2.1.4 | Admin page should be set up to allow searching and filtering log messages. | Should |

# APPENDIX A - GLOSSARY

UNC-CH UNC-Chapel Hill

IDM The Identity Management group at UNC-Chapel Hill

O365 Office 365

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

# APPENDIX B - UI Wireframes

## Self Service Landing Page

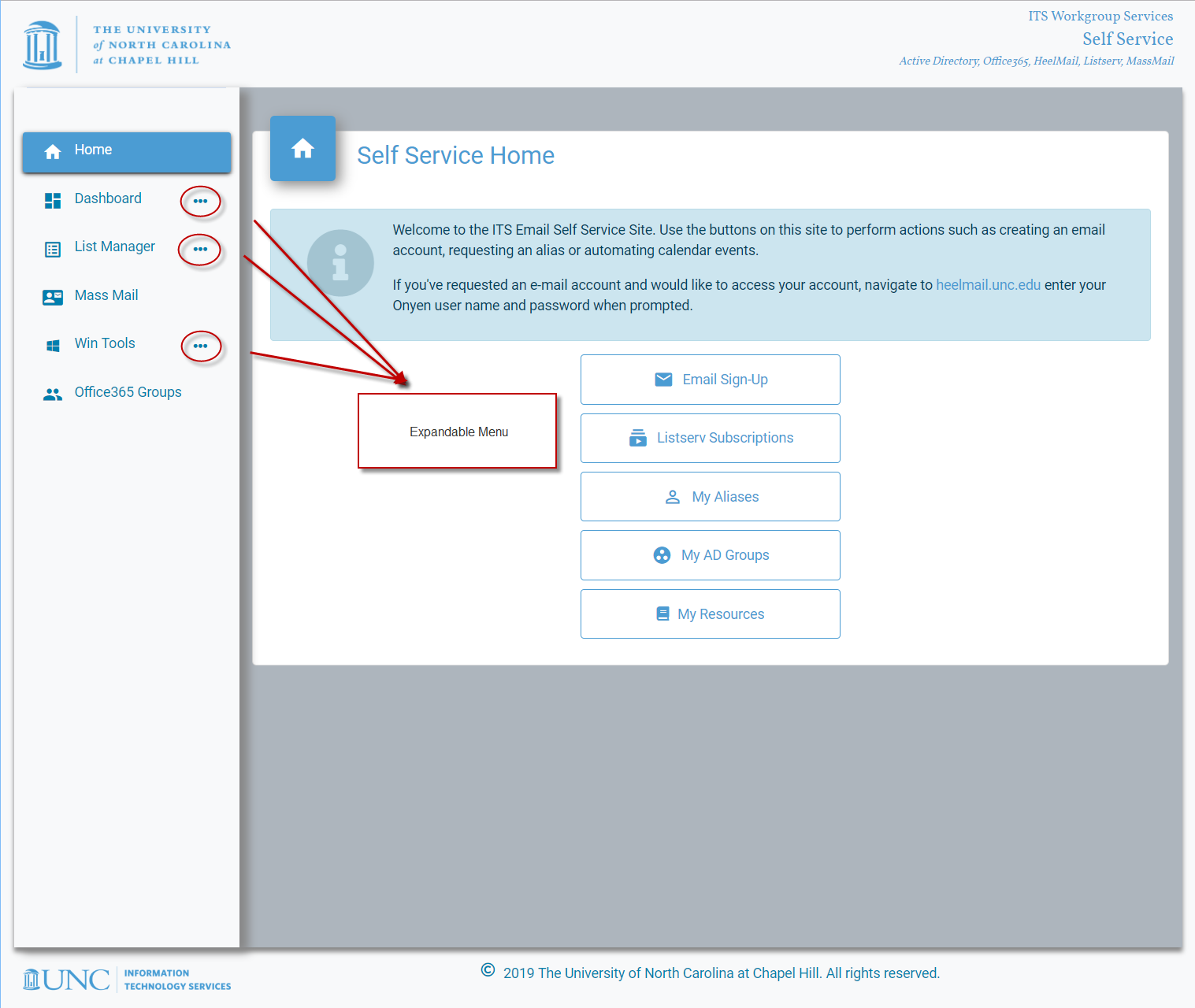


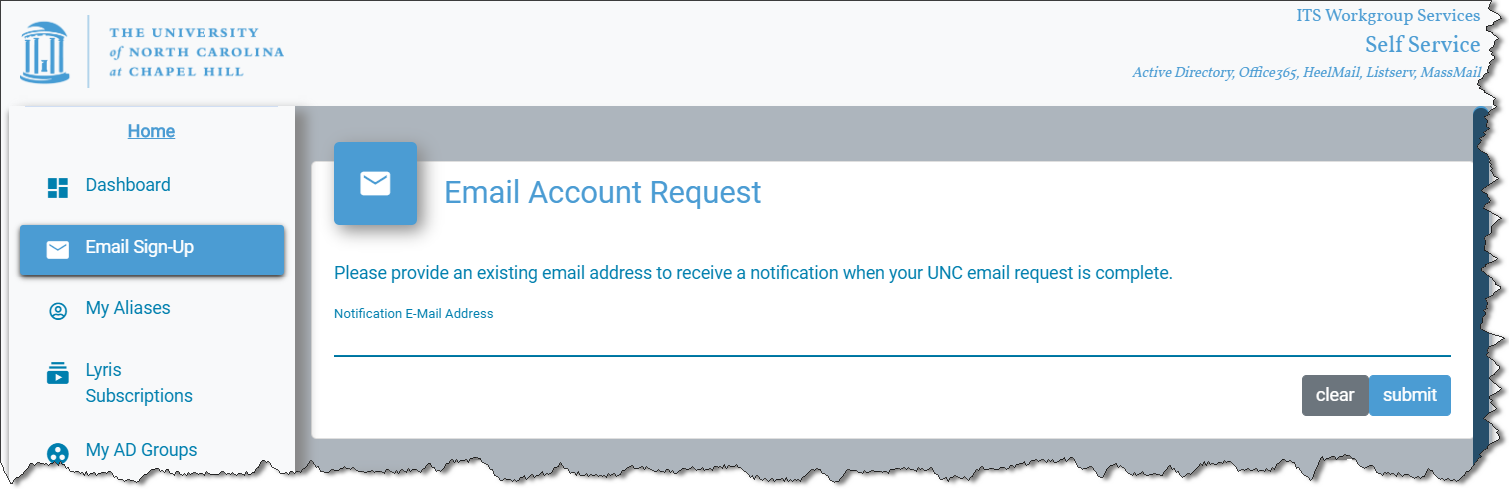
Exhibit 3 - Self Service Landing Page main screen

