|  |
| --- |
| Enterprise Tracking Application |
| Final Project Report |
| Prepared For: Perspecta  Location: Austin, TX |
| Date: 10/27/2020  Revised: 11/14/2020 |
| Angeleen Abesamis  Remnin Ferrer  James Fritz  Yeremy Joseph  Eric Leon |

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Comments | Author |
| 1.0 | 11/6/2020 | Original Submission |  |
|  |  |  |  |
|  |  |  |  |

# Executive Summary

This document serves as the final project report for the Enterprise Tracking Application (ETA) that is being developed by the University of Texas at Dallas (UTD) project team for Perspecta Inc. We will use this document to describe the complete project from start to finish.

# Table of Contents

[Revision History 2](#_Toc56272658)

[Executive Summary 3](#_Toc56272659)

[Table of Contents 4](#_Toc56272660)

[List of Figures 6](#_Toc56272661)

[List of Tables 8](#_Toc56272662)

[1. Introduction 10](#_Toc56272663)

[1.1 Purpose and Scope 10](#_Toc56272664)

[1.2 Product Overview 10](#_Toc56272665)

[1.3 Structure of the Document 10](#_Toc56272666)

[1.4 Terms, Acronyms, and Abbreviations 10](#_Toc56272667)

[2. Project Management Plan 11](#_Toc56272668)

[2.1 Project Organization 11](#_Toc56272669)

[2.2 Lifecycle Model 12](#_Toc56272670)

[2.3 Risk Analysis 13](#_Toc56272671)

[2.4 Hardware and Software Resource Requirements 14](#_Toc56272672)

[2.5 Deliverables and Schedule 15](#_Toc56272673)

[2.6 Monitoring, Reporting, and Controlling Mechanisms 17](#_Toc56272674)

[2.7 Professional Standards 18](#_Toc56272675)

[2.7.1 Member Expectations 18](#_Toc56272676)

[2.8 Evidence of Configuration Management 20](#_Toc56272677)

[2.9 Project Impact 24](#_Toc56272678)

[3. Requirements Specifications 25](#_Toc56272679)

[3.1 Stakeholders 25](#_Toc56272680)

[3.2 Use Case Models for Functional Requirements 25](#_Toc56272681)

[3.3 Use Case Model Rationale 62](#_Toc56272682)

[3.4. Non-functional requirements 62](#_Toc56272683)

[4. Architecture 63](#_Toc56272684)

[4.1 Architectural Style 63](#_Toc56272685)

[4.2 Architectural Model 63](#_Toc56272686)

[4.3 Technology, Software, and Hardware Used 82](#_Toc56272687)

[4.4 Rationale for the Architectural Style and Model 84](#_Toc56272688)

[5. Design 86](#_Toc56272689)

[5.1 GUI Design 86](#_Toc56272690)

[5.2 Static Model 98](#_Toc56272691)

[5.3 Dynamic Model 109](#_Toc56272692)

[5.4 Rationale for the Detailed Design Model 123](#_Toc56272693)

[5.5 System Traceability 123](#_Toc56272694)

[6. Test Plan 131](#_Toc56272695)

[6.1 Requirements/Specifications-Based System Level Test Cases 131](#_Toc56272696)

[6.2 Traceability of Test Cases to Use Cases 178](#_Toc56272697)

[6.3 Techniques for Test Generation 180](#_Toc56272698)

[6.4 Assessment of the Test Suite 181](#_Toc56272699)

[7. Acknowledgement 182](#_Toc56272700)

[8. References 183](#_Toc56272701)

# List of Figures

|  |  |  |
| --- | --- | --- |
| Figure | Description | Page |
| 3-1 | Use case 1. | 26 |
| 3-2 | Login page wireframe. | 27 |
| 3-3 | Home page wireframe. | 28 |
| 3-4 | Use case 2. | 30 |
| 3-5 | Use case 3. | 31 |
| 3-6 | Use case 4. | 32 |
| 3-7 | Use case 5. | 33 |
| 3-8 | Use case 6. | 36 |
| 3-9 | Use case 7. | 37 |
| 3-10 | Use case 8. | 38 |
| 3-11 | Use case 9. | 39 |
| 3-12 | Use case 10. | 42 |
| 3-13 | Use case 11. | 43 |
| 3-14 | Use case 12. | 44 |
| 3-15 | Use case 13. | 47 |
| 3-16 | Use case 14. | 48 |
| 3-17 | Use case 15. | 49 |
| 3-18 | Use case 16. | 50 |
| 3-19 | Use case 17. | 51 |
| 3-20 | Use case 18. | 53 |
| 3-21 | Use case 19. | 54 |
| 3-22 | Use case 20. | 55 |
| 3-23 | Use case 21. | 56 |
| 3-24 | Use case 22. | 58 |
| 3-25 | Use case 23. | 59 |
| 3-26 | Use case 24. | 60 |
| 3-27 | Use case 25. | 61 |
| 4-1 | 4+1 View Model Diagram. | 63 |
| 4-2 | Perspecta ETA N-tier Architecture. | 65 |
| 4-3 | User login activity diagram. | 66 |
| 4-4 | Add application activity diagram. | 67 |
| 4-5 | List applications activity diagram. | 68 |
| 4-6 | Update application activity diagram. | 69 |
| 4-7 | Delete application activity diagram. | 70 |
| 4-8 | Add data type activity diagram. | 71 |
| 4-9 | List data types activity diagram. | 72 |
| 4-10 | Update data type activity diagram. | 73 |
| 4-11 | Delete data type activity diagram. | 74 |
| 4-12 | Add data link activity diagram. | 75 |
| 4-13 | List data links activity diagram. | 76 |
| 4-14 | Delete data link activity diagram. | 77 |
| 4-15 | ETA architecture model diagram. | 78 |
| 4-16 | ETA component diagram. | 79 |
| 4-17 | ETA development package diagram. | 80 |
| 4-18 | ETA deployment diagram. | 81 |
| 4-19 | Hardware structure diagram. | 84 |
| 5-1 | Login Page Wireframe. | 86 |
| 5-2 | Home Page Wireframe. | 87 |
| 5-3 | List Applications Wireframe. | 88 |
| 5-4 | Add/Edit Applications Wireframe. | 89 |
| 5-5 | Delete Applications Wireframe. | 90 |
| 5-6 | List Data Types Wireframe. | 91 |
| 5-7 | Add Data Type Wireframe. | 92 |
| 5-8 | Edit Data Type Wireframe. | 93 |
| 5-9 | Delete Data Type Wireframe. | 94 |
| 5-10 | List Data Linkage Wireframe. | 95 |
| 5-11 | Add Data Linkage Wireframe. | 96 |
| 5-12 | Delete Data Linkage Wireframe. | 97 |
| 5-13 | System Front-End Class Diagram. | 99 |
| 5-14 | RESTful Services Class Diagram. | 100 |
| 5-15 | GUI Screen Class Diagram. | 101 |
| 5-16 | GUI Service Class Diagram. | 102 |
| 5-17 | Spring Boot Server Startup Class Diagram. | 103 |
| 5-18 | REST API User Authentication Class Diagram. | 104 |
| 5-19 | REST API Applications Class Diagram. | 105 |
| 5-20 | REST API Application Links Class Diagram. | 106 |
| 5-21 | User/Group ER Diagram. | 107 |
| 5-22 | Application Data ER Diagram. | 108 |
| 5-23 | GUI List Applications Sequence Diagram. | 109 |
| 5-24 | GUI Create/Update/Delete Applications Sequence Diagram. | 110 |
| 5-25 | User Login Sequence Diagram. | 111 |
| 5-26 | List Applications Sequence Diagram. | 112 |
| 5-27 | Add Application Sequence Diagram. | 113 |
| 5-28 | Edit Application Sequence Diagram. | 114 |
| 5-29 | Delete Application Sequence Diagram. | 115 |
| 5-30 | List Data Types Sequence Diagram. | 116 |
| 5-31 | Add Data Type Sequence Diagram. | 117 |
| 5-32 | Edit Data Type Sequence Diagram. | 118 |
| 5-33 | Delete Data Type Sequence Diagram. | 119 |
| 5-34 | List Data Linkages Sequence Diagram. | 120 |
| 5-35 | Add Data Linkage Sequence Diagram. | 121 |
| 5-36 | Delete Data Linkage Sequence Diagram. | 122 |

# List of Tables

|  |  |  |
| --- | --- | --- |
| Table | Description | Page |
| 2-1 | Project team members. | 11 |
| 2-2 | Project team member roles. | 12 |
| 2-3 | Project risks. | 13 |
| 3-1 | Use case 1. | 26 |
| 3-2 | Login UI requirements. | 28 |
| 3-3 | Main content home page content UI requirements. | 29 |
| 3-4 | Use case 2. | 30 |
| 3-5 | Use case 3. | 31 |
| 3-6 | Use case 4. | 32 |
| 3-7 | Use case 5. | 33 |
| 3-8 | Applications component GUI functional requirements. | 35 |
| 3-9 | Use case 6. | 36 |
| 3-10 | Use case 7. | 37 |
| 3-11 | Use case 8. | 38 |
| 3-12 | Use case 9. | 39 |
| 3-13 | Data types component UI functional requirements. | 41 |
| 3-14 | Use case 10. | 42 |
| 3-15 | Use case 11. | 43 |
| 3-16 | Use case 12. | 44 |
| 3-17 | Data links component UI functional requirements. | 46 |
| 3-18 | Use case 13. | 47 |
| 3-19 | User data functional requirements. | 47 |
| 3-20 | Use case 14. | 48 |
| 3-21 | Use case 15. | 49 |
| 3-22 | Use case 16. | 50 |
| 3-23 | Use case 17. | 51 |
| 3-24 | Applications web services functional requirements. | 52 |
| 3-25 | Use case 18. | 53 |
| 3-26 | Use case 19. | 54 |
| 3-27 | Use case 20. | 55 |
| 3-28 | Use case 21. | 56 |
| 3-29 | Applications web services functional requirements. | 57 |
| 3-30 | Use case 22. | 58 |
| 3-31 | Use case 23. | 59 |
| 3-32 | Use case 24. | 60 |
| 3-33 | Web services layer functional requirements. | 61 |
| 3-34 | Persistence layer functional requirements. | 62 |
| 3-35 | Use case 1 rationale. | 62 |
| 4-1 | Architecture Style Tradeoffs. | 85 |
| 5-1 | GUI Layout Requirements. | 123 |
| 5-2 | GUI Layout Traceability Matrix. | 124 |
| 5-3 | Applications GUI Requirements. | 125 |
| 5-4 | Applications GUI Traceability Matrix. | 125 |
| 5-5 | Data Types GUI Requirements. | 126 |
| 5-6 | Data Types GUI Traceability Matrix. | 126 |
| 5-7 | Data Linkages GUI Requirements. | 127 |
| 5-8 | Data Linkages GUI Traceability Matrix. | 127 |
| 5-9 | RESTful Web Services Requirements. | 128 |
| 5-10 | RESTful Web Services Traceability Matrix. | 128 |
| 5-11 | Persistence Requirements. | 129 |
| 6-1 | GUI Login and Layout Feature Set (FS1). | 131 |
| 6-2 | Applications Feature Set (FS2). | 145 |
| 6-3 | Data Types Feature Set (FS3). | 159 |
| 6-4 | Data Linkages Feature Set (FS4). | 169 |
| 6-5 | GUI Login and Layout Traceability. | 178 |
| 6-6 | Applications Traceability. | 178 |
| 6-7 | Data Types Traceability. | 179 |
| 6-8 | Data Linkages Traceability. | 179 |

# 1. Introduction

### 1.1 Purpose and Scope

The purpose of this final report for the ETA project sponsored by Perspecta, Inc. is to provide a final comprehensive project document which discusses the various phases of the project as well as the results and the learning outcomes of each.

### 1.2 Product Overview

TBD (including capabilities, scenarios for using the product, etc.)

The Perspecta ETA is a one-stop application that is designed to allow an authorized user to manage records for the Perspecta Enterprise Integration process through a convenient web interface. This application allows the user to store application integration records, data type records, and data linkage records. In addition to basic application integration records, the application also keeps track of what data types are included in each application record. The system also keeps track of data type information that has been moved from one application to another so that the source of a data type for an application record can be tracked.

### 1.3 Structure of the Document

To fully document all the design aspects, the software detailed design document contains the following subsections:

* Section 1, Introduction: This section...
* Section 2, Project Management Plan: This section...
* Section 3, Requirements Specification: This section...
* Section 4, Architecture: This section...
* Section 5, Detailed Design: This section...
* Section 6, Test Plan: This section...
* Acknowledgement: This section...
* References: The References section contains an IEEE style formatted list of the external sources of information the team used to create this document.

### 1.4 Terms, Acronyms, and Abbreviations

**CRUD:** Create, Retrieve, Update, and Delete database operations.

**DDD:** Detailed Design Document

**ETA:** Enterprise Tracking Application

**GUI:** Graphical User Interface

**SAD:** Software Architecture Document

**SRS:** Software Requirements Specification

**TPD:** Testing Plan Document

**UTD:** The University of Texas at Dallas

# 2. Project Management Plan

### 2.1 Project Organization

#### 2.1.1 Organizational Structure

Project Team Lead

The project team lead is responsible for tracking the team’s progress in accordance with the project schedule. The project team lead is also responsible for submitting all of the necessary deliverables to the teaching assistant by the required deadlines. The project team lead’s main goal is to make sure that the project is successful. They also communicate with the project sponsor to assure that the project goals are being met.

Software Architect

The software architects are responsible for creating the architecture for a system that will best suit the project needs. The architecture that they choose should be able to also meet project sponsor requirements and meet the project goals.

Software Developer

The software developers are responsible for developing the functionality we have stated in our requirements as well as integrate the corresponding technology to ensure that the application can carry out end-to-end tasks.

Database Developer

The database developers are responsible for implementing the database functionality that will be necessary to carry out the end-to-end tasks of the application.

Software Tester

The software testers are responsible for testing the application to ensure that it follows the quality that we promise to deliver to our project sponsor as well as carry out the functionality that we stated in our requirements.

#### 2.1.2 Project Team

|  |  |  |
| --- | --- | --- |
| Name | Title | Email |
| Angeleen Abesamis | Software Engineer | axa180073@utdallas.edu |
| Remnin Ferrer | Software Engineer | rxf170530@utdallas.edu |
| James Fritz | Software Engineer | jkf150030@utdallas.edu |
| Yeremy Joseph | Software Engineer | cwj180001@utdallas.edu |
| Eric Leon | Software Engineer | eal180003@utdallas.edu |

Table 2-1. Project team members

#### 2.1.3 Team Roles

Given the nature of the project, all team members will play some part of each role in the organizational structure apart from the project team lead. These team roles are indicated in the following table per team member.

|  |  |
| --- | --- |
| Role | Team Member Names |
| Project Team Lead | James Fritz |
| Software Architect | Angeleen Abesamis, Remnin Ferrer, James Fritz,  Yeremy Joseph, Eric Leon |
| Software Developer | Angeleen Abesamis, Remnin Ferrer, James Fritz,  Yeremy Joseph, Eric Leon |
| Database Developer | Angeleen Abesamis, Remnin Ferrer, James Fritz,  Yeremy Joseph, Eric Leon |
| Software Tester | Angeleen Abesamis, Remnin Ferrer, James Fritz,  Yeremy Joseph, Eric Leon |

Table 2-2. Project team member roles

### 2.2 Lifecycle Model

#### 2.2.1 Lifecycle Model Description

For this project we will be using a revised waterfall model as required by the university.

#### 2.2.2 Lifecycle Model Rationale

The reason for using a revised waterfall model has to do with both the nature of the projects and the amount of time available to see their completion. In this case, the revised waterfall model is a traditional iterative model, but it has been revised to allow for edits to be made to prior deliverables as subsequent iterations are completed.

### 2.3 Risk Analysis

#### 2.3.1 Project Risks

The possible project risks, likelihood of these risks arising, and proposed risk reduction strategies can be found in the following table.

|  |  |  |
| --- | --- | --- |
| Probability (L/M/H) | Risk Description and Impact | Mitigation Strategy and/or Contingency Plan |
| M | Hardware Accessibility: A team member that does not have access to a Windows PC or their machine becomes damaged | Borrow a laptop from UTD following their “Laptop Use & Checkout Policy” |
| L | Team member scheduling Conflicts: a member of the team is unable to attend a meeting | Sessions will be recorded, and any notes will be shared |
| L | Team lead scheduling conflicts: The team lead is unable to attend a meeting | The team lead will share the agenda to follow, if any, and designate an interim team lead |
| M | A team member becomes unavailable due to an emergency or medical issue which may impact the development phase | Sessions will be recorded, and any notes will be shared. Seek advice from course instructors |
| M | Sponsor scheduling conflicts: The sponsor is unable to attend a sponsor meeting | Attempt to reschedule the meeting prior to the next scheduled sponsor meeting |
| H | Complexity of the technology: Learning curve for new development technologies is steeper than expected | Designate additional resources to the most complex languages |
| L | Unrealistic delivery estimates: delivery targets are not being met | Re-estimate, put in additional hours, seek advice from course instructors |
| M | Project sponsor is unavailable to provide time-sensitive input | Seek advice from course instructors |
| M | Sponsor insists on new requirements | Estimate the effort and decline if necessary |
| L | Lack of effective communication or participation of a team member | Take disciplinary action as stated in section 8.1 |
| M | Design fails to address major requirements | Reassign additional team members to the project design |
| L | The product fails to meet expectations | Switch to bi-weekly meetings and additional hours until the project is back on track |
| M | The testing phase uncovers more errors/bugs than expected | Reassign additional team members where necessary |
| L | Team members do not work together efficiently | Reassign tasks and roles |

Table 2-3. Project risks

### 2.4 Hardware and Software Resource Requirements

#### 2.4.1 Required Hardware

* Windows PC
* Wireless Connection

#### 2.4.2 Required Software

* Windows Operating System
* Eclipse IDE
* Java Development Kit
* Microsoft SQL – Developer Edition
* Spring Boot Framework
* JavaScript Framework

#### 2.4.3 Learning Outcomes

1. Microsoft SQL Server – Ability to:
   1. Install an instance of SQL server
   2. Install SQL Server Management Studio
   3. Create objects and run CRUD scripts against those objects
2. Spring Boot – Ability to:
   1. Install Spring Boot into a selected IDE
   2. Create and configure a Spring Boot project
   3. Utilize Java within the project
3. Java – Ability to:
   1. Write clean Java code to create a middle layer and connect the application to the backend database
   2. Utilize OOP concepts and software design principles to meet Java programming best practices
4. Angular – Ability to:
   1. Configure and create an Angular project
   2. Write Typescript code
   3. Create and implement components, services, classes/models, and other Angular objects
   4. Implement routing and routing-guards
   5. Expand on HTML knowledge and integrate Angular functionality
   6. Expand on CSS knowledge and CSS libraries, e.g. Flexbox, Bootstrap, and CSS Grid
   7. Expand on modern JavaScript knowledge
5. RESTful Web Services – Ability to:
   1. Implement a RESTful architecture
   2. Create and handle RESTful methods

#### 2.4.3.1 Angeleen Abesamis

Learning outcomes.

#### 2.4.3.2 Remnin Ferrer

Learning outcomes.

#### 2.4.3.3 James Fritz

My primary focus during this project was on the RESTful API. I currently work full time as an application developer specializing in Java. The company where I work currently uses traditional data source methods, so the concept of RESTful web services was entirely new to me. I am looking forward to using Spring Boot, Maven Dependency Injection, JPA, Hibernate, and Swagger professionally. I also learned quite a bit about the Angular framework as well as HTML and CSS. This is a stark contrast to the Java Swing applications we're developing at my workplace for our in-house applications.

#### 2.4.3.4 Yeremy Joseph

Learning outcomes.

#### 2.4.3.5 Eric Leon

In order to accomplish my tasks for this project I had to learn, research, and implement various technologies including but not limited to Angular, Spring Boot, Java, Maven, and Microsoft SQL Server. My tasks mainly consisted in helping develop the front-end of the application using Angular. I learned how to use the Angular CLI to create projects, components, services, models, and other objects. I implemented routing and routing guards to secure access to elements of the UI. I understood how the component lifecycle works and how to use lifecycle hooks to make the application more efficient. In addition to this, I learned about databinding, directives, custom events, local references, dependency injection, observables, form handling, how to make http requests to work with the RESTful methods that the middle layer was exposing, and much more. Also, I was able to utilize Chrome dev tools from the console, application, network, and sources tabs and other web troubleshooting techniques to fix HTML, Typescript, and CSS code. Although my focus was Angular, I did implement a project in Spring Boot to help me test the front-end as it was being developed. There was very little exposure to SQL in this project, but I am very comfortable with this technology as I currently work as a SQL Developer. I truly enjoyed the many hours of coding spent with my team members in this project and plan to continue learning and developing applications using Angular and the knowledge gained from this project.

### 2.5 Deliverables and Schedule

#### 2.5.1 Project Deliverables

The following project deliverables are to be delivered to Perspecta, Inc.

* Project Plan
* Requirements
* Logical Architecture Diagram
* Physical Architecture Diagram
* Test Plan
* Final Report and Demonstration

The following project deliverables are to be delivered to the teaching assistant.

* Project management Plan
* Requirements Documentation
* Architecture Documentation
* Detailed Design Documentation
* Testing Plan
* Final Project Report
* Final Project Demonstration

#### 2.5.2 Project Schedule

The project team will meet with Dave Gibson from Perspecta (team sponsor) on Fridays from 3 p.m. to 5 p.m. through November 20th. Meetings with the team sponsor will take place on Cisco WebEx.

The following schedule is set to meet the delivery requirement:

1. Sponsor meeting to identify business problem 9/4
2. Project Management Plan 9/4
3. Sponsor meeting to identify requirements 9/11
4. Sponsor meeting to finalize requirements 9/18
5. Requirements Document 9/18
6. Sponsor meeting to identify design 9/25
7. Sponsor meeting to finalize design 10/2
8. Logical/Physical Architecture Diagram 10/2
9. Architecture Document 10/2
10. Sponsor meeting to review prototype design 10/9
11. Sponsor meeting to finalize prototype design 10/16
12. Detailed Design Document 10/16
13. Sponsor meeting to review implementation progress 10/23
14. Sponsor meeting to finalize implementation and review testing plan 10/30
15. Sponsor meeting to finalize testing plan 11/6
16. Testing Plan 11/6
17. Sponsor meeting to review feedback 11/13
18. Sponsor meeting to review revisions 11/20
19. Final Project Report and Demo 11/25

### 2.6 Monitoring, Reporting, and Controlling Mechanisms

#### 2.6.1 Management Reports

Performance Reports

Performance reports contain information about the project’s current state. This includes status and progress reports to keep track of on-going and finished tasks, as well as estimating progress in the future. Approved changes and status of risks and issues shall also be covered in the report. The Performance Reporting task will aid in tracking the current status of the project and keeping stakeholders and project members up to date.

#### 2.6.2 Project Monitoring and Control Mechanisms

Monitoring and Controlling Project Work

The team shall review and track current progress in the project and compare it to planned performance objectives to confirm that they are met. This comparison is included in the performance reports and will take place all throughout the project. This process will help both team members and stakeholders understand the project’s current state, as well as actions taken and projected dates for completion of tasks.

Monitoring Risk

The team will monitor the set of risks and agreed-upon risk response plans for some risks require an immediate response and to avoid a risk from becoming an issue. Excel will be used to manage and control risks. These responses can be modified as needed. The team leader will periodically work with the project members to identify and analyze new risks as these arise over time. This process shall mitigate the effects of risks, if not prevent them from occurring.

