STaff Ad Resource MANagement

(STARMAN)

Functional Specification

Robert Gerard Holohan

C00003778

Table of Contents

Table of Figures

1 Introduction

Functional managers within SETU Carlow Campus are responsible for the access to a range of different IT resources. These resources include, but are not limited to

- File storage folder on local File and Print solutions
- Mail distribution groups
- Phone dial-out privileges

ZZZZZ is a facility to request the provision and removal of access to specific resources with a functional area. The system will complement an existing staff account creation system.

1.1 Current situation

South-East Technological University (SETU) currently uses Active Directory (AD) for the local management of all its user and group objects. SETU also has an Azure Active Directory (AAD) instance. Individual users have access to a range of on-site and cloud resources, access to which is managed through AD and AAD respectively.

Locally, access to shared secured resources such as file storage and printing resources are managed through AD groups. Groups exist for specific functional areas such as Finance, HR, academic departments etc. Membership of these groups grants access to shared resources owned by the functional area head.

AD accounts for new staff are created through a bespoke system. The system processes individual requests from HR to create new accounts. This processing involves sending a series of emails to prospective new staff through their personal email. The emails request the staff member to provide a new secure password within a specified time frame to activate their accounts.

New staff are assigned a series of one or more default AD group memberships, based on their job titles.

Access to function-specific resources is currently managed manually. This can happen in one of a number of ways: -

- The function head send an email or makes a phone call to a staff member of Computing Services Carlow campus;
- A request is submitted to the Helpdesk ticketing system;
- Another member of staff in the functional area requests access via email or phone call;
- The new staff member, on realising they do not have access, requests access themselves;

Ordinarily, requests are only processed when received from the function manager. However, there is currently no definitive control to ensure that this is the case.

1.2 Issues

There are a number of issues with this current arrangement: -

- A new staff member does not have timely access to a resource they require to complete their work;
- Staff members may be given access to resources they do not need or should not have access to;
- Informal requests via email or phone call may not be actioned;
- When a staff member moves, their access to resources are not revoked;

2 System Overview

This application consists of two distinct parts. The first part, the front-end, is where the users will interact with the system to submit requests and monitor their progress and outcomes. The second part, the back-office element, will record all requests, process them and provide responses to enquiries about their status and progress.

2.1 Front-end system

The front-end system will be a web-based platform. The system will initially require the user to be authenticated through Microsoft Azure single sign-on. Access to the system will be managed through AAD group membership. These groups will be created in on-premise Active Directory in the existing infrastructure and synchronised to AAD through existing processes.

The web application will be developed using Microsoft C# and will be hosted on the Microsoft Internet Information Services (IIS) platform. User experience will be enhanced using JavaScript where appropriate.

The web application will operate on a secure HTTP (HTTPS) platform, using Microsoft Azure single sign-on authentication to validate users. Communications between the application and server will be authenticated using session tokens in line with industry best practice.

2.2 Back-office system

The back-office system will be a console application developed using Microsoft C#. This application will use Microsoft SQL Server for its database management. The application will run on a timed loop. The application will use a set of web services to carry out any necessary operations and interactions with both the database and the Microsoft AD environment.

The back-office system will check the system's database tables for outstanding requests. These requests will be processed appropriately with the outcomes stored in the database for reference. Successful completion of requests will be notified to the requesting end-user via email.

3 Users

There are two categories of users who will interact with the system.

3.1.1 Functional area managers

These users will have access to the system front-end to interact with the system as designed. Each user in this category will have access to all the functions of the front-end system. The user can apply the functionality to the staff and resources they manage and can view the status of their own requests and review their actions.

3.1.2 Administrators

The system will be administered by agreed technical support staff within the Computing Services department. These individuals will have access to the AD security groups that control system access. Key individuals in this group will also have access to the background data tables to maintain front-end user settings. These will include the resources managed by each end-user.

4 Functional Specification

There are a series of functions that the system must support. As the system will consist of two elements, a user front-end and a back-office system, the main functionalities will be separated here.

4.1 Overall functionality

The overarching functionality of the system is to provide the following features to its user base:-

- List resources managed by the user;
- List staff managed by the user;
- List existing resource allocations per user;
- Grant a managed staff member access to a specific resource;
- Revoke access to a specific resource from a managed staff member;
- Review the status of any/all requests made;
- Review a log of actions taken;
- Receive notification when a request is processed;

4.1.1 Front-end functionality

The user front-end will act as the interface between the user and the back-office system that processes requests.

