

## Leap Designs

As requested by QED Technologies:  
Eduvise

*Authors:*

Adam KWAN

Ahnaf AHMED

Austin SMITH

Darian SAMPARE

Hamzah MANSOUR

Jamie ST MARTIN

Sterling LAIRD

2019-03-28

## Revision History

2019-01-23 Requirements Document (RD) Version 0.1 - Leap Designs

- i. Document created
- ii. Initial features determined

2019-02-02 RD Version 0.9 - Leap Designs

- i. Features revised
- ii. Draft versions of requirements added

2019-02-05 RD Version 1.0 - Leap Designs

- i. Additional requirements added
- ii. Existing requirements revised
- iii. Context for features added
- iv. Submitted to QED Technologies and Colin Werner for review

2019-02-06 RD Version 1.1 - QED Technologies

- i. Comments from QED Technologies added

2019-02-24 RD Version 1.2 - Leap Designs

- i. Feedback from QED Technologies and Colin Werner incorporated
- ii. Requirements clarified and made unambiguous
- iii. Authentication feature and associated requirements added

2019-03-12 Requirements Specifications Document (RSD) Version 0.9 - Leap Designs

- i. Data dictionary, data flow diagrams, entity-relationship diagram, use case model, use cases, sequence diagrams, and UI models added
- ii. Demonstrated to QED Technologies for feedback

2019-03-19 RSD Version 1.0 - Leap Designs

- i. Separated use cases and sequence diagrams from system features
- ii. Updated user interface models based on feedback from QED Technologies
- iii. Improved clarity and consistency of use cases and requirements

- iv. Updated overview to reflect changes
- v. Submitted to QED Technologies and Colin Werner for review

2019-03-23 RSD Version 1.1 - QED Technologies

- i. Comments from QED Technologies added

2019-03-26 RSD Version 2.0 - Leap Designs

- i. Feedback from QED Technologies and Colin Werner incorporated
- ii. Updated entity-relationship diagram and data flow diagrams
- iii. Overhauled data dictionary
- iv. Revised use cases and UI models
- v. Added test cases, rationale and test cases to each requirement, and traceability matrix
- vi. Submitted to Colin Werner for review

# Contents

<b>Revision History</b>	<b>1</b>
<b>Table of Contents</b>	<b>3</b>
<b>1 Introduction</b>	<b>10</b>
1.1 Purpose . . . . .	10
1.2 Project Scope . . . . .	10
1.3 Overview . . . . .	10
<b>2 Overall Description</b>	<b>11</b>
2.1 Product Perspective . . . . .	11
2.2 Product Features . . . . .	11
2.3 User Classes and Characteristics . . . . .	12
2.3.1 Academic Advisor . . . . .	12
2.3.2 Student . . . . .	12
2.3.3 Referee . . . . .	12
2.4 Operating Environment . . . . .	12
2.5 Design and Implementation Constraints . . . . .	13
2.5.1 Storage . . . . .	13
2.5.2 External Application Interface . . . . .	13
2.5.3 Authentication System . . . . .	13
2.5.4 Information Security Policy . . . . .	13
2.5.5 Cost . . . . .	13
2.6 Assumption and Dependencies . . . . .	13
<b>3 System Features</b>	<b>13</b>
3.1 Searching . . . . .	14
3.1.1 Description and Priority . . . . .	14
3.1.2 Functional Requirements . . . . .	14
3.2 Adding Academic Advising Documents . . . . .	14
3.2.1 Description and Priority . . . . .	14
3.2.2 Functional Requirements . . . . .	14
3.3 Academic Advising Document Modification . . . . .	15
3.3.1 Description and Priority . . . . .	15
3.3.2 Functional Requirements . . . . .	15

3.4	Academic Advising Document Export . . . . .	16
3.4.1	Description and Priority . . . . .	16
3.4.2	Functional Requirements . . . . .	16
3.5	Academic Advising Document Viewing By Students . . . . .	17
3.5.1	Description and Priority . . . . .	17
3.5.2	Functional Requirements . . . . .	17
3.6	Academic Advising Document Viewing By Academic Advisors . . . . .	17
3.6.1	Description and Priority . . . . .	17
3.6.2	Functional Requirements . . . . .	17
3.7	Student Degree Planning . . . . .	18
3.7.1	Description and Priority . . . . .	18
3.7.2	Functional Requirements . . . . .	19
3.8	Authentication . . . . .	19
3.8.1	Description and Priority . . . . .	19
3.8.2	Functional Requirements . . . . .	20
<b>4</b>	<b>External Interface Requirements</b>	<b>20</b>
4.1	User Interfaces . . . . .	20
4.2	Hardware Interfaces . . . . .	20
4.3	Software Interfaces . . . . .	20
4.4	Communications Interfaces . . . . .	22
<b>5</b>	<b>Other Non-Functional Requirements</b>	<b>22</b>
5.1	Performance Requirements . . . . .	22
5.1.1	Capacity . . . . .	22
5.1.2	Speed . . . . .	22
5.2	Safety Requirements . . . . .	23
5.3	Security Requirements . . . . .	23
5.4	Software Quality Attributes . . . . .	24
5.4.1	Usability . . . . .	24
5.4.2	Availability . . . . .	24
5.4.3	Testability . . . . .	25
<b>6</b>	<b>Other Requirements</b>	<b>25</b>
<b>7</b>	<b>Diagrams</b>	<b>26</b>
7.1	Data Flow Diagrams . . . . .	26

7.1.1	Context Diagram . . . . .	27
7.1.2	DFD Level 1 . . . . .	28
7.1.3	DFD Level 2 Overview . . . . .	29
7.2	Data Dictionary . . . . .	35
7.3	Entity Relationship Diagram . . . . .	35
<b>8</b>	<b>Use Cases and User Interface Models</b>	<b>36</b>
8.1	Use Case Model . . . . .	36
8.2	UC-1.1: Searching . . . . .	38
8.3	UC-2.1: Adding Academic Advising Document . . . . .	42
8.4	UC-2.2: Requesting Referral . . . . .	46
8.5	UC-2.3: Submitting Referral . . . . .	49
8.6	UC-2.4: Approving Referral . . . . .	52
8.7	UC-3.1: Replacing Academic Advising Document . . . . .	56
8.8	UC-3.2: Adding Annotation . . . . .	59
8.9	UC-3.3: Modifying Annotation . . . . .	64
8.10	UC-3.4: Modifying Document Access . . . . .	68
8.11	UC-4.1: Document Exporting . . . . .	71
8.12	UC-5.1: Document Viewing . . . . .	74
8.13	UC-6.1: Getting Help . . . . .	77
8.14	UC-7.1: Creating Student Degree Map . . . . .	79
8.15	UC-7.2: Modifying Student Degree Map . . . . .	85
8.16	UC-8.1: Logging in . . . . .	89
<b>9</b>	<b>Forward Traceability</b>	<b>91</b>
9.1	Test Suite 1 - Searching . . . . .	92
9.2	Test Suite 2 - Adding Documents . . . . .	93
9.3	Test Suite 3 - Document Modification . . . . .	94
9.4	Test Suite 4 - Document Export . . . . .	95
9.5	Test Suite 5 - Document Viewing By Students . . . . .	96
9.6	Test Suite 6 - Document Viewing By Academic Advisors . . . . .	97
9.7	Test Suite 7 - Student Degree Planning . . . . .	97
9.8	Test Suite 8 - Authentication . . . . .	98
9.9	Test Suite 9 - External Interfaces . . . . .	99
9.10	Test Suite 10 - Non-Functional Requirements . . . . .	100
9.11	Test Suite 11 - Other . . . . .	102

## List of Tables

1	Glossary . . . . .	9
2	ERD Data Dictionary . . . . .	35
3	Use Case 1.1 - Searching . . . . .	38
4	Use Case 2.1 - Adding Academic Advising Document . . . . .	42
5	Use Case 2.2 - Requesting Referral . . . . .	46
6	Use Case 2.3 - Submitting Referral . . . . .	49
7	Use Case 2.4 - Approving Referral . . . . .	52
8	Use Case 3.1 - Replacing Academic Advising Document . . . . .	56
9	Use Case 3.2 - Adding Annotation . . . . .	59
10	Use Case 3.3 - Modifying Annotation . . . . .	64
11	Use Case 3.4 - Modifying Document Access . . . . .	68
12	Use Case 4.1 - Document Exporting . . . . .	71
13	Use Case 5.1 - Document Viewing . . . . .	74
14	Use Case 6.1 - Getting Help . . . . .	77
15	Use Case 7.1 - Creating Student Degree Map . . . . .	80
16	Use Case 7.2 - Modifying Student Degree Map . . . . .	85
17	Use Case 8.1 - Logging in . . . . .	89