## Dashboard

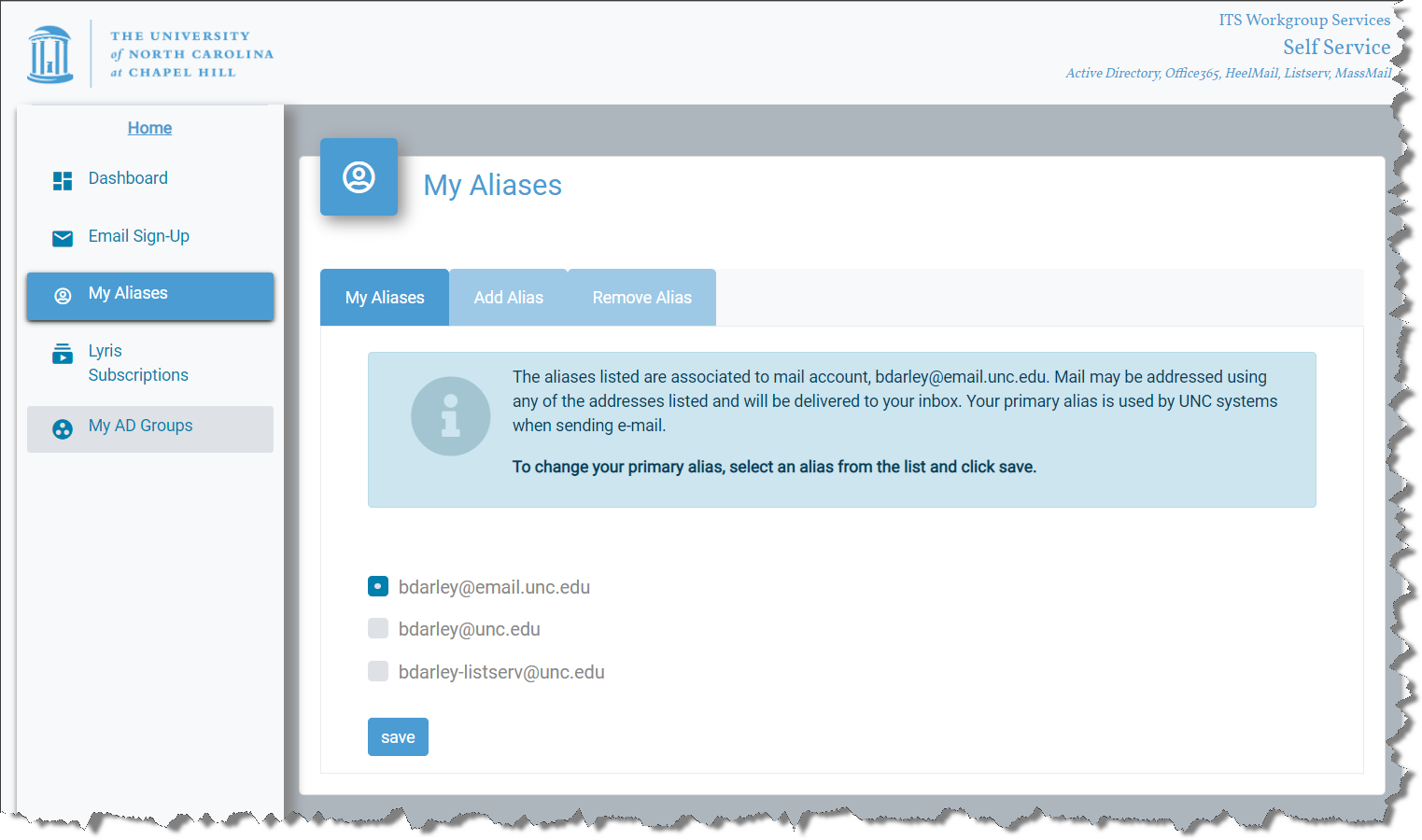


Exhibit 4 – Self Service Dashboard

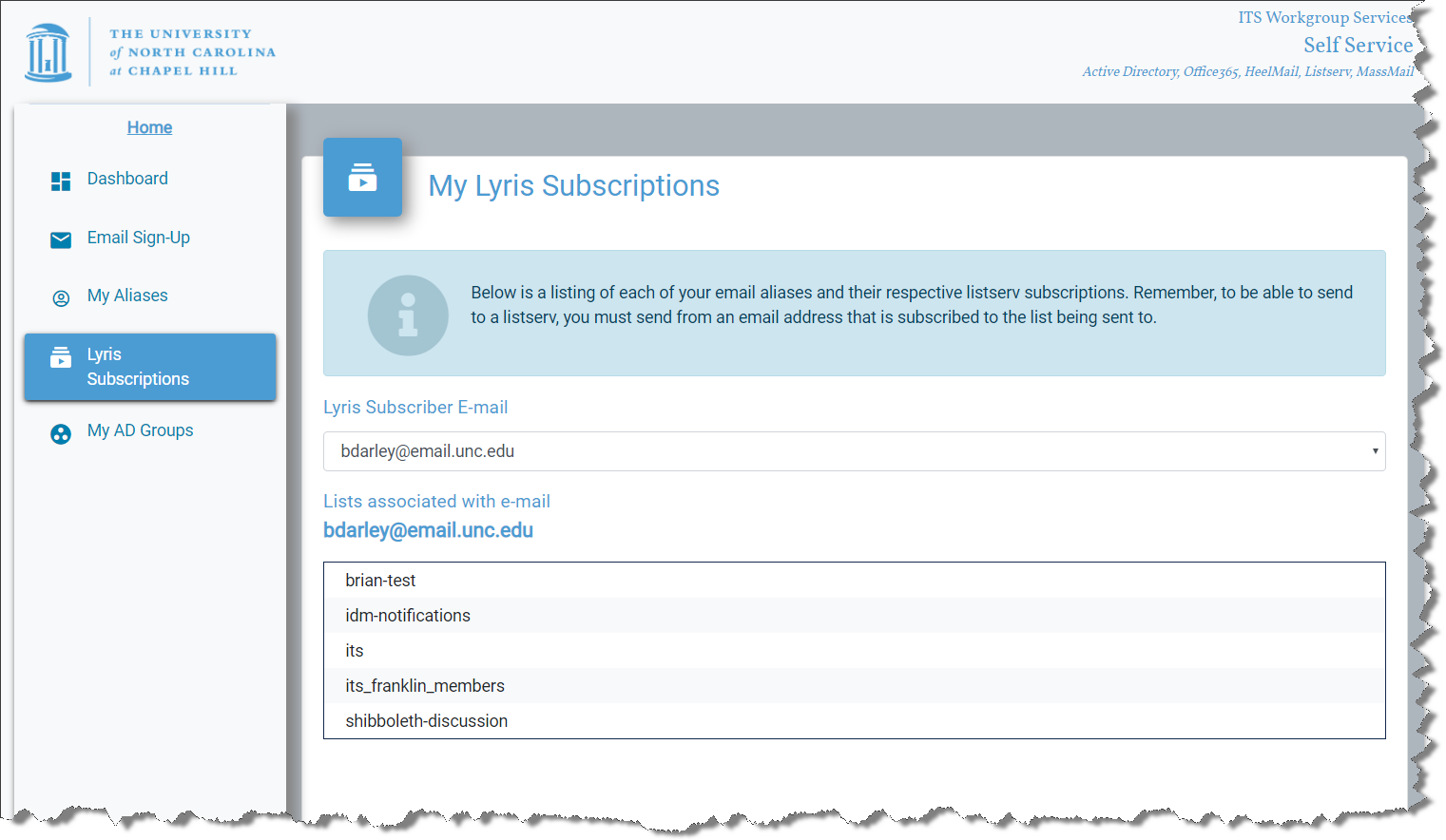
## Email Account Request



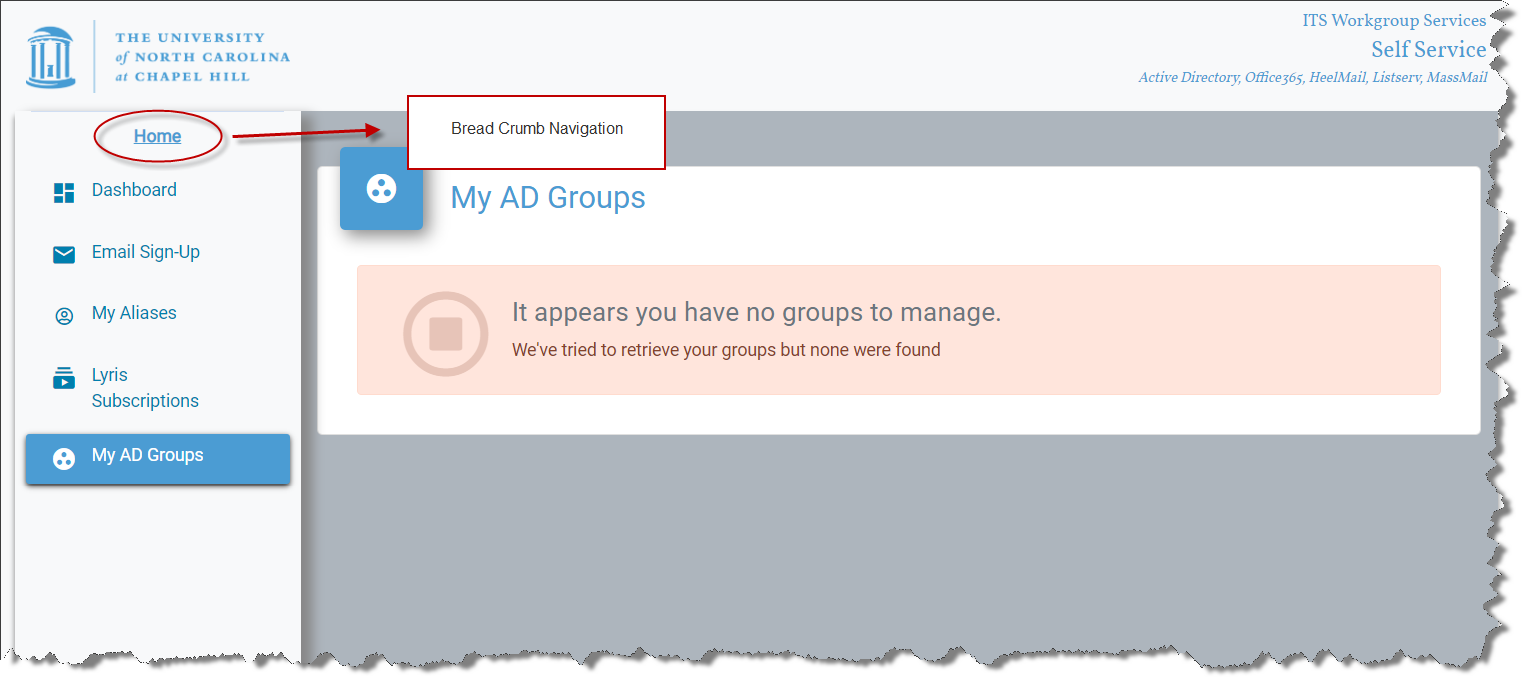
