|  |
| --- |
| Enterprise Tracking Application |
| Detailed Design Document |
| Prepared For: Perspecta  Location: Austin, TX |
| Date: 10/3/2020  Revised: 10/26/2020 |
| Angeleen Abesamis  Remnin Ferrer  James Fritz  Yeremy Joseph  Eric Leon |

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Comments | Author |
| 1.0 | 10/2/2020 | Original Submission |  |
| 2.0 | 10/26/2020 | Updated Web Services Class and Sequence Diagrams | James Fritz |
|  |  |  |  |

# Abstract

This document serves as the detailed design document for the Enterprise Tracking Application that is being developed by the University of Texas at Dallas project team for Perspecta Inc.

# Table of Contents

[Revision History 2](#_Toc54623068)

[Abstract 3](#_Toc54623069)

[Table of Contents 4](#_Toc54623070)

[List of Figures 5](#_Toc54623071)

[List of Tables 6](#_Toc54623072)

[1. Introduction 7](#_Toc54623073)

[1.1 Purpose 7](#_Toc54623074)

[1.2 Scope 7](#_Toc54623075)

[1.3 Definitions, Acronyms, and Abbreviations 7](#_Toc54623076)

[1.4 Overview 7](#_Toc54623077)

[2. GUI Design 9](#_Toc54623078)

[2.1 Login Page 9](#_Toc54623079)

[2.2 Home Page 10](#_Toc54623080)

[2.3 Applications Page 11](#_Toc54623081)

[2.4 Data Types Page 14](#_Toc54623082)

[2.5 Data Linkage Page 18](#_Toc54623083)

[3. Static Model 21](#_Toc54623084)

[3.1 System Class Diagrams 22](#_Toc54623085)

[3.2 Presentation Layer Class Diagrams 24](#_Toc54623086)

[3.3 Application Layer Class Diagrams 26](#_Toc54623087)

[3.4 Data Layer Class Diagrams 30](#_Toc54623088)

[4. Dynamic Model 32](#_Toc54623089)

[4.1 Presentation Layer Class Diagrams 32](#_Toc54623090)

[4.2 Application Layer Class Diagrams 34](#_Toc54623091)

[5. Rationale for the Detailed Design Model 45](#_Toc54623092)

[6. System Traceability 46](#_Toc54623093)

[6.1 Functional Requirements 46](#_Toc54623094)

[6.2 Non-Functional Requirements 49](#_Toc54623095)

[7. Evidence of Configuration Management 50](#_Toc54623096)

[8. References 51](#_Toc54623097)

# List of Figures

|  |  |  |
| --- | --- | --- |
| Figure | Description | Page |
| 2-1 | Login Page Wireframe | 8 |
| 2-2 | Home Page Wireframe | 9 |
| 2-3 | List Applications Wireframe | 10 |
| 2-4 | Add/Edit Applications Wireframe | 11 |
| 2-5 | Delete Applications Wireframe | 12 |
| 2-6 | List Data Types Wireframe | 13 |
| 2-7 | Add Data Type Wireframe | 14 |
| 2-8 | Edit Data Type Wireframe | 15 |
| 2-9 | Delete Data Type Wireframe | 16 |
| 2-10 | List Data Linkage Wireframe | 17 |
| 2-11 | Add Data Linkage Wireframe | 18 |
| 2-12 | Delete Data Linkage Wireframe | 19 |
| 3-1 | System Front-End Class Diagram | 21 |
| 3-2 | RESTful Services Class Diagram | 22 |
| 3-3 | GUI Screen Class Diagram | 23 |
| 3-4 | GUI Service Class Diagram | 24 |
| 3-5 | Spring Boot Server Startup Class Diagram | 25 |
| 3-6 | REST API User Authentication Class Diagram | 26 |
| 3-7 | REST API Applications Class Diagram | 27 |
| 3-8 | REST API Application Links Class Diagram | 28 |
| 3-9 | User/Group ER Diagram | 29 |
| 3-10 | Application Data ER Diagram | 30 |
| 4-1 | GUI List Applications Sequence Diagram | 31 |
| 4-2 | GUI Create/Update/Delete Applications Sequence Diagram | 32 |
| 4-3 | User Login Sequence Diagram | 23 |
| 4-4 | List Applications Sequence Diagram | 34 |
| 4-5 | Add Application Sequence Diagram | 35 |
| 4-6 | Edit Application Sequence Diagram | 36 |
| 4-7 | Delete Application Sequence Diagram | 37 |
| 4-8 | List Data Types Sequence Diagram | 38 |
| 4-9 | Add Data Type Sequence Diagram | 38 |
| 4-10 | Edit Data Type Sequence Diagram | 39 |
| 4-11 | Delete Data Type Sequence Diagram | 40 |
| 4-12 | List Data Linkages Sequence Diagram | 41 |
| 4-13 | Add Data Linkage Sequence Diagram | 42 |
| 4-14 | Delete Data Linkage Sequence Diagram | 43 |