## List of Figures

1	Data Flow Diagram Legend . . . . .	26
2	Context Diagram . . . . .	27
3	Data Flow Diagram Level 1 . . . . .	28
4	Data Flow Diagram Level 2 . . . . .	30
5	DFD Level 2: Top Left . . . . .	31
6	DFD Level 2: Top Right . . . . .	32

7	DFD Level 2: Bottom Left . . . . .	33
8	DFD Level 2: Bottom Right . . . . .	34
9	Entity Relationship Diagram . . . . .	36
10	Use Case Model . . . . .	37
11	Sequence Diagram 1.1 - Searching . . . . .	39
12	UI Model 1.1 - Searching . . . . .	41
13	Sequence Diagram 2.1 - Adding Academic Advising Document . . . . .	43
14	UI Model 2.1 - Adding Academic Advising Document . . . . .	45
15	Sequence Diagram 2.2 - Requesting Referral . . . . .	47
16	UI Model 2.2 - Requesting Referral . . . . .	48
17	Sequence Diagram 2.3 - Submitting Referral . . . . .	50
18	UI Model 2.3 - Submitting Referral . . . . .	51
19	Sequence Diagram 2.4 - Approving Referral . . . . .	53
20	UI Model 2.4 - Approving Referral . . . . .	55
21	Sequence Diagram 3.1 - Replacing Academic Advising Document . . . . .	57
22	UI Model 3.1 - Replace Academic Advising Document . . . . .	58
23	Sequence Diagram 3.2 - Adding Annotation . . . . .	60
24	UI Model 3.2 - Adding Annotation . . . . .	63
25	Sequence Diagram 3.3 - Modifying Annotation . . . . .	65
26	UI Model 3.3 - Modifying Annotation . . . . .	67
27	Sequence Diagram 3.4 - Modifying Document Access . . . . .	69
28	UI Model 3.4 - Modifying Document Access . . . . .	70
29	Sequence Diagram 4.1 - Document Exporting . . . . .	72
30	UI Model 4.1 - Document Exporting . . . . .	73
31	Sequence Diagram 5.1 - Document Viewing . . . . .	74
32	UI Model 5.1 - Document Viewing . . . . .	75
33	UI Model 5.1.2 - Document Viewing By Student . . . . .	76



34	Sequence Diagram 6.1 - Getting Help . . . . .	77
35	UI Model 6.1 - Getting Help . . . . .	79
36	Sequence Diagram 7.1 - Creating Student Degree Map . . . . .	81
37	UI Model 7.1 Creating Student Degree Map . . . . .	84
38	Sequence Diagram 7.2 - Modifying Student Degree Map . . . . .	86
39	UI Model 7.2 - Modifying Student Degree Map . . . . .	88
40	Sequence Diagram 8.1 - Logging In . . . . .	90
41	UI Model 8.1 Logging In . . . . .	91
42	Searching Test Matrix . . . . .	93
43	Adding Documents Test Matrix . . . . .	94
44	Document Modification Test Matrix . . . . .	95
45	Document Export Test Matrix . . . . .	96
46	Student Document Viewing Test Matrix . . . . .	96
47	Advisor Document Viewing Test Matrix . . . . .	97
48	Degree Planning Test Matrix . . . . .	98
49	Authentication Test Matrix . . . . .	99
50	External Requirements Test Matrix . . . . .	100
51	Non-Functional Requirements Test Matrix . . . . .	102
52	Other Requirements Test Matrix . . . . .	103

## Glossary of Terms

Academic advising document	Any file created by or uploaded to Eduvise, each of which pertains to and is associated with a student. This could be a referral document, a degree map, meeting notes, or another general document.
Academic advisor	An academic advisor working for the Faculty of Education at UVic.
Annotation	An additional comment on a document.
Course overview	Course information specifying a course's code, title, description, unit credit, and prerequisites.
Document	Any academic advising document.
Document category	One of "General Document", "Reference Letter", "Degree Map", or "Meeting Notes"
Matches	"Is identical to".
NetLink ID	A unique personal identifier which is used as a username for online services provided by UVic.
Professor profile	A read-only collection of references to professor record in the UVic internal records database.
Referee	A person who writes a reference letter for a student.
Reference letter	A letter of from a referee that assesses the capability of a student to be an educator.
Referral document	A reference letter. This is a subset of academic advising documents
Request For Proposal (RFP)	A document from QED Technologies requesting proposals for a new academic advising system for the Faculty of Education at UVic.
Student number	A unique numeric identifier given to each new student when enrolling at the University of Victoria in the form 'V' followed by 8 digits.
Student	A student enrolled in a program in the Faculty of Education at UVic.
Student degree map	An outline of classes arranged by semester for a student to take in order to graduate with the designed degree.
Student degree plan	A document that contains a degree plan designed by an academic advisor in Eduvise
Student profile	The collection of documents pertaining to a student and that student's student record.
Student record	The collection of entries pertaining to a student accessed through the UVic internal records database.
Student record entry	Any piece of information or file in a student's UVic internal record.
UVic internal records database	The existing database, managed by UVic, which contains every student record and information about every professor and course at UVic.

Table 1: Glossary

## 1 Introduction

### 1.1 Purpose

This requirements specifications document details the specifications of the requirements needed for QED Technologies' new academic advising system for the Faculty of Education at The University of Victoria. This project, titled "Eduvise", will be a software system that aims to increase the efficiency of academic advisors by providing them with a better system to store and manage student documents and records, and by providing easier access for students to see their own documents, records, and progress related to their degree. Eduvise aims to make the academic advising process of the Faculty of Education more efficient by removing the necessity of physical storage and manual retrieval of documents and records.

### 1.2 Project Scope

Eduvise's primary focus is to create a digital system for the viewing and manipulation of academic advising information for the Faculty of Education. The information includes students' correspondence with the Faculty of Education, documents and annotations added to Eduvise by academic advisors, and records from the University of Victoria (UVic) internal student records database.

A user of Eduvise will be in one of three classes: Student, Academic Advisor, or Referee. Each academic advisor will have full access to add, view, and modify documents and annotations pertaining to any student. Students will be able to view some of the information pertaining to them, including information about their academic progress. Referees will only submit reference letters for students. Eduvise will interact with its own database, UVic's internal student records database, and the UVic NetLink authentication server.

### 1.3 Overview

This document consists of seven main sections. This Section, "Introduction", has discussed the background to the report. Section 2, "Overall Description," describes Eduvise from a high level, including the product perspective, product features, user classes, operating environment, constraints on development, assumptions, and dependencies. Section 3, "System Features", outlines the core features that must be in Eduvise. This section includes a description of each feature, the level of priority of each feature, and the requirements associated with each of the system features. Section 4, "External Interface Requirements", describes logical characteristics between a proposed solution and the following four interfaces: user interface, hardware interface, software interface, and communication interface. Section 5, "Other Non-Functional Requirements", describes the performance requirements, safety requirements, security requirements, and software quality attributes that are necessary for the proposed solution. Section 6, "Other Requirements", describes all other requirements of a proposed solution not described in the rest of the requirement document. Section 7 "Diagrams", describes and shows the context diagram, data flow diagram level 1, data flow diagram level 2, data dictionary, and the entity relationship diagram. Section 8, "Use Cases", describes use

cases of Eduvise and provides a sequence diagram and user interface model for each described use case. Section 9, "Forward Traceability", describes test cases for each requirement and provides a traceability matrix for each feature.