## My Aliases



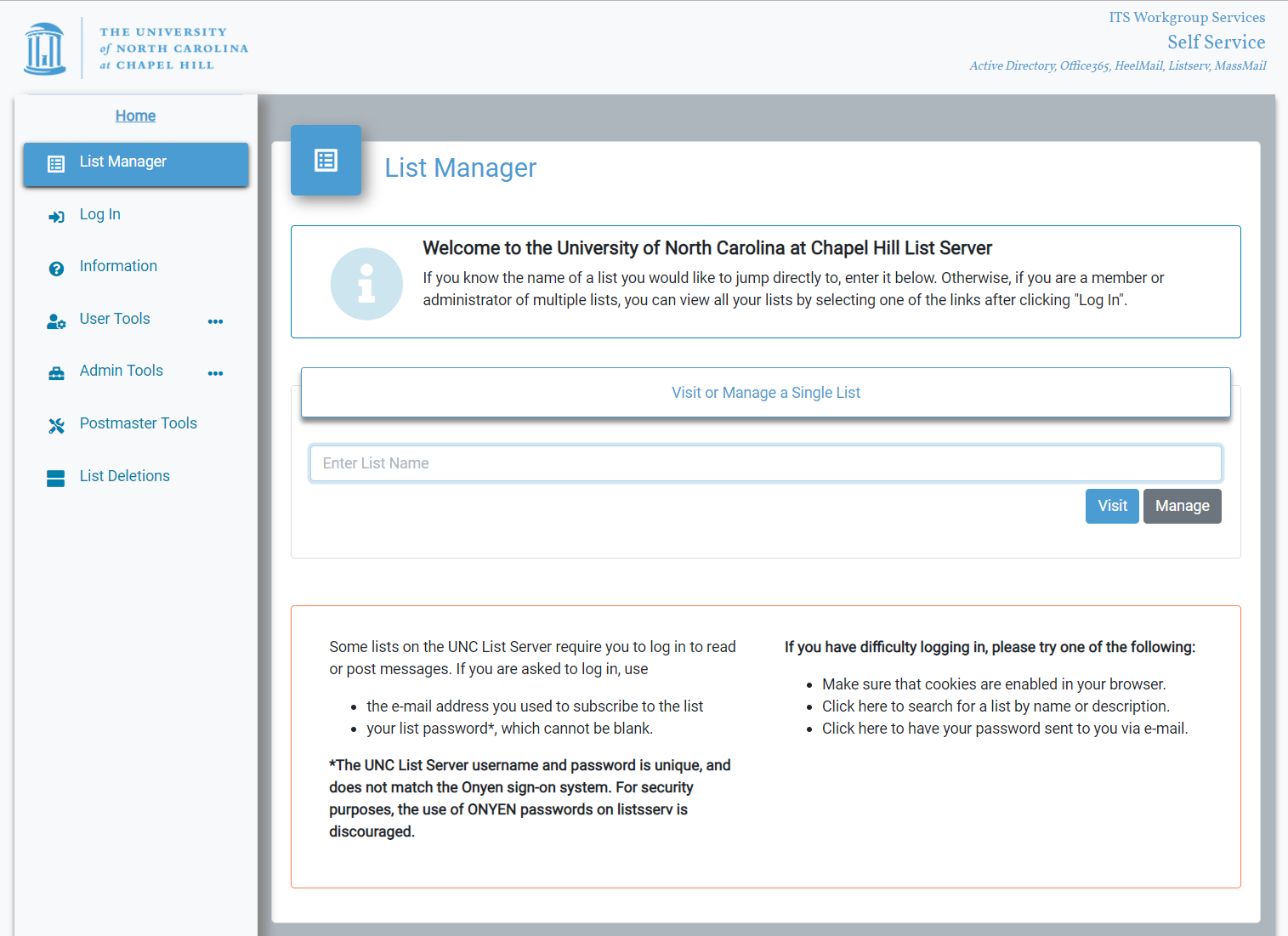
## 5.5.3 Lyris Subscriptions



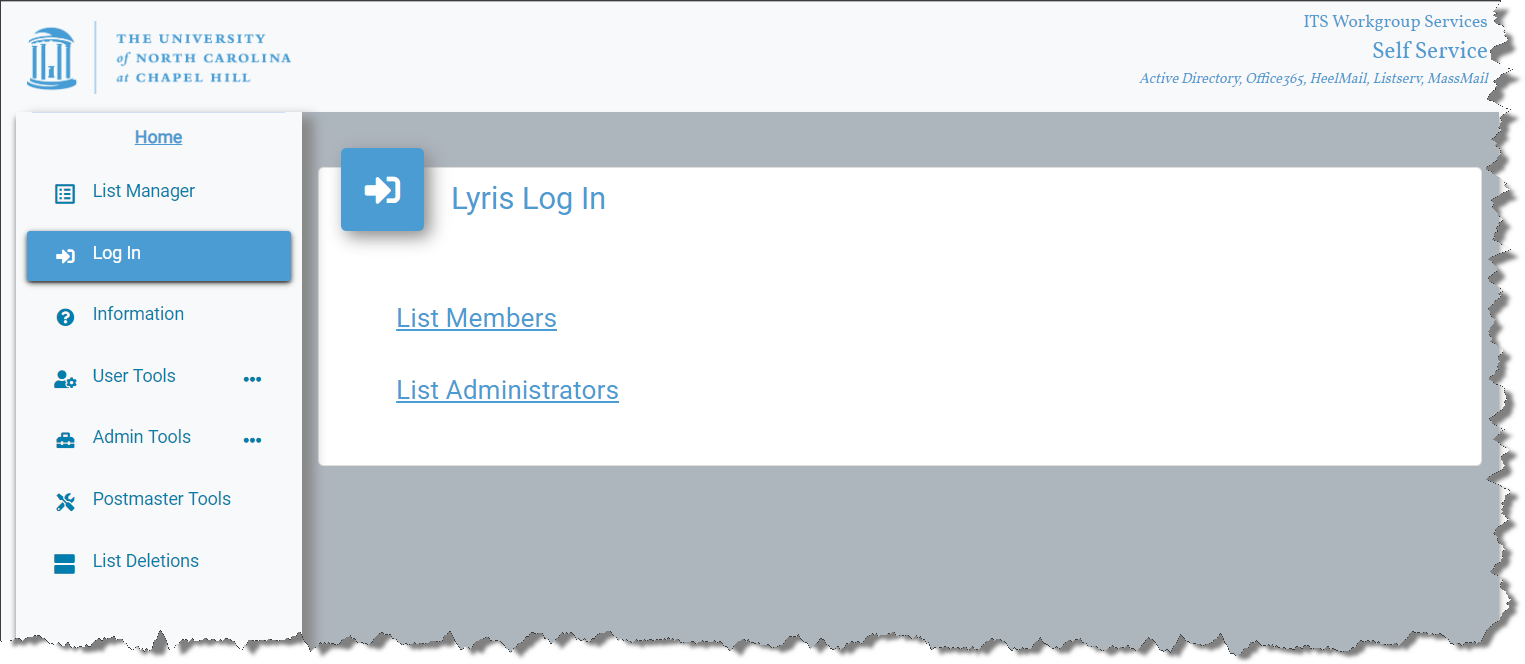
## My AD Groups



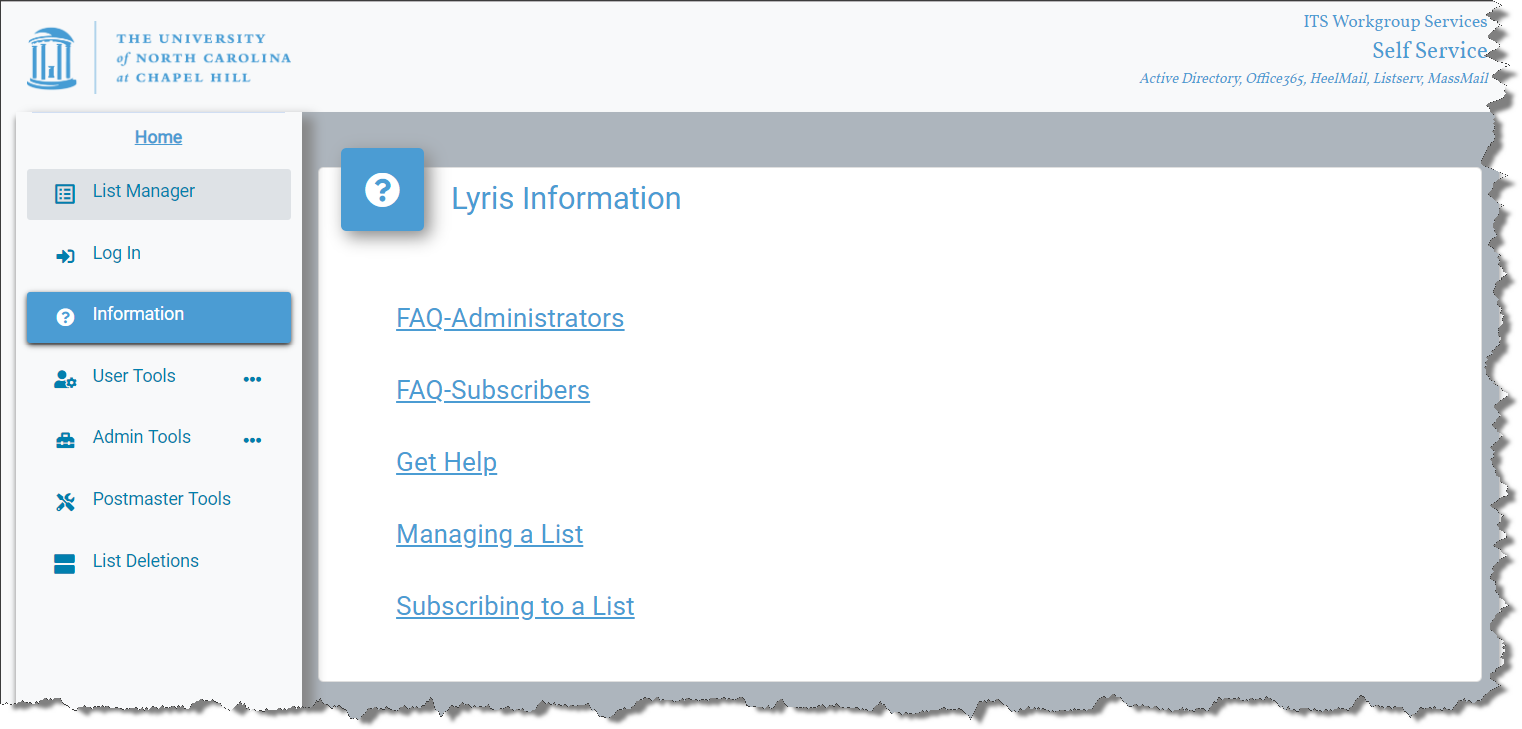