# List of Tables

|  |  |  |
| --- | --- | --- |
| Table | Description | Page |
| 6-1 | User Interface Functional Requirements | 45 |
| 6-2 | User Interface Traceability Matrix | 45 |
| 6-3 | RESTful Web Services Functional Requirements | 46 |
| 6-4 | RESTful Web Services Traceability Matrix | 46 |
| 6-5 | Persistence Functional Requirements | 47 |

# 1. Introduction

This Detailed Design Document (DDD) for the Enterprise Tracking Application (ETA) project sponsored by Perspecta Inc. provides the complete description of the system. This document follows the functional and non-functional requirements identified in the Software Requirements Specification (SRS) document of the project.

### 1.1 Purpose

The purpose of this document is to present a detailed description of the structure of the ETA system to satisfy the requirements outlined in the SRS. This document is intended for the University of Texas at Dallas (UTD) project team, who will use the designs as guidelines to implement the project. This document is also for the UTD faculty as it fulfills one of the requirements of the project. This document may also be used by designers who want to upgrade or modify the current design of the ETA system.

### 1.2 Scope

This document gives a detailed description of the software architecture of the ETA system. It specifies the structure and design of some of the modules discussed in the SRS. It also contains the sequence diagrams that outline how specific parts of the application will function. The class diagrams show how the programming team would implement the classes.

### 1.3 Definitions, Acronyms, and Abbreviations

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

**DDD:** Detailed Design Document

**ETA:** Enterprise Tracking Application

**GUI:** Graphical User Interface

**SRS:** Software Requirements Specification

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

### 1.4 Overview

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

* Section 1, Introduction: The introduction of the DDD provides an overview of the entire document. It includes the document introduction, purpose, scope, definitions, acronyms, and abbreviations, and the document structure of this Detailed Design Document.
* Section 2, GUI Design: This section describes the various screens the user will interact with when using the application. The screens are described using traditional wireframe drawings.
* Section 3, Static Model: The static model section includes the class diagrams for the system and the various components.
* Section 4, Dynamic Model: This section outlines how the classes in the static model will interact with one another. These interactions are described using sequence diagrams.
* Section 5, Rationale for the Detailed Design Model: This section offers a brief description of the rationale for the system design to be implemented.
* Section 6, System Traceability: This section outlines the traceability from the requirements specification to the detailed design model.
* Section 7, Evidence of Configuration Management: This section contains a single image depicting evidence that the Detailed Design Document has been placed under configuration management.
* Section 8, References: The References section contains an IEEE style formatted list of the external sources of information the team used to create this document.