## 2 Overall Description

### 2.1 Product Perspective

Eduvise is a software system that is currently being designed by Leap Designs for the Faculty of Education academic advising office at the University of Victoria. Eduvise will be a new and self-contained system that improves, augments, and partially automates the entirely manual academic advising process that is currently in place at the Faculty of Education academic advising office. Eduvise aims to increase academic advisor efficiency by allowing academic advisors to access student documents and records with ease. Additionally it aims to allow both students and academic advisors to view a student's academic progress and performance. Eduvise will store documents pertaining to a student's academic advising, and allow an academic advisor to view, modify, and append information to each student's profile and the contained student Eduvise documents.

### 2.2 Product Features

The features that Eduvise provides are:

i. Searching

The searching feature will allow an academic advisor to find UVic student profiles, professor information, and course information.

ii. Adding Academic Advising Documents

Eduvise will allow an academic advisor to upload academic advising documents to its database and will allow a referee to upload reference letters to its database.

iii. Academic Advising Document Modification

An academic advisor will be able to, through Eduvise, modify academic advising documents for their advising needs.

iv. Academic Advising Document Exporting

An academic advisor and students will have the ability to export academic advising documents through Eduvise.

v. Academic Advising Document Viewing by Students

A student will have access to view a limited selection of academic advising documents within the Eduvise database.

vi. Academic Advising Document Viewing by Academic Advisors

An academic advisor will have access to view every academic advising document stored in Eduvise's database and every student record stored in UVic's internal records database.

vii. Degree Planning

Eduvise will allow the use of academic advising documents and student records to assist students in planning their degree with an academic advisor.

viii. Authentication

Eduvise will authenticate users using UVic's NetLink credentials.

## **2.3 User Classes and Characteristics**

### **2.3.1 Academic Advisor**

The academic advisor user class consists of every academic advisor in the Faculty of Education. An academic advisor will use Eduvise to support their interactions with students. An academic advisor currently accesses student documents through a manual process which limits the academic advisor's efficiency. An academic advisor currently has no interaction with a software system for advising, but typically has some experience with other office-related software. An academic advisor will require some initial training to use Eduvise.

### **2.3.2 Student**

The student class consists of university students enrolled (currently or in the past three years) in any program in the Faculty of Education at UVic. A student will be able to track their degree progress and track goals set by an academic advisor. Currently, a student only has access to a physical transcript to track their degree progress. A student will be able to use Eduvise, including any features and capabilities available to them, without any previous knowledge, training, or experience related to Eduvise.

### **2.3.3 Referee**

The referee class consists of professionals overlooking the students' work experience. A referee will provide a reference letter for a student to an academic advisor by submitting the reference letter to Eduvise. A direct, secure submission method for the reference letter will prevent a student from viewing or altering the reference letter unless permitted to do so by an academic advisor.

## **2.4 Operating Environment**

Eduvise will operate to support the Faculty of Education academic advising office at UVic, provide a way for students to access their own degree progress and course performance, and allow referees to directly and securely submit documents to the academic advising office. Eduvise will operate on the existing hardware and operating systems in use by the Faculty of Education. The hardware Eduvise will operate on within the Faculty of Education is: 2009 20" iMacs, 2011 21.5" iMacs, 2011 27"

iMacs, and 2013 27" iMacs. The operating systems Eduvise will operate on in use in the faculty of education are: Windows 7, Windows 10, OS X 10.4 and above, and macOS 10.12 and above. Eduvise will need to work simultaneously with access to the UVic NetLink Authentication server and UVic Internal Records Database.

## **2.5 Design and Implementation Constraints**

### **2.5.1 Storage**

Student profiles and documents must be accessible for at least 3 years after a student has either graduated or last been enrolled in a Faculty of Education program.

### **2.5.2 External Application Interface**

Continuous access to UVic's internal student records database for student records is required. Continuous access to UVic's NetLink authentication server is required.

### **2.5.3 Authentication System**

Access to Eduvise must be authenticated through the use of a user's NetLink credentials.

### **2.5.4 Information Security Policy**

UVic has information security and privacy policies [1] to protect its students' information, which must be followed. These policies are described in detail in section 5.3 Security Requirements.

### **2.5.5 Cost**

The development of Eduvise must be completed within a \$30,000 budget.

## **2.6 Assumption and Dependencies**

Database Connection: Eduvise depends on access to UVic's internal records database for access to student records and professor and course information. Eduvise depends on access to UVic's NetLink authentication server for authentication.

Current Records Transfer: Every physical academic advising document currently stored by an academic advisor will need to be uploaded to the new Eduvise database.

## **3 System Features**

In section three, any reference to a student or an academic advisor pertains exclusively to a student or an academic advisor within the Faculty of Education at UVic. For further information on the roles of a student or an academic advisor in the report, refer to section 2.3 User Classes and Characteristics.

## 3.1 Searching

### 3.1.1 Description and Priority

The searching feature should be effective at finding a specific student profile, professor profile, or course overview stored by Eduvise or the UVic internal records database. This feature should allow an academic advisor to locate information using a variety of keywords and phrases that are related to the student profile, professor profile, or course overview they are trying to find. A search will return every student profile, professor profile, and course overview that pertains to the search query.

Priority: High

### 3.1.2 Functional Requirements

**REQ-1.1:** An academic advisor must be able to search for each student profile by name, NetLink ID, or student number.

Rationale: Academic advisors work with many students in the Faculty of Education, and require some way to easily find documents related to a student for reference or for an academic advising meeting. This business need was established in the RFP.

Test Cases: TC-1.1, TC-1.2, TC-1.3

**REQ-1.2:** An academic advisor must be able to search for each professor profile by name, faculty, or courses taught.

Rationale: Students take a wide variety of classes, and academic advisors need some way to sort through the large amount of courses offered by UVic. This business need was established in the RFP.

Test Cases: TC-1.4, TC-1.5, TC-1.6

**REQ-1.3:** An academic advisor must be able to search for each course overview by course code or department.

Rationale: When assisting students in planning their degree, academic advisors need to be able to parse through long lists of courses to help students pick appropriate courses. This business need was established in the RFP and expanded upon in a discussion with the clients on March 11, 2019.

Test Cases: TC-1.7, TC-1.8

## 3.2 Adding Academic Advising Documents

### 3.2.1 Description and Priority

The adding academic advising documents feature will allow an academic advisor to upload new academic advising documents pertaining to a student. This feature will also allow a referee to securely submit a reference letter for a student at the request of an academic advisor.

Priority: High

### 3.2.2 Functional Requirements

**REQ-2.1:** An academic advisor must be able to upload one or more academic advising documents pertaining to a student.

Rationale: Academic advisors currently store physical documents that inform their academic advising of students. These existing documents will need to be added to Eduvise. Also, new academic advising documents created in the future will similarly need to be added to Eduvise. These business needs were established in the RFP.

Test Cases: TC-2.1

**REQ-2.2:** A referee must be able to submit a reference letter for a student. (see EIR-4.1)

Rationale: A referee oversees a student during their work experience and can provide a reference letter that must be stored by Eduvise. This business need was established in the RFP.

Test Cases: TC-2.2

**REQ-2.3:** When uploading one or more academic advising documents pertaining to a student (see REQ-2.1), an academic advisor must be able to choose whether or not the student will have access to view each uploaded document.

Rationale: An academic advisor should be able to control the visibility of documents containing any sensitive information on a student's profile that they deem necessary for another academic advisor to view, but not for a student to view. This business need was established in the initial client meeting and was expanded on in the second client meeting.

Test Cases: TC-2.3

### **3.3 Academic Advising Document Modification**

#### **3.3.1 Description and Priority**

The academic advising document modification feature will allow an academic advisor to update and annotate existing academic advising documents. An academic advisor can replace a document with an updated version or add annotations to a document. These changes come about as the result of the academic advisor meeting with a student, acquiring new information, or communication with another academic advisor.

Priority: High

#### **3.3.2 Functional Requirements**

**REQ-3.1:** An academic advisor must be able to replace an existing academic advising document with a newer version.

Rationale: A student's path through university involves a lot of changes. An academic advisor needs a method to replace an old document with another one. This business need was established in the RFP.

Test Cases: TC-3.1

**REQ-3.2:** An academic advisor must be able to add a new annotation on an academic advising document.

Rationale: Each academic advisor needs to be able to make comments pertaining to specific documents. This business need was established in the RFP.

Test Cases: TC-3.2



**REQ-3.3:** An academic advisor must not be able to modify any annotation on an academic advising document while any other academic advisor is modifying any annotation on the same academic advising document.