## List Manager



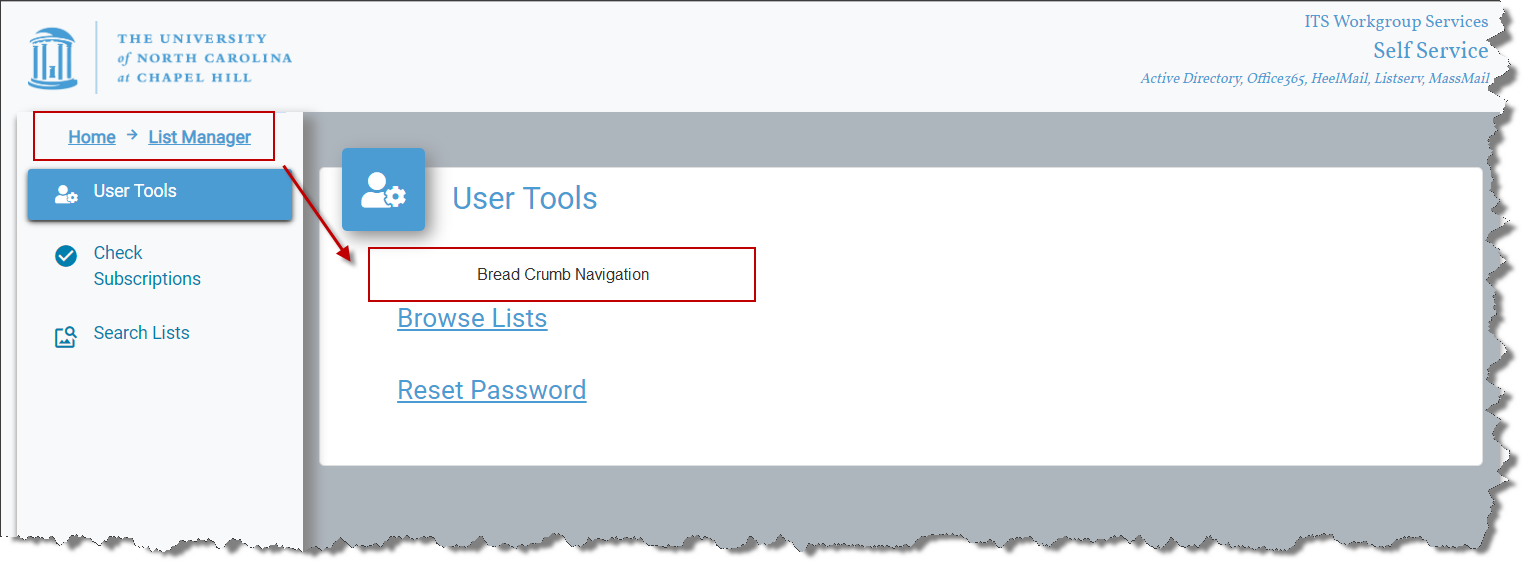
## Lyris List Login



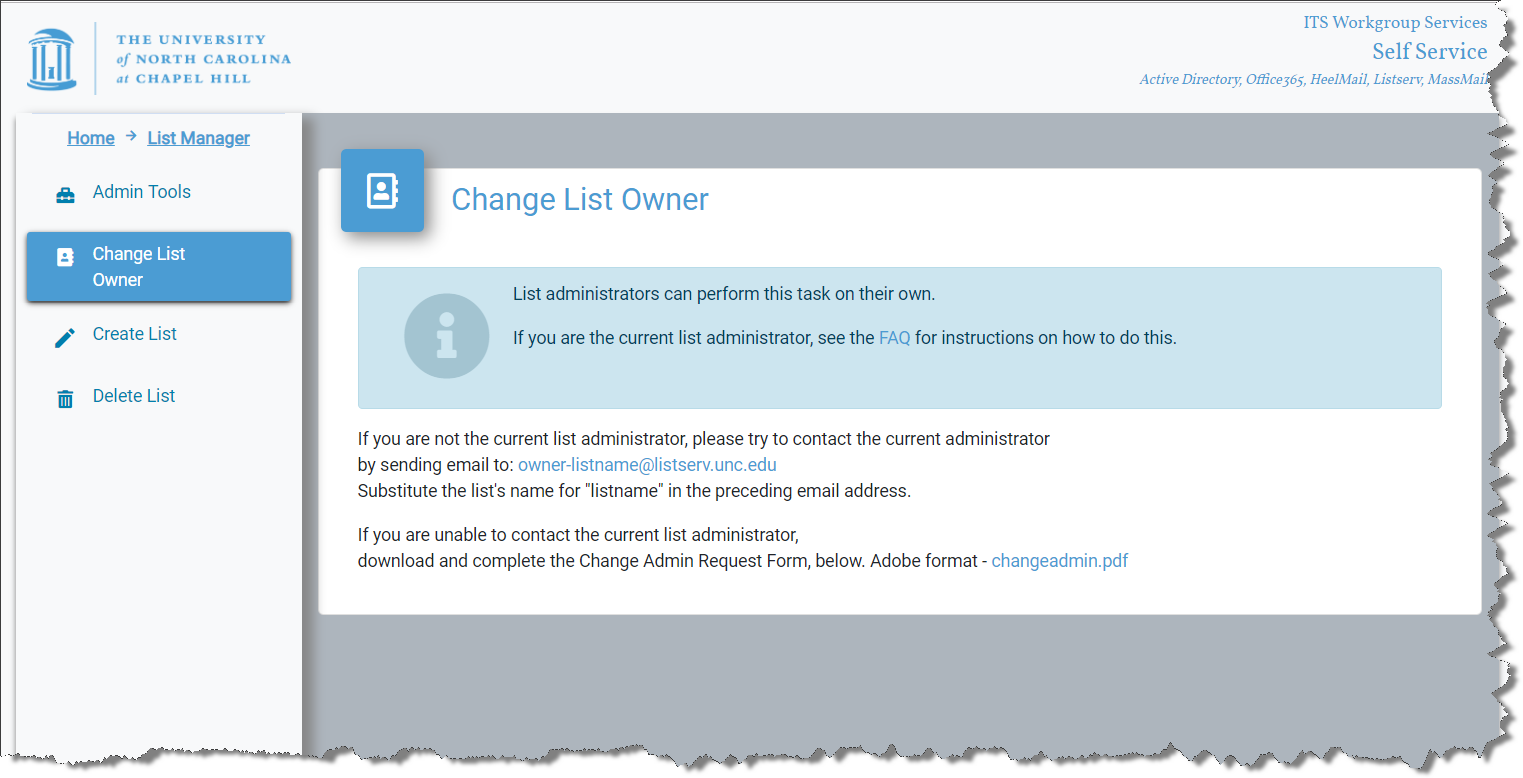
## Lyris Information



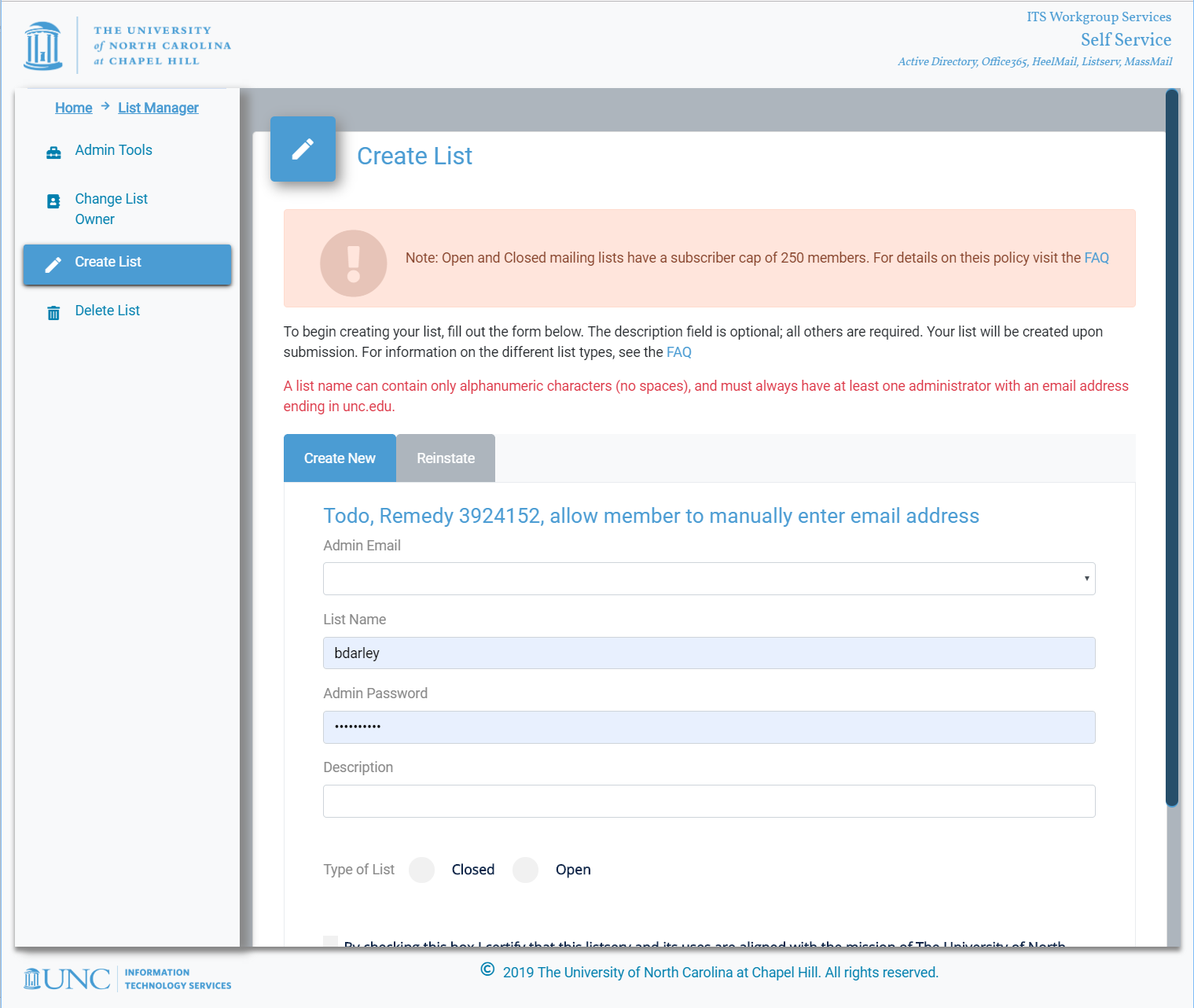
