Design Document

Arda Günay, Arif Akkaş, Ezgi Berengi, Kayra Şahin, Utku Üri

EasyRef

November 2019

Contents

1 Introduction

1.1 Information

EasyRef is a program that helps user to store and handle references. User can create references, search for a specific reference and edit it and export selected references.

1.2 Purpose

The purpose of this document is to explain the technical design of EasyRef application and provide an overview about EasyRef.

This document is describing functionalities which will be offered by EasyRef. Document is not involving any information about installation and configuration details.

1.3 Scope

Application Document is implemented based on the scope which explained in the Requirements Document.

1.4 Intended Audience

The intended audience for this document are lecturer Assistant Professor Kaya Oğuz, course assistants and team members.

2 Diagrams

This section shows diagrams to explain the design and structure of EasyRef.

2.1 Use Case Diagram

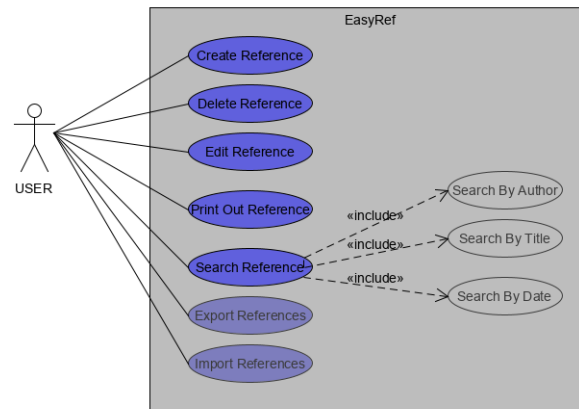EasyRef provides 7 main functions to the user.

Figure 1: Context as Use Case Diagram

2.2 Activity Diagrams

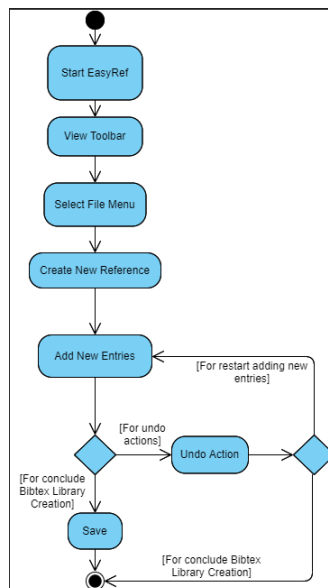Activity Diagrams explain each functionality EasyRef offers.
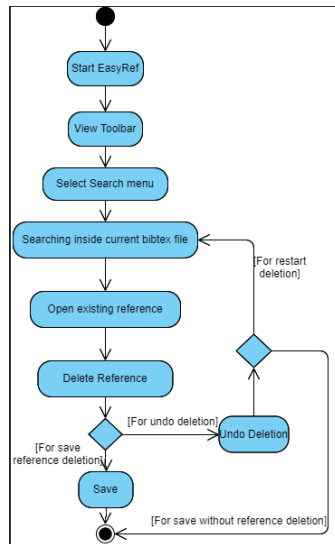

Figure 2: Create Reference Activity Diagram

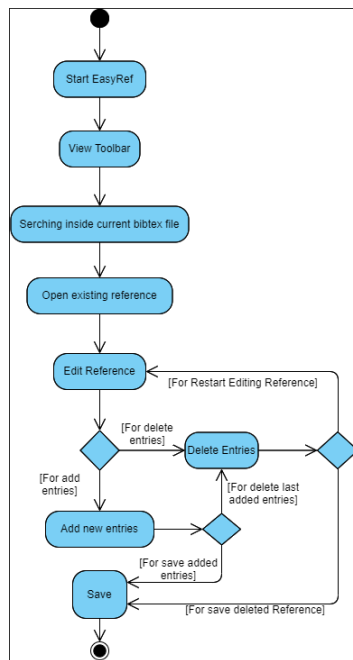Figure 3: Delete Reference Activity Diagram
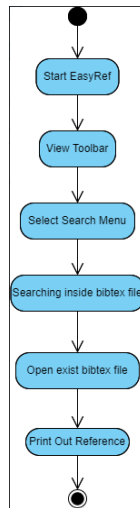

Figure 4: Edit Reference Activity Diagram

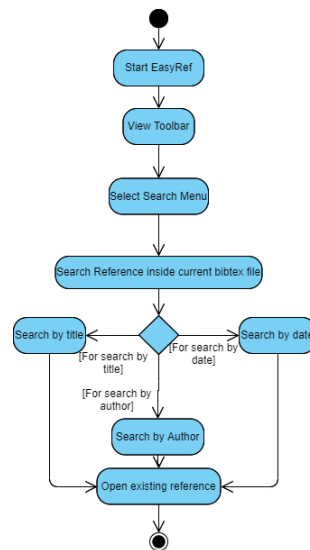Figure 5: Print Out Reference Activity Diagram