Rationale: Multiple academic advisors editing annotations on the same document concurrently could result in contradictory changes. This must be prevented to ensure correct annotation functionality. This business need was established in the initial client meeting.

Test Cases: TC-3.3

**REQ-3.4:** An academic advisor must be able to modify existing annotations on an academic advising document.

Rationale: An academic advisor needs to be able to change or extend any comments made on a document to keep up with new information. This business need was established in the RFP and expanded upon in the initial client meeting.

Test Cases: TC-3.4

**REQ-3.5:** An academic advisor must be able to modify access to an existing academic advising document from the student to which it pertains. (see REQ-2.3)

Rationale: Some academic advising documents, such as reference letters, are meant to be viewed by academic advisors only. Other academic advising documents, such as student transcripts, are meant to be viewed by both academic advisors and the student to which the document pertains. In some cases, an academic advisor may need to grant access or revoke access to the document from the student. This business need was established in the initial client meeting.

Test Cases: TC-3.5, TC-3.6

## **3.4 Academic Advising Document Export**

### **3.4.1 Description and Priority**

The academic advising document export feature will allow a student or an academic advisor to export chosen documents that they have access to in a printable or readable format.

Priority: Medium

### **3.4.2 Functional Requirements**

**REQ-4.1:** An academic advisor must be able to export one or more documents pertaining to a student in a single easily printable and readable file.

Rationale: An academic advisor frequently needs to provide academic advising documents to an external entity and as such will need a way to export those documents from Eduvise. This business need was established in the RFP.

Test Cases: TC-4.1

**REQ-4.2:** A student must be able to export one or more documents that they have access to view in a single easily printable and readable file.

Rationale: A student will at times need to provide their academic advising documents to an external entity and as such will need a way to export those documents from Eduvise. This business need was

established in the second client meeting.

Test Cases: TC-4.2

### **3.5 Academic Advising Document Viewing By Students**

#### **3.5.1 Description and Priority**

A student will need access to view only documents that they have been granted access to view (see REQ-2.3, REQ-3.5). Students should also be able to check on their current degree status and progress towards graduation via Eduvise.

Priority: Low

#### **3.5.2 Functional Requirements**

**REQ-5.1:** A student must be able to view their degree plan, including specific goals for the student that have been set with an academic advisor.

Rationale: A student should be able to see a course plan an advisor has designed for them. This business need was established in the RFP and refined in the second client meeting.

Test Cases: TC-5.1

**REQ-5.2:** A student must be able to view only the documents which they have been granted access to view. (see REQ-2.3, REQ-3.5)

Rationale: Some academic advising documents, such as reference letters, are meant to be viewed by academic advisors only. Other academic advising documents, such as degree plans, are meant to be viewed by an academic advisor or the student to which the document pertains. This business need was established in the initial client meeting.

Test Cases: TC-5.2

**REQ-5.3:** A student must not be able to view any document pertaining to a different student.

Rationale: Academic advising documents are confidential and should only be available to academic advisors and the student to which they pertain. This business need was established in the RFP.

Test Cases: TC-5.3

### **3.6 Academic Advising Document Viewing By Academic Advisors**

#### **3.6.1 Description and Priority**

An academic advisor should be able to view each academic advising document in Eduvise.

Priority: High

#### **3.6.2 Functional Requirements**

**REQ-6.1:** An academic advisor must be able to view each student's progress towards graduation.

Rationale: A critical part of the academic advising process is determining what a student needs to do to graduate from the Faculty of Education. This business need was established in the RFP.

Test Cases: TC-6.1

**REQ-6.2:** An academic advisor must be able to view each annotation added to each document by any academic advisor (see REQ-3.2).

Rationale: Multiple academic advisors may advise the same student at different times. Each academic advisor may wish to leave comments on the academic advising documents of the student. Each comment will be necessary for an academic advisor to appropriately advise the student in the future. This business need was established in the RFP and confirmed in RD1.1.

Test Cases: TC-6.2

**REQ-6.3:** An academic advisor must be able to view each academic advising document in Eduvise on each student profile.

Rationale: Academic advisors need to be able to see important documents pertaining to each student to improve advising of the student. This business need was established in the RFP.

Test Cases: TC-6.3

### **3.7 Student Degree Planning**

#### **3.7.1 Description and Priority**

An academic advisor can assist a student by helping to set academic goals for the student. These academic goals often take the form of notes that outline a set of criteria which the student is aiming to complete. An academic advisor can also help a student by planning out a degree schedule on a template calendar. The degree plan will show a student which courses should be taken each term to complete their degree.

Priority: Low

### 3.7.2 Functional Requirements

**REQ-7.1:** An academic advisor must be able to view specific goals for each student.

Rationale: When academic advisors meet with students, a common task is creating academic goals for students. These academic goals are kept in the student's file and are used in subsequent meetings to assess the student's progress towards the goal. Eduvise must contain the ability to create and view academic goals for students. This business need was established in the RFP, initial client meeting, and was clarified in the second client meeting.

Test Cases: TC-7.1

**REQ-7.2:** An academic advisor must be able to add specific goals for each student.

Rationale: As students progress through their degrees, academic advisors will often want to add new academic goals of a student to reflect new goals that the student has established. This business need was established in RD1.1.

Test Cases: TC-7.2

**REQ-7.3:** An academic advisor must be able to edit specific goals for each student.

Rationale: As students progress through their degrees, academic advisors will often want to update the academic goals of a student to reflect a student's changing academic priorities. This business need was established in RD1.1.

Test Cases: TC-7.3

**REQ-7.4:** An academic advisor must be able to create a personalized degree plan for each student.

Rationale: A student's courses is one of the main reasons they would go to an academic advisor. In order to assist students with this properly, academic advisors should be given support from Eduvise in creating a plan for students. These degree plans contain any classes the student may take. This business need was established in the RFP and refined in the second client meeting.

Test Cases: TC-7.4

**REQ-7.5:** An academic advisor must be able to modify any student's existing degree plan.

Rationale: A degree plan should not be set in stone, as students are free to take any courses that are available to them and these may not match up to an academic advisor's plan. They are also able to take a break and take courses at a later date if they feel overwhelmed. To account for this and any other changes, an academic advisor must be able to alter a degree plan at any time. This business need was established in the RFP and refined in the second client meeting.

Test Cases: TC-7.5

## 3.8 Authentication

### 3.8.1 Description and Priority

The authentication feature will verify a student or an academic advisor's identity before granting them access to Eduvise. The authentication feature will prevent others from accessing Eduvise.

Priority: High

### 3.8.2 Functional Requirements

**REQ-8.1:** A student or academic advisor must authenticate their identity by providing their UVic NetLink credentials.

Rationale: The information stored by Eduvise and accessible to a student or an academic advisor is confidential. Consequently, Eduvise must first authenticate that the user is who they claim to be. For consistency with other UVic services, authentication should be performed based on UVic NetLink IDs. This business need was established in the RFP.

Test Cases: TC-8.1, TC-8.2, TC-8.3

## 4 External Interface Requirements

### 4.1 User Interfaces

**EIR-1.1:** A user must always have access to help information and resources required to operate Eduvise.

Rationale: Users of Eduvise will not necessarily know what functionality is available or how to access it. Having help available to the user at all times will be necessary to provide them with that information. This business need was established in the first client meeting and confirmed in RD1.1.

Test Cases: TC-9.1

### 4.2 Hardware Interfaces

**EIR-2.1:** Eduvise must be accessible on any device required by the Faculty of Education's academic advising office. These devices are the following:

- 2009 20" iMacs
- 2011 21.5" iMacs
- 2011 27" iMacs
- 2013 27" iMacs

Rationale: Eduvise will primarily be used by academic advisors in the Faculty of Education's academic advising office, which utilizes only the hardware listed above. This business need was established in the initial client meeting and in RD1.1.

Test Cases: TC-9.2

### 4.3 Software Interfaces

**EIR-3.1:** Eduvise must have access to UVic's NetLink authentication server for authentication.

Rationale: In order to authenticate consistently with other UVic services, Eduvise will authenticate with UVic NetLink IDs. To do this, Eduvise must access the NetLink authentication server. This business need was established in RD1.1.