## Lyris User Tools



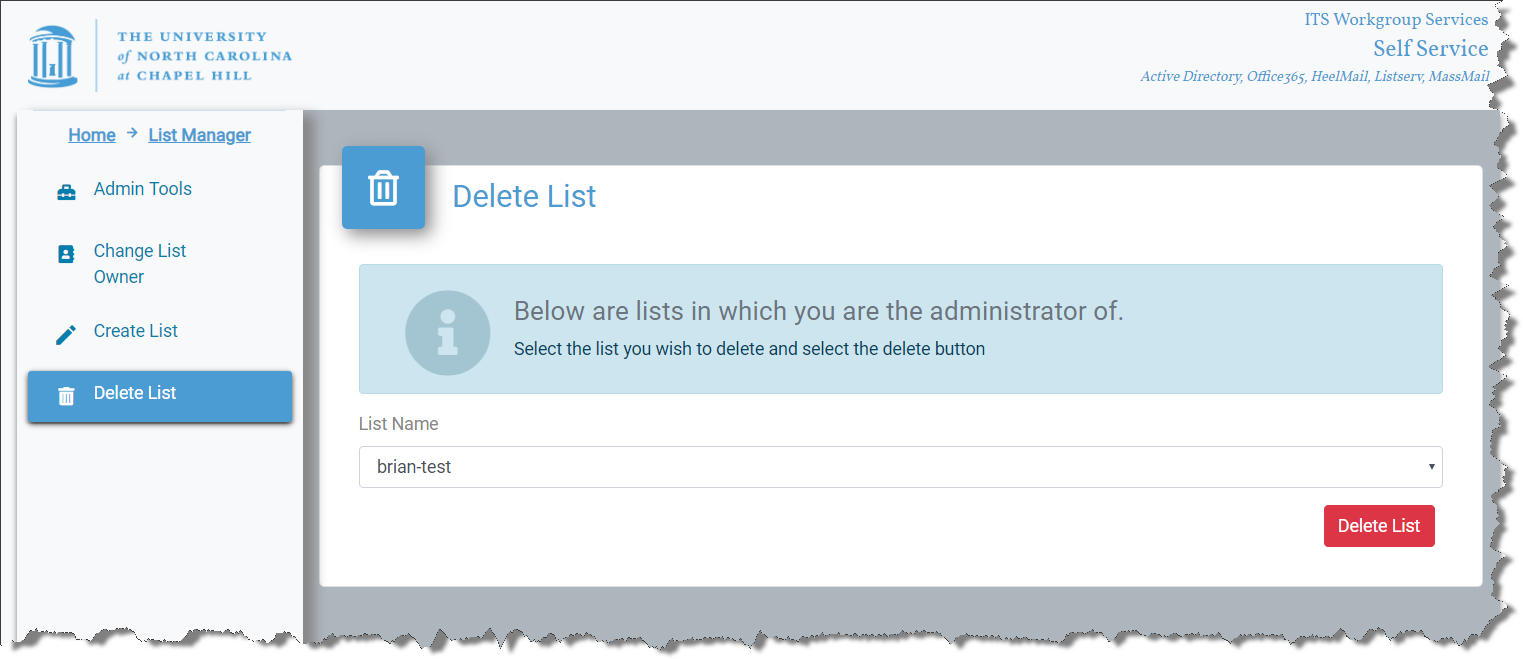
## Lyris Change List Owner



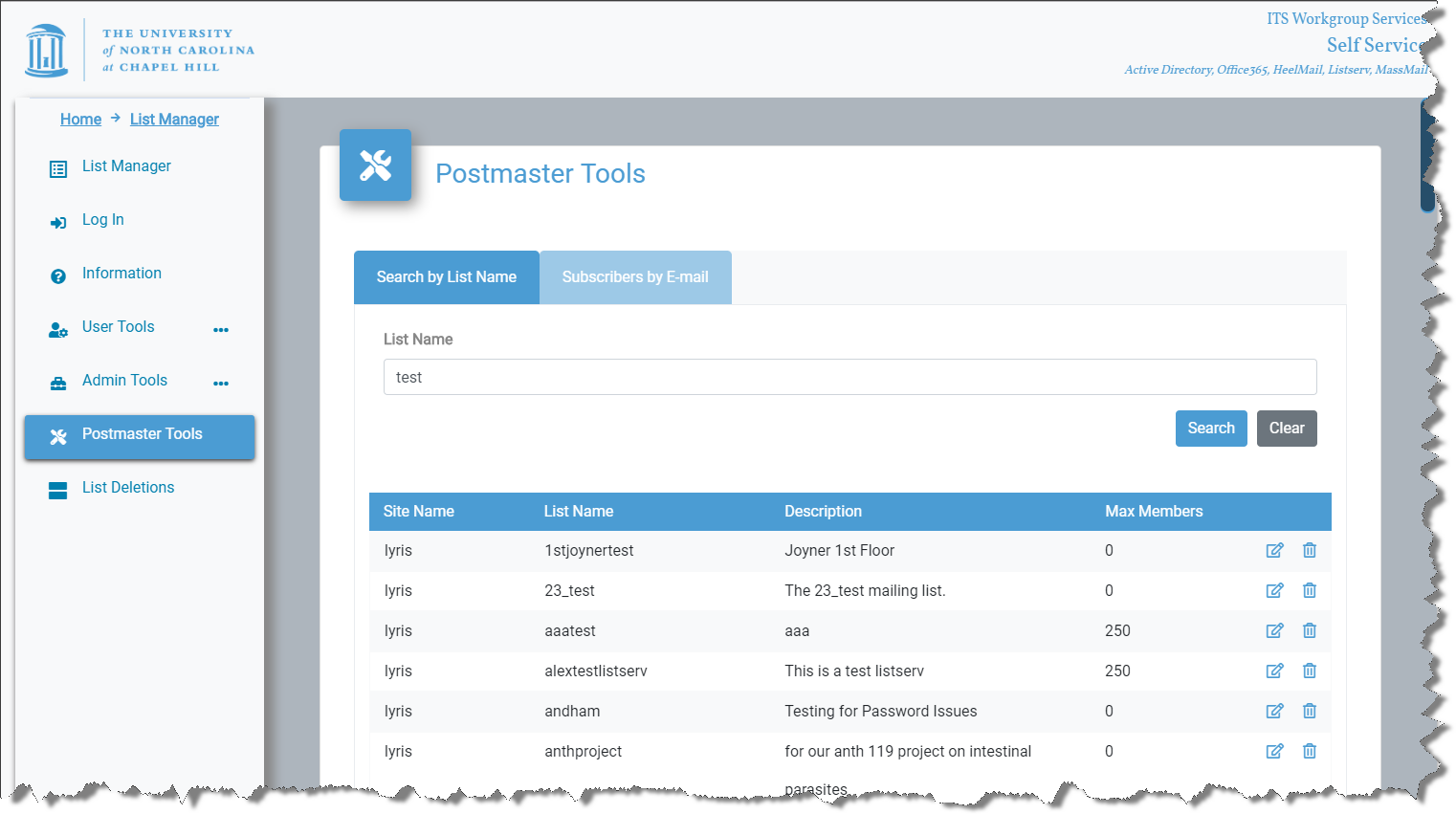
## Lyris Create List