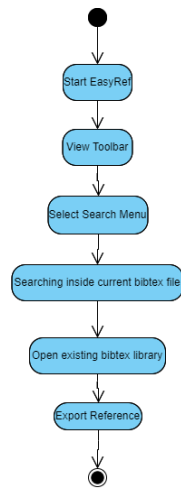Figure 6: Search Reference Activity Diagram
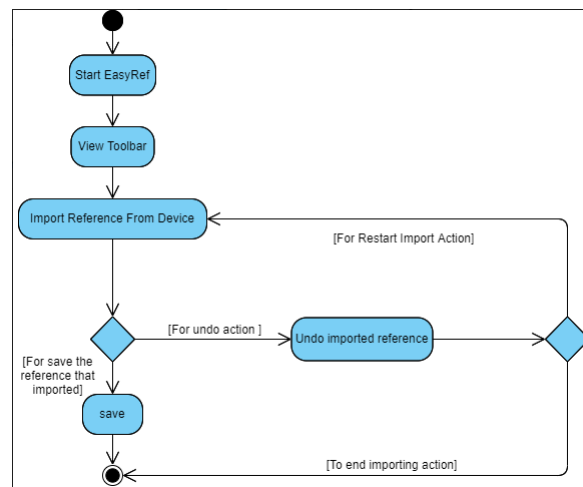
Figure 7: Export Reference Activity Diagram

Figure 8: Import Reference Activity Diagram

2.3 Sequence Diagram

Sequence Diagrams explain the system responses for a specific functional operation
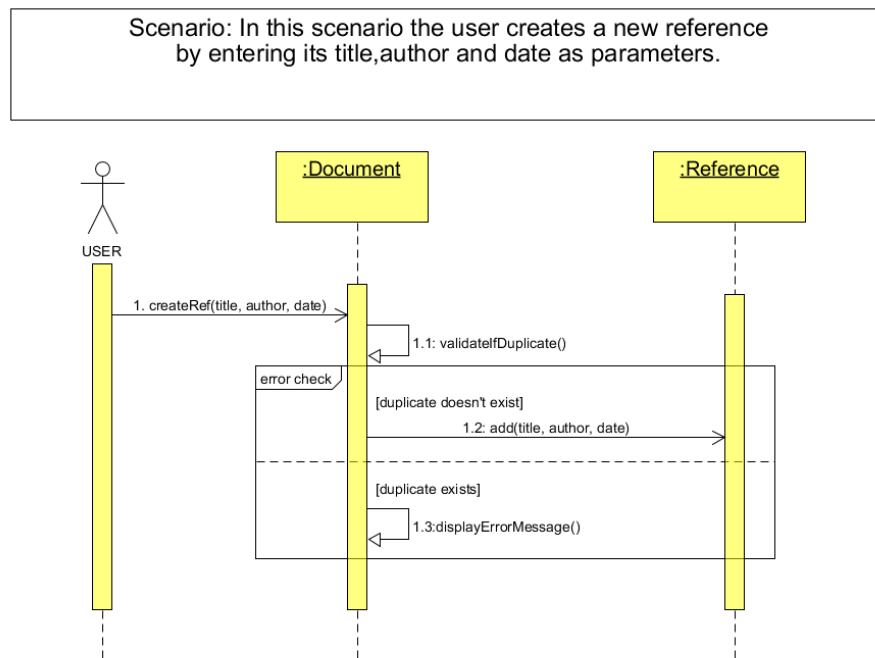