# 2. 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.

### 2.1 Login Page

|  |
| --- |
|  |
| Figure 2-1. Login Page Wireframe |

### 2.2 Home Page

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

### 2.3 Applications Page

#### 2.3.1 List Applications

|  |
| --- |
|  |
| Figure 2-3. List Applications Wireframe |

#### 2.3.2 Add/Edit Application

|  |
| --- |
|  |
| Figure 2-4. Add/Edit Applications Wireframe |

#### 2.3.3 Delete Application

|  |
| --- |
|  |
| Figure 2-5. Delete Applications Wireframe |

### 2.4 Data Types Page

#### 2.4.1 List Data Types

|  |
| --- |
|  |
| Figure 2-6. List Data Types Wireframe |

#### 2.4.2 Add Data Type

|  |
| --- |
|  |
| Figure 2-7. Add Data Type Wireframe |

#### 2.4.3 Edit Data Type

|  |
| --- |
|  |
| Figure 2-8. Edit Data Type Wireframe |

#### 2.4.4 Delete Data Type

|  |
| --- |
|  |
| Figure 2-9. Delete Data Type Wireframe |

### 2.5 Data Linkage Page

#### 2.5.1 List Data Linkage

|  |
| --- |
|  |
| Figure 2-10. List Data Linkage Wireframe |

#### 2.5.2 Add Data Linkage

|  |
| --- |
|  |
| Figure 2-11. Add Data Linkage Wireframe |

#### 2.5.3 Delete Data Linkage

|  |
| --- |
|  |
| Figure 2-12. Delete Data Linkage Wireframe |

# 3. 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.

### 3.1 System Class Diagrams

#### 3.1.1 System Front-End Class Diagram

|  |
| --- |
|  |
| Figure 3-1. System Front-End Class Diagram |

#### 3.1.2 System RESTful Service Class Diagram

|  |
| --- |
|  |
| Figure 3-2. RESTful Services Class Diagram |

### 3.2 Presentation Layer Class Diagrams

#### 3.2.1 GUI Screen Class Diagram

|  |
| --- |
|  |
| Figure 3-3. GUI Screen Class Diagram |

#### 3.2.2 GUI Service Class Diagram

|  |
| --- |
|  |
| Figure 3-4. GUI Service Class Diagram |

### 3.3 Application Layer Class Diagrams

#### 3.3.1 Spring Boot Server Startup Class Diagram

|  |
| --- |
|  |
| Figure 3-5. Spring Boot Server Startup Class Diagram |

#### 3.3.2 REST API User Authentication Class Diagram

|  |
| --- |
|  |
| Figure 3-6. REST API User Authentication Class Diagram |

#### 3.3.3 REST API List Applications Class Diagram

|  |
| --- |
|  |
| Figure 3-7. REST API Applications Class Diagram |

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

|  |
| --- |
|  |
| Figure 3-8. REST API Application Links Class Diagram |

### 3.4 Data Layer Class Diagrams

#### 3.4.1 Database User/Group ER Diagrams

|  |
| --- |
|  |
| Figure 3-9. User/Group ER Diagram |

#### 3.4.2 Database Application Data ER Diagram

|  |
| --- |
|  |
| Figure 3-10. Application Data ER Diagram |

# 4. 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.

### 4.1 Presentation Layer Class Diagrams

#### 4.1.1 GUI List Applications Sequence Diagrams

|  |
| --- |
|  |
| Figure 4-1. GUI List Applications Sequence Diagram |

#### 4.1.2 GUI CRUD Applications Sequence Diagrams

|  |
| --- |
|  |
| Figure 4-2. GUI Create/Update/Delete Applications Sequence Diagram |

### 4.2 Application Layer Class Diagrams

#### 4.2.1 User Login Sequence Diagram

|  |
| --- |
|  |
| Figure 4-3. User Login Sequence Diagram |

#### 4.2.2 Applications Sequence Diagrams

##### 4.2.2.1 List Applications Sequence Diagram

|  |
| --- |
|  |
| Figure 4-4. List Applications Sequence Diagram |

##### 4.2.2.2 Add Application Sequence Diagram

|  |
| --- |
|  |
| Figure 4-5. Add Application Sequence Diagram |

##### 4.2.2.3 Edit Application Sequence Diagram

|  |
| --- |
|  |
| Figure 4-6. Edit Application Sequence Diagram |

##### 4.2.2.4 Delete Application Sequence Diagram

|  |
| --- |
|  |
| Figure 4-7. Delete Application Sequence Diagram |

#### 4.2.3 Data Types Sequence Diagrams

##### 4.2.3.1 List Data Types Sequence Diagram

|  |
| --- |
|  |
| Figure 4-8. List Data Types Sequence Diagram |

##### 4.2.3.2 Add Data Type Sequence Diagram

|  |
| --- |
|  |
| Figure 4-9. Add Data Type Sequence Diagram |

##### 4.2.3.3 Edit Data Type Sequence Diagram

|  |
| --- |
|  |
| Figure 4-10. Edit Data Type Sequence Diagram |

##### 4.2.3.4 Delete Data Type Sequence Diagram

|  |
| --- |
|  |
| Figure 4-11. Delete Data Type Sequence Diagram |

#### 4.2.4 Data Linkage Sequence Diagrams

##### 4.2.4.1 List Data Linkages Sequence Diagram

|  |
| --- |
|  |
| Figure 4-12. List Data Linkages Sequence Diagram |

##### 4.2.4.2 Add Data Linkage Sequence Diagram

|  |
| --- |
|  |
| Figure 4-13. Add Data Linkage Sequence Diagram |

##### 4.2.4.3 Delete Data Linkage Sequence Diagram

|  |
| --- |
|  |
| Figure 4-14. Delete Data Linkage Sequence Diagram |

# 5. 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.

# 6. 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.

### 6.1 Functional Requirements

#### 6.1.1 Graphical User Interface 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 6-1. User Interface Functional Requirements

6.1.1.1 Graphical User Interface 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 6-2. User Interface Traceability Matrix

6.1.1.2 Graphical User Interface Traceability Details

* The System Front-End Class Diagram (Figure 3-1) and the GUI Screen Class Diagram (Figure 3-3) 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 2-1), the REST API User Authentication Class Diagram (Figure 3-6), and User Login Sequence Diagram (Figure 4-3).
* UI-FR-6, UI-FR-7, UI-FR-8, UI-FR-9, and UI-FR-10 are achieved by the Home Page Wireframe (Figure 2-2).