4.1.2 Back-office functionality

The back-office system will interrogate the system's database tables to identify and process requests for actions.

4.2 Use Cases

In software development terms, a use case describes how a system is used to achieve its functionality. Use cases outline the interactions between users (often referred to as actors) and the application being developed.

A use case consists of a descriptive diagram, highlighting the actions completed and the interactions required to make them happen. Figure 1 shows the Use Case Diagram for the End-User. A detailed description of each step is also included in the use case.

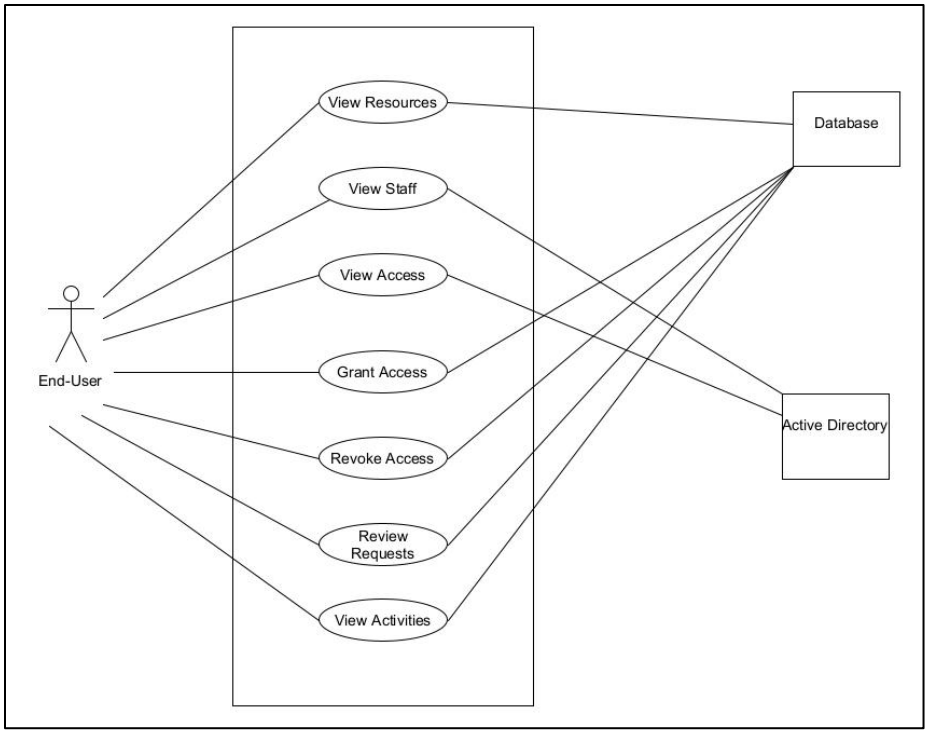

Figure 1: Use Case diagram - End-User

4.2.1 View Resources

Use Case	View Resources
Actors	End-user, database
Description	This process requests a list of the resources managed by the user from the database. The returned list will contain the names of all AD security groups the user is identified as a manager of. The list will include the group name and the AD description field for the group. Details of the request will be added to the activity log for the system. These details will include date/time, username, action taken and outcome
Success indicator	A list of one or more resources is displayed on the screen
Alternative options	It is possible that a new manager may not have any resources associated with them. In this case, the manager can submit a request to have resources assigned appropriately

4.2.2 View Staff

Use Case	View staff
Actors	End-user, AD
Description	This process requests a list of all staff users in AD who are indicated as being members of the department managed by the end-user. The list will contain the AD display name and description fields for each managed user. Details of the request will be added to the activity log for the system. These details will include date/time, username, action taken and outcome.
Success indicator	A list of one or more user objects is displayed on screen
Alternative options	It is possible that a list may not be displayed. There are a number of potential causes for this: - • The end-user's department name does not match that used in AD for other staff • Access to AD is not available at the time of submitting request • The end-user may not be assigned to the expected department (due to a change in role)

4.2.3 View Assignments

Use Case	View assignments
Actors	End-user, AD
Description	The end-user selects a specific resource they manage and a list of users with access to the resource is returned. This list may include people that are not part of the manager's staff. This will be due to some resources being shared between departments. The end-user will be able to see everyone with access to their resource but will only be able to revoke access for staff they manage. Removal of others from the list will have to be done by means of a helpdesk ticket. Details of the request will be added to the activity log for the system. These details will include date/time, username, action taken and outcome
Success indicator	A list of one or more resource users will be displayed on the screen
Alternative options	It is possible that no users have been assigned to a resource. This will be especially true of any new resources that have been created.