Test Cases: TC-9.3

**EIR-3.2:** Eduvise must be able to access the following information about each student from the UVic internal records database:

- Name
- Student number
- Contact information (home address, email address, and phone number)
- UVic academic transcript
- Academic transcripts from other institutions
- Medical records

Rationale: In order to adequately advise a student, an academic advisor must have access to the student's academic history, medical history as it pertains to their education, and other basic student records. This business need was established in the initial client meeting and RD1.1.

Test Cases: TC-9.7

**EIR-3.3:** Eduvise must be accessible on the following operating systems:

- Windows 7
- Windows 10
- OS X 10.4 and above
- macOS 10.12 and above

Rationale: Eduvise will primarily be used by academic advisors in the Faculty of Education's academic advising office, which utilizes only the operating systems listed above. This business need was established in the initial client meeting and in RD1.1.

Test Cases: TC-9.4

**EIR-3.4:** Eduvise must support the importing, viewing, and exporting of documents with each of the following file types:

- .doc
- .docx
- .odt
- .txt
- .pdf

Rationale: Eduvise will need to store primarily text-based documents, and will infrequently need to store image-based documents. The formats listed are sufficient for these needs. This business need was established in the initial client meeting and the details were confirmed in RD1.1.

Test Cases: TC-9.5

## 4.4 Communications Interfaces

**EIR-4.1:** Eduvise must support interfacing with an SMTP server to send emails to referees for submission of reference letters. (see REQ-2.2)

Rationale: The main method of communication between academic advisors and referees is via email. Since there cannot be a guarantee that referees will have access to other forms of communication, Eduvise must be able to support an email service to communicate with the referees. This business need was established in the initial client meeting.

Test Cases: TC-9.6

## 5 Other Non-Functional Requirements

### 5.1 Performance Requirements

#### 5.1.1 Capacity

At any given time, UVic has approximately 4000 students [2] who are currently enrolled in or have at some point in the past three years been enrolled in the Faculty of Education.

**NFR-1.1:** Eduvise must support storage of every academic advising document pertaining to each student in the Faculty of Education.

Rationale: To facilitate functionality such as uploading of documents or modifying annotations, Eduvise must be able to store the student documents so these actions can be executed. This must be true for all students in the Faculty of Education. This business need was established in the initial client meeting.

Test Cases: TC-10.1

**NFR-1.2:** Eduvise must support concurrent access from at least 200 users.

Rationale: Each academic advisor may be advising a student using Eduvise at once and many students may also be using Eduvise concurrently. There are approximately 4000 students [2] that may have documents in Eduvise at any one time, and each of them may wish to access their documents concurrently. The precise number of 200 concurrent users, was agreed upon in the second client meeting.

Test Cases: TC-10.2, TC-10.3, TC-10.4

#### 5.1.2 Speed

**NFR-1.3:** A request for authentication of any user must be approved or declined within three seconds.

Rationale: A user will wish to have feedback soon after making an authentication request to know that Eduvise has received their request. This business need was confirmed in RD1.1.

Test Cases: TC-10.2

**NFR-1.4:** A request to view any academic advising document must be given a response within three seconds.

Rationale: A user will wish to have feedback soon after making a request to view an academic

advising document to know that Eduvise has received their request. This business need was confirmed in RD1.1.

Test Cases: TC-10.3

**NFR-1.5:** Changes made more than 10 seconds before an academic advising document is accessed must be reflected at the time of access.

Rationale: When an academic advisor annotates or replaces an academic advising document, another academic advisor or a student may wish to view the changes soon afterwards. This business need was confirmed in RD1.1.

Test Cases: TC-10.4

## 5.2 Safety Requirements

**NFR-2.1:** Eduvise must backup all records and documents for data restoration in case of data loss or corruption.

Rationale: With the current physical system, if any documents are lost, there is no way to retrieve this data. Eduvise must be able to assure that if any documents are lost, then the documents will be able to be retrieved. This business need was established in the initial client meeting.

Test Cases: TC-10.5

## 5.3 Security Requirements

**NFR-3.1:** Eduvise must comply to BC's Freedom of Information and Protection of Privacy Act (FIPPA). [3]

Rationale: UVic must follow specific rules and regulations established by the government of British Columbia. Since Eduvise is being developed for UVic, it must also follow regulations. This business need was established in the first client meeting.

Test Cases: TC-10.6

**NFR-3.2:** Eduvise must comply to UVic's Protection of Privacy Policy. [4]

Rationale: UVic has defined guidelines for all applications that have access to UVic secured data. Eduvise must comply with all of these policies. This business need was established in the initial client meeting.

Test Cases: TC-10.7

**NFR-3.3:** Eduvise must comply to UVic's Records Management Policy. [5]

Rationale: UVic has defined guidelines for all applications that have access to UVic secured data. Eduvise must comply with all of these policies. This business need was established in the initial client meeting.

Test Cases: TC-10.8

**NFR-3.4:** No software system that is not Eduvise may have access to any academic advising document stored by Eduvise.

Rationale: Due to the potentially sensitive nature of the information that is available to the academic advisors about students, it is necessary that this information is accessible to exclusively the academic advisors. This business need was established in the initial client meeting.



Test Cases: TC-10.14

## 5.4 Software Quality Attributes

### 5.4.1 Usability

Eduvise must be easy and intuitive to use, so it can be used by both technical and non-technical personnel.

**NFR-4.1:** An academic advisor must require no more than two hours of initial training in the use of Eduvise from technical experts before they are able to correctly access, modify, and export any academic advising document.

Rationale: While most academic advisors will be proficient with most office software, Eduvise must not have such a high learning curve that it requires a long period of time for academic advisors to adjust to it. This business need was established in the initial client meeting and expanded upon in RD1.1.

Test Cases: TC-10.9

**NFR-4.2:** An academic advisor must require no more than two hours of support per month in the use of Eduvise from technical experts.

Rationale: The current physical system requires no maintenance time and no resources to keep it functional. Eduvise should minimize the amount of support required to keep it functional so no new resources will need to be allocated to it. This business need was established in the RFP and initial client meeting.

Test Cases: TC-10.10

**NFR-4.3:** Every student must require no training in the use of Eduvise from technical experts before they are able to correctly access and export any academic advising document that they have access to view.

Rationale: As there can be around 4000 students that have documents in Eduvise, it would be exceedingly impractical to provide training for each student. Since the resources don't exist to train each student, Eduvise must be accessible enough for a student to use without training. This business need was confirmed in RD1.1.

Test Cases: TC-10.11

### 5.4.2 Availability

**NFR-4.4:** An academic advisor must be guaranteed to have access to Eduvise on each weekday (Monday, Tuesday, Wednesday, Thursday, and Friday) from 7:00 am to 6:00 pm PST, with 99.9% uptime during these times.

Rationale: Academic advisors currently use the physical documents frequently throughout the entire day on each weekday. If the documents were unavailable during this time, academic advisors would have large delays in their work and certain meetings could be delayed. Eduvise would need high reliability to assure the academic advisors can execute their work uninterrupted. This business need was confirmed in RD1.1.

Test Cases: TC-10.12

### 5.4.3 Testability

**NFR-4.5:** Unit test coverage of Eduvise must be at least 70%.

Rationale: Eduvise will need to reach a level of software quality sufficient for a system used daily by multiple people. Eduvise will also likely have its features modified or expanded upon after implementation. Eduvise must be covered by unit tests that to ensure that changes to the software preserve its quality. This business need was confirmed in RD1.1.

Test Cases: TC-10.13

## 6 Other Requirements

**OREQ-1:** Each academic advising document must be accessible for a minimum of 3 years from the last date that the student the document pertains to is enrolled in any program at UVic.

Rationale: A student or academic advisor may need to access an academic advising document after leaving UVic. The business need was established in the RFP that each document must be stored for three years after graduation.

Test Cases: TC-11.1

**OREQ-2:** A student must be able to view each academic advising document accessible to that student without needing to be within the UVic telecommunication network.

Rationale: A student may wish to view their academic advising documents when they are at home, at work, or otherwise not on campus, so Eduvise needs to be available to them from outside the UVic network. This business need was established in the initial client meeting.

