|  |
| --- |
| Enterprise Tracking Application |
| Software Requirements Specification |
| Prepared For: Perspecta  Location: Austin, TX |
| Date: 9/7/2020  Revised: 11/9/2020 |
| Angeleen Abesamis  Remnin Ferrer  James Fritz  Yeremy Joseph  Eric Leon |

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Comments | Author |
| 1.0 | 9/18/2020 | Original Submission |  |
| 2.0 | 11/9/2020 | Requirements Updates |  |
|  |  |  |  |

# Abstract

This document serves as the software requirements specification 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 define what the application will do based on the information that has been provided by the customer.

Table of Contents

[Revision History 2](#_Toc55824616)

[Abstract 3](#_Toc55824617)

[Table of Contents 4](#_Toc55824618)

[List of Figures 5](#_Toc55824619)

[List of Tables 6](#_Toc55824620)

[1. Introduction 7](#_Toc55824621)

[1.1 Purpose 7](#_Toc55824622)

[1.2 Scope 7](#_Toc55824623)

[1.3 Definitions, Acronyms, and Abbreviations 7](#_Toc55824624)

[1.4 Overview 8](#_Toc55824625)

[2. Use Case Model for Functional Requirements 9](#_Toc55824626)

[2.1 Graphical User Interface Use Case Models 9](#_Toc55824627)

[2.2 Web Services Use Case Models 30](#_Toc55824628)

[2.3 Persistence Layer Use Case Models 45](#_Toc55824629)

[3. Use Case Model Rationale 46](#_Toc55824630)

[4. Non-Functional Requirements 47](#_Toc55824631)

[4.1 Usability 47](#_Toc55824632)

[4.2 Performance 47](#_Toc55824633)

[4.3 Portability 47](#_Toc55824634)

[4.4 Security 47](#_Toc55824635)

[5. Evidence of Configuration Management 48](#_Toc55824636)

[6. References 49](#_Toc55824637)

# List of Figures

|  |  |  |
| --- | --- | --- |
| Figure | Description | Page |
| 2-1 | Use case 1. | 9 |
| 2-2 | Login page wireframe. | 10 |
| 2-3 | Home page wireframe. | 11 |
| 2-4 | Use case 2. | 13 |
| 2-5 | Use case 3. | 14 |
| 2-6 | Use case 4. | 15 |
| 2-7 | Use case 5. | 16 |
| 2-8 | Use case 6. | 19 |
| 2-9 | Use case 7. | 20 |
| 2-10 | Use case 8. | 21 |
| 2-11 | Use case 9. | 22 |
| 2-12 | Use case 10. | 25 |
| 2-13 | Use case 11. | 26 |
| 2-14 | Use case 12. | 27 |
| 2-15 | Use case 13. | 30 |
| 2-16 | Use case 14. | 31 |
| 2-17 | Use case 15. | 32 |
| 2-18 | Use case 16. | 33 |
| 2-19 | Use case 17. | 34 |
| 2-20 | Use case 18. | 36 |
| 2-21 | Use case 19. | 37 |
| 2-22 | Use case 20. | 38 |
| 2-23 | Use case 21. | 39 |
| 2-24 | Use case 22. | 41 |
| 2-25 | Use case 23. | 42 |
| 2-26 | Use case 24. | 43 |
| 2-27 | Use case 25. | 45 |

# List of Tables

|  |  |  |
| --- | --- | --- |
| Table | Description | Page |
| 2-1 | Use case 1. | 9 |
| 2-2 | Login UI requirements. | 11 |
| 2-3 | Main content home page content UI requirements. | 12 |
| 2-4 | Use case 2. | 13 |
| 2-5 | Use case 3. | 14 |
| 2-6 | Use case 4. | 15 |
| 2-7 | Use case 5. | 16 |
| 2-8 | Applications component UI functional requirements. | 18 |
| 2-9 | Use case 6. | 19 |
| 2-10 | Use case 7. | 20 |
| 2-11 | Use case 8. | 21 |
| 2-12 | Use case 9. | 22 |
| 2-13 | Data types component UI functional requirements. | 24 |
| 2-14 | Use case 10. | 25 |
| 2-15 | Use case 11. | 26 |
| 2-16 | Use case 12. | 27 |
| 2-17 | Data links component UI functional requirements. | 29 |
| 2-18 | Use case 13. | 30 |
| 2-19 | User data functional requirements. | 31 |
| 2-20 | Use case 14. | 31 |
| 2-21 | Use case 15. | 32 |
| 2-22 | Use case 16. | 33 |
| 2-23 | Use case 17. | 34 |
| 2-24 | Applications web services functional requirements. | 35 |
| 2-25 | Use case 18. | 36 |
| 2-26 | Use case 19. | 37 |
| 2-27 | Use case 20. | 38 |
| 2-28 | Use case 21. | 39 |
| 2-29 | Applications web services functional requirements. | 40 |
| 2-30 | Use case 22. | 41 |
| 2-31 | Use case 23. | 42 |
| 2-32 | Use case 24. | 43 |
| 2-33 | Web services layer functional requirements. | 44 |
| 2-34 | Persistence layer functional requirements. | 45 |
| 3-1 | Use case 1 rationale. | 46 |