4.2.4 Grant Access

Use Case	Grant access
Actors	End-user, database
Description	The end-user selects a resource that they manage and one or more staff members they wish to grant access to. Once the user submits the request, a record is stored in the system's database for each user selected. These requests will be processed by the back-office application in due course. Details of the request will be added to the activity log for the system. These details will include date/time, username, action taken and outcome
Success indicator	A message on the screen will indicate the status of the processing of the request.
Alternative options	If the initial request submission fails for any reason, an appropriate error message will be displayed on the screen for the end-user to review and act upon.

4.2.5 Revoke Access

Use Case	Revoke Access
Actors	End-user, database
Description	The end-user selects a resource that they manage. A list of the users with access will be displayed. The end-user selects user(s) from the list. Once submitted, a record is stored in the database for each person selected from revocation. Once the user submits the request, a record is stored in the system's database for each user selected. These requests will be processed by the back-office application in due course. Details of the request will be added to the activity log for the system. These details will include date/time, username, action taken and outcome
Success indicator	A message on the screen will indicate the status of the processing of the request
Alternative options	If the initial request submission fails for any reason, an appropriate error message will be displayed on the screen for the end-user to review and act upon.

4.2.6 Review Requests

Use Case	Review Requests
Actors	End-user, database
Description	The end-user requests a status list of all requests submitted to date. This list will be in descending date and time order and will show the request type and the status of the operation. Details of the request will be added to the activity log for the system. These details will include date/time, username, action taken and outcome
Success indicator	A list of requests submitted by the end-user will appear on the screen.
Alternative options	A blank list may be returned if the user has not submitted any requests to date. A suitable message will be displayed explaining the outcome.

4.2.7 View Activities

Use Case	View Activities
Actors	End-user, database
Description	The end-user requests a list of all their activities to date. This list will be in descending date and time order and will show a summary of each activity completed within the system by the specific end-user. Details of the request will be added to the activity log for the system. These details will include date/time, username, action taken and outcome
Success indicator	A list of activities carried out by the end-user will appear on the screen
Alternative options	

As indicated earlier, there are two separate and distinct user types for this system. The second type of user is the Administrator. Figure 2 below shows the Use Case Diagram for the Administrator function of the system.