Monitoring Communications

The team shall ensure that the information needed from the client regarding project is concise and that these goals are met in the project. A meeting minutes for communications between the client and the team members shall be used, and a separate log that contains key information discussed in the meeting shall be used when necessary. This process will aid in each member’s understanding of information provided by the client and will lower the risk of any misunderstandings or mistakes when transcribing information into requirements.

Integrated Change Control

The Integrated Change Control process manages, and documents change requests for project deliverables and the project management plan. The process analyzes and determines if change requests are to be approved or rejected, then examines if changes to the project baseline is required upon implementing the change request. Having Integrated Change Control will drastically reduce possible risks associated with changes in the project, as well as provide some insight through analyzing the effects of implementing changes.

Scope Control

The Scope Control process monitors the project scope and manages any changes made to its baseline. This ensures that the scope baseline remains consistent all throughout the project, and that any changes to the baseline is implemented and documented.

Schedule Control

The Schedule Control process monitors the status of the project activities to update the progress of the project and modifies the schedule baseline if needed for the success of the project.

Quality Control

The Quality Control is the process of validating the information and implementing the appropriate quality steps to satisfy the project quality requirements. The team leader will evaluate and verify that the submitted information or report is accurate before including it in the overall project. The project team must determine the quality of each project deliverable.

### 2.7 Professional Standards

### 2.7.1 Member Expectations

Weekly Sponsor Meetings

* All members of the project team shall meet and participate weekly with the project sponsor.
* Weekly sponsor meetings will take place on Fridays at 3:00 PM and are scheduled for up to two hours in duration.
* Weekly sponsor meetings will utilize Cisco WebEx as the meeting platform.
* All members of the project team are expected to participate during the weekly sponsor meetings.
* All members are always expected to maintain professionalism during the sponsor meetings. This expectation extends to appropriate etiquette relating to the online meeting environment (e.g. keep mics muted when not talking).
* Team members shall notify the project team lead in advance with their reasoning if they will be unable to attend a weekly sponsor meeting. If the project team lead is unable to attend a weekly sponsor meeting, they will appoint a coordinator to take their place for the meeting and provide advance notice to the rest of the team with their reasoning.

Weekly Team Meetings

* All members of the project team shall meet and participate weekly to review the output from the weekly project sponsor meeting and determine the tasks to be completed for the week.
* Weekly team meetings will take place on Saturdays at 12:00 PM and are scheduled for up to two hours in duration.
* Weekly team meetings will utilize Cisco WebEx as the meeting platform.
* All members of the project team are expected to participate during the weekly team meeting.
* All members are always expected to maintain acceptable behavior during the weekly team meetings.
* Team members shall notify the project team lead in advance with their reasoning if they will be unable to attend a weekly team meeting. If the project team lead is unable to hold a weekly team meeting, they will appoint a coordinator to take their place for the meeting and provide advance notice to the rest of the team with their reasoning.
* Weekly team meetings may only be cancelled if suggested by the project team lead and agreed to by all team members. Cancellation by agreement of all team members is only possible if all members have either a) completed their tasks ahead of schedule or b) have all acknowledged that they are fully aware of their current expectations.

Unacceptable Behavior

Examples of unacceptable behavior include, but are not limited to:

* Not delivering on time.
* Delivering poor quality work.
* Missing team meetings.
* Being uncommunicative.
* Being unprepared for team meetings.
* Being disrespectful or rude.

Corrective Measures

* On the first occurrence of unacceptable behavior, determine the circumstances involved, resolve the problem, and document the event in the meeting minutes.
* On a second occurrence, notify the instructor of the problem. A meeting will be set up to evaluate the situation and resolve the problem.
* On a third occurrence, again notify the instructor of the problem. A meeting will be set up to evaluate the situation and resolve the problem. At this point, the team will have the \*option\* of removing the team member. If removed, then the team member receives a pro-rated grade based on the number of weeks they have participated in the group.

Valid reasons that must be considered include those listed for obtaining an incomplete standing in a course (illness, death in the family, travel for business or academic reasons, etc.).

### 2.8 Evidence of Configuration Management

#### 2.8.1 PMP Evidence of Configuration Management

Graphical user interface

Description automatically generated

#### 2.8.2 SRS Evidence of Configuration Management

Graphical user interface, text, application

Description automatically generated

#### 2.8.3 SAD Evidence of Configuration Management

Graphical user interface, text, application

Description automatically generated

#### 2.8.4 DDD Evidence of Configuration Management

Graphical user interface, text, application

Description automatically generated

#### 2.8.5 TPD Evidence of Configuration Management

Graphical user interface, text, application

Description automatically generated

#### 2.8.6 Final Report Evidence of Configuration Management

### 2.9 Project Impact

TBD. Impact of the project on individuals and organizations. (Include a description of what impact your project will have on individuals and society).

# 3. Requirements Specifications

### 3.1 Stakeholders

Prospective Users (Perspecta Inc. employees)

Enterprise Tracking Application Development Team

Dr. W. Eric Wong – Instructor

Dave Gibson – Sponsor

Linghuan Hu – Teaching Assistant

### 3.2 Use Case Models for Functional Requirements

#### 3.2.1. Graphical User Interface Use Case Model

##### 3.2.1.1 Login UI

###### 3.2.1.1.1 Login UI Feature Introduction

The user login interface will be used to verify access via permissions.

|  |  |
| --- | --- |
| Figure 3-1. Use case 1. | |
| Use Case ID: | UC1 |
| Use Case Name: | User Login/Logout |
| Actors: | Standard User, Admin User |
| Entry Condition(s): | User navigates to the web address of the application. |
| Basic Flow: | 1. The user enters their login id into the Login id text field. 2. The user enters their password into the Password field. 3. The user clicks the Ok button to submit their credentials for authentication and permissions. |
| Exit Condition(s): | The user provides credentials which have the appropriate permissions to access the home page for the web application. |
| Exceptions: | The user does not provide valid login credentials. The page will indicate that the login credentials are invalid. |
| Special Requirements: | The user credentials must exist in the database. |

Table 3-1. Use case 1.

###### 3.2.1.1.2 Wireframe

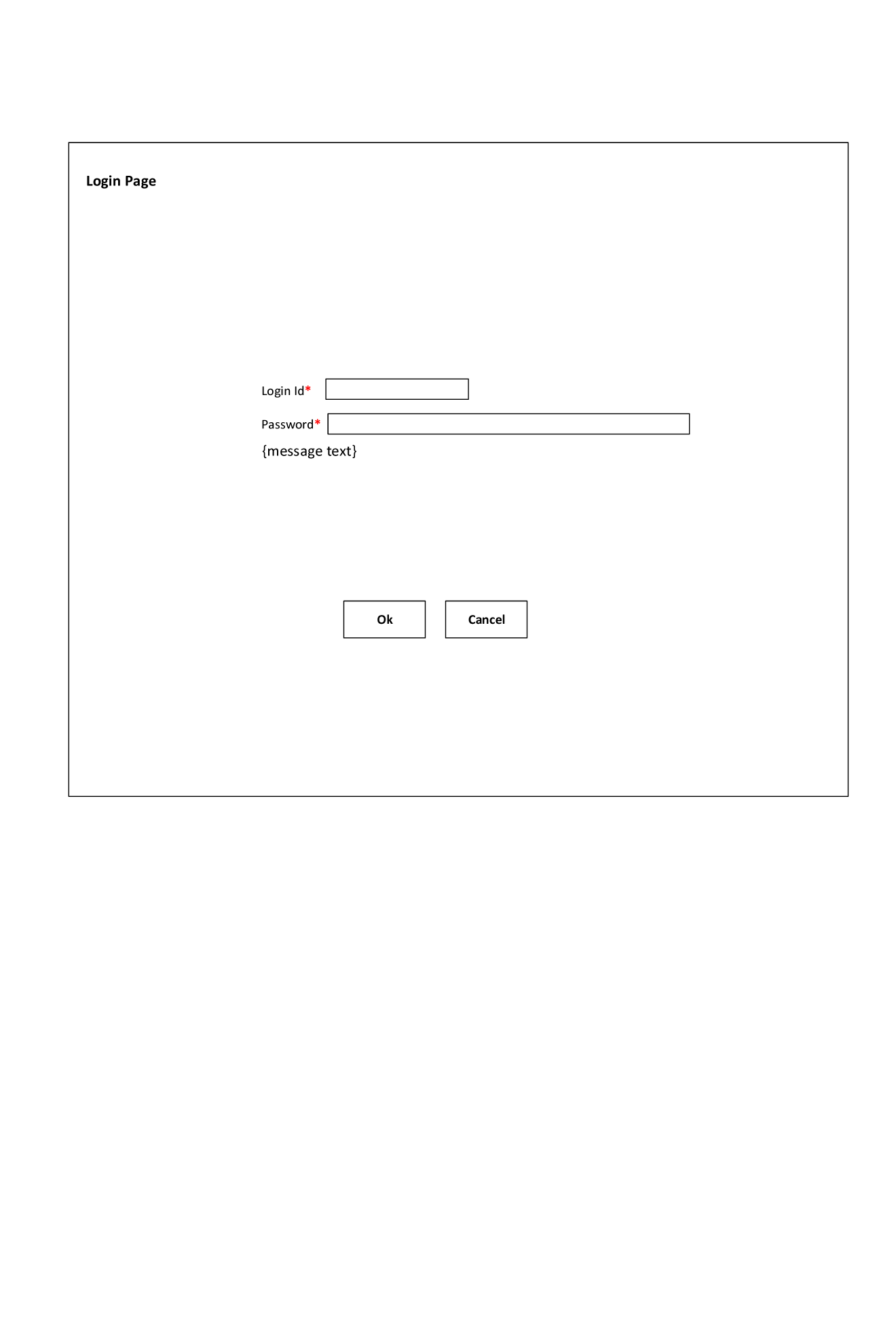


Figure 3-2. Login page wireframe.

###### 3.2.1.1.3 Associated Functional Requirements

**UI-FR-1**

The login user interface shall provide a text input field for the user to enter an email address as their login id.

**UI-FR-2**

The login user interface shall provide a text input field for the user to enter their password.

**UI-FR-3**

The login user interface shall provide a generic textual feedback message which will be used to notifying the user if their login attempt has failed. The message shall not provide specific information about the precise cause of the login attempt failure.

**UI-FR-4**

The login user interface shall provide a button that will allow the user to transmit their credentials for authentication.

**UI-FR-5**

The login user interface shall provide a button that will allow the user to cancel their login attempt and clear the fields.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req. ID | Type | Priority | Description | Owner |
| UI-FR-1 | Functional | Essential | Login email | JF, EL |
| UI-FR-2 | Functional | Essential | Login password | JF, EL |
| UI-FR-3 | Functional | Essential | Generic login failure message | JF, EL |
| UI-FR-4 | Functional | Essential | Login submit button | JF, EL |
| UI-FR-5 | Functional | Essential | Login cancel button | JF, EL |

Table 3-2. Login UI requirements.

##### 3.2.1.2 Main Content Page UI

###### 3.2.1.2.1 Main Content Page UI Feature Introduction

The home page the user will use to create, retrieve, update, and delete enterprise application integration records, related data types, and linkages of data types between application integration records.

3.2.1.2.1.1 Wireframe

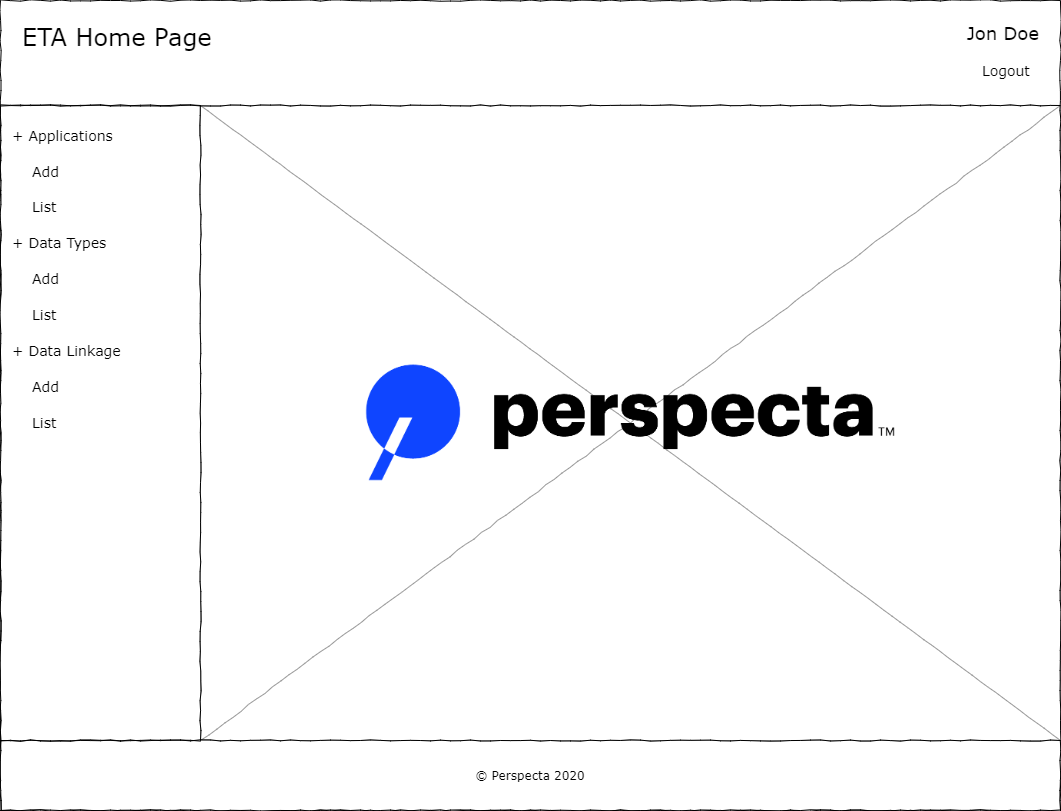


Figure 3-3. Home page wireframe.

3.2.1.2.1.2 Associated Functional Requirements

**UI-FR-6**

The home page user interface shall include a header area depicting the application title and the username.

**UI-FR-7**

The home page user interface shall include a footer area depicting the appropriate copyright information.

**UI-FR-8**

The home page user interface shall include a menu area on the left side of the main body.

**UI-FR-9**

The home page user interface left side menu shall include three expandable options, one for applications, one for data types, and one for data linkages. Each of these expandable options shall have permission-based options allowing the user to add new records or to list all of the current records.

**UI-FR-10**

The home page user interface shall include a main content area on the center-right area of the main body. The main content area shall feature a default image which will be replaced by the formatted table containing the selected record data.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req. ID | Type | Priority | Description | Owner |
| UI-FR-6 | Functional | Essential | Home page header area | JF, EL |
| UI-FR-7 | Functional | Essential | Home page footer area | JF, EL |
| UI-FR-8 | Functional | Essential | Home page menu area | JF, EL |
| UI-FR-9 | Functional | Essential | Home page menu functionality | JF, EL |
| UI-FR-10 | Functional | Essential | Home page main area | JF, EL |

Table 3-3. Main content home page content UI requirements.

###### 3.2.1.2.2 Main Content Page Application Features

|  |  |
| --- | --- |
| Figure 3-4. Use case 2. | |
| Use Case ID: | UC2 |
| Use Case Name: | View application records |
| Actors: | Standard User, Admin User |
| Entry Condition(s): | The user has logged in and has permission to view the records. |
| Basic Flow: | 1. The user expands Applications in the left side menu area. 2. The user selects the List option from the submenu in the left side menu area. 3. The application displays the relevant information from the database within the main content area of the page. |
| Exit Condition(s): | The user selects a different option from the left side menu or leaves the application. |
| Exceptions: | The user does not have sufficient permission.  There are no application records in the database. |
| Special Requirements: | The user must be logged in with valid credentials. |

Table 3-4. Use case 2.

|  |  |
| --- | --- |
| Figure 3-5. Use case 3. | |
| Use Case ID: | UC3 |
| Use Case Name: | Add application record |
| Actors: | Admin User |
| Entry Condition(s): | The user has logged in and has permission to add a new integration record. |
| Basic Flow: | 1. The user expands Applications in the left side menu area. 2. The user selects the Add option from the submenu in the left side menu area. 3. The application displays a page where the user may enter the relevant information for the application record. 4. The user selects the Submit button to add the application record. 5. The application displays a window with a message confirming the new application record was added. 6. The user clicks the Ok button to close the confirmation window. 7. The page reloads the list of application records. |
| Exit Condition(s): | The user has completed the application addition process, or they have clicked the Cancel button prior to the application record being added. |
| Exceptions: | 1. The user does not have sufficient permission. 2. The application record exists in the database. 3. The user entered invalid information. |
| Special Requirements: | The user must be logged in with valid credentials. |

Table 3-5. Use case 3.

|  |  |
| --- | --- |
| Figure 3-6. Use case 4. | |
| Use Case ID: | UC4 |
| Use Case Name: | Edit application record |
| Actors: | Admin User |
| Entry Condition(s): | 1. The user has logged in and has permission to view and edit an application record. 2. The user has displayed the existing application records in the main content area. |
| Basic Flow: | 1. The user selects the Update button in the row of the application record they wish to edit. 2. The application shows a window with the fields that have been pre-populated with the application record. 3. The user selects the Submit button to update the application record. 4. The application displays a window with a message confirming the application record was updated. 5. The user clicks the Ok button to close the confirmation window. 6. The page reloads the list of application records. |
| Exit Condition(s): | The user has completed the application update process, or they have clicked the Cancel button prior to the application record being updated. |
| Exceptions: | 1. The user does not have sufficient permission. 2. The changes would cause the application record to match another application record in the database. 3. The user entered invalid information. |
| Special Requirements: | The user must be logged in with valid credentials. |

Table 3-6. Use case 4.

|  |  |
| --- | --- |
| Figure 3-7. Use case 5. | |
| Use Case ID: | UC5 |
| Use Case Name: | Delete application record |
| Actors: | Admin User |
| Entry Condition(s): | 1. The user has logged in and has permission to view and delete an application record. 2. The user has displayed the existing application records in the main content area. |
| Basic Flow: | 1. The user selects the Delete button in the row of the application record they wish to delete. 2. The application shows a window asking the user to confirm they would like to delete the application record. 3. The user selects the Submit button to delete the application record. 4. The application displays a window with a message confirming the application record was deleted. 5. The user clicks the Ok button to close the confirmation window. 6. The page reloads the list of application records. |
| Exit Condition(s): | The user has completed or cancelled the deletion by pressing the Ok button or the Cancel button respectively. |
| Exceptions: | The user does not have sufficient permission. |
| Special Requirements: | The user must be logged in with valid credentials. |

Table 3-7. Use case 5.

3.2.1.2.2.1 Associated Functional Requirements

**UI-FR-11**

The application shall display the application records in the database within a formatted table located within the main content area when the user clicks List from the Applications submenu.

**UI-FR-11.1**

The table rows shall include the application id, the application name, the application description, the application’s data types, an update button, and a delete button for admin users.

**UI-FR-11.2**

The table rows shall include the application id, the application name, the application description, and the application’s data types for standard users.

**UI-FR-12**

The application shall allow an admin user to add a new application record via a form located within the main content area when the user clicks Add from the Applications submenu.

**UI-FR-12.1**

The new application form shall include a text field for the application name, a text field for the application description, and a multi-select dropdown for the data types. The form shall also include buttons to submit or cancel the action.

**UI-FR-12.2**

The multi-select dropdown shall include pre-populated values from the data types in the database.

**UI-FR-13**

The application shall allow an admin user to update an existing application record via a form located within the main content area. The form shall be displayed when the user clicks the Update button located within the table row of the application to be updated.

**UI-FR-13.1**

The edit application form shall include a text field for the application name, a text field for the application description, and a multi-select dropdown for the data types. The form shall also include buttons to submit or cancel the action.

**UI-FR-13.2**

The multi-select dropdown shall include pre-populated values from the data types in the database.

**UI-FR-13.3**

The form fields shall be pre-populated with the current database application record information.

**UI-FR-14**

The application shall allow an admin user to delete an existing application record by clicking the Delete button located within the table row of the application to be deleted.

**UI-FR-14.1**

The application shall prompt the user to confirm that they wish to delete the application record.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req. ID | Type | Priority | Description | Owner |
| UI-FR-11 | Functional | Essential | List applications | JF, EL |
| UI-FR-11.1 | Functional | Essential | Applications table content – admin user | JF, EL |
| UI-FR-11.2 | Functional | Essential | Applications table content – standard user | JF, EL |
| UI-FR-12 | Functional | Essential | Add application | JF, EL |
| UI-FR-12.1 | Functional | Essential | Add application form content | JF, EL |
| UI-FR-12.2 | Functional | Essential | Add application populated multi-select data types from db | JF, EL |
| UI-FR-13 | Functional | Essential | Update application | JF, EL |
| UI-FR-13.1 | Functional | Essential | Update application form content | JF, EL |
| UI-FR-13.2 | Functional | Essential | Update application populated multi-select data types from db | JF, EL |
| UI-FR-13.3 | Functional | Essential | Update form fields populated with application data | JF, EL |
| UI-FR-14 | Functional | Essential | Delete application | JF, EL |
| UI-FR-14.1 | Functional | Essential | Delete application confirmation | JF, EL |

Table 3-8. Applications component GUI functional requirements.

###### 3.2.1.2.3 Main Content Page Data Types Features

|  |  |
| --- | --- |
| Figure 3-8. Use case 6. | |
| Use Case ID: | UC6 |
| Use Case Name: | View data type records |
| Actors: | Standard User, Admin User |
| Entry Condition(s): | The user has logged in and has permission to view the records. |
| Basic Flow: | 1. The user expands Data Types in the left side menu area. 2. The user selects the List option from the submenu in the left side menu area. 3. The application displays the relevant information from the database within the main content area of the page. |
| Exit Condition(s): | The user selects a different option from the left side menu or leaves the application. |
| Exceptions: | 1. The user does not have sufficient permission. 2. There are no data type records in the database. |
| Special Requirements: | The user must be logged in with valid credentials. |

Table 3-9. Use case 6.

|  |  |
| --- | --- |
| Figure 3-9. Use case 7. | |
| Use Case ID: | UC7 |
| Use Case Name: | Add data type record |
| Actors: | Admin User |
| Entry Condition(s): | The user has logged in and has permission to add a new integration record. |
| Basic Flow: | 1. The user expands Data Types in the left side menu area. 2. The user selects the Add option from the submenu in the left side menu area. 3. The application displays a page where the user may enter the relevant information for the data type record. 4. The user selects the Submit button to add the data type record. 5. The application displays a window with a message confirming the new data type record was added. 6. The user clicks the Ok button to close the confirmation window. 7. The page reloads the list of data type records. |
| Exit Condition(s): | The user has completed the data type addition process, or they have clicked the Cancel button prior to the data type record being added. |
| Exceptions: | 1. The user does not have sufficient permission. 2. The data type record exists in the database. 3. The user entered invalid information. |
| Special Requirements: | The user must be logged in with valid credentials. |

Table 3-10. Use case 7.

|  |  |
| --- | --- |
| Figure 3-10. Use case 8. | |
| Use Case ID: | UC8 |
| Use Case Name: | Edit data type record |
| Actors: | Admin User |
| Entry Condition(s): | 1. The user has logged in and has permission to view and edit a data type record. 2. The user has displayed the existing data type records in the main content area. |
| Basic Flow: | 1. The user selects the Update button in the row of the data type record they wish to edit. 2. The application shows a window with the fields that have been pre-populated with the data type record. 3. The user selects the Submit button to update the data type record. 4. The application displays a window with a message confirming the data type record was updated. 5. The user clicks the Ok button to close the confirmation window. 6. The page reloads the list of data type records. |
| Exit Condition(s): | The user has completed the data type update process, or they have clicked the Cancel button prior to the data type record being updated. |
| Exceptions: | 1. The user does not have sufficient permission. 2. The changes would cause the data type record to match another data type record in the database. 3. The user entered invalid information. |
| Special Requirements: | The user must be logged in with valid credentials. |

Table 3-11. Use case 8.

|  |  |
| --- | --- |
| Figure 3-11. Use case 9. | |
| Use Case ID: | UC9 |
| Use Case Name: | Delete data type record |
| Actors: | Admin User |
| Entry Condition(s): | 1. The user has logged in and has permission to view and delete a data type record. 2. The user has displayed the existing data type records in the main content area. |
| Basic Flow: | 1. The user selects the Delete button in the row of the data type record they wish to delete. 2. The application shows a window asking the user to confirm they would like to delete the data type record. 3. The user selects the Submit button to delete the data type record. 4. The application displays a window with a message confirming the data type record was deleted. 5. The user clicks the Ok button to close the confirmation window. 6. The page reloads the list of data type records. |
| Exit Condition(s): | The user has completed or cancelled the deletion by pressing the Ok button or the Cancel button respectively. |
| Exceptions: | 1. The user does not have sufficient permission. 2. The data type is constrained to application records or application linkage records in the database. |
| Special Requirements: | The user must be logged in with valid credentials. |

Table 3-12. Use case 9.