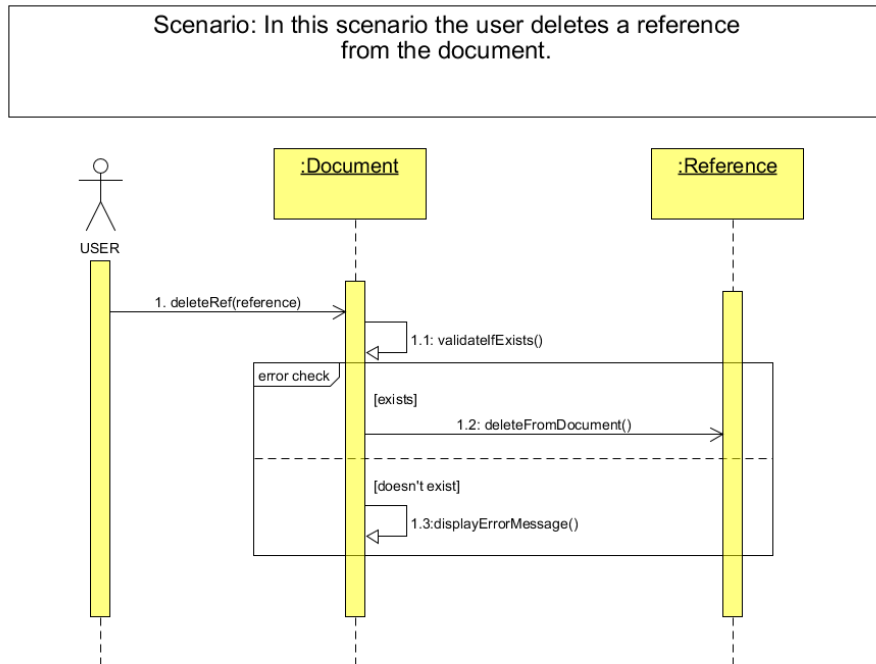Figure 9: Creating Reference Scenario Sequence Diagram

Scenario: In this scenario the user deletes a reference
from the document.

:Document

:Reference

USER

1. deleteRef(reference)

1.1: validateIfExists()

error check

[exists]

1.2: deleteFromDocument()

[doesn't exist]

1.3:displayErrorMessage()

Figure 10: Deleting Reference Scenario Sequence Diagram

Scenario: In this scenario the user edits a reference created formerly.

USER

:Document

:Reference

1. editRef()

1.1: validateIfExists()

error check

[exists]

1.2: edit()
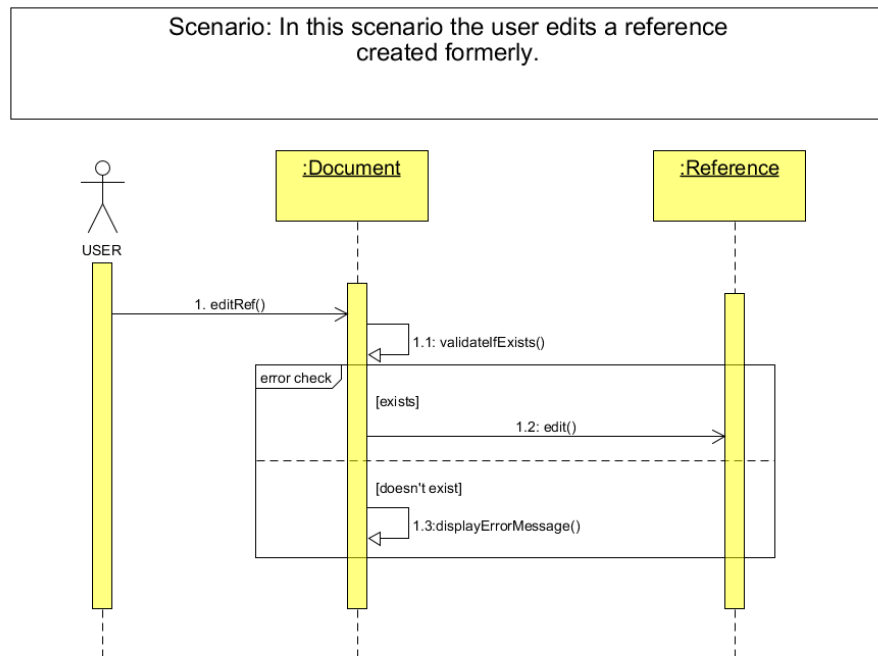
[doesn't exist]

1.3:displayErrorMessage()