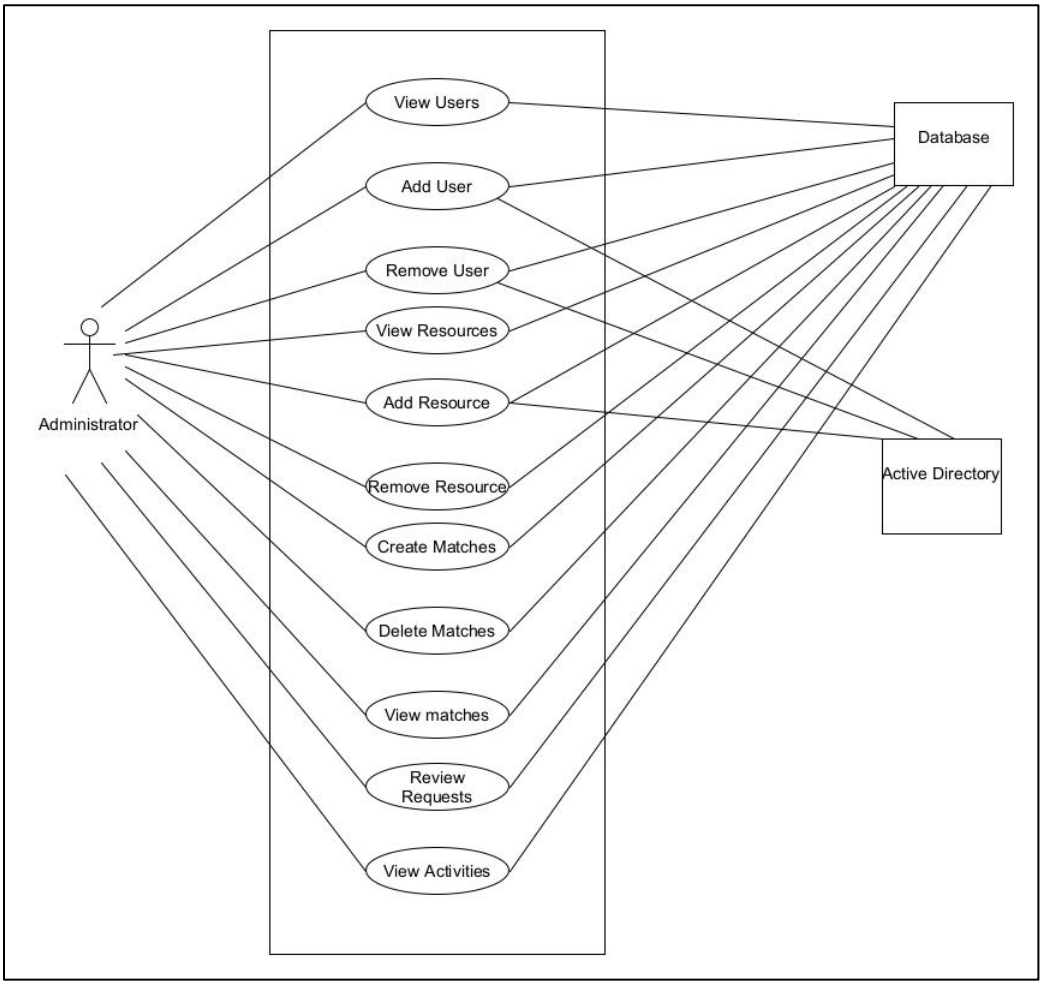

Figure 2: Use Case Diagram – Administrator

4.2.8 View Users

Use Case	View Users
Actors	Administrator, database
Description	The administrator requests a list of existing system users. This list will be in alphabetical order and will include job title and department.
Success indicator	A list of users will appear on the screen.
Alternative options	An error message will appear if a list is not generated (No users in the system, no access to database)

4.2.9 Add User

Use Case	Add User
Actors	Administrator, database, Active Directory
Description	The administrator will review a list of users in Active Directory and will select the one(s) to be added to the system. The selected user will be added to the user database. Details of the request will be added to the activity log for the system. These details will include date/time, username, action taken and outcome
Success indicator	A message on the screen indicates success. The new user will also be visible in the list of users
Alternative options	In the event of a failure to add a user, a detailed message will be displayed. This may be due to the existence of a user with the same name, the database may not be available

4.2.10 Remove User

Use Case	Remove User
Actors	Administrator, database, mail server
Description	The administrator will review a list of users and select one or more for removal. The list of managed resources will be reviewed and any entries for the user(s) will be removed. Once this is done, the individual user will be removed. An email notification will be sent to the removed user. Details of the request will be added to the activity log for the system. These details will include date/time, username, action taken and outcome
Success indicator	A message on the screen indicates success. The removed user(s) will also no longer appear in the list of users.
Alternative options	In the event of a failure to remove a user, a detailed message will be displayed. This may be due to the database not being available.

4.2.11 View Resources

Use Case	View Resources
Actors	Administrator, database
Description	The administrator requests a list of current resources within the system. A list of resources is returned from the database. This list is in alphabetical order and includes details of the resource owner. Details of the request will be added to the activity log for the system. These details will include date/time, username, action taken and outcome
Success indicator	A list of one or more resources is displayed on the screen.
Alternative options	If a list is not generated, a relevant message is displayed on the screen (no resources exist, database access issue)

4.2.12 Add Resource

Use Case	Add Resource
Actors	Administrator, database, Active Directory
Description	The administrator will review a list of resources in Active Directory. From this list, the administrator will select one or more resources to be added to the system. Details of the request will be added to the activity log for the system. These details will include date/time, username, action taken and outcome
Success indicator	A message is displayed on the screen. The resource can also bee seen in a list of resources
Alternative options	In the event of a failure to add a resource, a detailed message will be displayed. This may be due to the database not being available or that the resource already exists.

4.2.13 Remove Resource

Use Case	Remove Resource
Actors	Administrator, database
Description	The administrator reviews a list of resources in the database and selects one or more for removal. Any user matches to a particular resource will be removed from the database. A notification email will be sent to each user who managed the deleted resource. Once this is complete, the resource will be removed. Details of the request will be added to the activity log for the system. These details will include date/time, username, action taken and outcome
Success indicator	A message of success will be displayed on the screen.
Alternative options	In the event of a failure to remove a resource, a detailed message will be displayed. This may be due to the database not being available.