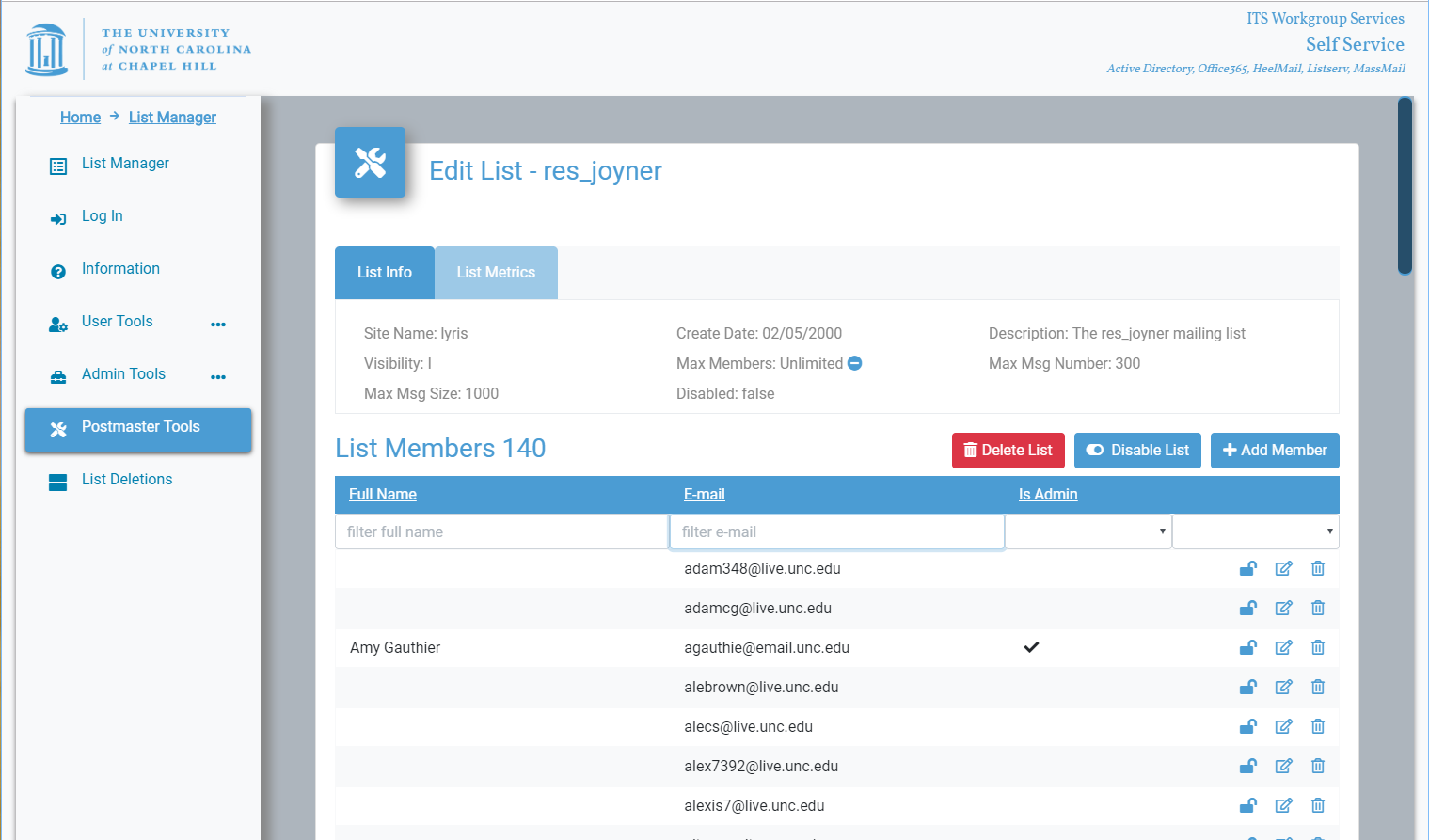


## Lyris Delete List

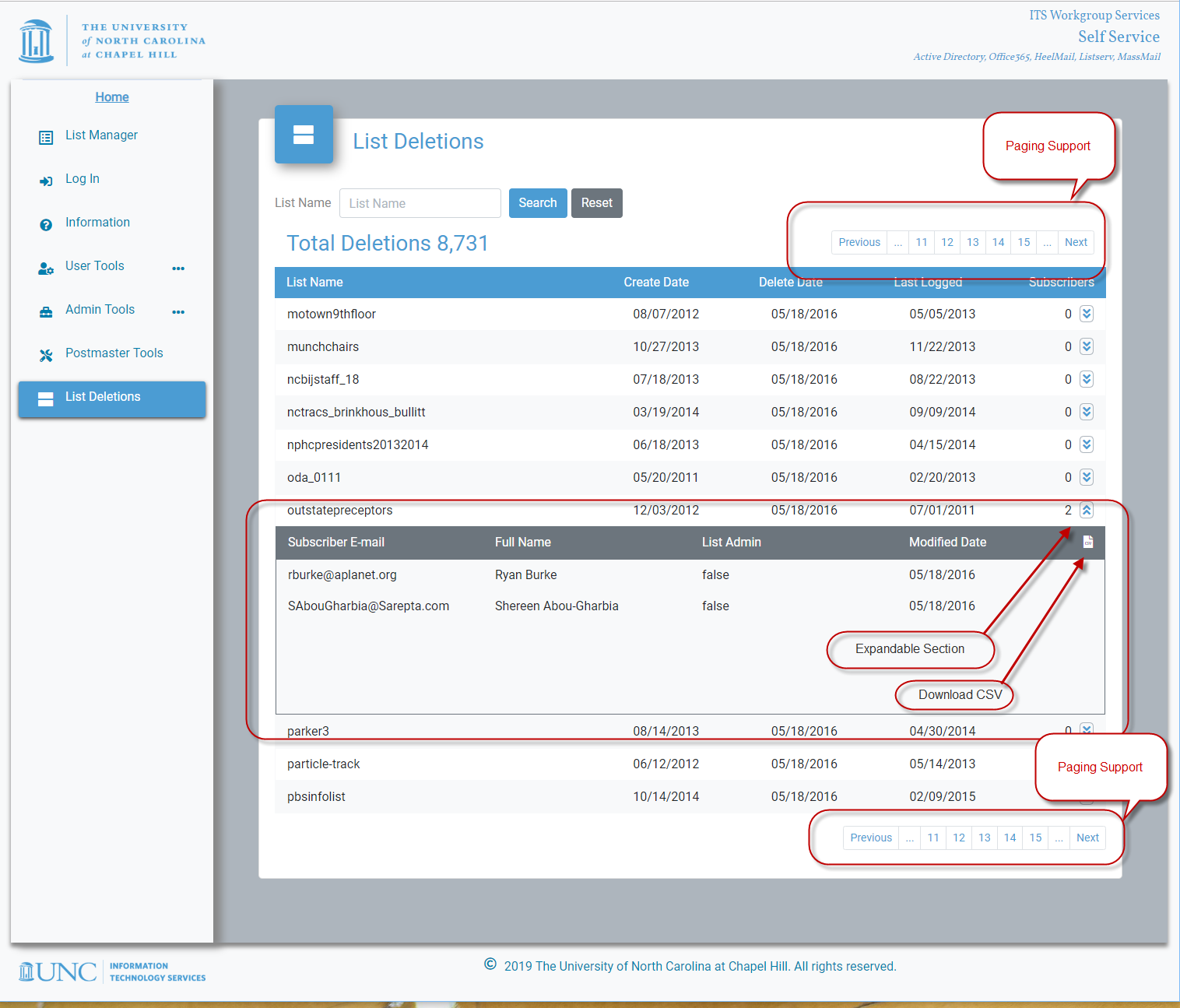


## Lyris Postmaster Tools

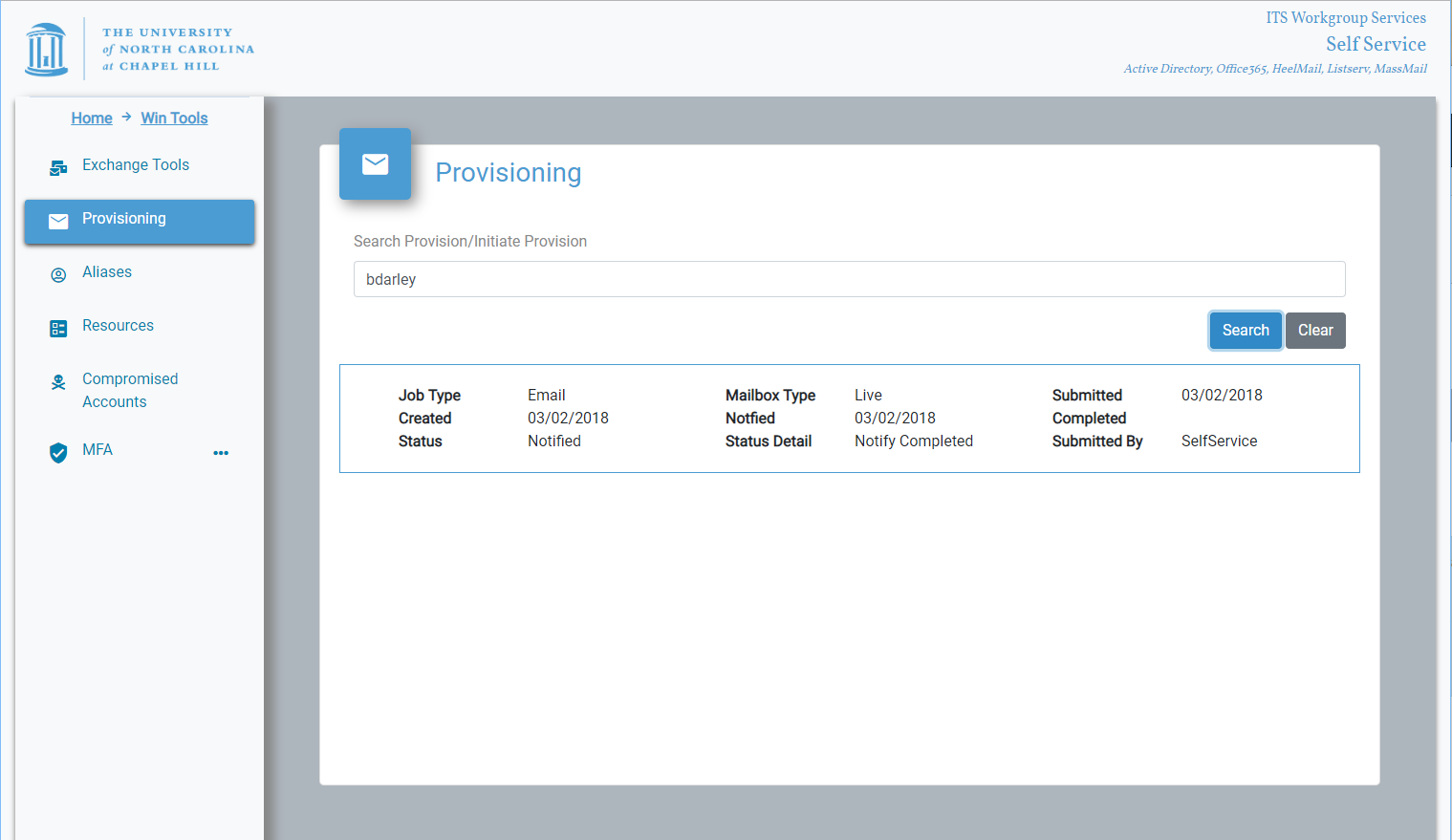


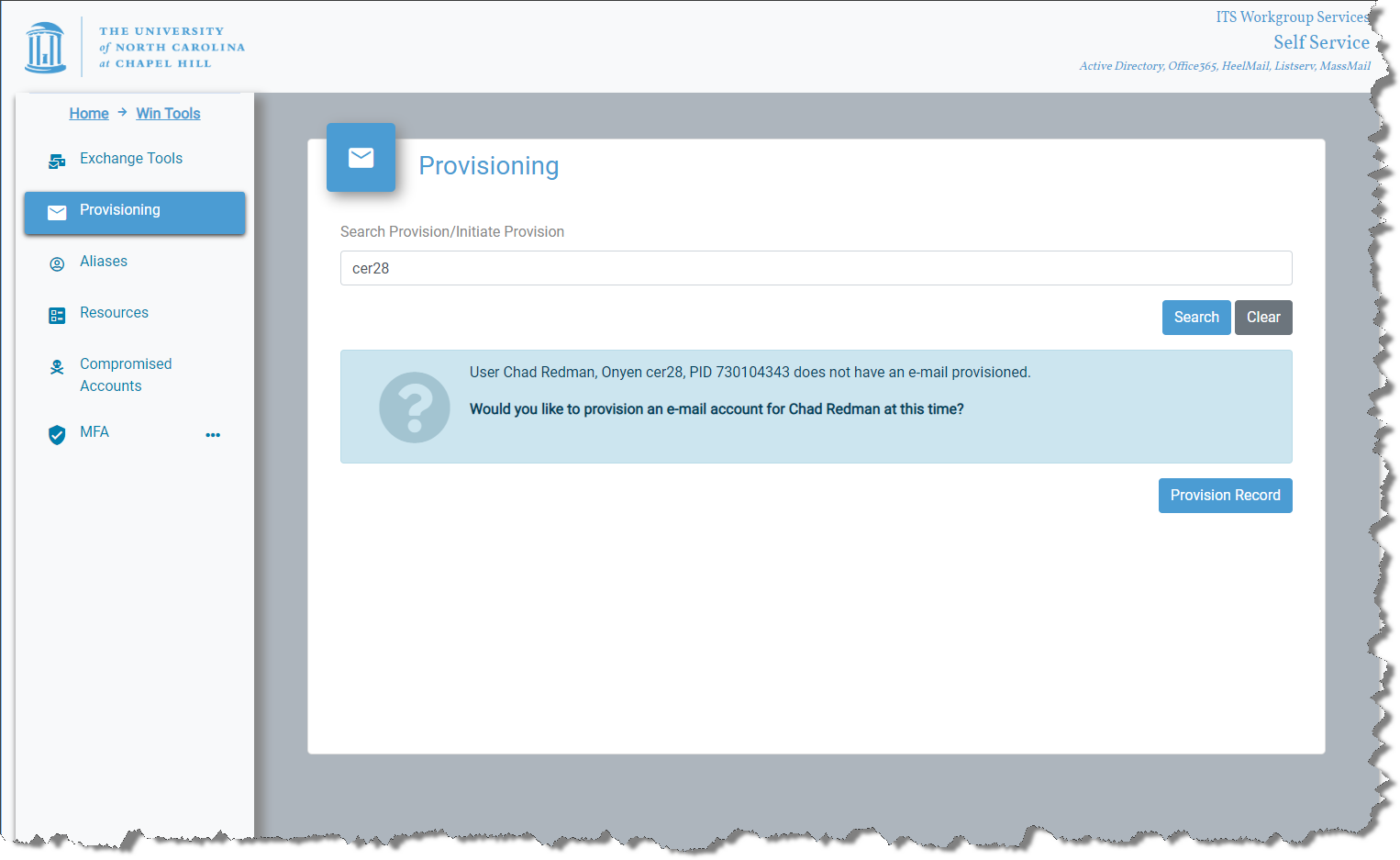