3.2.1.2.3.1 Associated Functional Requirements

**UI-FR-15**

The application shall display the data type records in the database within a formatted table located within the main content area when the user clicks List from the Data Type submenu.

**UI-FR-15.1**

The table rows shall include the data type id, the data type name, an update button, and a delete button for admin users.

**UI-FR-15.2**

The table rows shall include the data type id and the data type name for standard users.

**UI-FR-16**

The application shall allow an admin user to add a new data type record via a form located within the main content area when the user clicks Add from the Data Type submenu.

**UI-FR-16.1**

The new data type form shall include a text field for the data type name. The form shall also include buttons to submit or cancel the action.

**UI-FR-17**

The application shall allow an admin user to update an existing data type record via a form located within the main content area. The form shall be displayed when the user clicks the Update button located within the table row of the data type to be updated.

**UI-FR-17.1**

The edit data type form shall include a text field for the application name. The form shall also include buttons to submit or cancel the action.

**UI-FR-17.2**

The form field shall be pre-populated with the current database data type record information.

**UI-FR-18**

The application shall allow an admin user to delete an existing data type record by clicking the Delete button located within the table row of the data type to be deleted.

**UI-FR-18.1**

**The application shall prevent deletion of a data type that is constrained to an application.**

**UI-FR-18.2**

The application shall prompt the user to confirm that they wish to delete the data type record.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req. ID | Type | Priority | Description | Owner |
| UI-FR-15 | Functional | Essential | List data types | JF, EL |
| UI-FR-15.1 | Functional | Essential | Data type table content – admin user | JF, EL |
| UI-FR-15.2 | Functional | Essential | Data type table content – standard user | JF, EL |
| UI-FR-16 | Functional | Essential | Add data type | JF, EL |
| UI-FR-16.1 | Functional | Essential | Add data type form content | JF, EL |
| UI-FR-17 | Functional | Essential | Update data type | JF, EL |
| UI-FR-17.1 | Functional | Essential | Update data type form content | JF, EL |
| UI-FR-17.2 | Functional | Essential | Update form fields populated with data type data | JF, EL |
| UI-FR-18 | Functional | Essential | Delete data type | JF, EL |
| UI-FR-18.1 | Functional | Essential | Delete data type constraint protection | JF, EL |
| UI-FR-18.2 | Functional | Essential | Delete data type confirmation | JF, EL |

Table 3-13. Data types component UI functional requirements.

###### 3.2.1.2.4 Main Content Page Data Linkage Features

|  |  |
| --- | --- |
| Figure 3-12. Use case 10. | |
| Use Case ID: | UC10 |
| Use Case Name: | View data linkage records |
| Actors: | Standard User, Admin User |
| Entry Condition(s): | The user has logged in and has permission to view the records. |
| Basic Flow: | 1. The user expands Data Linkage in the left side menu area. 2. The user selects the List option from the submenu in the left side menu area. 3. The application displays the relevant information from the database within the main content area of the page. |
| Exit Condition(s): | The user selects a different option from the left side menu or leaves the application. |
| Exceptions: | 1. The user does not have sufficient permission. 2. There are no data linkage records in the database. |
| Special Requirements: | The user must be logged in with valid credentials. |

Table 3-14. Use case 10.

|  |  |
| --- | --- |
| Figure 3-13. Use case 11. | |
| Use Case ID: | UC11 |
| Use Case Name: | Add data linkage record |
| Actors: | Admin User |
| Entry Condition(s): | The user has logged in and has permission to add a new data linkage record. |
| Basic Flow: | 1. The user expands Data Linkage in the left side menu area. 2. The user selects the Add option from the submenu in the left side menu area. 3. The application displays a page where the user may enter the relevant information for the data linkage record. 4. The user selects the Submit button to add the data linkage record. 5. The application displays a window with a message confirming the new data linkage record was added. 6. The user clicks the Ok button to close the confirmation window. 7. The page reloads the list of data linkage records. |
| Exit Condition(s): | The user has completed the data linkage addition process, or they have clicked the Cancel button prior to the data linkage record being added. |
| Exceptions: | 1. The user does not have sufficient permission. 2. The data linkage record exists in the database. 3. The user entered invalid information. |
| Special Requirements: | The user must be logged in with valid credentials. |

Table 3-15. Use case 11.

|  |  |
| --- | --- |
| Figure 3-14. Use case 12. | |
| Use Case ID: | UC12 |
| Use Case Name: | Delete data linkage record |
| Actors: | Admin User |
| Entry Condition(s): | 1. The user has logged in and has permission to view and delete a data linkage record. 2. The user has displayed the existing data linkage records in the main content area. |
| Basic Flow: | 1. The user selects the Delete button in the row of the data linkage record they wish to delete. 2. The application shows a window asking the user to confirm they would like to delete the data linkage record. 3. The user selects the Submit button to delete the data linkage record. 4. The application displays a window with a message confirming the data linkage record was deleted. 5. The user clicks the Ok button to close the confirmation window. 6. The page reloads the list of data linkage records. |
| Exit Condition(s): | The user has completed or cancelled the deletion by pressing the Ok button or the Cancel button respectively. |
| Exceptions: | The user does not have sufficient permission. |
| Special Requirements: | The user must be logged in with valid credentials. |

Table 3-16. Use case 12.

3.2.1.2.4.1 Associated Functional Requirements

**UI-FR-19**

The application shall display the data linkage records in the database within a formatted table located within the main content area when the user clicks List from the Data Linkage submenu.

**UI-FR-19.1**

The table rows shall include the source application id, the source application name, the destination application id, the destination application name, the data type id, the data type name, an update button, and a delete button for admin users.

**UI-FR-19.2**

The table rows shall include the source application id, the source application name, the destination application id, the destination application name, the data type id, and the data type name for standard users.

**UI-FR-20**

The application shall allow an admin user to add a new data linkage record via a form located within the main content area when the user clicks Add from the Data Linkage submenu.

**UI-FR-20.1**

The new data linkage form shall include a dropdown for the name of the source application, a dropdown for the name of the destination application, and a dropdown for the data types associated with the selected source application. The form shall also include buttons to submit or cancel the action.

**UI-FR-20.2**

The source and destination application dropdown boxes shall include pre-populated values from the applications in the database.

**UI-FR-20.3**

The data type dropdown shall populate values when the source application is selected from the data types in the database that are constrained to the selected source application.

**UI-FR-21**

The application shall allow an admin user to delete an existing data linkage record by clicking the Delete button located within the table row of the data linkage to be deleted.

**UI-FR-21.1**

The application shall prompt the user to confirm that they wish to delete the data type record.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req. ID | Type | Priority | Description | Owner |
| UI-FR-19 | Functional | Essential | List data links | JF, EL |
| UI-FR-19.1 | Functional | Essential | Data link table content – admin user | JF, EL |
| UI-FR-19.2 | Functional | Essential | Data link table content – standard user | JF, EL |
| UI-FR-20 | Functional | Essential | Add data link | JF, EL |
| UI-FR-20.1 | Functional | Essential | Add data link form content | JF, EL |
| UI-FR-20.2 | Functional | Essential | Add data link form populated applications | JF, EL |
| UI-FR-20.3 | Functional | Essential | Add data link form populated data types | JF, EL |
| UI-FR-21 | Functional | Essential | Delete data link | JF, EL |
| UI-FR-21.1 | Functional | Essential | Delete data link confirmation | JF, EL |

Table 3-17. Data links component UI functional requirements.

#### 3.2.2 Web Services Use Case Model

##### 3.2.2.1 Web Services Introduction

The web services shall receive either POST, GET, PUT or DELETE requests based on user’s actions and input, and shall return appropriate results.

##### 3.2.2.2 User Data Web Services

|  |  |
| --- | --- |
| Figure 3-15. Use case 13. | |
| Use Case ID: | UC13 |
| Use Case Name: | View User Data |
| Actors: | System, Web Service |
| Entry Condition(s): | The user inputs username and password in the login page and attempts to login. |
| Basic Flow: | 1. The system sends a GET request to the web service with the username and password provided by the user 2. The web service returns appropriate user records, if any, based on search criteria. 3. The system grants the user access and navigates them to the main content page. |
| Exit Condition(s): | The user is granted or denied access to the home page. |
| Exceptions: | The user’s username and password does not match any records from the database. |
| Special Requirements: | n/a |

Table 3-18. Use case 13.

###### 3.2.2.2.1 Associated Functional Requirements

**WS-FR-1**

**The application shall send GET requests to the web service to verify user credentials when the user attempts to login.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req. ID | Type | Priority | Description | Owner |
| WS-FR-1 | Functional | Essential | View user data send GET | RF |

Table 3-19. User data functional requirements.

##### 3.2.2.3 Application Web Services

|  |  |
| --- | --- |
| Figure 3-16. Use case 14. | |
| Use Case ID: | UC14 |
| Use Case Name: | View Application Record |
| Actors: | User, Web Service |
| Entry Condition(s): | The user is logged in, has valid permissions and is in the home page. |
| Basic Flow: | 1. The user sends a GET request to the web service by selecting List from the Applications menu. 2. The web service returns a list of all applications. 3. The application displays the results of the GET request in the main content area. |
| Exit Condition(s): | The list of application records is successfully displayed in the main content area. |
| Exceptions: | The database lacks information needed.  The web service encounters an error with the GET request. |
| Special Requirements: | The user is logged in and must have valid credentials. |

Table 3-20. Use case 14.

|  |  |
| --- | --- |
| Figure 3-17. Use case 15. | |
| Use Case ID: | UC15 |
| Use Case Name: | Add Application Record |
| Actors: | User, Web Service |
| Entry Condition(s): | The user has selected the “Submit” button with values inputted in the Add Application page. |
| Basic Flow: | 1. The user sends a POST request after selecting the submit button in the Add Application page. 2. The web service adds the application record into the database. |
| Exit Condition(s): | The service successfully stores the new application record into the database. |
| Exceptions: | The user provides invalid input. The web service encounters an error with the GET request. |
| Special Requirements: | The user is logged in, must have valid credentials, and has sufficient permissions to add applications. |

Table 3-21. Use case 15.

|  |  |
| --- | --- |
| Figure 3-18. Use case 16. | |
| Use Case ID: | UC16 |
| Use Case Name: | Edit Application Record |
| Actors: | User, Web Service |
| Entry Condition(s): | The user has selected the “Save” button with values inputted in the Update Application Page. |
| Basic Flow: | 1. The user sends a PUT request after selecting the save button in the Edit Application page. 2. The web service edits the data of the targeted application record with the data provided by the user. |
| Exit Condition(s): | The service successfully updates the application record. |
| Exceptions: | The user provides invalid input. The web service encounters an error with the PUT request. |
| Special Requirements: | The user is logged in, must have valid credentials, and has limited permission to edit applications. |

Table 3-22. Use case 16.

|  |  |
| --- | --- |
| Figure 3-19. Use case 17. | |
| Use Case ID: | UC17 |
| Use Case Name: | Delete Application Record |
| Actors: | User, Web Service |
| Entry Condition(s): | The user has selected the “Delete” button on an application record from the list of applications. |
| Basic Flow: | 1. The user sends a DELETE request after selecting the delete button beside an application on the list of applications. 2. The web service deletes the application record from the database. |
| Exit Condition(s): | The service successfully deletes the application record. |
| Exceptions: | The user has insufficient permissions. The database lacks information needed. The web service encounters an error with the DELETE request. |
| Special Requirements: | The user is logged in, must have valid credentials, and has sufficient permissions to delete applications. |

Table 3-23. Use case 17.

###### 3.2.2.2.1 Associated Functional Requirements

**WS-FR-2**

**The application shall send GET requests to the web service when the user selects List from the Application submenu.**

**WS-FR-3**

**The application shall send POST requests to the web service when the user selects Submit button from the Add Application Page.**

**WS-FR-4**

**The application shall send PUT requests to the web service when the user selects Save button from the Update Application page.**

**WS-FR-5**

**The application shall send DELETE requests to the web service when the user selects Delete button from the list of applications.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req. ID | Type | Priority | Description | Owner |
| WS-FR-2 | Functional | Essential | Retrieve applications | AA |
| WS-FR-3 | Functional | Essential | Create application | AA |
| WS-FR-4 | Functional | Essential | Update application | AA |
| WS-FR-5 | Functional | Essential | Delete application | AA |

Table 3-24. Applications web services functional requirements.

##### 3.2.2.4 Data Type Web Services

|  |  |
| --- | --- |
| Figure 3-20. Use case 18. | |
| Use Case ID: | UC18 |
| Use Case Name: | View Data Type Record |
| Actors: | User, Web Service |
| Entry Condition(s): | The user is logged in, has valid permissions and is in the home page. |
| Basic Flow: | 1. The user sends a GET request to the web service by selecting List from the Data Types menu. 2. The web service returns a list of all data types. 3. The application displays the results of the GET request in the main content area. |
| Exit Condition(s): | The list of data type records is successfully displayed in the main content area. |
| Exceptions: | The database lacks information needed.  The web service encounters an error with the GET request. |
| Special Requirements: | The user is logged in and must have valid credentials. |

Table 3-25. Use case 18.

|  |  |
| --- | --- |
| Figure 3-21. Use case 19. | |
| Use Case ID: | UC19 |
| Use Case Name: | Add Data Type Record |
| Actors: | User, Web Service |
| Entry Condition(s): | The user has selected the “Submit” button with values inputted in the Add Data Type page. |
| Basic Flow: | 1. The user sends a POST request after selecting the submit button in the Add Data Type page. 2. The web service adds the data type record into the database. |
| Exit Condition(s): | The service successfully stores the new data type record into the database. |
| Exceptions: | The user provides invalid input. The web service encounters an error with the GET request. |
| Special Requirements: | The user is logged in, must have valid credentials, and has sufficient permissions to add data type. |

Table 3-26. Use case 19.

|  |  |
| --- | --- |
| Figure 3-22. Use case 20. | |
| Use Case ID: | UC20 |
| Use Case Name: | Edit Data Type Record |
| Actors: | User, Web Service |
| Entry Condition(s): | The user has selected the “Save” button with values inputted in the Update Data Type Page. |
| Basic Flow: | 1. The user sends a PUT request after selecting the save button in the Update Data Type page. 2. The web service edits the data of the targeted data type record with the data provided by the user. |
| Exit Condition(s): | The service successfully updates the data type record. |
| Exceptions: | The user provides invalid input. The web service encounters an error with the PUT request. |
| Special Requirements: | The user is logged in, must have valid credentials, and has sufficient permission to view data type. |

Table 3-27. Use case 20.

|  |  |
| --- | --- |
| Figure 3-23. Use case 21. | |
| Use Case ID: | UC21 |
| Use Case Name: | Delete Data Type Record |
| Actors: | User, Web Service |
| Entry Condition(s): | The user has selected the “Delete” button on a data type record from the list of data types. |
| Basic Flow: | 1. The user sends a DELETE request after selecting the delete button beside a data type on the list of data linkages. 2. The web service deletes the data type record from the database. |
| Exit Condition(s): | The service successfully deletes the data type record. |
| Exceptions: | The user has insufficient permissions. The database lacks information needed. The web service encounters an error with the DELETE request. |
| Special Requirements: | The user is logged in, must have valid credentials, and has sufficient permission to delete data types. |

Table 3-28. Use case 21.

###### 3.2.2.3.1 Associated Functional Requirements

**WS-FR-6**

**The application shall send GET requests to the web service when the user selects List from the Data Type submenu.**

**WS-FR-7**

**The application shall send POST requests to the web service when the user selects Submit button from the Add Data Type Page.**

**WS-FR-8**

**The application shall send PUT requests to the web service when the user selects Save button from the Update Data Type page.**

**WS-FR-9**

**The application shall send DELETE requests to the web service when the user selects Delete button from the list of data types.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req. ID | Type | Priority | Description | Owner |
| WS-FR-6 | Functional | Essential | Retrieve data types | RF |
| WS-FR-7 | Functional | Essential | Create data type | RF |
| WS-FR-8 | Functional | Essential | Update data type | RF |
| WS-FR-9 | Functional | Essential | Delete data type | RF |

Table 3-29. Applications web services functional requirements.

##### 3.2.2.3 Data Linkage Web Services

|  |  |
| --- | --- |
| Figure 3-24. Use case 22. | |
| Use Case ID: | UC22 |
| Use Case Name: | View Data Linkage Record |
| Actors: | User, Web Service |
| Entry Condition(s): | The user is logged in, has valid permissions and is in the home page. |
| Basic Flow: | 1. The user sends a GET request to the web service by selecting List from the Data Linkage menu. 2. The web service returns a list of all data linkages. 3. The application displays the results of the GET request in the main content area. |
| Exit Condition(s): | The list of data linkage records is successfully displayed in the main content area. |
| Exceptions: | The user has insufficient permissions. The database lacks information needed. The web service encounters an error with the GET request. |
| Special Requirements: | The user is logged in and must have valid credentials. |

Table 3-30. Use case 22.

|  |  |
| --- | --- |
| Figure 3-25. Use case 23. | |
| Use Case ID: | UC23 |
| Use Case Name: | Add Data Linkage Record |
| Actors: | User, Web Service |
| Entry Condition(s): | The user has selected the “Submit” button with values inputted in the Add Data Linkage page. |
| Basic Flow: | 1. The user sends a POST request after selecting the submit button in the Add Data Linkage page. 2. The web service adds the data linkage record into the database. |
| Exit Condition(s): | The service successfully stores the new data linkage record into the database. |
| Exceptions: | The user has insufficient permissions. The database lacks information needed. The web service encounters an error with the POST request. |
| Special Requirements: | The user is logged in, must have valid credentials, and has sufficient permissions to add data linkages. |

Table 3-31. Use case 23.

|  |  |
| --- | --- |
| Figure 3-26. Use case 24. | |
| Use Case ID: | UC24 |
| Use Case Name: | Delete Data Linkage Record |
| Actors: | User, Web Service |
| Entry Condition(s): | The user has selected the “Delete” button on a data link record from the list of data linkage. |
| Basic Flow: | 1. The user sends a DELETE request after selecting the delete button beside a data link on the list of data linkages. 2. The web service deletes the data linkage record from the database. |
| Exit Condition(s): | The service successfully deletes the data linkage record. |
| Exceptions: | The user has insufficient permissions. The database lacks information needed. The web service encounters an error with the DELETE request. |
| Special Requirements: | The user is logged in, must have valid credentials, and has sufficient permission to delete data linkages. |

Table 3-32. Use case 24.

###### 3.2.2.4.1 Associated Functional Requirements

**WS-FR-10**

**The application shall send GET requests to the web service when the user selects List from the Data Linkage submenu.**

**WS-FR-11**

**The application shall send POST requests to the web service when the user selects Submit button from the Add Data Linkage Page.**

**WS-FR-12**

**The application shall send DELETE requests to the web service when the user selects Delete button from the list of data linkage.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req. ID | Type | Priority | Description | Owner |
| WS-FR-10 | Functional | Essential | Retrieve data linkages | JY |
| WS-FR-11 | Functional | Essential | Create data linkage | JY |
| WS-FR-12 | Functional | Essential | Delete data linkage | JY |

Table 3-33. Web services layer functional requirements.

#### 3.2.3 Persistence Layer Use Case Model

##### 3.2.3.1 Persistence Layer Introduction

The persistence layer shall provide information about the database and common CRUD functionalities.

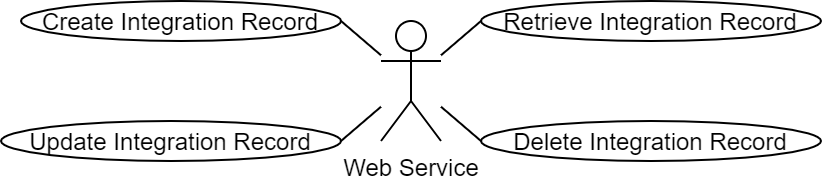


Figure 3-27. Use case 25.

##### 3.2.3.2 Associated Functional Requirements

**PL-FR-1**

**The database shall be able to create entries.**

**PL-FR-2**

**The database shall be able to retrieve entries.**

**PL-FR-3**

**The database shall be able to update entries.**

**PL-FR-4**

**The database shall be able to delete entries.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req. ID | Type | Priority | Description | Owner |
| PL-FR-1 | Functional | Essential | Database Create | RF, AA |
| PL-FR-2 | Functional | Essential | Database Retrieve | RF, AA |
| PL-FR-3 | Functional | Essential | Database Update | RF, AA |
| PL-FR-4 | Functional | Essential | Database Delete | RF, AA |

Table 3-34. Persistence layer functional requirements.

### 3.3 Use Case Model Rationale

|  |  |
| --- | --- |
| UC13 | The application shall send GET requests to the web service to verify user credentials when the user attempts to login. |
| 1. Will the credentials be encrypted on the back end? 2. Will we implement the authentication layer directly or will we use a third-party source? 3. Will it be possible to add new users via the application interface? | |
| Assumptions   * A server containing the necessary credentials will be available upon deployment. | |
| Option 1 | Basic authentication using username and password. |
| Option 2 | JSON token-based authentication via OAuth2.0 protocol. |
| Option 3 | XML token-based authentication via SAML 2.0 protocol. |
| Choice | Option 1 |
| Rationale | Option 1 is the best option due to time constraints and because the customer intends to add their own authentication measures after the application has been deployed. |

Table 3-35. Use case 1 rationale.

### 3.4. Non-functional requirements

#### 3.4.1 Usability

NFR1: The site shall function as a single-page application per the client’s request. The integration data will be displayed using a sortable, table style grid.

#### 3.4.2 Performance

NFR2: The site shall respond to user input within a reasonable amount of time.

#### 3.4.3 Portability

NFR3: The application shall function in modern web browsers.

#### 3.4.4 Security

NFR4: The system shall encrypt and store user credentials in an authentication database per the client’s request.

# 4. Architecture

### 4.1 Architectural Style

The development team has chosen to use a layered architecture style to implement the ETA. The layered architecture style supports the application by utilizing an application logic layer to serve as the data controller. This will enable the application to use CRUD operations on the data within the data layer while remaining on a single page in the presentation layer.

### 4.2 Architectural Model

The architectural model used in this document is based on the 4+1 View Model of Software Architecture by Philippe B. Kruchten. The 4+1 View Model of Software Architecture is a model for describing the architecture of software-intensive systems, based on the use of multiple, concurrent views.