# 1. Introduction

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, and abbreviations, references, and a sectional overview of the SRS. The aim of this document is to gather and analyze and give an in-depth insight of the complete Enterprise Tracking Application software by defining the problem statement in detail. Nevertheless, it also concentrates on the capabilities required by stakeholders and their needs while defining high-level product features. The detailed requirements by Perspecta are provided in this document.

### 1.1 Purpose

The purpose of the SRS is to provide a detailed overview of the ETA software product parameters and goals as per the requirements set by Perspecta Inc. This document describes the requirements for the project's user interface, hardware, and software. It defines how Perspecta and the project team see the product and its functionality. The SRS will help the design and development throughout the software development lifecycle (SDLC) process. This document is intended for the UTD project team, who will use the requirements as a guide to ensure project accuracy and completeness. This document is also for the UTD faculty as it fulfills one of the requirements of the project.

### 1.2 Scope

Under the direction of UTD faculty, the UTD Senior Project Team is tasked with producing and delivering the ETA product for Perspecta Inc. The product will manage application integration records throughout the company which will allow project integration tracking capabilities for the Perspecta Enterprise. The product also provides emphasis on permission-based access, ensuring that only users with appropriate credentials can access the information. The three main components of the project are a graphical user interface that displays application integration records and allows the user input, a web service layer that allows communication between users and the server through RESTful API methods, and an MSSQL persistence layer that stores integration record data which will be provided by the customer.

### 1.3 Definitions, Acronyms, and Abbreviations

* **API** – Application Programming Interface
* **Auth** – Authentication
* **ETA** – Enterprise Tracking Application
* **FR** – Functional Requirement
* **NFR** – Non-Functional Requirement
* **PL** – Persistence Layer
* **REST** – Representational State Transfer
* **SDLC** – Software Development Lifecycle
* **UC** – Use Case
* **UI** – User Interface
* **WS** – Web Services Layer

### 1.4 Overview

The SRS is structured as follows:

* Section 1, Introduction. The introduction of the SRS provides an overview of the entire document. It includes the introduction, purpose, scope, and structure of the document.
* Section 2, Use Case Model for Functional Requirements. This section is broken into the major components of the project. It contains the use cases for the project that will be used to identify the project functionality and the resulting functional requirements. The use cases outline the basic flow of the project functions.
* Section 3, Use Case Model Rationale. This section briefly describes the reasoning behind the structure of our use cases.
* Section 4, Non-Functional Requirements. This section contains the non-functional requirements that we will reference when choosing the architecture for the project.
* Section 5, Evidence of Configuration Management: This section contains a single image depicting evidence that the SRS has been placed under configuration management.
* Section 6, References: The References section contains an IEEE style formatted list of the external sources of information the team used to create this document.

# 2. Use Case Model for Functional Requirements

### 2.1 Graphical User Interface Use Case Models

Requirements table lookup acronym: UI

#### 2.1.1 Login UI

##### 2.1.1.1 Login UI Feature Introduction

The user login interface to verify access via permissions.

|  |  |
| --- | --- |
| Figure 2-1. 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 2-1. Use case 1.