4.2.14 Create Matches

Use Case	Create Matches
Actors	Administrator, database
Description	The administrator reviews a list of resources within the database. When a resource is selected, a list of users within the database is displayed. One or more users will be selected and assigned to manage the selected resource. An email notification will be sent to the selected user. Details of the request will be added to the activity log for the system. These details will include date/time, username, action taken and outcome
Success indicator	A message of success will be displayed on the screen. The match can also be viewed in the list of matches.
Alternative options	In the event of a failure to create a match, a detailed message will be displayed. This may be due to the database not being available.

4.2.15 Delete Matches

Use Case	Delete Matches
Actors	Administrator, database
Description	The administrator will review a list of matches. One or more matches will be selected for deletion. Once deleted, an email notification will be sent to the user. Details of the request will be added to the activity log for the system. These details will include date/time, username, action taken and outcome
Success indicator	A list of matches will appear on the screen. This will be sorted in resource and end-user order.
Alternative options	In the event of a failure to delete a match, a detailed message will be displayed. This may be due to the database not being available.

4.2.16 View Matches

Use Case	View Matches
Actors	Administrator, database
Description	The administrator requests a list of existing matches from the database. The list will be sorted in order of resource name and username. Details of the request will be added to the activity log for the system. These details will include date/time, username, action taken and outcome
Success indicator	A list of matches appears on the screen.
Alternative options	In the event of a failure to retrieve a list of matches, a detailed message will be displayed. This may be due to the database not being available or no matches existing in the system.

4.2.17 Review Requests

Use Case	Review Requests
Actors	Administrator, database
Description	The administrator requests a status list of all requests submitted to date. This list will be in descending date and time order and will show the request type and the status of the operation. Details of the request will be added to the activity log for the system. These details will include date/time, username, action taken and outcome
Success indicator	A list of requests submitted by the end-user will appear on the screen.
Alternative options	A blank list may be returned if the user has not submitted any requests to date. A suitable message will be displayed explaining the outcome.

4.2.18 View Activities

Use Case	View Activities
Actors	Administrator, database
Description	The end-user requests a list of all their activities to date. This list will be in descending date and time order and will show a summary of each activity completed within the system by the specific end-user. Details of the request will be added to the activity log for the system. These details will include date/time, username, action taken and outcome
Success indicator	A list of activities carried out by the end-user will appear on the screen
Alternative options	A blank list may be returned if the user has not submitted any requests to date or there is an issue with the database. A suitable message will be displayed explaining the outcome.