#### 6.1.2 RESTful Web Services Requirements

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

Table 6-3. RESTful Web Services Functional Requirements

6.1.2.1 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-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 6-4. RESTful Web Services Traceability Matrix

6.1.2.2 RESTful Web Services Traceability Details

* The Web Service 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).

#### 6.1.3 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 6-5. Persistence Functional Requirements

6.1.3.1 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).

### 6.2 Non-Functional Requirements

#### 6.2.1 Usability

The single page format of the application and table-style display for data is achieved through Login Page Wireframe (Figure 2-1), Home Page Wireframe (Figure 2-2), System Front-End Class Diagram (Figure 3-1), both GUI Class Diagrams (Figures 3-3 and 3-4), and all wireframes associated with the Applications Page (Figure 2-3 through Figure 2-12).

#### 6.2.2 Performance

The site responsiveness within a reasonable amount of time is achieved through RESTful services as shown in the GUI Service Class Diagram (Figure 3-4), the REST API Applications Class Diagram (Figure 3-7), and the REST API Application Links Class Diagram (Figure 3-8). These elements combine to produce a single page, responsive application.

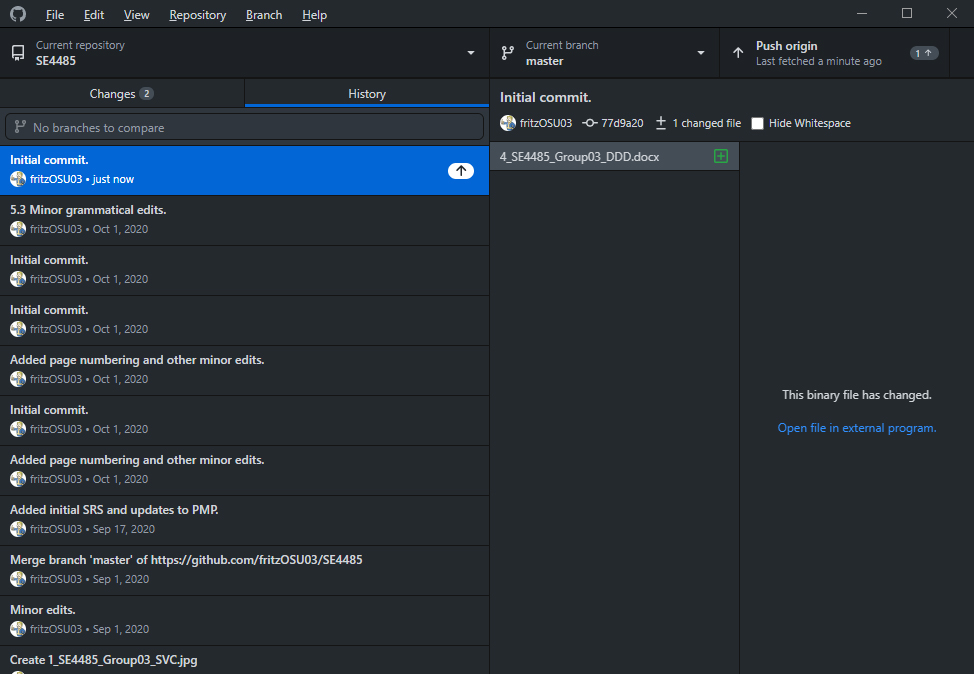
#### 6.2.3 Portability

The functionality in modern web browsers is achieved through the Angular framework depicted in the System Front-End Class Diagram (Figure 3-1). 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.

#### 6.2.4 Security

The user credential encryption and storage in an authentication database is achieved through the REST API User Authentication Class Diagram (Figure 3-6). Additional security is achieved through the single-page application design, which is accomplished by the Angular Framework (Figure 3-1).

# 7. Evidence of Configuration Management



# 8. References

[IEEE] IEEE Draft Standard for Information Technology--Systems Design--Software Design Descriptions," in *IEEE Unapproved Draft Std P1016/D5.0, Dec 2005*, vol., no., 2005.

END OF DOCUMENT