#### 4.2.1 4+1 Architecture View Model Diagram

|  |
| --- |
|  |
| Figure 4-1. 4+1 View Model Diagram. |

#### 4.2.2 System Architecture Models/Views

The Perspecta ETA is divided into layers based on the N-tier architecture. The layering model is based on a responsibility layering strategy that associates each layer with a particular responsibility.

This structure has been chosen because it isolates various system responsibilities from one another in order to support the RESTful API as well as system development.

Each layer has specific responsibilities.

* The presentation layer deals with the presentation logic and with rendering the functions within the page.
* The application logic layer manages access to the data layer as well as the information contained within the data layer.
* The data layer deals with managing the database records and is being provided by the customer.

##### 4.2.2.1 Logical View

###### 4.2.2.1.1 Overview

The Logical View describes the design's object model.

|  |
| --- |
|  |
| Figure 4-2. Perspecta ETA N-tier Architecture. |

##### 4.2.2.2 Process View

###### 4.2.2.2.1 Overview

The Process View is used to show the high-level structure and composition of the application processes that are available to the user.

###### 4.2.2.2.2 Login Process

|  |
| --- |
|  |
| Figure 4-3. User login activity diagram. |

###### 4.2.2.2.3 Add Application Process

|  |
| --- |
|  |
| Figure 4-4. Add application activity diagram. |

###### 4.2.2.2.4 List Applications Process

|  |
| --- |
|  |
| Figure 4-5. List applications activity diagram. |

###### 4.2.2.2.5 Update Application Process

|  |
| --- |
|  |
| Figure 4-6. Update application activity diagram. |

###### 4.2.2.2.6 Delete Application Process

|  |
| --- |
|  |
| Figure 4-7. Delete application activity diagram. |

###### 4.2.2.2.7 Add Data Type Process

|  |
| --- |
|  |
| Figure 4-8. Add data type activity diagram. |

###### 4.2.2.2.8 List Data Types Process

|  |
| --- |
|  |
| Figure 4-9. List data types activity diagram. |

###### 4.2.2.2.9 Update Data Type Process

|  |
| --- |
|  |
| Figure 4-10. Update data type activity diagram. |

###### 4.2.2.2.10 Delete Data Type Process

|  |
| --- |
|  |
| Figure 4-11. Delete data type activity diagram. |

###### 4.2.2.2.11 Add Data Link Process

|  |
| --- |
|  |
| Figure 4-12. Add data link activity diagram. |

###### 4.2.2.2.12 List Data Links Process

|  |
| --- |
|  |
| Figure 4-13. List data links activity diagram. |

###### 4.2.2.2.13 Delete Data Link Process

|  |
| --- |
|  |
| Figure 4-14. Delete data link activity diagram. |

##### 4.2.2.3 Development View

###### 4.2.2.3.1 Overview

The Development View describes the layers and subsystems of the application.

###### 4.2.2.3.2 Architecture Model Diagram

|  |
| --- |
| Diagram  Description automatically generated |
| Figure 4-15. ETA architecture model diagram. |

###### 4.2.2.3.3 Component Diagram

|  |
| --- |
|  |
| Figure 4-16. ETA component diagram. |

###### 4.2.2.3.4 Development Package Diagram

|  |
| --- |
|  |
| Figure 4-17. ETA development package diagram. |

##### 4.2.2.4 Physical View

###### 4.2.2.4.1 Overview

The Physical View describes the mapping of the software onto the hardware and shows the system's distributed aspects.

###### 4.2.2.4.2 Physical Deployment Diagram

|  |
| --- |
|  |
| Figure 4-18. ETA deployment diagram. |

### 4.3 Technology, Software, and Hardware Used

#### 4.3.1 Technology

* Spring Boot – Spring Boot is an open source Java-based framework used to create Micro Services.
* Java – Java is a class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible.
* RESTful Web Services – Representational State Transfer (REST) is an architectural style that specifies constraints, such as the uniform interface, that if applied to a web service induce desirable properties, such as performance, scalability, and modifiability, that enable services to work best on the Web.
* Angular – Angular is a TypeScript-based open-source web application framework led by the Angular Team at Google and by a community of individuals and corporations.
* Typescript – TypeScript is an open-source programming language developed and maintained by Microsoft. It is a strict syntactical superset of JavaScript and adds optional static typing to the language. TypeScript is designed for development of large applications and transcompiles to JavaScript.
* JavaScript – JavaScript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm.
* Microsoft SQL Server – Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network.
* Structured Query Language (SQL) – SQL is a domain-specific language used in programming and designed for managing data held in a relational database management system, or for stream processing in a relational data stream management system.

#### 4.3.2 Software

* Spring Boot – Spring Boot is an open source Java-based framework used to create Micro Services.
* Java – Java is a class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible.
* RESTful Web Services – Representational State Transfer (REST) is an architectural style that specifies constraints, such as the uniform interface, that if applied to a web service induce desirable properties, such as performance, scalability, and modifiability, that enable services to work best on the Web.
* Angular – Angular is a TypeScript-based open-source web application framework led by the Angular Team at Google and by a community of individuals and corporations.
* Typescript – TypeScript is an open-source programming language developed and maintained by Microsoft. It is a strict syntactical superset of JavaScript and adds optional static typing to the language. TypeScript is designed for development of large applications and transcompiles to JavaScript.
* JavaScript – JavaScript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm.
* Microsoft SQL Server – Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network.
* Structured Query Language (SQL) – SQL is a domain-specific language used in programming and designed for managing data held in a relational database management system, or for stream processing in a relational data stream management system.
* Node.js – Node.js is an open-source, cross-platform, back end, JavaScript runtime environment that executes JavaScript code outside a web browser.
* Angular CLI – The Angular CLI is a command-line interface tool that you use to initialize, develop, scaffold, and maintain Angular applications directly from a command shell.
* Eclipse IDE – Eclipse is an integrated development environment used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment.
* Maven – Maven is a build automation tool used primarily for Java projects. Maven can also be used to build and manage projects written in C#, Ruby, Scala, and other languages. The Maven project is hosted by the Apache Software Foundation, where it was formerly part of the Jakarta Project.
* Microsoft SQL Server Management Studio (SSMS) – SSMS is a software application developed by Microsoft that is used to configure, manage, and administer all components within Microsoft SQL Server. This tool includes both script editors and graphical tools which work with objects and features of the server.

#### 4.3.3 Hardware

|  |
| --- |
|  |
| Figure 4-19. Hardware structure diagram. |

* Local System – User’s device with existing operating system and web browser to access the Enterprise Tracking Application website.
* Application Server – An existing web server provided by the client that provides the RESTful web service and front-end controls.
* Database Management System Server – An existing database server provided by the client.

### 4.4 Rationale for the Architectural Style and Model

#### 4.4.1 Data Flow Architectural Style

This Architectural style consists of filters that perform transformations on data received and pipes which serve as connectors for the stream of the data being transformed. They have shown to be highly effective in Embedded system software design, where the system is manipulated by process control variable data, and applications that aim to maintain specified properties of the outputs of the process at given reference values. Although there are apparent advantages to this style, it is not suitable for this project because it only works in one direction and we would have to loop through the same algorithm which will cause extreme overhead of data transformation between filters, and implementation complexity. This style is not suitable for dynamic interactions that will be needed in this project therefore we have decided NOT to take this architectural style approach.

#### 4.4.2 Model-View-Controller Architectural Style

The model-view-controller (MVC) has been the traditional architectural pattern for developing applications having user interface. MVC renders testability, maintainability and scalability. It also satisfies Single Responsibility Principle of SOLID. Some of the disadvantages, however, are that the View depends both on Controller and Model so when the updating the UI data the view might have to go to the model for some functionality and to the controller for other, and that the Model just does too much work from getting the data from the network/database to informing the Controller about whether it could get the data or not, and even preparing the result to be displayed on the View. Other architectural styles have come up with solutions to these issues. For these reasons and because the sponsor has advised us against this style, we decided not to implement the Model-View-Controller Architectural style.

#### 4.4.3 Layered Architectural Style

The development team selected the Layered Architectural style to accommodate for functionalities and RESTful services that the utilized by the application. The architectural style is comprised of layers perform specific roles in the application and contain components and modules with similar functionalities. It provides an abstract view of the system and defines responsibilities and relationships between each layer. High level requirements can be easily implemented as separate layers, which will streamline development by minimizing dependencies, as well as improve testability, maintainability and reusability through layer isolation. This also allows high cohesion due to well-defined functional boundaries and low coupling from abstraction and event-based communication. While the architecture has innate performance issues, the trade-off for ease of implementation and improvement to many nonfunctional requirements puts it above the other architectural styles considered for this software.

#### 4.4.4 Reason for Architecture Selection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Data Flow | MVC | Layered Architecture |
| Modifiability | Algorithms | - | + | + |
| Data Representation | -- | + | ++- |
| Enhanceability | Add Function | +- | ++- | ++ |
| Performance | Space | ++ | + | - |
| Time | + | + | - |
| Reusability |  | -- | + | + |
| Intuitiveness |  | - | + | +- |
| Scalability |  | -- | - | ++ |
| Security |  | + | - | ++ |

Table 4-1. Architecture Style Tradeoffs.

NFR1: Layered architecture is a key component of a RESTful architecture. Applying an Angular front end will ensure that the single-page application requirement is met.

NFR2 The system has been implemented using a RESTful architecture which will improve performance by minimizing the need to reload unnecessary web pages.

NFR3: This system has been implemented in Java and utilizes OOP with ADT which will allow it to function on different systems.

NFR4: A layered architecture implementation will allow the application to function securely by allowing users data access through a middleware layer. The single-page web design will also contribute to security by preventing sensitive data from being cached within the browser history.

# 5. Design

### 5.1 GUI Design

The GUI Design section contains various wireframes that depict the text boxes, drop-down menus, and buttons for their related functions. The first two subsections display the initial Login Page wireframe and the default Home Page wireframe. The remaining subsections are divided based on pages that encompass similar functionalities. The Applications, Data Type, and Data Linkage wireframes show the associated records, as well as additional windows and buttons for record management.

#### 5.1.1 Login Page

|  |
| --- |
|  |
| Figure 5-1. Login Page Wireframe. |

#### 5.1.2 Home Page

|  |
| --- |
|  |
| Figure 5-2. Home Page Wireframe. |

#### 5.1.3 Applications Page

##### 5.1.3.1 List Applications

|  |
| --- |
|  |
| Figure 5-3. List Applications Wireframe. |

##### 5.1.3.2 Add/Edit Application

|  |
| --- |
|  |
| Figure 5-4. Add/Edit Applications Wireframe. |

##### 5.1.3.3 Delete Application

|  |
| --- |
|  |
| Figure 5-5. Delete Applications Wireframe. |

#### 5.1.4 Data Types Page

##### 5.1.4.1 List Data Types

|  |
| --- |
|  |
| Figure 5-6. List Data Types Wireframe. |

##### 5.1.4.2 Add Data Type

|  |
| --- |
|  |
| Figure 5-7. Add Data Type Wireframe. |

##### 5.1.4.3 Edit Data Type

|  |
| --- |
|  |
| Figure 5-8. Edit Data Type Wireframe. |

##### 5.1.4.4 Delete Data Type

|  |
| --- |
|  |
| Figure 5-9. Delete Data Type Wireframe. |

#### 5.1.5 Data Linkage Page

##### 5.1.5.1 List Data Linkage

|  |
| --- |
|  |
| Figure 5-10. List Data Linkage Wireframe. |

##### 5.1.5.2 Add Data Linkage

|  |
| --- |
|  |
| Figure 5-11. Add Data Linkage Wireframe. |

##### 5.1.5.3 Delete Data Linkage

|  |
| --- |
|  |
| Figure 5-12. Delete Data Linkage Wireframe. |

### 5.2 Static Model

The Static Model section contains diagrams that model the structure of the application and its components. The first subsection shows System Class Diagrams that address the Front-End and RESTful Service classes. The remaining subsections show class diagrams for each layer in the N-Tier architecture, which are the GUI, Application, and Database. Each diagram contains dependencies and responsibilities of their associated classes.

#### 5.2.1 System Class Diagrams

##### 5.2.1.1 System Front-End Class Diagram

|  |
| --- |
|  |
| Figure 5-13. System Front-End Class Diagram. |

##### 5.2.1.2 System RESTful Service Class Diagram

|  |
| --- |
|  |
| Figure 5-14. RESTful Services Class Diagram. |

#### 5.2.2 Presentation Layer Class Diagrams

##### 5.2.2.1 GUI Screen Class Diagram

|  |
| --- |
|  |
| Figure 5-15. GUI Screen Class Diagram. |

##### 5.2.2.2 GUI Service Class Diagram

|  |
| --- |
|  |
| Figure 5-16. GUI Service Class Diagram. |

#### 5.2.3 Application Layer Class Diagrams

##### 5.2.3.1 Spring Boot Server Startup Class Diagram

|  |
| --- |
|  |
| Figure 5-17. Spring Boot Server Startup Class Diagram. |

##### 5.2.3.2 REST API User Authentication Class Diagram

|  |
| --- |
|  |
| Figure 5-18. REST API User Authentication Class Diagram. |

##### 5.2.3.3 REST API List Applications Class Diagram

|  |
| --- |
|  |
| Figure 5-19. REST API Applications Class Diagram. |

##### 5.2.3.4 REST API List Application Links Class Diagram

|  |
| --- |
|  |
| Figure 5-20. REST API Application Links Class Diagram. |

#### 5.2.4 Data Layer Class Diagrams

##### 5.2.4.1 Database User/Group ER Diagrams

|  |
| --- |
|  |
| Figure 5-21. User/Group ER Diagram. |

##### 5.2.4.2 Database Application Data ER Diagram

|  |
| --- |
|  |
| Figure 5-22. Application Data ER Diagram. |

### 5.3 Dynamic Model

The Dynamic Model section contains sequence diagrams that show interactions of classes derived from the static model. This section is divided similarly to the Static model, which is through the layers it represents. The Presentation Layer and Application Layer subsections contain the order of instantiation and message relationships associated with performing a given function.

#### 5.3.1 Presentation Layer Class Diagrams

##### 5.3.1.1 GUI List Applications Sequence Diagrams

|  |
| --- |
|  |
| Figure 5-23. GUI List Applications Sequence Diagram. |

##### 5.3.1.2 GUI CRUD Applications Sequence Diagrams

|  |
| --- |
|  |
| Figure 5-24. GUI Create/Update/Delete Applications Sequence Diagram. |

#### 5.3.2 Application Layer Class Diagrams

##### 5.3.2.1 User Login Sequence Diagram

|  |
| --- |
|  |
| Figure 5-25. User Login Sequence Diagram. |

##### 5.3.2.2 Applications Sequence Diagrams

###### 5.3.2.2.1 List Applications Sequence Diagram

|  |
| --- |
|  |
| Figure 5-26. List Applications Sequence Diagram. |

###### 5.3.2.2.2 Add Application Sequence Diagram

|  |
| --- |
|  |
| Figure 5-27. Add Application Sequence Diagram. |

###### 5.3.2.2.3 Edit Application Sequence Diagram

|  |
| --- |
|  |
| Figure 5-28. Edit Application Sequence Diagram. |

###### 5.3.2.2.4 Delete Application Sequence Diagram

|  |
| --- |
|  |
| Figure 5-29. Delete Application Sequence Diagram. |

##### 5.3.2.3 Data Types Sequence Diagrams

###### 5.3.2.3.1 List Data Types Sequence Diagram

|  |
| --- |
|  |
| Figure 5-30. List Data Types Sequence Diagram. |

###### 5.3.2.3.2 Add Data Type Sequence Diagram

|  |
| --- |
|  |
| Figure 5-31. Add Data Type Sequence Diagram. |

###### 5.3.2.3.3 Edit Data Type Sequence Diagram

|  |
| --- |
|  |
| Figure 5-32. Edit Data Type Sequence Diagram. |

###### 5.3.2.3.4 Delete Data Type Sequence Diagram

|  |
| --- |
|  |
| Figure 5-33. Delete Data Type Sequence Diagram. |

##### 5.3.2.4 Data Linkage Sequence Diagrams

###### 5.3.2.4.1 List Data Linkages Sequence Diagram

|  |
| --- |
|  |
| Figure 5-34. List Data Linkages Sequence Diagram. |

###### 5.3.2.4.2 Add Data Linkage Sequence Diagram

|  |
| --- |
|  |
| Figure 5-35. Add Data Linkage Sequence Diagram. |

###### 5.3.2.4.3 Delete Data Linkage Sequence Diagram

|  |
| --- |
|  |
| Figure 5-36. Delete Data Linkage Sequence Diagram. |

### 5.4 Rationale for the Detailed Design Model

For this project, we were tasked with creating a single page web application that is capable of functioning in any web browser. Furthermore, we were asked to implement RESTful services to support the web application that is designed to work with an existing Microsoft SQL Server database. Given these specific requirements, we chose the Angular Framework for the web application because it is compatible with all the current most popular web browsers and because it is purpose-built for implementing single-page web applications. The single page capabilities of the Angular Framework also add to the system’s overall security by keeping specific details of the user’s actions while using the site functions out of their browser cache and browsing history. Spring Boot is being utilized as the foundation for the requested RESTful services because it is one of the most popular Micro Services frameworks in the industry today and because of the portability that it brings to the table by running on Java.

### 5.5 System Traceability

The System Traceability section displays how the requirements from the SRS are being satisfied throughout the design document. The section is split into functional and non-functional requirements subsections. The functional requirements subsection is divided into three other subsections that contain requirements associated with the GUI, Web Services, and the Database.

#### 5.5.1 Functional Requirements

##### 5.5.1.1 GUI Layout Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req. ID | Type | Priority | Description | Owner |
| UI-FR-1 | Functional | Essential | Login email | JF, EL |
| UI-FR-2 | Functional | Essential | Login password | JF, EL |
| UI-FR-3 | Functional | Essential | Generic login failure message | JF, EL |
| UI-FR-4 | Functional | Essential | Login submit button | JF, EL |
| UI-FR-5 | Functional | Essential | Login cancel button | JF, EL |
| UI-FR-6 | Functional | Essential | Home page header area | JF, EL |
| UI-FR-7 | Functional | Essential | Home page footer area | JF, EL |
| UI-FR-8 | Functional | Essential | Home page menu area | JF, EL |
| UI-FR-9 | Functional | Essential | Home page menu functionality | JF, EL |
| UI-FR-10 | Functional | Essential | Home page main area | JF, EL |

Table 5-1. GUI Layout Requirements.

###### 5.5.1.1.1 GUI Layout Traceability Matrix

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Req. ID | Fig. 2-1 | Fig. 2-2 | Fig. 3-1 | Fig. 3-3 | Fig. 3-6 | Fig. 4-3 |
| UI-FR-1 | X |  | X | X | X | X |
| UI-FR-2 | X |  | X | X | X | X |
| UI-FR-3 | X |  | X | X | X | X |
| UI-FR-4 | X |  | X | X | X | X |
| UI-FR-5 | X |  | X | X | X | X |
| UI-FR-6 |  | X | X | X |  |  |
| UI-FR-7 |  | X | X | X |  |  |
| UI-FR-8 |  | X | X | X |  |  |
| UI-FR-9 |  | X | X | X |  |  |
| UI-FR-10 |  | X | X | X |  |  |

Table 5-2. GUI Layout Traceability Matrix

###### 5.5.1.1.2 Graphical User Interface Traceability Details

* The System Front-End Class Diagram (Figure 5-13) and the GUI Screen Class Diagram (Figure 5-15) encompass all Graphical User Interface Requirements.
* UI-FR-1, UR-FR-2, UR-FR-3, UR-FR-4, and UR-FR-5 are achieved by the Login Page Wireframe (Figure 5-1), the REST API User Authentication Class Diagram (Figure 5-18), and User Login Sequence Diagram (Figure 5-25).
* UI-FR-6, UI-FR-7, UI-FR-8, UI-FR-9, and UI-FR-10 are achieved by the Home Page Wireframe (Figure 5-2).

##### 5.5.1.2 Applications GUI Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req. ID | Type | Priority | Description | Owner |
| UI-FR-11 | Functional | Essential | List applications | JF, EL |
| UI-FR-11.1 | Functional | Essential | Applications table content – admin user | JF, EL |
| UI-FR-11.2 | Functional | Essential | Applications table content – standard user | JF, EL |
| UI-FR-12 | Functional | Essential | Add application | JF, EL |
| UI-FR-12.1 | Functional | Essential | Add application form content | JF, EL |
| UI-FR-12.2 | Functional | Essential | Add application populated multi-select data types from db | JF, EL |
| UI-FR-13 | Functional | Essential | Update application | JF, EL |
| UI-FR-13.1 | Functional | Essential | Update application form content | JF, EL |
| UI-FR-13.2 | Functional | Essential | Update application populated multi-select data types from db | JF, EL |
| UI-FR-13.3 | Functional | Essential | Update form fields populated with application data | JF, EL |
| UI-FR-14 | Functional | Essential | Delete application | JF, EL |
| UI-FR-14.1 | Functional | Essential | Delete application confirmation | JF, EL |

Table 5-3. Applications GUI Requirements.

###### 5.5.1.2.1 Applications GUI Traceability Matrix

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Req. ID | Fig. 2-3 | Fig. 2-4 | Fig. 2-5 | Fig. 2-6 | Fig. 2-7 | Fig. 2-8 | Fig. 2-9 | Fig. 2-10 | Fig. 2-11 | Fig. 2-12 | Fig. 3-2 | Fig. 3-7 | Fig. 3-8 | Fig. 4-4 | Fig. 4-5 | Fig. 4-6 | Fig. 4-7 | Fig. 4-8 | Fig. 4-9 | Fig. 4-10 | Fig. 4-11 | Fig. 4-12 | Fig. 4-13 | Fig. 4-14 |
| UI-FR-11 | X |  |  |  |  |  |  |  |  |  | X | X |  | X |  |  |  |  |  |  |  |  |  |  |
| UI-FR-11.1 |  | X |  |  |  |  |  |  |  |  | X |  |  |  | X |  |  |  |  |  |  |  |  |  |
| UI-FR-11.2 |  | X |  |  |  |  |  |  |  |  | X |  |  |  |  | X |  |  |  |  |  |  |  |  |
| UI-FR-12 |  |  | X |  |  |  |  |  |  |  | X |  |  |  |  |  | X |  |  |  |  |  |  |  |
| UI-FR-12.1 |  |  |  | X |  |  |  |  |  |  | X |  |  |  |  |  |  | X |  |  |  |  |  |  |
| UI-FR-12.2 |  |  |  |  | X |  |  |  |  |  | X |  |  |  |  |  |  |  | X |  |  |  |  |  |
| UI-FR-13 |  |  |  |  |  | X |  |  |  |  | X |  |  |  |  |  |  |  |  | X |  |  |  |  |
| UI-FR-13.1 |  |  |  |  |  |  | X |  |  |  | X |  |  |  |  |  |  |  |  |  | X |  |  |  |
| UI-FR-13.2 |  |  |  |  |  |  |  | X |  |  | X |  | X |  |  |  |  |  |  |  |  | X |  |  |
| UI-FR-13.3 |  |  |  |  |  |  |  |  | X |  | X |  | X |  |  |  |  |  |  |  |  |  | X |  |
| UI-FR-14 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UI-FR-14.1 |  |  |  |  |  |  |  |  |  | X | X |  | X |  |  |  |  |  |  |  |  |  |  | X |

Table 5-4. Applications GUI Traceability Matrix.