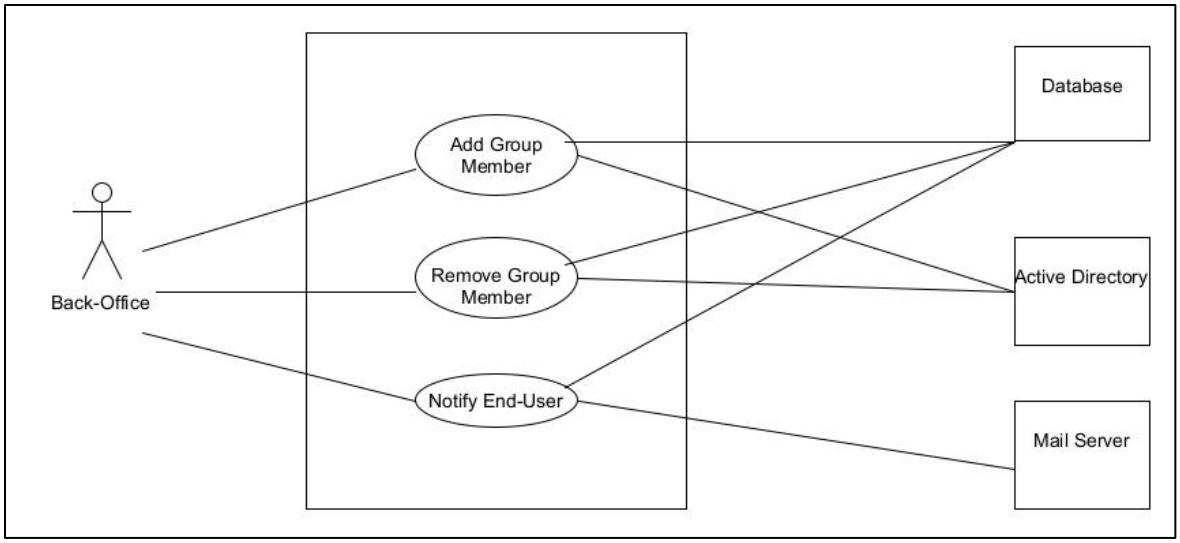

Figure 3: Use Case Diagram - Back-office

4.2.19 Add Group Member

Use Case	Add Group Member
Actors	Back-office system, Web service, AD
Description	The back-office application queries a database table and processes the resulting record set (if any). A web service call to add a user to the indicated AD security group is made for each record returned in the query. Details of the request will be added to the activity log for the system. These details will include date/time, username, action taken and outcome
Success indicator	The web service returns a response to indicate the status of the request. This response will be stored in the database in the request record. If the response indicates success, a corresponding flag will be set on the record.
Alternative options	The result of the web service request is stored in the relevant database record. If the request is unsuccessful, the record will be reprocessed on the next iteration of the back-office process.

4.2.20 Remove Group Member

Use Case	Remove Group Member
Actors	Back-office system, web service, AD
Description	The back-office application queries a database table and processes the resulting record set (if any). A web service call to remove a user from the indicated AD security group is made for each record returned in the query. Details of the request will be added to the activity log for the system. These details will include date/time, username, action taken and outcome
Success indicator	The web service returns a response to indicate the status of the request. This response will be stored in the database in the request record. If the response indicates success, a corresponding flag will be set on the record.
Alternative options	The result of the web service request is stored in the relevant database record. If the request is unsuccessful, the record will be reprocessed on the next iteration of the back-office process.

4.2.21 Notify User

Use Case	Notify User
Actors	Back-office system, web service
Description	The back-office application queries a series of database tables and processes the resulting record set (if any). The records will be grouped by end user and an email will be sent to each end user found. The email will indicate the actions requested and their successful completion. If there is more than one action for a specific end user, a table of actions and outcomes will be contained in the email. On successful submission of the email, each relevant record will be updated to reflect that an email has been sent. This will prevent duplicate emails being sent. Details of the request will be added to the activity log for the system. These details will include date/time, username, action taken and outcome
Success indicator	The web service returns a response to indicate the status of the request. This response will be stored in the database in the request record. If the response indicates success, a corresponding flag will be set on the record.
Alternative options	The result of the web service request is stored in the relevant database record. If the request is unsuccessful, the record will be reprocessed on the next iteration of the back-office process.

5 FURPS+

FURPS is an acronym for **F**unctionality, **U**sability, **R**eliability, **P**erformance and **S**upportability. This section of the document deals with each of these heading in more details. The + symbol is appended to the term to cover any other requirements of functionality that does not fall into the previous 5 categories.

5.1 Functionality

This category deals with the feature set and functionality of the required system.

This system is designed to provide functional managers (or their identified agents) with the ability to indirectly grant access to managed resources to their staff. By using this system, access can be granted without the user needing privileged access or specialised training in the background systems.

The key functionalities required for the end user are:-

1. View who currently has access to a specific resource
2. Request access to a resource for specified individuals
3. Revoke access to a resource for specified individuals
4. Track the progress of previous requests

5.2 Usability

The system is required to have a user-friendly interface that allows the user to request access to resources they manage, and no other resources. The interface should be form based, with operational choices made using keyboard and mouse as appropriate. The system must be able to identify the user, know what resources the control and what staff they can grant access to.

The system should provide a level of on-screen help and guidance where necessary and appropriate.

The system will deliver this functionality by use of background data tables of end-users and their managed resources. Staff users will be identified by means of agreed standard user properties within the existing AD infrastructure.

5.3 Reliability

This category covers the overall stability and availability of the system. Users will have an expectation that the system be available to them whenever possible. The following list are the expected reliability markers for the system: -

- The system should be available during reasonable working hours – 8am to 6pm Monday to Friday;
- Responses to user and resource list requests should be answered within 15 seconds;
- Assignment and de-assignment of resource access should be completed within 15 minutes of request being submitted;

To ensure this, industry standard development and hosting platforms will be used.