Test Cases: TC-11.2

## 7 Diagrams

### 7.1 Data Flow Diagrams

The data flow diagrams (DFDs) outline the flow of data into and out of Eduvise. The context diagram is from a high level perspective showing the main entities and what data they send and receive from Eduvise. The DFD level 1 shows the data flow between entities and the system processes within Eduvise. The DFD level 2 shows the data flowing between the different steps of the processes and the entities interacting with the system. The following legend is used for the context diagram, data flow diagram (DFD) level 1, and DFD level 2.

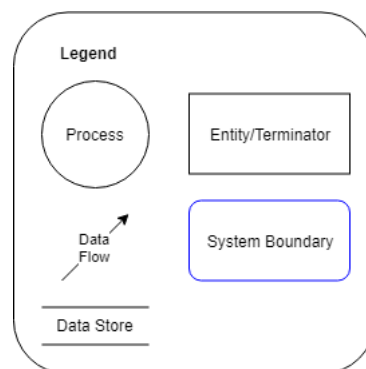


Figure 1: Data Flow Diagram Legend

### 7.1.1 Context Diagram

The context diagram provides a high-level overview of the inputs and outputs of Eduvise.

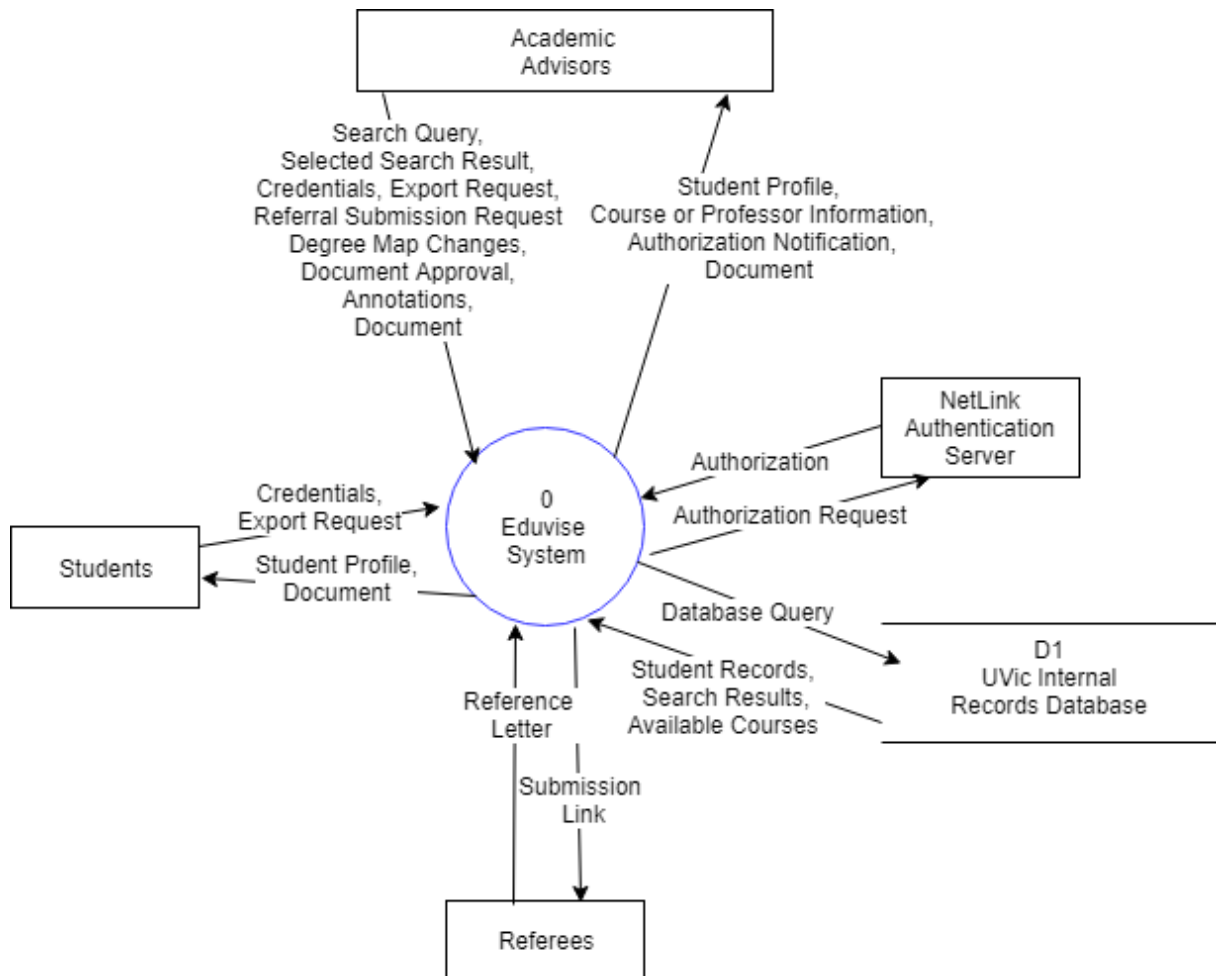


Figure 2: Context Diagram

### 7.1.2 DFD Level 1

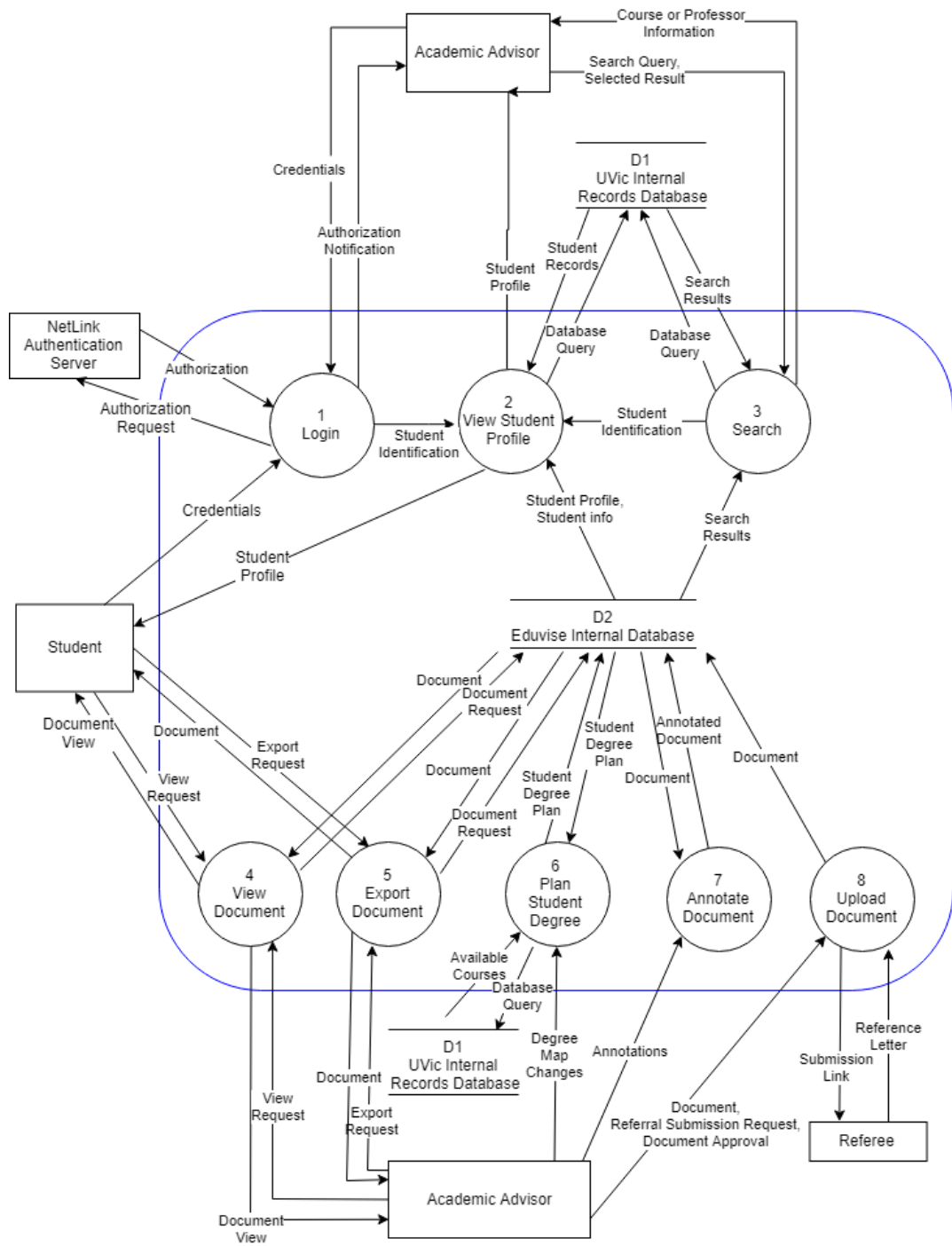


Figure 3: Data Flow Diagram Level 1

### 7.1.3 DFD Level 2 Overview

This is the overview of DFD level 2, The red lines indicate the split of the four zoomed in quadrants on the following four pages in the following order: top left, top right, bottom left, bottom right.