##### 5.5.1.3 Data Types GUI Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req. ID | Type | Priority | Description | Owner |
| UI-FR-15 | Functional | Essential | List data types | JF, EL |
| UI-FR-15.1 | Functional | Essential | Data type table content – admin user | JF, EL |
| UI-FR-15.2 | Functional | Essential | Data type table content – standard user | JF, EL |
| UI-FR-16 | Functional | Essential | Add data type | JF, EL |
| UI-FR-16.1 | Functional | Essential | Add data type form content | JF, EL |
| UI-FR-17 | Functional | Essential | Update data type | JF, EL |
| UI-FR-17.1 | Functional | Essential | Update data type form content | JF, EL |
| UI-FR-17.2 | Functional | Essential | Update form fields populated with data type data | JF, EL |
| UI-FR-18 | Functional | Essential | Delete data type | JF, EL |
| UI-FR-18.1 | Functional | Essential | Delete data type constraint protection | JF, EL |
| UI-FR-18.2 | Functional | Essential | Delete data type confirmation | JF, EL |

Table 5-5. Data Types GUI Requirements.

###### 5.5.1.3.1 Data Types GUI Traceability Matrix

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Req. ID | Fig. 2-3 | Fig. 2-4 | Fig. 2-5 | Fig. 2-6 | Fig. 2-7 | Fig. 2-8 | Fig. 2-9 | Fig. 2-10 | Fig. 2-11 | Fig. 2-12 | Fig. 3-2 | Fig. 3-7 | Fig. 3-8 | Fig. 4-4 | Fig. 4-5 | Fig. 4-6 | Fig. 4-7 | Fig. 4-8 | Fig. 4-9 | Fig. 4-10 | Fig. 4-11 | Fig. 4-12 | Fig. 4-13 | Fig. 4-14 |
| UI-FR-15 | X |  |  |  |  |  |  |  |  |  | X | X |  | X |  |  |  |  |  |  |  |  |  |  |
| UI-FR-15.1 |  | X |  |  |  |  |  |  |  |  | X |  |  |  | X |  |  |  |  |  |  |  |  |  |
| UI-FR-15.2 |  | X |  |  |  |  |  |  |  |  | X |  |  |  |  | X |  |  |  |  |  |  |  |  |
| UI-FR-16 |  |  | X |  |  |  |  |  |  |  | X |  |  |  |  |  | X |  |  |  |  |  |  |  |
| UI-FR-16.1 |  |  |  | X |  |  |  |  |  |  | X |  |  |  |  |  |  | X |  |  |  |  |  |  |
| UI-FR-17 |  |  |  |  | X |  |  |  |  |  | X |  |  |  |  |  |  |  | X |  |  |  |  |  |
| UI-FR-17.1 |  |  |  |  |  | X |  |  |  |  | X |  |  |  |  |  |  |  |  | X |  |  |  |  |
| UI-FR-17.2 |  |  |  |  |  |  | X |  |  |  | X |  |  |  |  |  |  |  |  |  | X |  |  |  |
| UI-FR-18 |  |  |  |  |  |  |  | X |  |  | X |  | X |  |  |  |  |  |  |  |  | X |  |  |
| UI-FR-18.1 |  |  |  |  |  |  |  |  | X |  | X |  | X |  |  |  |  |  |  |  |  |  | X |  |
| UI-FR-18.2 |  |  |  |  |  |  |  |  |  | X | X |  | X |  |  |  |  |  |  |  |  |  |  | X |

Table 5-6. Data Types GUI Traceability Matrix.

##### 5.5.1.4 Data Linkages GUI Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req. ID | Type | Priority | Description | Owner |
| UI-FR-19 | Functional | Essential | List data links | JF, EL |
| UI-FR-19.1 | Functional | Essential | Data link table content – admin user | JF, EL |
| UI-FR-19.2 | Functional | Essential | Data link table content – standard user | JF, EL |
| UI-FR-20 | Functional | Essential | Add data link | JF, EL |
| UI-FR-20.1 | Functional | Essential | Add data link form content | JF, EL |
| UI-FR-20.2 | Functional | Essential | Add data link form populated applications | JF, EL |
| UI-FR-20.3 | Functional | Essential | Add data link form populated data types | JF, EL |
| UI-FR-21 | Functional | Essential | Delete data link | JF, EL |
| UI-FR-21.1 | Functional | Essential | Delete data link confirmation | JF, EL |

Table 5-7. Data Linkages GUI Requirements.

###### 5.5.1.4.1 Data Linkages GUI Traceability Matrix

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Req. ID | Fig. 2-3 | Fig. 2-4 | Fig. 2-5 | Fig. 2-6 | Fig. 2-7 | Fig. 2-8 | Fig. 2-9 | Fig. 2-10 | Fig. 2-11 | Fig. 2-12 | Fig. 3-2 | Fig. 3-7 | Fig. 3-8 | Fig. 4-4 | Fig. 4-5 | Fig. 4-6 | Fig. 4-7 | Fig. 4-8 | Fig. 4-9 | Fig. 4-10 | Fig. 4-11 | Fig. 4-12 | Fig. 4-13 | Fig. 4-14 |
| UI-FR-19 | X |  |  |  |  |  |  |  |  |  | X | X |  | X |  |  |  |  |  |  |  |  |  |  |
| UI-FR-19.1 |  | X |  |  |  |  |  |  |  |  | X |  |  |  | X |  |  |  |  |  |  |  |  |  |
| UI-FR-19.2 |  | X |  |  |  |  |  |  |  |  | X |  |  |  |  | X |  |  |  |  |  |  |  |  |
| UI-FR-20 |  |  | X |  |  |  |  |  |  |  | X |  |  |  |  |  | X |  |  |  |  |  |  |  |
| UI-FR-20.1 |  |  |  | X |  |  |  |  |  |  | X |  |  |  |  |  |  | X |  |  |  |  |  |  |
| UI-FR-20.2 |  |  |  |  | X |  |  |  |  |  | X |  |  |  |  |  |  |  | X |  |  |  |  |  |
| UI-FR-20.3 |  |  |  |  |  | X |  |  |  |  | X |  |  |  |  |  |  |  |  | X |  |  |  |  |
| UI-FR-21 |  |  |  |  |  |  | X |  |  |  | X |  |  |  |  |  |  |  |  |  | X |  |  |  |
| UI-FR-21.1 |  |  |  |  |  |  |  | X |  |  | X |  | X |  |  |  |  |  |  |  |  | X |  |  |

Table 5-8. Data Linkages GUI Traceability Matrix.

##### 5.5.1.5 RESTful Web Services Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req. ID | Type | Priority | Description | Owner |
| WS-FR-1 | Functional | Essential | View user data send GET | RF |
| WS-FR-2 | Functional | Essential | View applications send GET | AA |
| WS-FR-3 | Functional | Essential | Create applications send POST | AA |
| WS-FR-4 | Functional | Essential | Edit applications send PUT | AA |
| WS-FR-5 | Functional | Essential | Delete applications send DELETE | AA |
| WS-FR-6 | Functional | Essential | View data types send GET | RF |
| WS-FR-7 | Functional | Essential | Create data types send POST | RF |
| WS-FR-8 | Functional | Essential | Edit data types send PUT | RF |
| WS-FR-9 | Functional | Essential | Delete data types send DELETE | RF |
| WS-FR-10 | Functional | Essential | View data linkage send GET | JY |
| WS-FR-11 | Functional | Essential | Create data linkage send POST | JY |
| WS-FR-12 | Functional | Essential | Delete data linkage send DELETE | JY |

Table 5-9. RESTful Web Services Requirements.

###### 5.5.1.5.1 RESTful Web Services Traceability Matrix

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Req. ID | Fig. 2-3 | Fig. 2-4 | Fig. 2-5 | Fig. 2-6 | Fig. 2-7 | Fig. 2-8 | Fig. 2-9 | Fig. 2-10 | Fig. 2-11 | Fig. 2-12 | Fig. 3-2 | Fig. 3-7 | Fig. 3-8 | Fig. 4-4 | Fig. 4-5 | Fig. 4-6 | Fig. 4-7 | Fig. 4-8 | Fig. 4-9 | Fig. 4-10 | Fig. 4-11 | Fig. 4-12 | Fig. 4-13 | Fig. 4-14 |
| WS-FR-1 | X |  |  |  |  |  |  |  |  |  | X | X |  | X |  |  |  |  |  |  |  |  |  |  |
| WS-FR-1 | X |  |  |  |  |  |  |  |  |  | X | X |  | X |  |  |  |  |  |  |  |  |  |  |
| WS-FR-2 |  | X |  |  |  |  |  |  |  |  | X |  |  |  | X |  |  |  |  |  |  |  |  |  |
| WS-FR-3 |  | X |  |  |  |  |  |  |  |  | X |  |  |  |  | X |  |  |  |  |  |  |  |  |
| WS-FR-4 |  |  | X |  |  |  |  |  |  |  | X |  |  |  |  |  | X |  |  |  |  |  |  |  |
| WS-FR-5 |  |  |  | X |  |  |  |  |  |  | X |  |  |  |  |  |  | X |  |  |  |  |  |  |
| WS-FR-6 |  |  |  |  | X |  |  |  |  |  | X |  |  |  |  |  |  |  | X |  |  |  |  |  |
| WS-FR-7 |  |  |  |  |  | X |  |  |  |  | X |  |  |  |  |  |  |  |  | X |  |  |  |  |
| WS-FR-8 |  |  |  |  |  |  | X |  |  |  | X |  |  |  |  |  |  |  |  |  | X |  |  |  |
| WS-FR-9 |  |  |  |  |  |  |  | X |  |  | X |  | X |  |  |  |  |  |  |  |  | X |  |  |
| WS-FR-10 |  |  |  |  |  |  |  |  | X |  | X |  | X |  |  |  |  |  |  |  |  |  | X |  |
| WS-FR-11 |  |  |  |  |  |  |  |  |  | X | X |  | X |  |  |  |  |  |  |  |  |  |  | X |

Table 5-10. RESTful Web Services Traceability Matrix.

###### 5.5.1.5.2 RESTful Web Services Traceability Details

* The RESTful Services Class Diagram (Figure 3-2) encompasses all Web Service requirements.
* The REST API List Application Links Class Diagram (Figure 3-8) encompasses all Data Linkage requirements (Web Services Functional Requirements 9-11).
* WS-FR-1 is achieved by the List Applications Wireframe (Figure 2-3), the List Applications Class Diagram (Figure 3-7) and the List Applications Sequence Diagram (Figure 4-4).
* WS-FR-2 is achieved by the Add/Edit Applications Wireframe (Figure 2-4) and the Add Application Sequence Diagram (Figure 4-5).
* WS-FR-3 is achieved by the Add/Edit Applications Wireframe (Figure 2-4) and the Edit Application Sequence Diagram (Figure 4-6).
* WS-FR-4 is achieved by the Delete Applications Wireframe (Figure 2-5) and the Delete Application Sequence Diagram (Figure 4-7).
* WS-FR-5 is achieved by the List Data Types Wireframe (Figure 2-6) and the List Data Types Sequence Diagram (Figure 4-8).
* WS-FR-6 is achieved by the Add Data Type Wireframe (Figure 2-7) and the Add Data Type Sequence Diagram (Figure 4-9).
* WS-FR-7 is achieved by the Edit Data Type Wireframe (Figure 2-8) and the Edit Data Type Sequence Diagram (Figure 4-10).
* WS-FR-8 is achieved by the Delete Data Type Wireframe (Figure 2-9) and the Delete Data Type Sequence Diagram (Figure 4-11).
* WS-FR-9 is achieved by the List Data Linkage Wireframe (Figure 2-10) and the List Data Type Sequence Diagram (Figure 4-12).
* WS-FR-10 is achieved by the Add Data Linkage Wireframe (Figure 2-11) and the Add Data Linkage Sequence Diagram (Figure 4-13).
* WS-FR-11 is achieved by the Delete Data Linkage Wireframe (Figure 2-12) and the Delete Data Linkage Sequence Diagram (Figure 4-14).

##### 5.5.1.6 Persistence Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req. ID | Type | Priority | Description | Owner |
| PL-FR-1 | Functional | Essential | Database Create | RF, AA |
| PL-FR-2 | Functional | Essential | Database Retrieve | RF, AA |
| PL-FR-3 | Functional | Essential | Database Update | RF, AA |
| PL-FR-4 | Functional | Essential | Database Delete | RF, AA |

Table 5-11. Persistence Requirements.

###### 5.5.1.6.1 Persistence Traceability

PL-FR-1, PL-FR-2, PL-FR-3, and PL-FR-4 are all achieved through the User/Group ER Diagram (Figure 3-9), the Application Data ER Diagram (Figure 3-10), and the GUI CRUD Applications Sequence Diagram (Figure 4-1).

#### 5.5.2 Non-Functional Requirements

##### 5.5.2.1 Usability

The single page format of the application and table-style display for data is achieved through Login Page Wireframe (Figure 3-2), Home Page Wireframe (Figure 3-3), System Front-End Class Diagram (Figure 5-13), both GUI Class Diagrams (Figures 5-15 and 5-16), and all wireframes associated with the Applications Page (Figure 5-3 through Figure 5-12).

##### 5.5.2.2 Performance

The site responsiveness within a reasonable amount of time is achieved through RESTful services as shown in the GUI Service Class Diagram (Figure 5-16), the REST API Applications Class Diagram (Figure 5-19), and the REST API Application Links Class Diagram (Figure 5-20). These elements combine to produce a single page, responsive application.

##### 5.5.2.3 Portability

The functionality in modern web browsers is achieved through the Angular framework depicted in the System Front-End Class Diagram (Figure 5-13). The RESTful services described in section 6.2.2 are server-side components that function independently from the browser. Further, the Spring Boot Framework is built in Java, which is a cross-platform computing environment that should allow it to function in any server environment that has Java installed.

##### 5.5.2.4 Security

The user credential encryption and storage in an authentication database is achieved through the REST API User Authentication Class Diagram (Figure 5-18). Additional security is achieved through the single-page application design, which is accomplished by the Angular Framework (Figure 4-2).

# 6. Test Plan

### 6.1 Requirements/Specifications-Based System Level Test Cases

This section contains various test cases divided into subsections based on system functionalities, namely the GUI, Applications, Data Types, and Data Linkages subsections. Each subsection contains feature sets, test cases, and test procedures associated with the functionality.

#### 6.1.1 GUI Login and Layout Test Cases

##### 6.1.1.1 GUI Login and Layout Feature Set (FS1)

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Objective | Pri. | Contents |
| H.1 | User login | H | TC-2.1.1 through TC-2.1.3 |
| H.2 | User logout | H | TC-2.1. 4 through TC-2.1.7 |
| H.3 | Login page layout | Bm | TC-2.1.8 |
| H.4 | Logout page layout | Bm | TC-2.1.9 |
| H.5 | Main page layout | Bm | TC-2.1.10 |
| H.6 | Error page | Bm | TC-2.1.11 |
| H.7 | List page layout | Bm | TC-2.1.12 through TC-2.1.14 |
| H.8 | Add page layout | Bm | TC-2.1.15 through TC-2.1.17 |
| H.9 | Delete object layout | Bm | TC-2.1.18 |

Table 6-1. GUI Login and Layout Feature Set (FS1).

##### 6.1.1.2 GUI Login and Layout Test Cases

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.1.1** | **Purpose:** to test the reaction to an invalid combination of username and password values when attempting to login to the application. |
| **Priority: H** |
| Tracing: (FS1).H.1 |
| **Preconditions:** | A sample username with invalid value invalidUser  A sample password with invalid value xyz  A sample username with valid value jonny.jones@xxxx.com  A sample password with valid value password  A combination of valid username and valid password must not be included. |
| **Input:** | Insert the sample combination and start the analysis. |
| **Expected Result:** | The display shows “The login credentials you have entered are invalid” and the input fields are cleared. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.1.2** | **Purpose:** to test the reaction to a blank username, or password, or both when attempting to login into application. |
| **Priority: H** |
| Tracing: (FS1).H.1 |
| **Preconditions:** | A sample username with valid value jonny.jones@xxxx.com  A sample password with valid value password  A combination of valid username and valid password must not be included. |
| **Input:** | Insert only username, only password, or none. |
| **Expected Result:** | The display shows “The login credentials you have entered are invalid” and input fields are cleared. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.1.3** | **Purpose:** to test the reaction to a valid combination of username and password values when attempting to login into application. |
| **Priority: H** |
| Tracing: (FS1).H.1 |
| **Preconditions:** | A sample username with valid value jonny.jones@xxxx.com  A sample password with valid value password |
| **Input:** | Enter username and password. |
| **Expected Result:** | User is granted access to the main page of the system. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.1.4** | **Purpose:** to test the reaction of a user clicking the logout link on the main page. |
| **Priority: H** |
| Tracing: (FS1).H.2 |
| **Preconditions:** | User is logged in.  User is in the main page of the system.  User clicks the logout link from the header. |
| **Input:** | n/a |
| **Expected Result:** | User is directed to the logged-out confirmation page. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.1.5** | **Purpose:** to test the reaction of a user clicking the logout link on the ‘Applications’ > ‘Add’ page. |
| **Priority: H** |
| Tracing: (FS1).H.2 |
| **Preconditions:** | User is logged in.  User is in the ‘Applications’ > ‘Add’ page of the system.  User clicks the logout link from the header. |
| **Input:** | n/a |
| **Expected Result:** | User is directed to the logged-out confirmation page. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.1.6** | **Purpose:** to test the reaction of a user clicking the logout link on the ‘Data Types’ > ‘Add’ page. |
| **Priority: H** |
| Tracing: (FS1).H.2 |
| **Preconditions:** | User is logged in.  User is in the ‘Data Types’ > ‘Add’ page of the system.  User clicks the logout link from the header. |
| **Input:** | n/a |
| **Expected Result:** | User is directed to the logged-out confirmation page. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.1.7** | **Purpose:** to test the reaction of a user clicking the logout link on the ‘Data Linkages’ > ‘Add’ page. |
| **Priority: H** |
| Tracing: (FS1).H.2 |
| **Preconditions:** | User is logged in.  User is in the ‘Data Linkages’ > ‘Add’ page of the system.  User clicks the logout link from the header. |
| **Input:** | n/a |
| **Expected Result:** | User is directed to the logged-out confirmation page. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.1.8** | **Purpose:** to test that the login page HTML DOM elements are properly placed, named, and styled. |
| **Priority: Bm** |
| Tracing: (FS1).H.3 |
| **Preconditions:** | None. |
| **Input:** | n/a |
| **Expected Result:** | Correct login page layout. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.1.9** | **Purpose:** to test that the logout page HTML DOM elements are properly placed, named, and styled. |
| **Priority: Bm** |
| Tracing: (FS1).H.4 |
| **Preconditions:** | User has valid credentials and access to the application. |
| **Input:** | n/a |
| **Expected Result:** | Correct logout page layout. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.1.10** | **Purpose:** to test that the main page HTML DOM elements are properly placed, named, and styled. |
| **Priority: Bm** |
| Tracing: (FS1).H.5 |
| **Preconditions:** | User has valid credentials and access to the application. |
| **Input:** | n/a |
| **Expected Result:** | Correct main page layout. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.1.11** | **Purpose:** to test that an error page id displayed when necessary. |
| **Priority: Bm** |
| Tracing: (FS1).H.6 |
| **Preconditions:** | Incorrect URI is entered in a browser. |
| **Input:** | n/a |
| **Expected Result:** | Error page is displayed. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.1.12** | **Purpose:** to test that the ‘- Applications’ > ‘List’ page HTML DOM elements are properly placed, named, and styled. |
| **Priority: Bm** |
| Tracing: (FS1).H.7 |
| **Preconditions:** | User has valid credentials and access to the application. |
| **Input:** | n/a |
| **Expected Result:** | Correct ‘- Applications’ > 'List’ page layouts |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.1.13** | **Purpose:** to test that the ‘- Data Types’ > ‘List’ page HTML DOM elements are properly placed, named, and styled. |
| **Priority: Bm** |
| Tracing: (FS1).H.7 |
| **Preconditions:** | User has valid credentials and access to the application. |
| **Input:** | n/a |
| **Expected Result:** | Correct ‘- Data Types’ > 'List’ page layout. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.1.14** | **Purpose:** to test that the ‘- Data Linkages’ > ‘List’ page HTML DOM elements are properly placed, named, and styled. |
| **Priority: Bm** |
| Tracing: (FS1).H.7 |
| **Preconditions:** | User has valid credentials and access to the application. |
| **Input:** | n/a |
| **Expected Result:** | Correct ‘- Data Linkages’ > 'List’ page layout. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.1.15** | **Purpose:** to test that the ‘- Applications’ > ‘Add’ page HTML DOM elements are properly placed, named, and styled. |
| **Priority: Bm** |
| Tracing: (FS1).H.8 |
| **Preconditions:** | User has valid credentials and access to the application. |
| **Input:** | n/a |
| **Expected Result:** | Correct ‘- Applications’ > ‘Add’ page layout |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.1.16** | **Purpose:** to test that the ‘- Data Types’ > ‘Add’ page HTML DOM elements are properly placed, named, and styled. |
| **Priority: Bm** |
| Tracing: (FS1).H.8 |
| **Preconditions:** | User has valid credentials and access to the application. |
| **Input:** | n/a |
| **Expected Result:** | Correct ‘- Data Types’ > ‘Add’ page layout. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.1.17** | **Purpose:** to test that the ‘- Data Linkages’ > ‘Add’ page HTML DOM elements are properly placed, named, and styled. |
| **Priority: Bm** |
| Tracing: (FS1).H.8 |
| **Preconditions:** | User has valid credentials and access to the application. |
| **Input:** | n/a |
| **Expected Result:** | Correct ‘- Data Linkages’ > ‘Add’ page layout. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.1.18** | **Purpose:** to test that the Delete confirmation modal window HTML DOM elements are properly placed, named, and styled. |
| **Priority: Bm** |
| Tracing: (FS1).H.9 |
| **Preconditions:** | User has valid credentials and access to the application. |
| **Input:** | n/a |
| **Expected Result:** | Correct Delete confirmation modal window layout. |