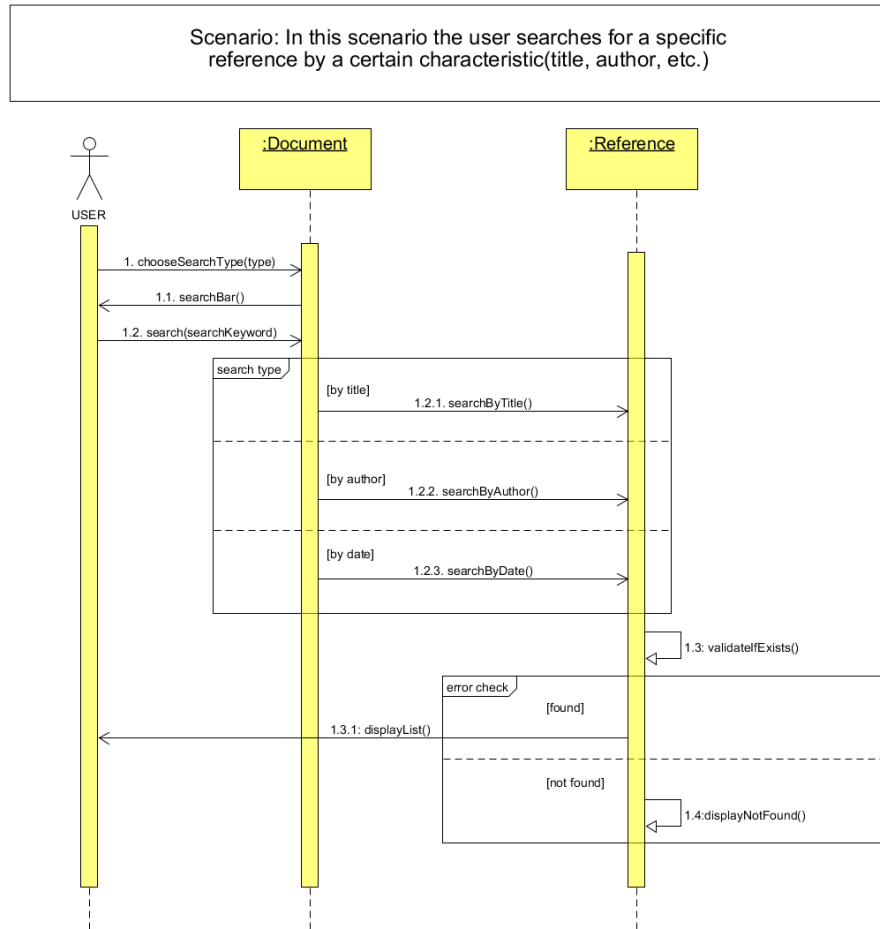Figure 11: Editing Reference Scenario Sequence Diagram

10

Figure 12: Searching Reference Scenario Sequence Diagram

2.4　UML Class Diagram

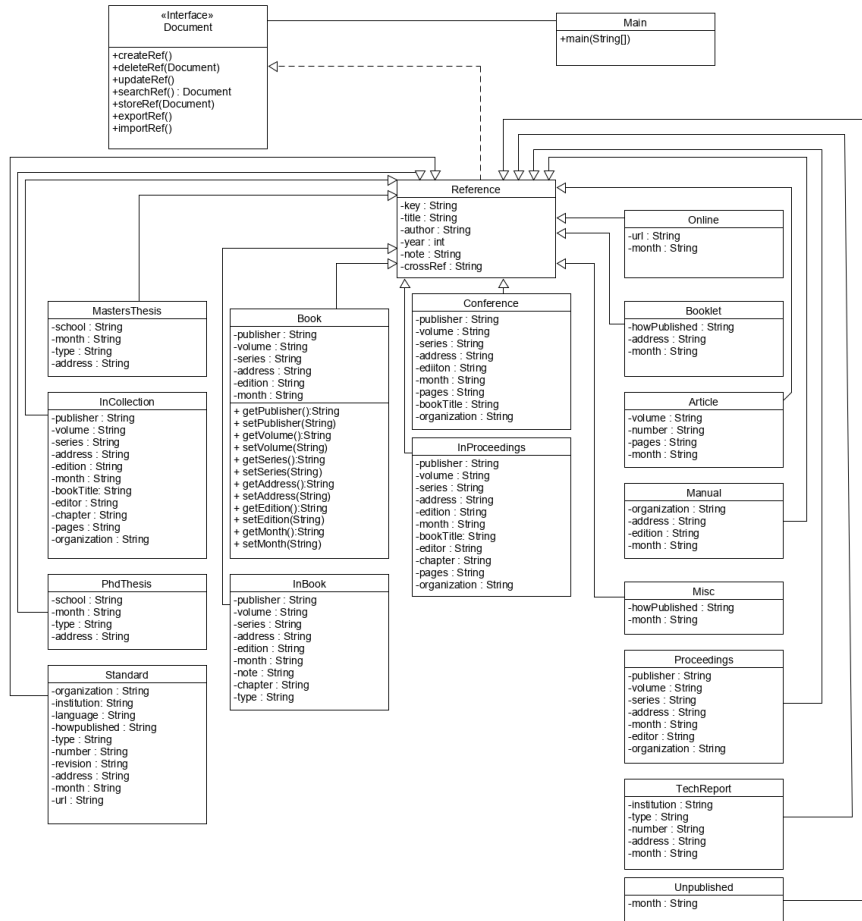Class Diagram represents structural organization which will be used during project implementation, by using UML Class Diagram.

Figure 13: Class Diagram of EasyRef