5.4 Performance

Performance refers to the speed of the system. It refers to the response times that a user should expect from the system. As there are two main aspects to this system, there will be two areas of performance. The first area will be the performance of the user front-end. The performance of the front-end will be dependent on several different factors including: -

- Web server specification and load
    - Based on prior levels of requests, it is estimated that there will be approximately 10 requests per day at the start of each semester
- Database server specification and load
    - Each request for a list or assignment/de-assignment of resources will require the execution of 1-3 SQL statements – 30 per day
    - Each iteration of the back-office system will require 4-6 SQL statements per iteration – 36 per day
- Efficiency of design and implementation of the software solution

The second area is the back-office system also has several dependencies: -

- Frequency of iteration of the looping strategies of the software
- Performance of the Domain Controller(s) on the AD network

5.4.1 End-User request response times

Table 1 outlines the desired response times for each request submitted by an end-user.

Table 1: End-User request response times

Action	Time
View Resources	15 seconds
View Staff	15 seconds
View Access	15 seconds
Grant Access	5 seconds
Revoke Access	5 seconds
Review Requests	15 seconds
View Activities	30 seconds

5.4.2 Administrator response times

Table 2 outlines the desired response times for each request submitted by an administrator.

Table 2: Administrator request response times

Action	Time
View Users	15 seconds
Add User	5 seconds
Remove User	5 seconds
View Resources	15 seconds
Add Resource	15 seconds
Remove Resource	15 seconds
Create Matches	5 seconds
Delete Matches	5 seconds
View Matches	15 seconds
Review Requests	15 seconds
View Activities	30 seconds

5.4.3 Back-office response times

Table 3 outlines the desired response times for each request processed by the back-office application. These times refer to the time taken to complete the task by the console application. They do not refer to the time taken from the submission of the request by the user to completion.

Table 3: Back-office response times

Action	Time
Add Group Membership	15 seconds
Remove Group Member	15 seconds
Notify End-User	15 seconds

5.5 Supportability

This category deals with the ongoing maintenance and expansion of the system. As end-users become familiar with the functionality of the system, additional access requirements may be identified. The system should be designed to ensure that additional resource assignments can be made available with the minimum amount of redesign and redevelopment.

To improve the supportability of the system, a set of technical and user documentation will be produced. This will be available to designated support personnel as required.

All produced code will be fully and unambiguously documented to facilitate maintenance by third parties.

5.6 +

This category covers any non-functional aspects and requirements of the system.

5.6.1 Security and Access Control

As the system will provide end-users with the ability to grant access to resources to other individuals, it is essential that the level of security and access control on the system are as high as possible and meet with industry standard practices.

Initial access to the system will be managed using Microsoft Azure single sign-on technology. Only designated individuals will be able to access the user front-end. This will be controlled by means of AD security group membership. Each identified user will be added to a pre-defined security group by the system administrator(s). This group will be the only ones with access to log into the system itself.

The individual users and the areas the manage will be identified initially through contact with the HR department.

Once logged in, the individual user will be identified by the system through their login credentials. These details will be used to determine what resources the user manages. The user will only be able to grant/revoke and monitor access to the resources the manage.

The front-end application will be hosted on a secured web server. This will employ secure HTTP using a commercially granted 2048 bit certificate.

The front-end application will only be accessible from within the University network infrastructure and will also be restricted to a specific range of IP addresses to further limit access and risk.

All actions by users will be logged. The log will include date/time, username, action taken and outcome. All activities within this log will be available to view by technical support and each individual user can view their own actions. These actions will be grouped by date/time and action type.

6 Testing

As with all software systems, it will be necessary to carry out extensive testing of all aspects of the system. The software application will be tested manually at all stages of development. This will involve running each functional element of the code multiple times and observing the results where relevant. This project consists of three distinctly separate elements, and each will require its own testing processes.

Final testing of the system will be carried out by a combination of technical staff and end users with no technical background.

6.1 User Interface (UI)

As each aspect of the interface is built, its functionality will be manually tested. The function of each aspect of the interface will be noted. These notes will include details of the initial system state, what should happen when the interface element is activated and what the actual outcome is. This will be a completely manual process and screen shots will be kept of the different stages of testing. Failures in testing will be logged as part of the verification process.

During testing, the applications adherence to the defined performance criteria will be tested at all stages. Details of the performance found will be recorded and proof of testing. This information will also act as a performance benchmark for the product as it transitions to a live application.