##### 6.1.1.3 GUI Login and Layout Test Procedures

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TPID** | **Objective and Priority** | | | **Est. Dur.:** |
| TP-H-1 | The purpose of this test procedure is to test the way the system handles invalid, blank, and valid user credentials in the login page.  Priority: H | | | 3 minutes |
| **Start up:** Open a browser and navigate to the login page. Place the username and password in the labeled input fields with the following values:  1) username: invalidUser / jonny.jones@xxxx.com password: xyz / password  2) username: jonny.jones@xxxx.com password: password  3) username: jonny.jones@xxxx.com password: password | | | | |
| **Relationships to other procedures:** None | | | | |
| **Test Log** | | | | |
| Date: | Initials: | Test item: | | OK/Not OK |
|  |  |  | |  |
| **Comments:** | | | | |
| **Procedure** | | | | |
| **Step No.**  **Test Case** | **Activities** | **Examination of Result** | **Actual Results** | **Test Result** |
| 1 | Enter a valid username and invalid password and click login. |  |  |  |
| TC-2.1.1 | Enter an invalid username and valid password and click login.  Enter an invalid username and invalid password and click login. | Check that the display shows “The login credentials you have entered are invalid” and input fields are cleared. |
| 2 | Enter a valid username and blank password and click login. |  |  |  |
| TC-2.1.2 | Enter a blank username and valid password and click login.  Leave both username and password blank and click login. | Check that the display shows “The login credentials you have entered are invalid” and input fields are cleared. |
| 3 | Enter the provided credentials. Wait for the system’s reaction. |  |  |  |
| TC-2.1.3 |  | Check that you are been granted access to the application. |
| **Stop and Wrap Up:** Logout from the application. Clean cache. Exit browser. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TPID** | **Objective and Priority** | | | **Est. Dur.:** |
| TP-H-2 | The purpose of this test procedure is to test the way the system handles a logout request from the different pages in the system.  Priority: H | | | 3 minutes |
| **Start up:** Login to the system using username: jonny.jones@xxxx.com password: password | | | | |
| **Relationships to other procedures:** None | | | | |
| **Test Log** | | | | |
| Date: | Initials: | Test item: | | OK/Not OK |
|  |  |  | |  |
| **Comments:** | | | | |
| **Procedure** | | | | |
| **Step No.**  **Test Case** | **Activities** | **Examination of Result** | **Actual Results** | **Test Result** |
| 1 | Enter the provided credentials. Make sure the current page is the main page. |  |  |  |
| TC-2.2.4 |  | User is directed to the logout page and session is cleared. |
| 2 | Enter the provided credentials. Navigate to the ‘Add’ page from the link under ‘ - Applications’ in the navigation menu. |  |  |  |
| TC-2.2.5 |  | User is directed to the logout page and session is cleared. |
| 3 | Enter the provided credentials. Navigate to the ‘Add’ page from the link under ‘- Data Types’ in the navigation menu. |  |  |  |
| TC-2.2.6 |  | User is directed to the logout page and session is cleared. |
| 4 | Enter the provided credentials. Navigate to the ‘Add’ page from the link under ‘- Data Linkage’ in the navigation menu. |  |  |  |
| TC-2.2.7 |  | User is directed to the logout page and session is cleared. |
| **Stop and Wrap Up:** Clean cache. Exit browser. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TPID** | **Objective and Priority** | | | **Est. Dur.:** |
| TP-H-3 | The purpose of this test procedure is to test the layout of the Login page  Priority: Bm | | | 2 minutes |
| **Start up:** Open a browser and navigate to the login page. | | | | |
| **Relationships to other procedures:** None | | | | |
| **Test Log** | | | | |
| Date: | Initials: | Test item: | | OK/Not OK |
|  |  |  | |  |
| **Comments:** | | | | |
| **Procedure** | | | | |
| **Step No.**  **Test Case** | **Activities** | **Examination of Result** | **Actual Results** | **Test Result** |
| 1 | Click on the login button. |  |  |  |
| TC-2.1.8 |  | Blue header at the top with ‘Login Page’ message on the left. White background on the main section. Left-aligned Perspecta logo under the header. Centered Login form with username and password inputs that have placeholder text and labels. Under the form a green ‘Login’ button next to a blue ‘Cancel’ button. A ‘The login credentials you have entered are invalid’ message is displayed when login credentials are incorrect or blank and the Login button is clicked. |
| **Stop and Wrap Up:** Logout from the application. Clean cache. Exit browser. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TPID** | **Objective and Priority** | | | **Est. Dur.:** |
| TP-H-4 | The purpose of this test procedure is to test the layout of the Logout page  Priority: Bm | | | 2 minutes |
| **Start up:** Login to the system using username: jonny.jones@xxxx.com password: password. | | | | |
| **Relationships to other procedures:** None | | | | |
| **Test Log** | | | | |
| Date: | Initials: | Test item: | | OK/Not OK |
|  |  |  | |  |
| **Comments:** | | | | |
| **Procedure** | | | | |
| **Step No.**  **Test Case** | **Activities** | **Examination of Result** | **Actual Results** | **Test Result** |
| 1 | Enter the provided credentials. Click the Login button. Click the Logout link from the header. |  |  | Passed |
| TC-2.1.9 |  | White background with a centered ‘You have successfully logged out!’ message is displayed and under it a ‘Click here to return to the login page’ message. The word ‘here’ is a link that takes the user back to the login page. |
| **Stop and Wrap Up:** Logout from the application. Clean cache. Exit browser. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TPID** | **Objective and Priority** | | | **Est. Dur.:** |
| TP-H-5 | The purpose of this test procedure is to test the layout of the Main page  Priority: Bm | | | 2 minutes |
| **Start up:** Login to the system using username: jonny.jones@xxxx.com password: password. | | | | |
| **Relationships to other procedures:** None | | | | |
| **Test Log** | | | | |
| Date: | Initials: | Test item: | | OK / Not OK |
|  |  |  | |  |
| **Comments:** | | | | |
| **Procedure** | | | | |
| **Step No.**  **Test Case** | **Activities** | **Examination of Result** | **Actual Results** | **Test Result** |
| 1 | Enter the provided credentials. Click the Login button. |  |  |  |
| TC-2.1.10 |  | White background on the main section. A blue header with ‘ETA Home Page’ on the left, <Username> on the right, under username a ‘logout’ link. A blue footer with a centered Perspecta copyright. A navigation menu on the left side between header and footer containing the options: ‘+ Application’, ‘+ Data Types’, + ‘Data Linkages’. These menus should expand, the ‘+’ will now be a ‘-’ and each will contain submenus of: ‘Add’ and ‘List’. An image of Perspecta’s logo in the center of the screen. |
| **Stop and Wrap Up:** Logout from the application. Clean cache. Exit browser. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TPID** | **Objective and Priority** | | | **Est. Dur.:** |
| TP-H-6 | The purpose of this test procedure is to test the Error page and its layout when a URI is entered with an incorrect path.  Priority: Bm | | | 2 minutes |
| **Start up:** Navigate to http://localhost:4200/invalid. | | | | |
| **Relationships to other procedures:** None | | | | |
| **Test Log** | | | | |
| Date: | Initials: | Test item: | | OK/Not OK |
|  |  |  | |  |
| **Comments:** | | | | |
| **Procedure** | | | | |
| **Step No.**  **Test Case** | **Activities** | **Examination of Result** | **Actual Results** | **Test Result** |
| 1 | Open a browser and enter the URI provided |  |  |  |
| TC-2.1.11 |  | An Error page is opened with an error message displayed on the top left. |
| **Stop and Wrap Up:** Clean cache. Exit browser. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TPID** | **Objective and Priority** | | | **Est. Dur.:** |
| TP-H-7 | The purpose of this test procedure is to test the layout of the ‘List’ page  Priority: Bm | | | 2 minutes |
| **Start up:** Login to the system using username: jonny.jones@xxxx.com password: password | | | | |
| **Relationships to other procedures:** None | | | | |
| **Test Log** | | | | |
| Date: | Initials: | Test item: | | OK/Not OK |
|  |  |  | |  |
| **Comments:** | | | | |
| **Procedure** | | | | |
| **Step No.**  **Test Case** | **Activities** | **Examination of Result** | **Actual Results** | **Test Result** |
| 1 | Enter the provided credentials. Click the Login button. Navigate to ‘- Applications’ > ‘List’. |  |  |  |
| TC-2.2.12 |  | Header, footer, and navigation menu remain as described in TP-H-5. Perspecta logo is replaced by a table in the main section. The table contains a header and six columns with a green ‘Update’ button and a red ‘Delete’ button in the last two columns of each row. |
| 2 | Enter the provided credentials. Click the Login button. Navigate to ‘- Data Types’ > ‘List’. |  |  |  |
| TC-2.2.13 |  | Header, footer, and navigation menu remain as described in TP-H-5. Perspecta logo is replaced by a table in the main section. The table contains a header and four columns with a green ‘Update’ button and a red ‘Delete’ button in the last two columns of each row. |
| 3 | Enter the provided credentials. Click the Login button. Navigate to ‘- Data Linkage’ > ‘List’. |  |  |  |
| TC-2.2.14 |  | Header, footer, and navigation menu remain as described in TP-H-5. Perspecta logo is replaced by a table in the main section. The table contains a header and four columns with a red ‘Delete’ button in the last column of each row. |
| **Stop and Wrap Up:** Logout from the application. Clean cache. Exit browse. | | | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TPID** | | **Objective and Priority** | | | | **Est. Dur.:** | |
| TP-H-8 | | The purpose of this test procedure is to test the layout of the 'Add’ page  Priority: Bm | | | | 2 minutes | |
| **Start up:** Login to the system using username: jonny.jones@xxxx.com password: password | | | | | | | |
| **Relationships to other procedures:** None | | | | | | | |
| **Test Log** | | | | | | | |
| Date: | | Initials: | Test item: | | | OK/Not OK | |
|  | |  |  | | |  | |
| **Comments:** | | | | | | | |
| **Procedure** | | | | | | | |
| **Step No.**  **Test Case** | **Activities** | | | **Examination of Result** | **Actual Results** | | **Test Result** |
| 1 | Enter the provided credentials. Click the Login button. Navigate to ‘- Applications’ > ‘Add’. | | |  |  | |  |
| TC-2.2.15 |  | | | Header and footer remain the same as described in TP-H-5. The title is changed to ‘ETA’. The navigation menu is removed and the Perspecta logo is replaced by a form in the main section. The form contains two text fields. The first one is labeled ‘Application Name’ and the second one is labeled ‘Description’. Below the text fields there is a blue button that expands a ‘Data Types’ dropdown menu containing checkboxes. Below the dropdown menu there are two buttons aligned horizontally, on the left there is a green ‘Save’ button, and on the right, there is a blue ‘Cancel’ button. On top of the form there is a legend with text value of ‘Application’ with a horizontal line below. |
| 2 | Enter the provided credentials. Click the Login button. Navigate to ‘- Data Types’ > ‘Add’. | | |  |  | |  |
| TC-2.2.16 |  | | | Header and footer remain the same as described in TP-H-5. The title is changed to ‘ETA’. The navigation menu is removed and the Perspecta logo is replaced by a form in the main section. The form contains a text field label ‘Data Type’. Below the text field there are two buttons aligned horizontally, on the left there is a green ‘Save’ button, and on the right, there is a blue ‘Cancel’ button. On top of the form there is a legend with text value of ‘Data Type’ with a horizontal line below. |
| 3 | Enter the provided credentials. Click the Login button. Navigate to ‘- Data Linkage’ > ‘Add’. | | |  |  | |  |
| TC-2.2.17 |  | | | Header and footer remain the same as described in TP-H-5. The title is changed to ‘ETA’. The navigation menu is removed and the Perspecta logo is replaced by a form in the main section. The form contains three dropdown menus aligned vertically. The first dropdown is labeled ‘Source Application’ the second is labeled ‘Data Type’ and the third is labeled ‘Destination Application’. Under the dropdowns there are two buttons aligned horizontally, on the left there is a green ‘Save’ button, and on the right, there is a blue ‘Cancel’ button. On top of the form there is a legend with text value of ‘Data Linkage’ with a horizontal line below. |  | |  |
| **Stop and Wrap Up:** Logout from the application. Clean cache. Exit browse. | | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TPID** | **Objective and Priority** | | | **Est. Dur.:** | | |
| TP-H-9 | The purpose of this test procedure is to test the Error page and its layout when a URI is entered with an incorrect path.  Priority: Bm | | | 2 minutes | | |
| **Start up:** Login to the system using username: jonny.jones@xxxx.com password: password | | | | | | |
| **Relationships to other procedures:** None | | | | | | |
| **Test Log** | | | | | | |
| Date: | Initials: | Test item: | | OK/Not OK | |
|  |  |  | |  | |
| **Comments:** | | | | | | |
| **Procedure** | | | | | | |
| **Step No.**  **Test Case** | **Activities** | **Examination of Result** | **Actual Results** | **Test Result** | | |
| 1 | Enter the provided credentials. Click the Login button. Navigate to a list and click a ‘Delete’ button. |  |  |  | | |
| TC-2.1.18 |  | A small modal window is opened with white background, a ‘Confirm delete?’ message and two buttons under the message. Left button is red with white ‘Yes’ message, and the right button is green with white ‘No’ message. The rest of the page is greyed out. |  |  | | |
| **Stop and Wrap Up:** Logout from the application. Clean cache. Exit browser. | | | | |

#### 6.1.2 Applications Test Cases

##### 6.1.2.1 Applications Feature Set (FS2)

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Objective | Pri. | Contents |
| A.1 | Retrieve Applications | H | TC-2.2.1 through TC-2.2.4 |
| A.2 | Add Applications | H | TC-2.2.5 through TC-2.2.13 |
| A.3 | Update Applications | H | TC-2.2.14 through TC-2.2.22 |
| A.4 | Delete Applications | H | TC-2.2.23 through TC-2.2.29 |

Table 6-2. Applications Feature Set (FS2).

##### 6.1.2.2 Applications Test Cases

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.1** | **Purpose:** to test the ability of the system to detect an admin user and allow the option to retrieve application records. |
| **Priority: H** |
| Tracing: (FS2).A.1 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The ‘Applications’ submenu in the navigation area contains an option to ‘List’ applications. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.2** | **Purpose:** to test the ability of an admin user to retrieve application records and view them in the main content area of the ETA homepage. |
| **Priority: H** |
| Tracing: (FS2).A.1 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The main content area of the Application List displays the application records in the database within a formatted table. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.3** | **Purpose:** to test the ability of the system to detect a standard user and allow the option to retrieve application records. |
| **Priority: H** |
| Tracing: (FS2).A.1 |
| **Preconditions:** | The user must be logged in.  The user must have standard level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The ‘Applications’ submenu in the navigation area contains an option to ‘List’ applications. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.4** | **Purpose:** to test the ability of a standard user to retrieve application records and view them in the main content area of the ETA homepage. |
| **Priority: H** |
| Tracing: (FS2).A.1 |
| **Preconditions:** | The user must be logged in.  The user must have standard level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The main content area of the Application List displays the application records in the database within a formatted table. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.5** | **Purpose:** to test the ability of the system to detect an admin user and allow the option to create a new application record. |
| **Priority: H** |
| Tracing: (FS2).A.2 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The ‘Applications’ submenu in the navigation area contains an option to ‘Add’ applications. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.6** | **Purpose:** to test the ability of an admin user to create a new application record with the following fields: Application Name, Description, and Data Types. |
| **Priority: H** |
| Tracing: (FS2).A.2 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | Application Name, Description, and Data Types. |
| **Expected Result:** | The new application is added, and the user is returned to the List page. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.7** | **Purpose:** to test the ability of the system to prevent the creation of an application with a non-unique application name. |
| **Priority: H** |
| Tracing: (FS2).A.2 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | An application name that exists in the database. |
| **Expected Result:** | The page to add an application displays a message that the application name exists in the database. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.8** | **Purpose:** to test the ability of the system to display warning messages when user attempts to create an application with empty fields. |
| **Priority: H** |
| Tracing: (FS2).A.2 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The main content area displays warning messages, “Application Name and Description must be filled out!” when the ‘Save’ button is clicked. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.9** | **Purpose:** to test the ability of the system to display a warning message if the user clicks the ‘Save’ button without an application name when creating an application. |
| **Priority: H** |
| Tracing: (FS2).A.2 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The main content area displays warning message: “Application Name and Description must be filled out!” |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.10** | **Purpose:** to test the ability of the system to display a warning message if the user clicks the ‘Save’ button without a description when creating an application. |
| **Priority: H** |
| Tracing: (FS2).A.2 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The main content area displays warning message: “Application Name and Description must be filled out!” |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.11** | **Purpose:** to test the ability of the system to create an application without selected data types. |
| **Priority: H** |
| Tracing: (FS2).A.2 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The new application is added, and the user is returned to the List page. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.12** | **Purpose:** to test the ability of the system to return to the List page if the admin user clicks the “Cancel” button prior to the application record being created. |
| **Priority: H** |
| Tracing: (FS2).A.2 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | Nothing is changed in the database and the user is returned to the List page. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.13** | **Purpose:** to test the ability of the system to detect a standard user and disallow the ability to create a new application record. |
| **Priority: H** |
| Tracing: (FS2).A.2 |
| **Preconditions:** | The user must be logged in.  The user must have standard level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The option to add an application is not visible in the “Applications” submenu of the navigation menu. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.14** | **Purpose:** to test the ability of the system to detect an admin user and allow the ability to update an application record. |
| **Priority: H** |
| Tracing: (FS2).A.3 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | A button to update an application is visible in the table for each listed application in the main content area. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.15** | **Purpose:** to test the ability of an admin user to update an application record. |
| **Priority: H** |
| Tracing: (FS2).A.3 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | An application name that does not exist in the database. |
| **Expected Result:** | The application is updated, and the user is returned to the List page. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.16** | **Purpose:** to test the ability of the system to prevent the update to a non-unique application record. |
| **Priority: H** |
| Tracing: (FS2).A.3 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | An application name that exists in the database. |
| **Expected Result:** | The page to update the application displays a message that the application name exists in the database. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.17** | **Purpose:** to test the ability of the system to display warning messages when user attempts to update an application with empty fields. |
| **Priority: H** |
| Tracing: (FS2).A.3 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The main content area displays warning messages, “Application Name and Description must be filled out!” when the ‘Save’ button is clicked. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.18** | **Purpose:** to test the ability of the system to display a warning message if the user clicks the ‘Save’ button without an application name when updating an application. |
| **Priority: H** |
| Tracing: (FS2).A.3 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The main content area displays warning message: “Application Name and Description must be filled out!” |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.19** | **Purpose:** to test the ability of the system to display a warning message if the user clicks the ‘Save’ button without a description when updating an application |
| **Priority: H** |
| Tracing: (FS2).A.3 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The main content area displays warning message: “Application Name and Description must be filled out!” |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.20** | **Purpose:** to test the ability of the system to successfully update an application without selected data types. |
| **Priority: H** |
| Tracing: (FS2).A.3 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The application is updated, and the user is returned to the List page. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.21** | **Purpose:** to test the ability of the system to return to the List page if the admin user clicks the “Cancel” button prior to the application record being updated. |
| **Priority: H** |
| Tracing: (FS2).A.3 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | Nothing is changed in the database and the user is returned to the List page. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.22** | **Purpose:** to test the ability of the system to detect a standard user and disallow the ability to update an application record. |
| **Priority: H** |
| Tracing: (FS2).A.3 |
| **Preconditions:** | The user must be logged in.  The user must have standard level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The option to update an application is not visible in the table for any listed application in the main content area. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.23** | **Purpose:** to test the ability of the system to detect an admin user and allow the ability to delete an application record. |
| **Priority: H** |
| Tracing: (FS2).A.4 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | A button to delete an application is visible in the table for each listed application in the main content area. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.24** | **Purpose:** to test the ability of an admin user to delete an application record. |
| **Priority: H** |
| Tracing: (FS2).A.4 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The application is deleted, and the user is returned to the List. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.25** | **Purpose:** to test the ability of the system to confirm that the admin user wants to delete an application record. |
| **Priority: H** |
| Tracing: (FS2).A.4 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | A confirmation message is shown to the user for them to confirm they want to delete the application. The prompt should include ‘Yes’ and ‘No’ options. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.26** | **Purpose:** to test the ability of the system to respond to an affirmative delete confirmation response. |
| **Priority: H** |
| Tracing: (FS2).A.4 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The application is deleted, and the user is returned to the List page. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.27** | **Purpose:** to test the ability of the system to respond to a negative delete confirmation response. |
| **Priority: H** |
| Tracing: (FS2).A.4 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The application is not deleted, and the user is returned to the List page. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.28** | **Purpose:** to test the ability of the system to delete any data linkages related to the deleted source or destination application. |
| **Priority: H** |
| Tracing: (FS2).A.4 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | Data linkages related to the deleted application are also deleted. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.2.29** | **Purpose:** to test the ability of the system to detect a standard user and disallow the ability to delete an application record. |
| **Priority: H** |
| Tracing: (FS2).A.4 |
| **Preconditions:** | The user must be logged in.  The user must have standard level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The option to delete an application is not visible in the table for any listed application in the main content area. |