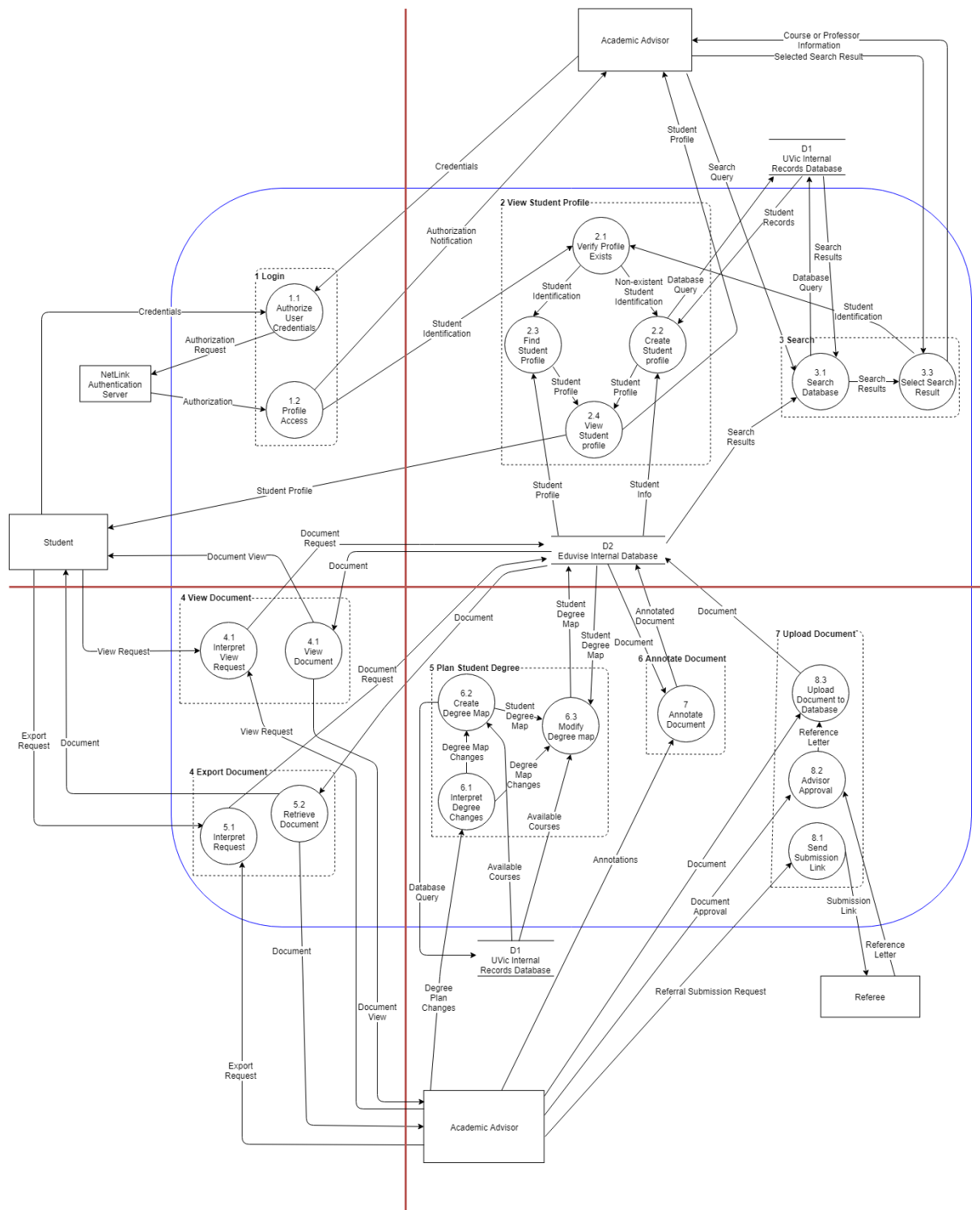


Figure 4: Data Flow Diagram Level 2

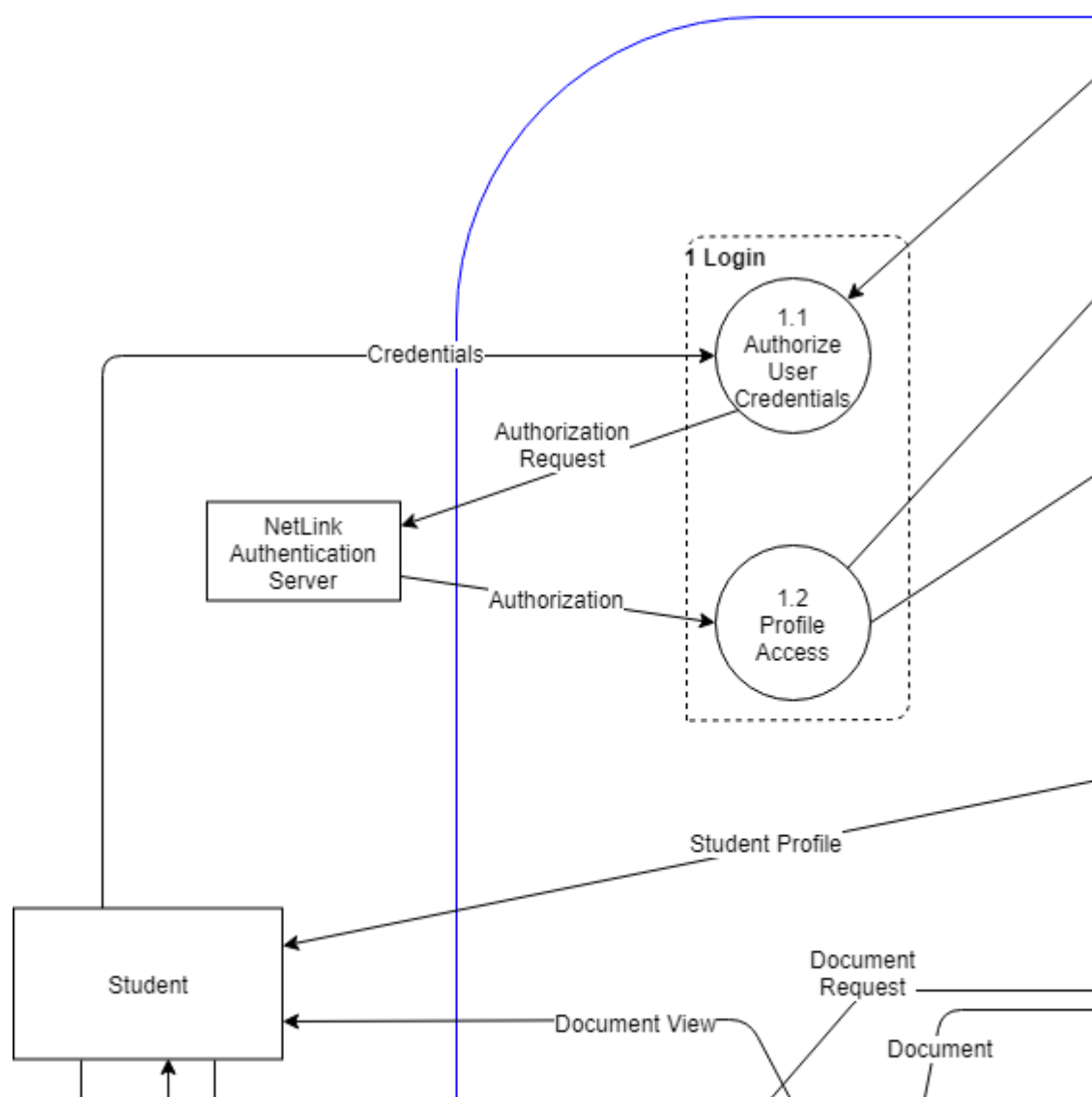


Figure 5: DFD Level 2: Top Left