6.2 Console Application (CA)

The console application uses data from the application database and interacts with Active Directory to gather information and implement changes. As the CA is being developed, each stage will be tested manually. This will involve stepping through the code to monitor the changes in variables and the results of database operations.

The database tables will be monitored to ensure that the expected changes (and only the expected changes) are being made. This will involve manually running SQL scripts against the database table using Microsoft SQL Management Studio.

The CA will be written to output details of its progress to the console, and this will be monitored during all aspects of testing. Screen shots will be kept during testing to verify results.

As with the UI, during testing, the applications adherence to the defined performance criteria will be tested at all stages. Details of the performance found will be recorded and proof of testing. This information will also act as a performance benchmark for the product as it transitions to a live application.

6.3 Web Services Platform (WSP)

Both the UI and CA consume services from the WSP. By testing the UI and CA fully, the WSP elements will also be tested.

6.4 Setting up a test platform

As this system is intended to be implemented in an existing live Active Directory (AD) environment, I determined that it would be best to carry out all testing on a live AD platform. SETU Carlow campus still maintains its ITCARLOW AD platform. To this end, I devised a strategy to test the application within this environment.

To facilitate my project, I requested permission from the Computing Services Manager at the Carlow Campus to use a specific area of the existing AD tree to my purposes. I requested an OU to be created and a standard user account be created and granted management rights to the specified OU.

7 Inspiration

In my daily work, I regularly meet people who are frustrated with a lack of access to resources they require. This can be anything from access to shared files, mailboxes, printing resources or even door access.

This situation has led me to a point where I am attempting to empower people with the facilities they need to grant access to resources they manage. Why should a manager have to wait for someone outside their department to grant access to a resource that the manager ultimately “owns”.

All of this has given me the inspiration to provide a simple solution that is always at the user’s fingertips to manage at least some of the things that make their life trickier on a daily basis.

8 Conclusion

In conclusion, this product should provide the end user with a simple to use and stable platform to locally manage access to network resources for staff. The process should be simple to use and allow for access requests to be processed more efficiently than by existing request methods outside the department. Above all, the system should provide a safe and secure means of maintaining user level resource access.

As access to the system is managed through Microsoft Azure Active Directory, the platform will be secure and protected using industry standard security methodologies.

There are no data entry fields in the system. Users can only make selections of the actions they wish to take within a confined environment. Users will only be able to interact with their own staff and resources. All of this will help to significantly reduce the risk of data corruption or compromise through the interface at any time.

The use of an API to handle all the back-office interactions ensures that the system can be operated by any authorised user with no need for special privileges.

Each API request will be secured using session tokens. Any request without a recognised valid session token will be ignored. Tokens are created and destroyed by the application as required.

The use of parameterised SQL for all database transactions will ensure that the integrity of the data platform the system needs. By protecting this, the Active Directory infrastructure the system maintains will also be protected.

The implementation of a full audit trail of all requests processed, and actions taken by all users will ensure that all events will be traceable. Not only does this provide a level of monitoring for errant actions, but it also can be used when determining the events leading up to any possible resource access issues or changes that were unexpected.

9 Glossary of Terms

AAD	Azure Active Directory
AD	Active Directory
CA	Console Application – the element of the project that processes user's requests stored in the database and ensures their completion as appropriate
DC	Domain Controller
ITCARLOW	Institute of Technology Carlow
MIIS	Microsoft Internet Information Services
SETU	South-East Technological University
UI	User Interface – This is the main application front-end that users have access to
WSP	Web Service Platform – the element of the project that handles all interactions between the Console Application and Active Directory and the system database

10 Declaration on plagiarism

Student	Robert Gerard Holohan – C00003778
Tutor	Dr. Joseph Kehoe
University	South East Technological University, Carlow Campus
Project Title	STaff Ad Resource MANagement (STARMAN)
Submission Date	April 2024

I declare that this research project titled STaff Ad Resource MANagement (STARMAN)

has been written by me under the supervision of Dr Joseph Kehoe.

This document was not presented in any previous research papers for the award of a bachelor's degree to the best of my knowledge. The work is entirely mine, and I accept full responsibility for any errors that might be found in this report. At the same time, the reference to publish materials had been duly acknowledged. I have provided a complete table of references for all works and sources used in the preparation of this document. I understand that failure to conform with the Institute's regulations governing plagiarism represents a serious offense.