##### 6.1.2.3 Applications Test Procedures

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TPID** | | **Objective and Priority** | | | | **Est. Dur.:** | |
| TP-A-1 | | The purpose of this test procedure is to test the ability of an admin user to adequately create and retrieve applications.  Priority: H | | | | 10 mins | |
| **Start up:** Login to the ETA site using the admin level permissions listed below.  Username: jonny.jones@xxxx.com, Password: password | | | | | | | |
| **Relationships to other procedures:** None | | | | | | | |
| **Test Log** | | | | | | | |
| Date: | | Initials: | Test item: | | | OK/Not OK | |
|  | |  |  | | |  | |
| **Comments:** | | | | | | | |
| **Procedure** | | | | | | | |
| **Step No.**  **Test Case** | **Activities** | | | **Examination of Result** | **Actual Results** | | **Test Result** |
| 1 | Click and Inspect Application submenu. | | |  |  | |  |
| TC-2.2.1 |  | | | Check that the Application submenu contains a List option. |  | |  |
| 2 | Click List from the Application submenu. | | |  |  | |  |
| TC-2.2.2 |  | | | Check that the main content area is filled with a table containing the applications from the database. |  | |  |
| 3 | Inspect the Application submenu. | | |  |  | |  |
| TC-2.2.5 |  | | | Check that the Application submenu contains an Add option. |  | |  |
| 4 | Click Add from the Application submenu. | | |  |  | |  |
| TC-2.2.6 |  | | | Check that a new page is displayed to add an application. |  | |  |
| 5 | Enter Application Name X in the Application Name field, Description in the Description field, select data type(s) from the multi-select dropdown, and click the Save button | | |  |  | |  |
| TC-2.2.6 |  | | | Check for return to the Application List with an addition confirmation message. |  | |  |
| 6 | Click add from the Application  submenu. | | |  |  | |  |
| TC-2.2.6 |  | | | Check that a new page is displayed to add an Application. |  | |  |
| 7  TC-2.2.8 | Click the Save button with completely empty fields. | | | Check that a warning message is shown for empty fields. |  | |  |
| 8 | Enter Application Name X in the Application Name field, Description in the Description field, select data type(s) from the multi-select dropdown, and click the Save button | | |  |  | |  |
| TC-2.2.7 |  | | | Check that a failure message is shown for a non-unique value. |  | |  |
| 9  TC-2.2.9 | Skip Application Name field, enter Description in the Description field, select data type(s) from the multi-select dropdown, and click the Save button. | | | Check that a warning message is shown for a missing field. |  | |  |
| 10  TC-2.2.10 | Enter Application Name Y in the Application Name field, skip description field, select data type(s) from the multi-select dropdown, and click the Save button. | | | Check that a warning message is shown for a missing field. |  | |  |
| 11  TC-2.2.11 | Enter Application Name Y in the Application Name field and Description in the Description field, skip data type selection, and click the Save button. | | | Check for return to the Application List with an addition confirmation message. |  | |  |
| 12  TC-2.2.6 | Click add from the Application  submenu. | | | Check that a new page is displayed to add an application. |  | |  |
| 13 | Click the Cancel button. | | |  |  | |  |
| TC-2.2.12 |  | | | Check for return to the Application List without an addition. |  | |  |
| **Stop and Wrap Up:** Logout from the application. Clean cache. Exit browser. | | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TPID** | | **Objective and Priority** | | | **Est. Dur.:** | |
| TP-A-2 | | The purpose of this test procedure is to test the ability of an admin user to adequately update applications.  Priority: H | | | 15 min | |
| **Start up:** Login to the ETA site using the admin level permissions listed below.  Username: jonny.jones@xxxx.com, Password: password  Application: ‘Application name X’, ‘Application name Y’ | | | | | | |
| **Relationships to other procedures:** Assumes successful completion of TP-A-1. | | | | | | |
| **Test Log** | | | | | | |
| Date: | | Initials: | Test item: | | OK/Not OK | |
|  | |  |  | |  | |
| **Comments:** | | | | | | |
| **Procedure** | | | | | | |
| **Step No.**  **Test Case** | **Activities** | | **Examination of Result** | **Actual Results** | | **Test Result** |
| 1 | Click List from the Applications submenu. | |  |  | |  |
| TC-2.2.14 |  | | Check that the table in the main content area contains applications with an Update button for each. |  | |  |
| 2 | Click Update for Application X. | |  |  | |  |
| TC-2.2.15 |  | | Check that a new page is displayed to update the application. |  | |  |
| 3 | Enter Application Name Z in the Application Name field and click the Save button | |  |  | |  |
| TC-2.2.15 |  | | Check for return to the Application List with an update confirmation message. |  | |  |
| 4 | Click Update for Application Z. | |  |  | |  |
| TC-2.2.15 |  | | Check that a new page is displayed to update the Application. |  | |  |
| 5 | Enter Application Name Y in the Application field and click the Save button. | |  |  | |  |
| TC-2.2.16 |  | | Check that a failure message is shown for a non-unique value. |  | |  |
| 6 | Enter Application Name Z in the Application Name field, enter Description in the Description field, and click the Update button. | |  |  | |  |
| TC-2.2.15 |  | | Check for return to the Application List with an update confirmation message. |  | |  |
| 7  TC-2.2.15 | Click Update for Application Z. | | Check that a new page is displayed to update the Application. |  | |  |
| 8  TC-2.2.20 | Unselect all checked data types if any and click the Update button. | | Check for return to the Application List with an update confirmation message. |  | |  |
| 9  TC-2.2.15 | Click Update for Application Z. | | Check that a new page is displayed to update the Application. |  | |  |
| 10  TC-2.2.18 | Remove Application Name in the Application Name field and click the Update button. | | Check that a warning message is shown for a missing field. |  | |  |
| 11  TC-2.2.19 | Remove Description in the Description field and click the Update button. | | Check that a warning message is shown for a missing field. |  | |  |
| 12  TC-2.2.17 | Click the Save button with completely empty fields. | | Check that a warning message is shown for empty fields. |  | |  |
| 13 | Click the Cancel button. | |  |  | |  |
| TC-2.2.21 |  | | Check for return to the ETA homepage without an update. |  | |  |
| **Stop and Wrap Up:** Logout from the application. Clean cache. Exit browser. | | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TPID** | **Objective and Priority** | | | **Est. Dur.:** |
| TP-A-3 | The purpose of this test procedure is to test the ability of an admin user to adequately delete applications.  Priority: H | | | 5 min |
| **Start up:** Login to the ETA site using the admin level permissions listed below.  Username: jonny.jones@xxxx.com, Password: password  Data Types: ‘Data Type X’, ‘Data Type Y’  Applications: ‘Application A’ with ‘Data Type X’ and ‘Data Type Y’, ‘Application B’  Data Linkage: ‘Data Type Y’ is linked from ‘Application A’ to ‘Application B’ | | | | |
| **Relationships to other procedures:** Assumes successful completion of TP-A-1 and TP-A-2. | | | | |
| **Test Log** | | | | |
| Date: | Initials: | Test item: | | OK/Not OK |
|  |  |  | |  |
| **Comments:** | | | | |
| **Procedure** | | | | |
| **Step No.**  **Test Case** | **Activities** | **Examination of Result** | **Actual Results** | **Test Result** |
| 1 | Click List from the Application submenu. |  |  |  |
| TC-2.2.23 |  | Check that the table in the main content area contains Application with a Delete button for each. |  |  |
| 2 | Click Delete for Application A. |  |  |  |
| TC-2.2.25 |  | Check for prompt to confirm delete. |  |  |
| 3 | Click No to deny the delete. |  |  |  |
| TC-2.2.27 |  | Check for return to the Application List without a delete. |  |  |
| 4 | Click Delete for Application A and click Yes to approve the delete. |  |  |  |
| TC-2.2.25  TC-2.2.26  TC-2.2.24 |  | Check for return to the Application List with a delete confirmation message. |  |  |
| 5 | Click List from the Data Linkage submenu. |  |  |  |
| TC-2.2.28 |  | Check that the table in the main content area does not contain the data linkages related to the deleted source or destination application. |  |  |
| **Stop and Wrap Up:** Remove test data linkages, applications, and data types. Logout from the application. Clean cache. Exit browser. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TPID** | **Objective and Priority** | | | **Est. Dur.:** |
| TP-A-4 | The purpose of this test procedure is to test all abilities related to applications for a standard user.  Priority: H | | | 5 min |
| **Start up:** Login to the ETA site using the standard level permissions listed below.  Username: sally.thomas@xxxx.com, Password: password | | | | |
| **Relationships to other procedures:** None | | | | |
| **Test Log** | | | | |
| Date: | Initials: | Test item: | | OK/Not OK |
|  |  |  | |  |
| **Comments:** | | | | |
| **Procedure** | | | | |
| **Step No.**  **Test Case** | **Activities** | **Examination of Result** | **Actual Results** | **Test Result** |
| 1 | Click and Inspect ‘Application’ submenu. |  |  |  |
| TC-2.2.3 |  | Check that the ‘Application‘ submenu contains a ‘List’ option. |  |  |
| 2 | Inspect the ‘Application’ submenu. |  |  |  |
| TC-2.2.13 |  | Check that the ‘Application’ submenu does not contain an ‘Add’ option. |  |  |
| 3 | Click ‘List’ from the ‘Application’ submenu. |  |  |  |
| TC-2.2.4 |  | Check that the main content area is filled with a table containing the Application from the database. |  |  |
| 4 | Inspect Application List for Update buttons. |  |  |  |
| TC-2.2.22 |  | Check that the records do not contain ‘Update’ buttons. |  |  |
| 5 | Inspect Application List for Delete buttons. |  |  |  |
| TC-2.2.29 |  | Check that the records do not contain ‘Delete’ buttons. |  |  |
| **Stop and Wrap Up:** Logout from the application. Clean cache. Exit browser. | | | | |

#### 6.1.3 Data Types Test Cases

##### 6.1.3.1 Data Types Feature Set (FS3)

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Objective | Pri. | Contents |
| D.1 | Retrieve Data Types | H | TC-2.3.1 through TC-2.3.4 |
| D.2 | Add Data Type | H | TC-2.3.5 through TC-2.3.9 |
| D.3 | Update Data Type | H | TC-2.3.10 through TC-2.3.14 |
| D.4 | Delete Data Type | H | TC-2.3.15 through TC-2.3.22 |

Table 6-3. Data Types Feature Set (FS3).

##### 6.1.3.2 Data Types Test Cases

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.3.1** | **Purpose:** to test the ability of the system to detect an admin user and allow the option to retrieve data type records. |
| **Priority: H** |
| Tracing: (FS3).D.1 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The ‘Data Types’ submenu in the navigation area contains an option to ‘List’ data types. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.3.2** | **Purpose:** to test the ability of an admin user to retrieve data type records and view them in the main content area of the ETA homepage. |
| **Priority: H** |
| Tracing: (FS3).D.1 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The main content area of the Data Types list displays the data type records in the database within a formatted table. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.3.3** | **Purpose:** to test the ability of the system to detect a standard user and allow the option to retrieve data type records. |
| **Priority: H** |
| Tracing: (FS3).D.1 |
| **Preconditions:** | The user must be logged in.  The user must have standard level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The ‘Data Types’ submenu in the navigation area contains an option to ‘List’ data types. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.3.4** | **Purpose:** to test the ability of a standard user to retrieve data type records and view them in the main content area of the ETA homepage. |
| **Priority: H** |
| Tracing: (FS3).D.1 |
| **Preconditions:** | The user must be logged in.  The user must have standard level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The main content area of the Data Types list page displays the data type records in the database within a formatted table. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.3.5** | **Purpose:** to test the ability of the system to detect an admin user and allow the option to create a new data type record. |
| **Priority: H** |
| Tracing: (FS3).D.2 |
| **Preconditions:** | The user must be logged in.  The user must have standard level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The ‘Data Types’ submenu in the navigation area contains an option to ‘Add’ a data type. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.3.6** | **Purpose:** to test the ability of an admin user to create a new data type record. |
| **Priority: H** |
| Tracing: (FS3).D.2 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | A data type name that does not exist in the database. |
| **Expected Result:** | The new data type is added and the user is returned to the List page. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.3.7** | **Purpose:** to test the ability of the system to prevent the creation of a non-unique data type record. |
| **Priority: H** |
| Tracing: (FS3).D.2 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | A data type name that exists in the database. |
| **Expected Result:** | The page to add a data type displays a message that the data type description exists in the database. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.3.8** | **Purpose:** to test the ability of the system to return to the List page if the admin user clicks the “Cancel” button prior to the data type record being created. |
| **Priority: H** |
| Tracing: (FS3).D.2 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | Nothing is changed in the database and the user is returned to the List page. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.3.9** | **Purpose:** to test the ability of the system to detect a standard user and disallow the ability to create a new data type record. |
| **Priority: H** |
| Tracing: (FS3).D.2 |
| **Preconditions:** | The user must be logged in.  The user must have standard level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The option to add a data type is not visible in the “Data Types” submenu of the navigation menu. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.3.10** | **Purpose:** to test the ability of the system to detect an admin user and allow the ability to update a data type record. |
| **Priority: H** |
| Tracing: (FS3).D.3 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | A button to update a data type is visible in the table for each listed data type in the main content area. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.3.11** | **Purpose:** to test the ability of an admin user to update a data type record. |
| **Priority: H** |
| Tracing: (FS3).D.3 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | A data type name that does not exist in the database. |
| **Expected Result:** | The data type is updated and the user is returned to the List page. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.3.12** | **Purpose:** to test the ability of the system to prevent the update to a non-unique data type record. |
| **Priority: H** |
| Tracing: (FS3).D.3 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | A data type description that exists in the database. |
| **Expected Result:** | The page to update the data type displays a message that the data type description exists in the database. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.3.13** | **Purpose:** to test the ability of the system to return to the List page if the admin user clicks the “Cancel” button prior to the data type record being updated. |
| **Priority: H** |
| Tracing: (FS3).D.3 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | Nothing is changed in the database and the user is returned to the List page. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.3.14** | **Purpose:** to test the ability of the system to detect a standard user and disallow the ability to update a data type record. |
| **Priority: H** |
| Tracing: (FS3).D.3 |
| **Preconditions:** | The user must be logged in.  The user must have standard level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The option to update a data type is not visible in the table for any listed data type in the main content area. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.3.15** | **Purpose:** to test the ability of the system to detect an admin user and allow the ability to delete a data type record. |
| **Priority: H** |
| Tracing: (FS3).D.4 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | A button to delete a data type is visible in the table for each listed data type in the main content area. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.3.16** | **Purpose:** to test the ability of an admin user to delete a data type record. |
| **Priority: H** |
| Tracing: (FS3).D.4 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The data type is deleted and the user is returned to the List page. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.3.17** | **Purpose:** to test the ability of the system to confirm that the admin user wants to delete a data type record. |
| **Priority: H** |
| Tracing: (FS3).D.4 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | A confirmation message is shown to the user for them to confirm they want to delete the data type. The prompt should include ‘Yes’ and ‘No’ options. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.3.18** | **Purpose:** to test the ability of the system to respond to an affirmative delete confirmation. |
| **Priority: H** |
| Tracing: (FS3).D.4 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The data type is deleted and the user is returned to the List page. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.3.19** | **Purpose:** to test the ability of the system to respond to a negative delete confirmation. |
| **Priority: H** |
| Tracing: (FS3).D.4 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The data type is not deleted and the user is returned to the List page. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.3.20** | **Purpose:** to test the ability of the system to prevent the deletion of a data type record that is connected to an application record. |
| **Priority: H** |
| Tracing: (FS3).D.4 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The main content area of the List page displays a message that the data type cannot be deleted because it is being used by an application or data linkage. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.3.21** | **Purpose:** to test the ability of the system to prevent the deletion of a data type record that is connected to a data linkage record. |
| **Priority: H** |
| Tracing: (FS3).D.4 |
| **Preconditions:** | The user must be logged in.  The user must have admin level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The main content area of the List page displays a message that the data type cannot be deleted because it is being used by an application or data linkage. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.3.22** | **Purpose:** to test the ability of the system to detect a standard user and disallow the ability to delete a new data type record. |
| **Priority: H** |
| Tracing: (FS3).D.4 |
| **Preconditions:** | The user must be logged in.  The user must have standard level permissions. |
| **Input:** | n/a |
| **Expected Result:** | The option to delete a data type is not visible in the table for any listed data type in the main content area. |

##### 6.1.3.3 Data Types Test Procedures

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TPID** | **Objective and Priority** | | | **Est. Dur.:** |
| TP-D-1 | The purpose of this test procedure is to test the ability of an admin user to adequately create and retrieve data types.  Priority: H | | | 5 min |
| **Start up:** Login to the ETA site using the admin level permissions listed below.  Username: jonny.jones@xxxx.com, Password: password | | | | |
| **Relationships to other procedures:** None | | | | |
| **Test Log** | | | | |
| Date: | Initials: | Test item: | | OK/Not OK |
|  |  |  | |  |
| **Comments:** | | | | |
| **Procedure** | | | | |
| **Step No.**  **Test Case** | **Activities** | **Examination of Result** | **Actual Results** | **Test Result** |
| 1 | Click and Inspect Data Types submenu. |  |  |  |
| TC-2.3.1 |  | Check that the Data Types submenu contains a List option. |
| 2 | Click List from the Data Types submenu. |  |  |  |
| TC-2.3.2 |  | Check that the main content area is filled with a table containing the data types from the database. |
| 3 | Inspect the Data Types submenu. |  |  |  |
| TC-2.3.5 |  | Check that the Data Types submenu contains an Add option. |
| 4 | Click Add from the Data Types submenu. |  |  |  |
| TC-2.3.6 |  | Check that a new page is displayed to add a data type. |
| 5 | Enter Data Type X in the Data Type field and click the Save button. |  |  |  |
| TC-2.3.6 |  | Check for return to the List page with an addition confirmation message. |
| 6 | Click Add from the Data Types submenu. |  |  |  |
| TC-2.3.6 |  | Check that a new page is displayed to add a data type. |
| 7 | Enter Data Type X in the Data Type field and click the Save button. |  |  |  |
| TC-2.3.7 |  | Check that a failure message is shown for a non-unique value. |
| 8 | Click the Cancel button. |  |  |  |
| TC-2.3.8 |  | Check for return to the List page without an addition. |
| **Stop and Wrap Up:** Logout from the application. Clean cache. Exit browser. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TPID** | **Objective and Priority** | | | **Est. Dur.:** |
| TP-D-2 | The purpose of this test procedure is to test the ability of an admin user to adequately update data types.  Priority: H | | | 10 min |
| **Start up:** Login to the ETA site using the admin level permissions listed below.  Username: jonny.jones@xxxx.com, Password: password  Data Types: ‘Data Type X’, ‘Data Type Y’ | | | | |
| **Relationships to other procedures:** Assumes successful completion of TP-D-1. | | | | |
| **Test Log** | | | | |
| Date: | Initials: | Test item: | | OK/Not OK |
|  |  |  | |  |
| **Comments:** | | | | |
| **Procedure** | | | | |
| **Step No.**  **Test Case** | **Activities** | **Examination of Result** | **Actual Results** | **Test Result** |
| 1 | Click List from the Data Types submenu. |  |  |  |
| TC-2.3.10 |  | Check that the table in the main content area contains data types with an Update button for each. |
| 2 | Click Update for Data Type X. |  |  |  |
| TC-2.3.11 |  | Check that a new page is displayed to update the data type. |
| 3 | Enter Data Type Z in the Data Type field and click the Save button. |  |  |  |
| TC-2.3.11 |  | Check for return to the List page with an update confirmation message. |
| 4 | Click List from Data Types submenu and click Update for Data Type Z. |  |  |  |
| TC-2.3.11 |  | Check that a new page is displayed to update the data type. |
| 5 | Enter Data Type Y in the Data Type field and click the Save button. |  |  |  |
| TC-2.3.12 |  | Check that a failure message is shown for a non-unique value. |
| 6 | Click List from Data Types submenu and click Update for Data Type Z. |  |  |  |
| TC-2.3.11 |  | Check that a new page is displayed to update the data type. |
| 7 | Click the Cancel button. |  |  |  |
| TC-2.3.13 |  | Check for return to the List page without an update. |
| **Stop and Wrap Up:** Logout from the application. Clean cache. Exit browser. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TPID** | **Objective and Priority** | | | **Est. Dur.:** |
| TP-D-3 | The purpose of this test procedure is to test the ability of an admin user to adequately delete data types.  Priority: H | | | 10 min |
| **Start up:** Login to the ETA site using the admin level permissions listed below.  Username: jonny.jones@xxxx.com, Password: password  Data Types: ‘Data Type X’, ‘Data Type Y’  Applications: ‘Application A’ with ‘Data Type X’ and ‘Data Type Y’, ‘Application B’  Data Linkage: ‘Data Type Y’ is linked from ‘Application A’ to ‘Application B’ | | | | |
| **Relationships to other procedures:** Assumes successful completion of TP-D-1 and TP-D-2. | | | | |
| **Test Log** | | | | |
| Date: | Initials: | Test item: | | OK/Not OK |
|  |  |  | |  |
| **Comments:** | | | | |
| **Procedure** | | | | |
| **Step No.**  **Test Case** | **Activities** | **Examination of Result** | **Actual Results** | **Test Result** |
| 1 | Click List from the Data Types submenu. |  |  |  |
| TC-2.3.15 |  | Check that the table in the main content area contains data types with a Delete button for each. |
| 2 | Click Delete for Data Type Z. |  |  |  |
| TC-2.3.17 |  | Check for prompt to confirm delete. |
| 3 | Click No to deny the delete. |  |  |  |
| TC-2.3.19 |  | Check for return to the List page without a delete. |
| 4 | Click List from Data Types submenu, click Delete for Data Type Z, and click Yes to approve the delete. |  |  |  |
| TC-2.3.16  TC-2.3.18 |  | Check for return to the List page with a delete confirmation message. |
| 5 | Click List from Data Types submenu and click Delete for Data Type X. |  |  |  |
| TC-2.3.20 |  | Check that a failure message is shown due to an application constraint. |
| 6 | Click List from Data Types submenu and click Delete for Data Type Y. |  |  |  |
| TC-2.3.21 |  | Check that a failure message is shown due to a data linkage constraint. |
| **Stop and Wrap Up:** Remove test data linkages, applications, and data types. Logout from the application. Clean cache. Exit browser. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TPID** | **Objective and Priority** | | | **Est. Dur.:** |
| TP-D-4 | The purpose of this test procedure is to test all abilities related to data types for a standard user.  Priority: H | | | 5 min |
| **Start up:** Login to the ETA site using the standard level permissions listed below.  Username: sally.thomas@xxxx.com, Password: password | | | | |
| **Relationships to other procedures:** None | | | | |
| **Test Log** | | | | |
| Date: | Initials: | Test item: | | OK/Not OK |
|  |  |  | |  |
| **Comments:** | | | | |
| **Procedure** | | | | |
| **Step No.**  **Test Case** | **Activities** | **Examination of Result** | **Actual Results** | **Test Result** |
| 1 | Click and Inspect ‘Data Types’ submenu. |  |  |  |
| TC-2.3.3 |  | Check that the ‘Data Types’ submenu contains a ‘List’ option. |
| 2 | Inspect the ‘Data Types’ submenu. |  |  |  |
| TC-2.3.9 |  | Check that the ‘Data Types’ submenu does not contain an ‘Add’ option. |
| 3 | Click ‘List’ from the ‘Data Types’ submenu. |  |  |  |
| TC-2.3.4 |  | Check that the main content area is filled with a table containing the data types from the database. |
| 4 | Inspect Data Types List for Update buttons. |  |  |  |
| TC-2.3.14 |  | Check that the records do not contain ‘Update’ buttons. |
| 5 | Inspect Data Types List for Delete buttons. |  |  |  |
| TC-2.3.22 |  | Check that the records do not contain ‘Delete’ buttons. |
| **Stop and Wrap Up:** Logout from the application. Clean cache. Exit browser. | | | | |

#### 6.1.4 Data Linkages Test Cases

##### 6.1.4.1 Data Linkages Feature Set (FS4)

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Objective | Pri. | Contents |
| L.1 | Retrieve Data Linkage | H | TC-2.4.1 through TC-2.4.2  TC-2.4.9 through TC-2.4.10  TC-2.4.16 through TC-2.4.17 |
| L.2 | Add Data Linkage | H | TC-2.4.3 through TC-2.4.8  TC-2.4.18 |
| L.3 | Delete Data Linkage | H | TC-2.4.11 through TC-2.4.15  TC-2.4.19 |

Table 6-4. Data Linkages Feature Set (FS4).