## Lyris List Deletions

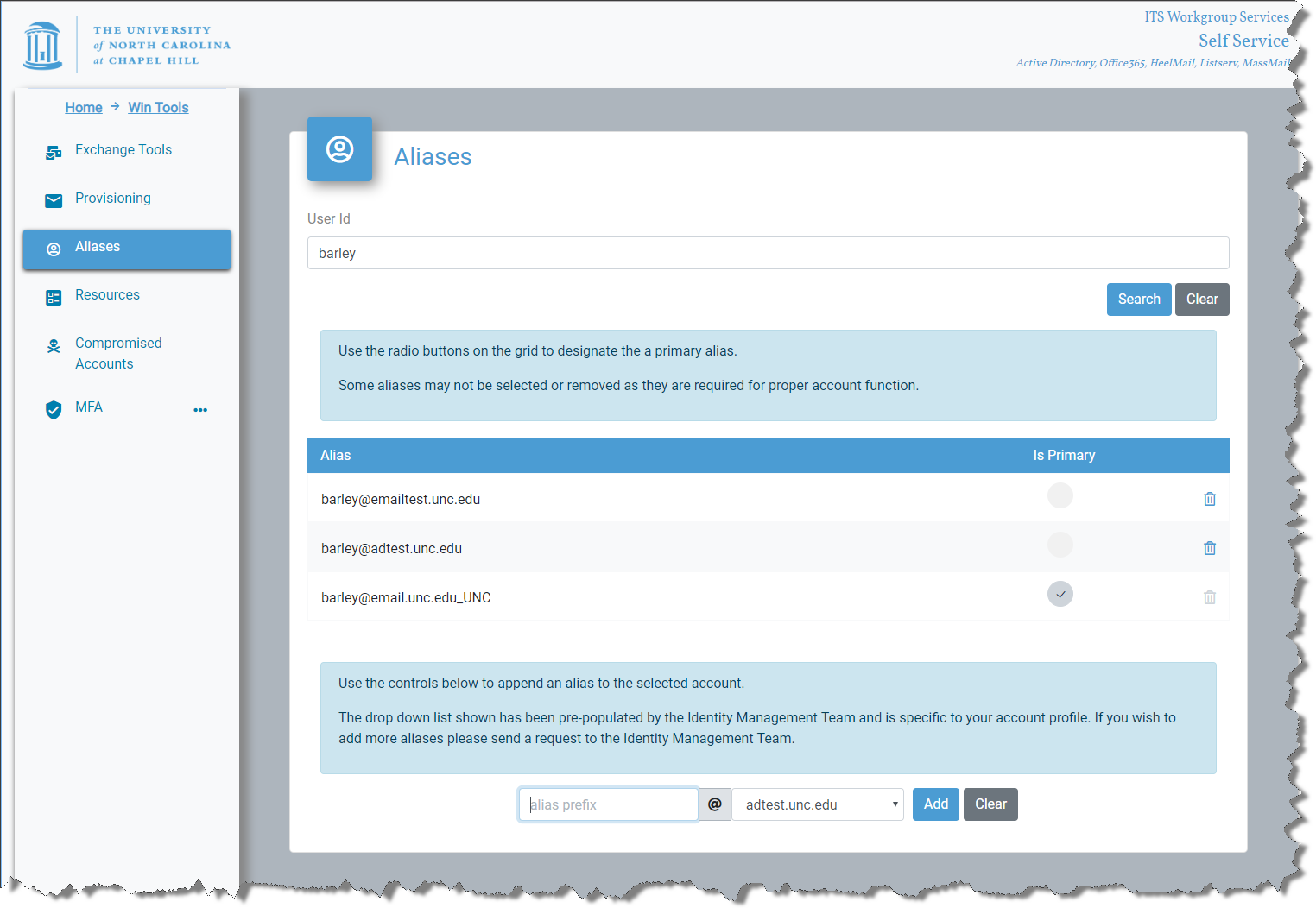


## Win-Tools Exchange Tools

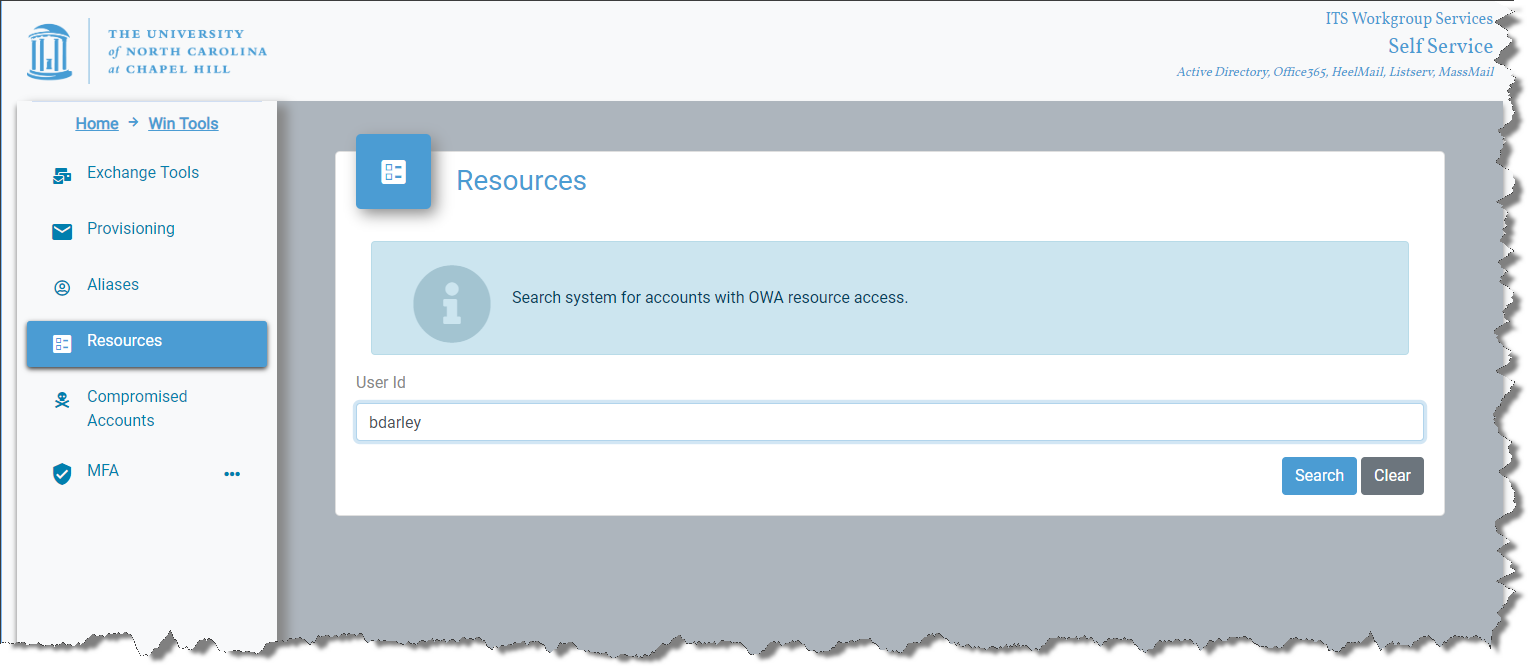




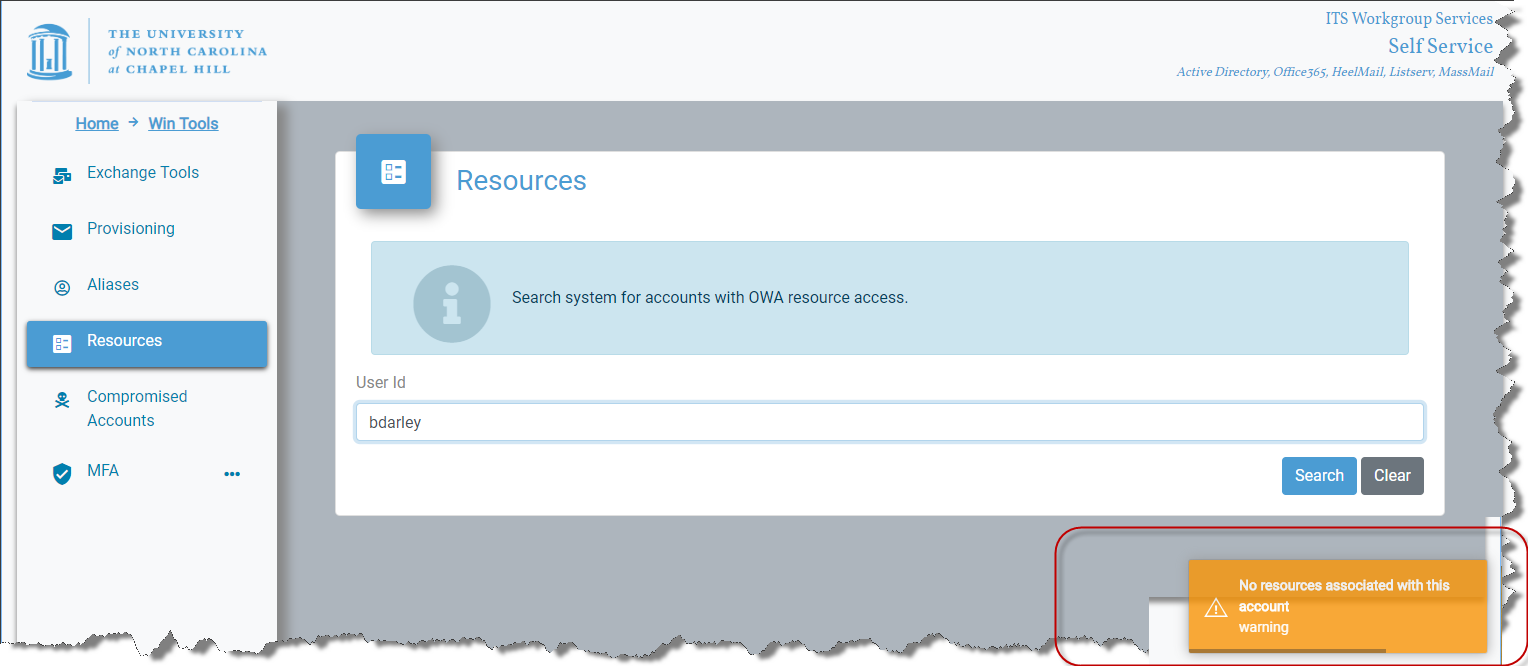