##### 2.1.1.2 Wireframe

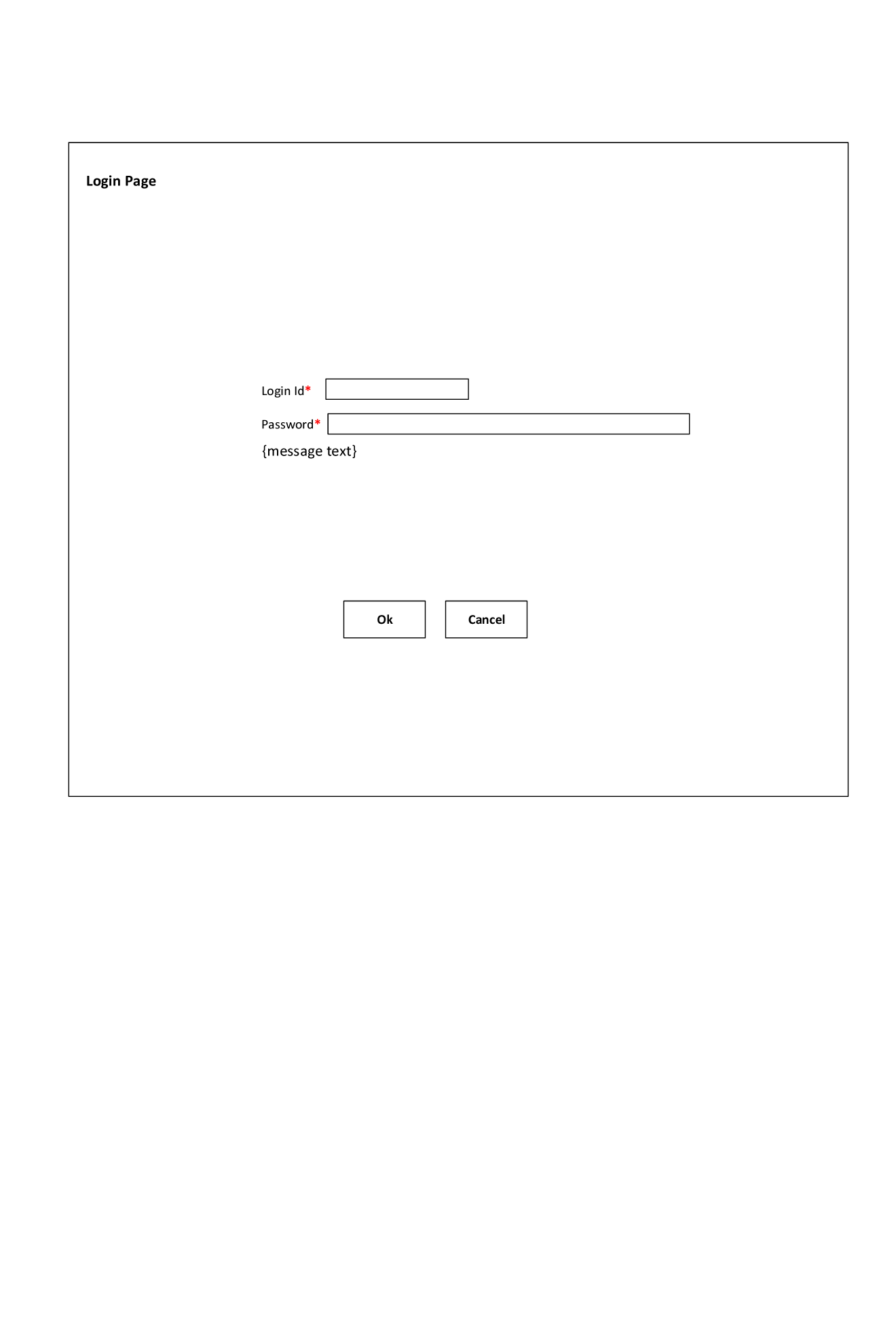


Figure 2-2. Login page wireframe.