##### 6.1.4.2 Data Linkages Test Cases

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.4.1** | **Purpose:** to test if the system can detect an admin user and allow an admin user to view the list of data linkages. |
| **Priority: H** |
| Tracing: (FS4).L.1 |
| **Preconditions:** | A user is logged in and has admin-level permissions. |
| **Input:** | N/A |
| **Expected Result:** | The application displays a “List” option under the Data Linkages sub header. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.4.2** | **Purpose:** to test if the system can retrieve a list of data linkages for an admin user. |
| **Priority: H** |
| Tracing: (FS4).L.1 |
| **Preconditions:** | A user is logged in and has admin-level permissions. The user also has the “List” option available to them under the Data Linkages sub header. |
| **Input:** | A user selects the “List” option under the Data Linkages sub header. |
| **Expected Result:** | The application displays a list of data linkages in the main content area. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.4.3** | **Purpose:** to test if the system can detect an admin user and allow an admin user to create a data linkage entry. |
| **Priority: H** |
| Tracing: (FS4).L.1 |
| **Preconditions:** | A user is logged in and has admin-level permissions. |
| **Input:** | N/A |
| **Expected Result:** | The application displays an “Add” option under the Data Linkages sub header. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.4.4** | **Purpose:** to test if the system can create a data linkage entry for an admin user. |
| **Priority: H** |
| Tracing: (FS4).L.1 |
| **Preconditions:** | A user is logged in and has admin-level permissions. The user also has the “Add” option available to them under the Data Linkages sub header. |
| **Input:** | A user selects the “Add” option under the Data Linkages sub header. |
| **Expected Result:** | A new data linkage entry is generated, and the user is returned to the homepage where a list of data linkages is displayed. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.4.5** | **Purpose:** to test if the system can navigate to the home page if an admin user selects the “Cancel” option in the Create Data Linkage page |
| **Priority: H** |
| Tracing: (FS4).L.1 |
| **Preconditions:** | A user is logged in and has admin-level permissions. The user is also in the Create Data Linkage Page. |
| **Input:** | A user selects the “Cancel” option in the Create Data Linkage page. |
| **Expected Result:** | A new data linkage entry is not generated, and the user is returned to the home page where a list of data linkages is displayed. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.4.6** | **Purpose:** to test if the system can disallow the admin user from creating a data linkage entry that already exists. |
| **Priority: H** |
| Tracing: (FS4).L.1 |
| **Preconditions:** | A user is logged in and has admin-level permissions. The user is also in the Create Data Linkage Page and has entered values from an entry that already exists. |
| **Input:** | A user selects the “Add” option in the Create Data Linkage page. |
| **Expected Result:** | A new data linkage entry is not generated, and the user is notified with a message displayed in the current page. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.4.7** | **Purpose:** to test if the system can disallow the admin user from creating a data linkage entry that has the same Source Application and Destination Application values. |
| **Priority: H** |
| Tracing: (FS4).L.1 |
| **Preconditions:** | A user is logged in and has admin-level permissions. The user is also in the Create Data Linkage Page and has entered the same values for Source Application and Destination Application. |
| **Input:** | A user selects the “Add” option in the Create Data Linkage page. |
| **Expected Result:** | A new data linkage entry is not generated, and the user is notified with a message displayed in the current page. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.4.8** | **Purpose:** to test if the system can disallow the admin user from creating a data linkage entry that has one or more missing input in Source Application, Data Type, and Destination Application values. |
| **Priority: H** |
| Tracing: (FS4).L.1 |
| **Preconditions:** | A user is logged in and has admin-level permissions. The user is also in the Create Data Linkage Page and has left one or more values empty. |
| **Input:** | A user selects the “Add” option in the Create Data Linkage page. |
| **Expected Result:** | A new data linkage entry is not generated, and the user is notified with a message displayed in the current page. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.4.9** | **Purpose:** to test if the system can display the correct list of Applications for Source Applications and Destination Applications when creating a data linkage entry. |
| **Priority: H** |
| Tracing: (FS4).L.1 |
| **Preconditions:** | A user is logged in and has admin-level permissions. The user is also in the Create Data Linkage Page. |
| **Input:** | A user selects either the Source Application or Destination Application drop-down menu. |
| **Expected Result:** | The drop-down menu displays the correct list of Source Applications and Destination Applications |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.4.10** | **Purpose:** to test if the system can display the correct list of Data Types when creating a data linkage entry. |
| **Priority: H** |
| Tracing: (FS4).L.1 |
| **Preconditions:** | A user is logged in and has admin-level permissions. The user is also in the Create Data Linkage Page and has selected a source application from the Source Application drop-down menu. |
| **Input:** | A user selects the data type drop-down menu. |
| **Expected Result:** | The drop-down menu displays the correct list of Data Types that apply to the selected source application. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.4.11** | **Purpose:** to test if the system can detect an admin user and allow an admin user to delete a data linkage entry. |
| **Priority: H** |
| Tracing: (FS4).L.2 |
| **Preconditions:** | A user is logged in and has admin-level permissions. The user also has the list of data linkages in the main content page. |
| **Input:** | N/A |
| **Expected Result:** | The application displays a Delete button beside each of the data linkages in the list. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.4.12** | **Purpose:** to test if the system can delete a data linkage entry for an admin user. |
| **Priority: H** |  |
| Tracing: (FS4).L.2 |  |
| **Preconditions:** | A user is logged in and has admin-level permissions. |
| **Input:** | N/A |
| **Expected Result:** | The data linkage entry is removed, and the user is returned to the homepage where the list of data linkages is displayed. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.4.13** | **Purpose:** to test if the system can display a confirmation window when the user selects the Delete button beside an entry from the list of data linkages. |
| **Priority: H** |
| Tracing: (FS4).L.2 |
| **Preconditions:** | A user is logged in and has admin-level permissions. The user also has the list of data linkages in the main content page. |
| **Input:** | The user selects the Delete button beside an entry from the list of data linkages. |
| **Expected Result:** | The application displays confirmation window asking if the user wants to delete the data linkage entry. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.4.14** | **Purpose:** to test if the system can detect confirmation from the admin user if they select the “Yes” option from the Confirm Delete window. |
| **Priority: H** |
| Tracing: (FS4).L.2 |
| **Preconditions:** | A user is logged in and has admin-level permissions. The user is also currently prompted with a confirmation window for the deletion of a data linkage entry. |
| **Input:** | The user selects the “Yes” option from the confirmation window. |
| **Expected Result:** | The data linkage entry is removed, and the user is returned to the homepage where the list of data linkages is displayed. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.4.15** | **Purpose:** to test if the system can detect negation from the admin user if they select the “No” option from the Confirm Delete window. |
| **Priority: H** |  |
| Tracing: (FS4).L.2 |  |
| **Preconditions:** | A user is logged in and has admin-level permissions. The user is also currently prompted with a confirmation window for the deletion of a data linkage entry. |
| **Input:** | The user selects the “No” option from the confirmation window. |
| **Expected Result:** | The data linkage entry is not removed, and the user is returned to the homepage where the list of data linkages is displayed. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.4.16** | **Purpose:** to test if the system can detect a standard user and allow a standard user to view the list of data linkages. |
| **Priority: H** |
| Tracing: (FS4).L.3 |
| **Preconditions:** | A user is logged in and has standard-level permissions. |
| **Input:** | N/A |
| **Expected Result:** | The application displays a “List” option under the Data Linkages sub header. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.4.17** | **Purpose:** to test if the system can retrieve a list of data linkages for a standard user. |
| **Priority: H** |
| Tracing: (FS4).L.3 |
| **Preconditions:** | A user is logged in and has standard-level permissions. The user also has the “List” option available to them under the Data Linkages sub header. |
| **Input:** | A user selects the “List” option under the Data Linkages sub header. |
| **Expected Result:** | The application displays a list of data linkages in the main content area. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.4.18** | **Purpose:** to test if the system can detect a standard user and disallow a standard user to create a data linkage entry. |
| **Priority: H** |
| Tracing: (FS4).L.3 |
| **Preconditions:** | A user is logged in and has standard-level permissions. |
| **Input:** | N/A |
| **Expected Result:** | The application does not display an “Add” option under the Data Linkages sub header. |

|  |  |
| --- | --- |
| **Test Case ID:**  **TC-2.4.19** | **Purpose:** to test if the system can detect a standard user and disallow a standard user to delete a data linkage entry. |
| **Priority: H** |
| Tracing: (FS4).L.3 |
| **Preconditions:** | A user is logged in and has standard-level permissions. The user also has the list of data linkages in the main content page. |
| **Input:** | N/A |
| **Expected Result:** | The application does not display a Delete button beside each of the data linkages in the list. |

##### 6.1.4.3 Data Linkages Test Procedures

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TPID** | | **Objective and Priority** | | | | **Est. Dur.:** | |
| TP-L-1 | | The purpose of this test procedure is to test the ability of an admin user to adequately create and retrieve data linkages  Priority: H | | | | 10 min | |
| **Start up:** Login to ETA site using the account with admin-level permissions listed below:  Username: jonny.jones@xxxx.com, Password: password | | | | | | | |
| **Relationships to other procedures:** None | | | | | | | |
| **Test Log** | | | | | | | |
| Date: | | Initials: | Test item: | | | OK/Not OK | |
|  | |  |  | | |  | |
| **Comments:** | | | | | | | |
| **Procedure** | | | | | | | |
| **Step No.**  **Test Case** | **Activities** | | | **Examination of Result** | **Actual Results** | | **Test Result** |
| 1 | Select and Inspect Data Linkage submenu | | |  |  | |  |
| TC-2.4.1 |  | | | Check if the sub header contains List option |
| 2 | Select List from Data Linkage submenu | | |  |  | |  |
| TC-2.4.2 |  | | | Check if a list of data linkages is displayed into the main content area |
| 3 | Select and Inspect Add from Data Linkage submenu | | |  |  | |  |
| TC-2.4.3 |  | | | Check if the sub header contains Add option |
| 4 | Select and put input in Source Applications drop-down menu | | |  |  | |  |
| TC-2.4.10 |  | | | Check if correct applications are displayed in Source Applications drop-down menu |
| 5 | Select and put input Data Type drop-down menu | | |  |  | |  |
| TC-2.4.9 |  | | | Check if correct data types for specific application from Source Applications is displayed in Data Type drop-down menu |
| 6  TC-2.4.10 | Select and put input in Destination Application drop-down menu | | | Check if correct applications are displayed in Destination Applications drop-down menu |  | |  |
| 7  TC-2.4.4 | Select Save button from the Add Data Linkage page | | | Check if data linkage entry was created |  | |  |
| 8  TC-2.4.3 | Select Add from Data Linkage submenu | | | Check if sub header contains Add option |  | |  |
| 9  TC-2.4.8 | Select Save button from the Add Data Linkage page | | | Check if message is displayed and no data linkage entry was created |  | |  |
| 10  TC-2.4.5 | Select Cancel from Data Linkage submenu | | | Check if redirected into home page and list is displayed in main content homepage |  | |  |
| 11  TC-2.4.4 | Select Add from Data Linkage submenu | | | Check if sub header contains Add option |  | |  |
| 12 | Select and input values in Data Type, and the same Application values in Source and Destination drop-down menu | | | N/A |  | |  |
| 13  TC-2.4.7 | Select Save button from the Add Data Linkage page | | | Check if message is displayed and no data linkage entry was created |  | |  |
| 14 | Select and input a combination of values that already exist in the list for Source, Data Type, and Application drop-down menu | | | N/A |  | |  |
| 15  TC-2.4.6 | Select Save button from the Add Data Linkage page | | | Check if message is displayed and no data linkage entry was created |  | |  |
| **Stop and Wrap Up:** Logout from the application. Clean cache. Exit browser. | | | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TPID** | **Objective and Priority** | | | **Est. Dur.:** |
| TP-L-2 | The purpose of this test procedure is to test the ability of an admin user to adequately delete data linkages  Priority: H | | | 5 min |
| **Start up:** Login to ETA site using the account with admin-level permissions listed below:  Username: jonny.jones@xxxx.com, Password: password | | | | |
| **Relationships to other procedures:** None | | | | |
| **Test Log** | | | | |
| Date: | Initials: | Test item: | | OK/Not OK |
|  |  |  | |  |
| **Comments:** something here. | | | | |
| **Procedure** | | | | |
| **Step No.**  **Test Case** | **Activities** | **Examination of Result** | **Actual Results** | **Test Result** |
| 1 | Select List option from Data Linkage sub header |  |  |  |
| TC-2.4.11 |  | Check if delete option is available from list of data linkages is displayed in the main content area |
| 2 | Select and Inspect Delete button for one of the entries from the list of data linkages |  |  |  |
| TC-2.4.13 |  | Check if a confirmation window is displayed |
| 3 | Select No from the confirmation window |  |  |  |
| TC-2.4.15 |  | Check if returned to main content page with list of data linkages displayed |
| 4 | Select Delete button for one of the entries from the list of data linkages |  |  |  |
| TC-2.4.11 |  | Check if a confirmation window is displayed |
| 5 | Select Yes from the confirmation window |  |  |  |
| TC-2.4.14 |  | Check if returned to main content page with list of data linkages displayed. |
| 6  TC-2.4.12 | Inspect main content page | Check if deleted data linkage entry does not exist in list of data linkages |  |  |
| **Stop and Wrap Up:** Logout from the application. Clean cache. Exit browser. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TPID** | **Objective and Priority** | | | **Est. Dur.:** |
| TP-D-4 | The purpose of this test procedure is to test all abilities of a standard user associated with data linkages  Priority: H | | | 5 min |
| **Start up:** Login to the ETA site using the standard level permissions listed below.  Username: sally.thomas@xxxx.com, Password: password | | | | |
| **Relationships to other procedures:** None | | | | |
| **Test Log** | | | | |
| Date: | Initials: | Test item: | | OK/Not OK |
|  |  |  | |  |
| **Comments:** | | | | |
| **Procedure** | | | | |
| **Step No.**  **Test Case** | **Activities** | **Examination of Result** | **Actual Results** | **Test Result** |
| 1 | Select and Inspect Data Linkage submenu. |  |  |  |
| TC-2.3.16 |  | Check that the Data Linkages submenu contains a ‘List’ option. |
| 2 | Inspect the Data Linkage submenu |  |  |  |
| TC-2.3.18 |  | Check that the Data Linkages submenu does not contain an Add option. |
| 3 | Select List option from Data Linkage sub menu |  |  |  |
| TC-2.3.17 |  | Check that the main content area is filled with a table containing the data linkages list from the database. |
| 5 | Inspect list of Data Linkages for delete button |  |  |  |
| TC-2.3.19 |  | Check that the entries do not have a Delete button |
| **Stop and Wrap Up:** Logout from the application. Clean cache. Exit browser. | | | | |

### 6.2 Traceability of Test Cases to Use Cases

This section contains various traceability matrices that contain information on how use cases are fulfilled through test cases. The matrices are split into subsections based on system functionalities.

#### 6.2.1 GUI Login and Layout Traceability

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| UCID | Description | TC-2.1.1 | TC-2.1.2 | TC-2.1.3 | TC-2.1.4 | TC-2.1.5 | TC-2.1.6 | TC-2.1.7 | TC-2.1.8 | TC-2.1.9 |
| UC1 | GUI – User Login/Logout | X | X | X | X | X | X | X | X | X |
| UC13 | REST – User Data | X | X | X |  |  |  |  |  |  |

Table 6-5. GUI Login and Layout Traceability.

#### 6.2.2 Applications Traceability

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| UCID | Description | TC-2.2.1 | TC-2.2.2 | TC-2.2.3 | TC-2.2.4 | TC-2.2.5 | TC-2.2.6 | TC-2.2.7 | TC-2.2.8 | TC-2.2.9 | TC-2.2.10 | TC-2.2.11 | TC-2.2.12 | TC-2.2.13 | TC-2.2.14 | TC-2.2.15 | TC-2.2.16 | TC-2.2.17 | TC-2.2.18 | TC-2.2.19 | TC-2.2.20 | TC-2.2.21 | TC-2.2.22 | TC-2.2.23 | TC-2.2.24 | TC-2.2.25 | TC-2.2.26 | TC-2.2.27 | TC-2.2.28 | TC-2.2.29 |
| UC2 | GUI – List | X |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC3 | GUI – Add |  |  |  |  | X |  | X | X | X | X |  | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC4 | GUI – Update |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  | X | X | X |  | X | X |  |  |  |  |  |  |  |
| UC5 | GUI – Delete |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  | X | X | X |  | X |
| UC15 | REST – List |  | X |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC16 | REST – Add |  |  |  |  |  | X |  |  |  |  | X |  | X |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |
| UC17 | REST – Update |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC18 | REST – Delete |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  | X |  | X |  |
| UC25 | Database CRUD |  | X |  | X |  | X |  |  |  |  | X |  | X |  | X | X |  |  |  | X |  |  |  | X |  | X |  | X |  |

Table 6-6. Applications Traceability.

#### 6.2.3 Data Types Traceability

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| UCID | Description | TC-2.3.1 | TC-2.3.2 | TC-2.3.3 | TC-2.3.4 | TC-2.3.5 | TC-2.3.6 | TC-2.3.7 | TC-2.3.8 | TC-2.3.9 | TC-2.3.10 | TC-2.3.11 | TC-2.3.12 | TC-2.3.13 | TC-2.3.14 | TC-2.3.15 | TC-2.3.16 | TC-2.3.17 | TC-2.3.18 | TC-2.3.19 | TC-2.3.20 | TC-2.3.21 | TC-2.3.22 |
| UC6 | GUI – List | X |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC7 | GUI – Add |  |  |  |  | X |  |  | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC8 | GUI – Update |  |  |  |  |  |  |  |  |  | X |  |  | X | X |  |  |  |  |  |  |  | X |
| UC9 | GUI – Delete |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  | X | X | X |  |  |  |
| UC18 | REST – List |  | X |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC19 | REST – Add |  |  |  |  |  | X | X |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC20 | REST – Update |  |  |  |  |  |  |  |  |  |  | X | X |  | X |  |  |  |  |  |  |  | X |
| UC21 | REST – Delete |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  | X | X | X |  |  |
| UC25 | Database CRUD |  | X |  | X |  | X |  |  | X |  | X |  |  | X |  | X |  | X |  | X | X | X |

Table 6-7. Data Types Traceability.

#### 6.2.4 Data Linkages Traceability

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| UCID | Description | TC-2.4.1 | TC-2.4.2 | TC-2.4.3 | TC-2.4.4 | TC-2.4.5 | TC-2.4.6 | TC-2.4.7 | TC-2.4.8 | TC-2.4.9 | TC-2.4.10 | TC-2.4.11 | TC-2.4.12 | TC-2.4.13 | TC-2.4.14 | TC-2.4.15 | TC-2.4.16 | TC-2.4.17 | TC-2.4.18 | TC-2.4.19 |
| UC10 | GUI – List | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |
| UC11 | GUI – Add |  |  | X |  | X | X | X | X | X | X |  |  |  |  |  |  |  | X |  |
| UC12 | GUI – Delete |  |  |  |  |  |  |  |  |  |  | X |  | X | X | X |  |  |  | X |
| UC22 | REST – List |  | X |  |  |  | X | X | X | X | X |  |  |  |  |  |  | X |  |  |
| UC23 | REST – Add |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC24 | REST – Delete |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |
| UC25 | Database CRUD |  | X |  | X |  | X | X | X | X | X |  | X |  |  |  |  | X | X | X |

Table 6-8. Data Linkages Traceability.

### 6.3 Techniques for Test Generation

#### 6.3.1 Test Generation Techniques

Given the nature of this project, we felt it necessary to utilize Behavior-Driven Development (BDD) testing techniques. We chose this approach because of how well suited this method is for testing web applications that are largely based on use cases. By breaking the application down into four feature sets, we were able to identify the test cases that would need to be tested in order to ensure that the application is both complete and accurate. The test procedures would then be generated based on specific functions of the application, which ensures that multiple test cases are evaluated in a single test procedure. Our test procedures are all black-box in nature because the behavior of the individual functions is contained within the application layer which is not be visible to the tester. Additional reasons for using black-box testing are the visual nature of the output and the testing process remaining high-level since the application does not require code inspection to test. These reasons are also why we opted for manual testing over automated testing, which is more efficient and effective considering the limited number of inputs and outputs which are easy to verify visually.

#### 6.3.2 Testing Quality Measurement Criteria

In addition to the functional requirements that will be used to test the overall quality of the application, the following non-functional criteria will also be used as sources for measuring testing quality due to the behavioral nature of the application.

Portability – The application will be tested for functionality across multiple operating systems and browsers to ensure that it is as functional and as portable as possible.

Reliability – The actual results of the tests remain consistent with the predicted results throughout multiple tests. Inputs that are alike in predicted results yield the same results. For example, two different administrator accounts allowing the same actions will find the same conclusions.

Security – The test cases and test procedures verify the information entered by users to ensure that only users with proper credentials and permissions can login and perform permission-locked actions. These measures start at the user login and remain throughout the application in the presentation layer as well as the web service layer.

Performance – The test cases are designed to follow the normal path of execution that a user might follow. This application of the test procedures will identify any performance issues that might exist. Given that this application is based on a network connection, performance issues that are the result of network connectivity problems cannot be tested by the testing team.

Testability – The necessary results for the test cases and test procedures are easily confirmed due to the visual nature of the application. The inputs for tests are either controlled by the test procedures or limited by the application, which helps narrow down the number of required inputs.

Usability – The test cases are designed to be as thorough as possible and to ensure that all of the functions provide appropriate feedback to the user.

### 6.4 Assessment of the Test Suite

TBD. (Which metrics were used for such assessment?)

# 7. Acknowledgement

To all of us who made this possible, we are grateful.

We would like to thank The University of Texas at Dallas for arranging this wonderful opportunity for us. We are also grateful to Perspecta Inc. for sponsoring this project.

We take this opportunity to express our profound gratitude and deep regard to our mentor, Dave Gibson, for his exemplary guidance, monitoring, and continuous encouragement throughout the course of this project. The help and guidance given by him is something we will all be able to look back on with fondness throughout our careers.

We would like to express our heartfelt gratitude to our professor, Dr. Eric Wong, and our teaching assistant, Linghuan Hu, for their kind support and inspiration which have immensely strengthened our confidence during this project.

We also take this opportunity to express a deep sense of gratitude to our Team Leaders, James Fritz and Eric Leon, for their cordial support, valuable information, and guidance, which helped the rest of the team in completing this task throughout the various stages.

We are obliged to each member of our team for the valuable information each of us provided. We are grateful for each other’s cooperation during this project. We are thankful to each other for the continual constructive criticism and invaluable suggestions and help, which benefited each of us a lot during this period.

# 8. References

*IEEE Recommended Practice for Software Requirements Specifications*, in IEEE Std 830-1998, vol., no., pp.1-40, 20 Oct. 1998, doi: 10.1109/IEEESTD.1998.88286. [Online]. Available: https://ieeexplore.ieee.org/servlet/opac?punumber=5841

*IEEE Standard for Information Technology--Systems Design--Software Design Descriptions*, in IEEE STD 1016-2009 , vol., no., pp.1-35, 20 July 2009, doi: 10.1109/IEEESTD.2009.5167255. [Online]. Available: https://ieeexplore.ieee.org/servlet/opac?punumber=5167253

*IEEE Standard for Software and System Test Documentation*, in IEEE Std 829-2008, vol., no., pp.1-150, 18 July 2008, doi: 10.1109/IEEESTD.2008.4578383. [Online]. Available: https://ieeexplore.ieee.org/servlet/opac?punumber=4578271

*ISO/IEC Standard for Systems and Software Engineering - Recommended Practice for Architectural Description of Software-Intensive Systems*, in ISO/IEC 42010 IEEE Std 1471-2000 First edition 2007-07-15, vol., no., pp.1-24, 15 July 2007, doi: 10.1109/IEEESTD.2007.386501. [Online]. Available: https://ieeexplore.ieee.org/servlet/opac?punumber=4278470

*ISO/IEC/IEEE International Standard - Systems and software engineering - Life cycle processes - Project management*, in ISO/IEC/IEEE 16326:2019(E) , vol., no., pp.1-42, 13 Dec. 2019, doi: 10.1109/IEEESTD.2019.8932690. [Online]. Available: https://ieeexplore.ieee.org/servlet/opac?punumber=8932688

*ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering*, in ISO/IEC/IEEE 29148:2018(E), vol., no., pp.1-104, 30 Nov. 2018, doi: 10.1109/IEEESTD.2018.8559686. [Online]. Available: https://ieeexplore.ieee.org/servlet/opac?punumber=8559684

*ISO/IEC/IEEE International Standard - Software and systems engineering --Software testing --Part 1: Concepts and definitions*, in *ISO/IEC/IEEE 29119-1:2013(E)*, vol., no., pp.1-64, 1 Sept. 2013, doi: 10.1109/IEEESTD.2013.6588537. [Online]. Available: https://ieeexplore.ieee.org/servlet/opac?punumber=6588535

*ISO/IEC/IEEE International Standard - Software and systems engineering --Software testing --Part 2: Test processes*, in *ISO/IEC/IEEE 29119-2:2013(E)*, vol., no., pp.1-68, 1 Sept. 2013, doi: 10.1109/IEEESTD.2013.6588543. [Online]. Available: https://ieeexplore.ieee.org/servlet/opac?punumber=6588541

*ISO/IEC/IEEE International Standard - Software and systems engineering --Software testing --Part 3: Test documentation*, in *ISO/IEC/IEEE 29119-3:2013(E)*, vol., no., pp.1-138, 1 Sept. 2013, doi: 10.1109/IEEESTD.2013.6588540. [Online]. Available: https://ieeexplore.ieee.org/servlet/opac?punumber=6588538

*ISO/IEC/IEEE International Standard - Software and systems engineering--Software testing--Part 4: Test techniques*, in *ISO/IEC/IEEE 29119-4:2015*, vol., no., pp.1-149, 8 Dec. 2015, doi: 10.1109/IEEESTD.2015.7346375. [Online]. Available: https://ieeexplore.ieee.org/servlet/opac?punumber=7346373

P. B. Kruchten, *The 4+1 View Model of architecture*, in IEEE Software, vol. 12, no. 6, pp. 42-50, Nov. 1995, doi: 10.1109/52.469759. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=469759&isnumber=9910

J. Maréchaux. *Developing a J2EE Architecture with Rational Software Architect Using the Rational Unified Process*. IBM Developer. [Online]. Available: https://www.ibm.com/developerworks/rational/library/05/0816\_Louis/index.html (accessed Sep. 23, 2020).

M. Shaw, *Comparing Architectural Design Styles*, in IEEE Software, vol. 12, no. 6, pp. 27-41, Nov. 1995, doi: 10.1109/52.469758.

END OF DOCUMENT