## Win-Tools Provisioning



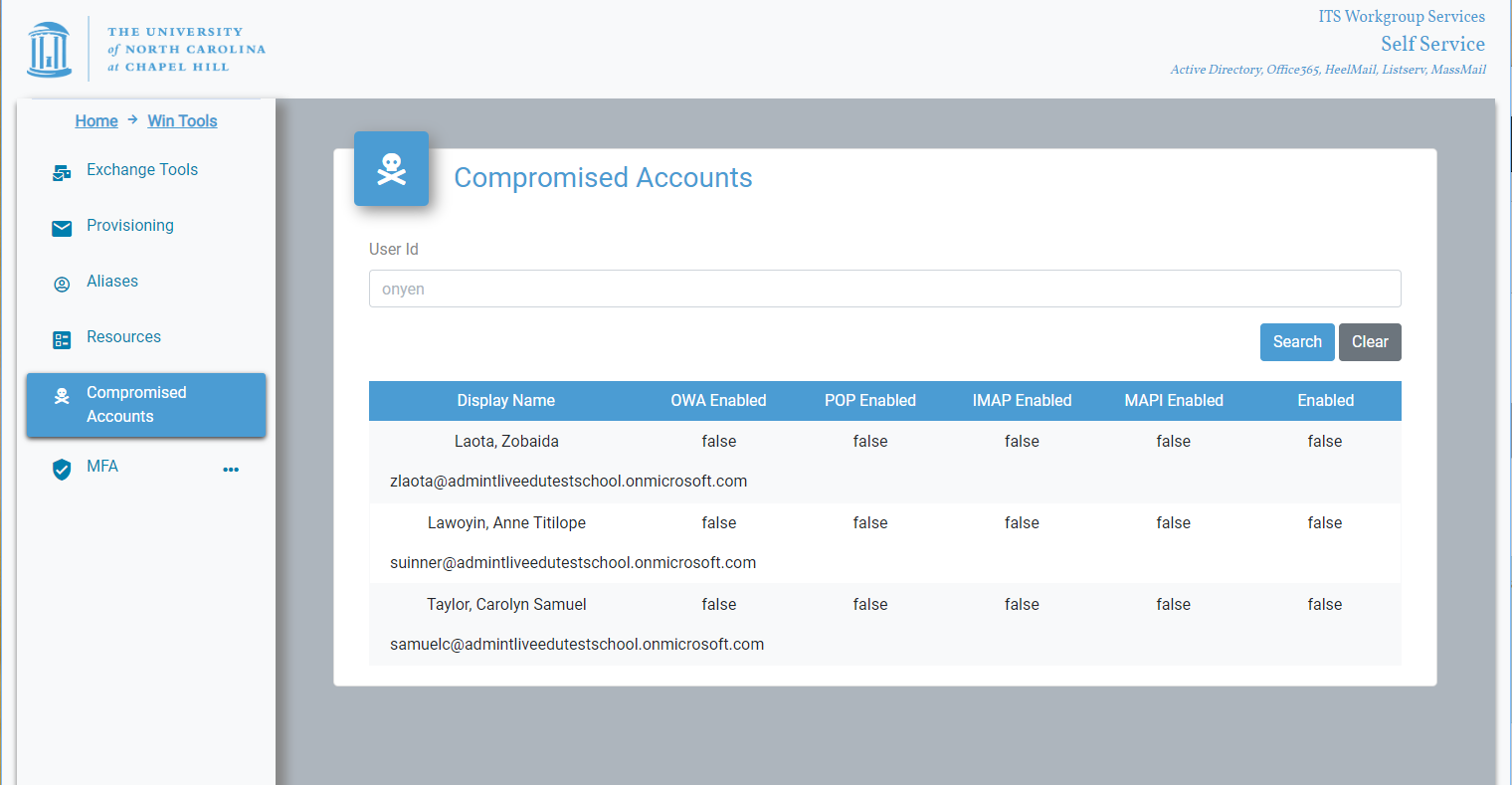
## Win-Tools Aliases



## Win-Tools Resources



## Win-Tools Compromised Accounts



## Win-Tools MFA