##### 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 2-2. Login UI requirements.

#### 2.1.2 Main Content Page UI

##### 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.

###### 2.1.2.1.1 Wireframe

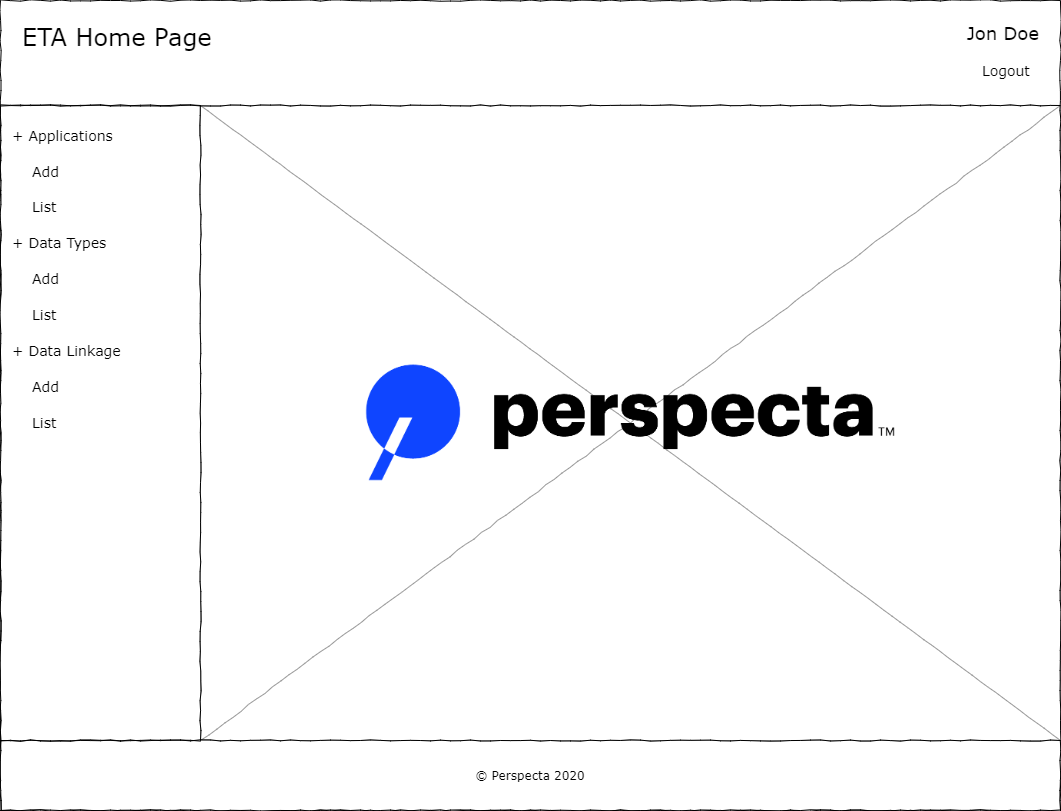


Figure 2-3. Home page wireframe.

###### 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 2-3. Main content home page content UI requirements.

##### 2.1.2.2 Main Content Page Application Features

|  |  |
| --- | --- |
| Figure 2-4. 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 2-4. Use case 2.

|  |  |
| --- | --- |
| Figure 2-5. 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 2-5. Use case 3.

|  |  |
| --- | --- |
| Figure 2-6. 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 2-6. Use case 4.

|  |  |
| --- | --- |
| Figure 2-7. 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 2-7. Use case 5.

###### 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 2-8. Applications component GUI functional requirements.

##### 2.1.2.3 Main Content Page Data Types Features

|  |  |
| --- | --- |
| Figure 2-8. 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 2-9. Use case 6.

|  |  |
| --- | --- |
| Figure 2-9. 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 2-10. Use case 7.

|  |  |
| --- | --- |
| Figure 2-10. 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 2-11. Use case 8.

|  |  |
| --- | --- |
| Figure 2-11. 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 2-12. Use case 9.

###### 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 2-13. Data types component UI functional requirements.

##### 2.1.2.4 Main Content Page Data Linkage Features

|  |  |
| --- | --- |
| Figure 2-12. 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 2-14. Use case 10.

|  |  |
| --- | --- |
| Figure 2-13. 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 2-15. Use case 11.

|  |  |
| --- | --- |
| Figure 2-14. 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 2-16. Use case 12.

###### 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 2-17. Data links component UI functional requirements.

### 2.2 Web Services Use Case Models

Requirements table lookup acronym: WS

#### 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.

##### 2.2.2 User Data Web Services

|  |  |
| --- | --- |
| Figure 2-15. 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 2-18. Use case 13.

##### *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 2-19. User data functional requirements.

##### 2.2.3 Application Web Services

|  |  |
| --- | --- |
| Figure 2-16. 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 2-20. Use case 14.

|  |  |
| --- | --- |
| Figure 2-17. 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 2-21. Use case 15.

|  |  |
| --- | --- |
| Figure 2-18. 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 2-22. Use case 16.

|  |  |
| --- | --- |
| Figure 2-19. 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 2-23. Use case 17.

##### *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 2-24. Applications web services functional requirements.

##### 2.2.4 Data Type Web Services

|  |  |
| --- | --- |
| Figure 2-20. 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 2-25. Use case 18.

|  |  |
| --- | --- |
| Figure 2-21. 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 2-26. Use case 19.

|  |  |
| --- | --- |
| Figure 2-22. 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 2-27. Use case 20.

|  |  |
| --- | --- |
| Figure 2-23. 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 2-28. Use case 21.

##### *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 2-29. Applications web services functional requirements.

##### 2.2.3 Data Linkage Web Services

|  |  |
| --- | --- |
| Figure 2-24. 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 2-30. Use case 22.

|  |  |
| --- | --- |
| Figure 2-25. 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 2-31. Use case 23.

|  |  |
| --- | --- |
| Figure 2-26. 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 2-32. Use case 24.

##### *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 2-33. Web services layer functional requirements.

### 2.3 Persistence Layer Use Case Models

Requirements table lookup acronym: PL

#### 2.3.1 Persistence Layer Introduction

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

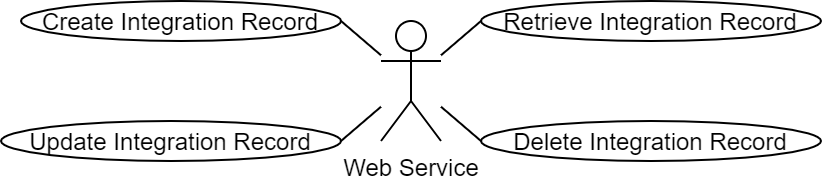


Figure 2-27. Use case 25.

#### 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 2-34. Persistence layer functional requirements.

# 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 3-1. Use case 1 rationale.

# 4. Non-Functional Requirements

### 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.

### 4.2 Performance

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

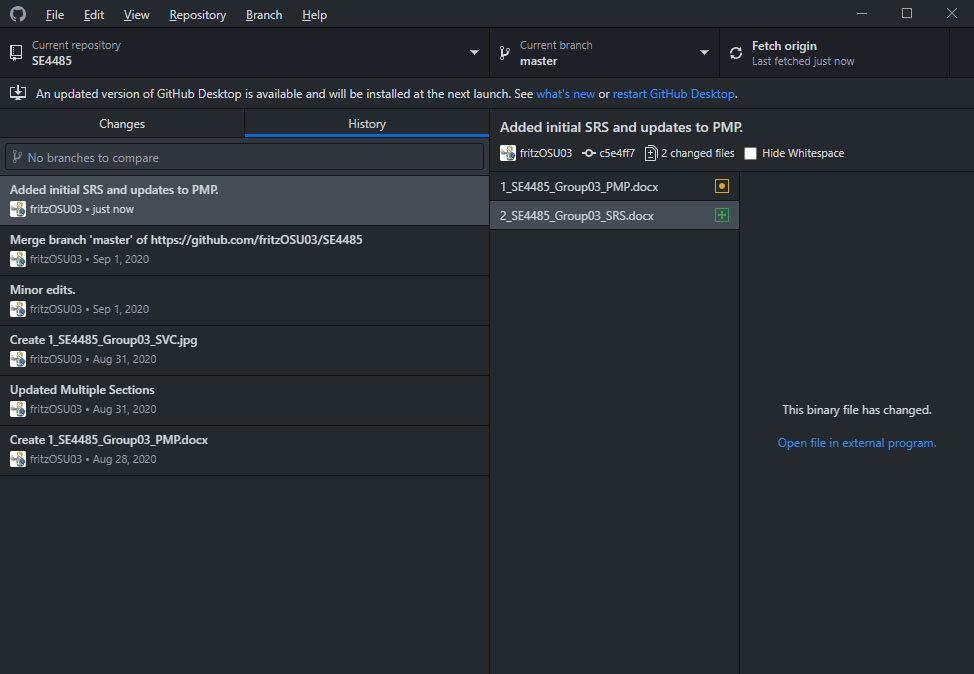
### 4.3 Portability

NFR3: The application shall function in modern web browsers.

### 4.4 Security

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

# 5. Evidence of Configuration Management



# 6. References

*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

*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

END OF DOCUMENT