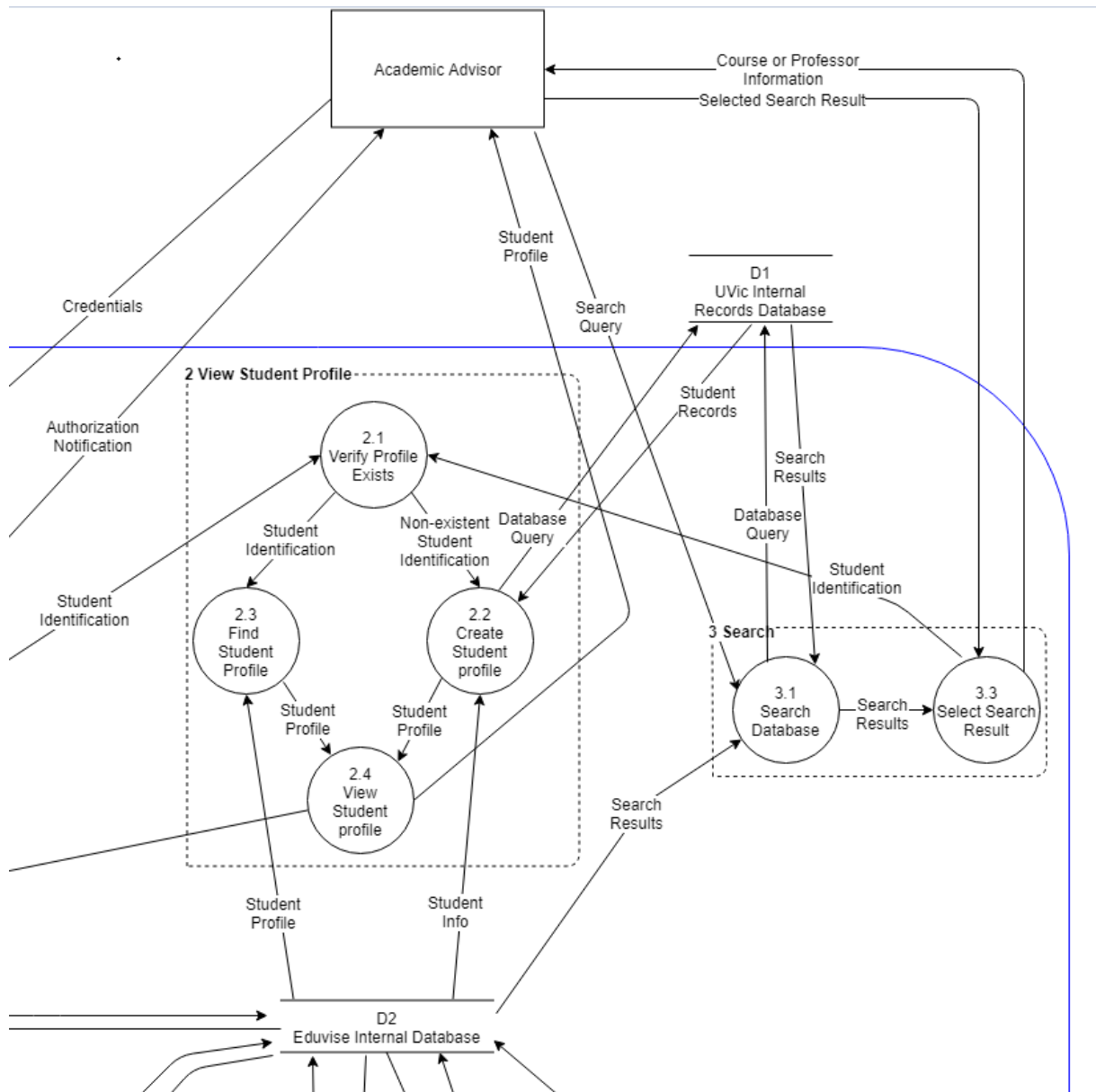


Figure 6: DFD Level 2: Top Right

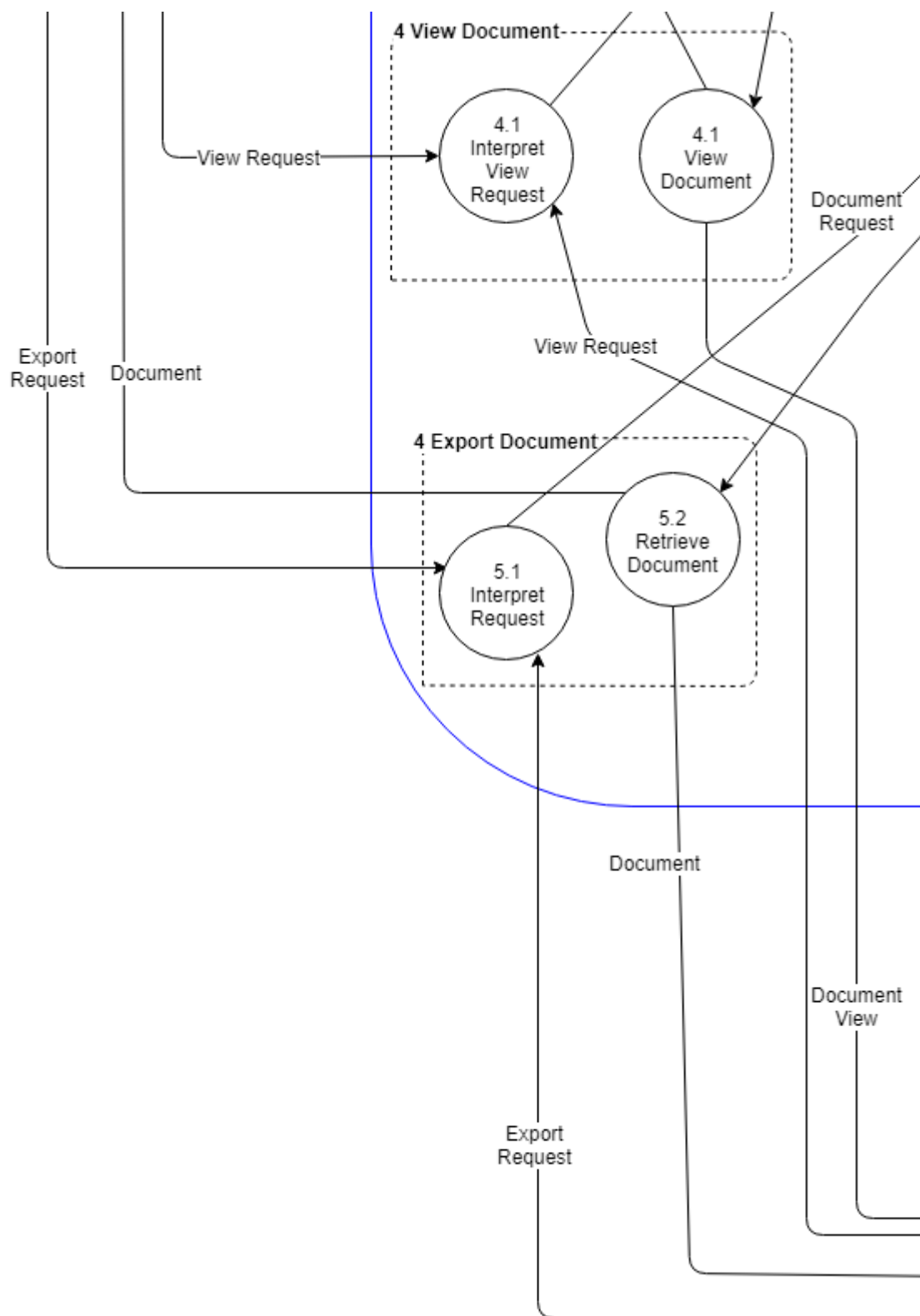


Figure 7: DFD Level 2: Bottom Left

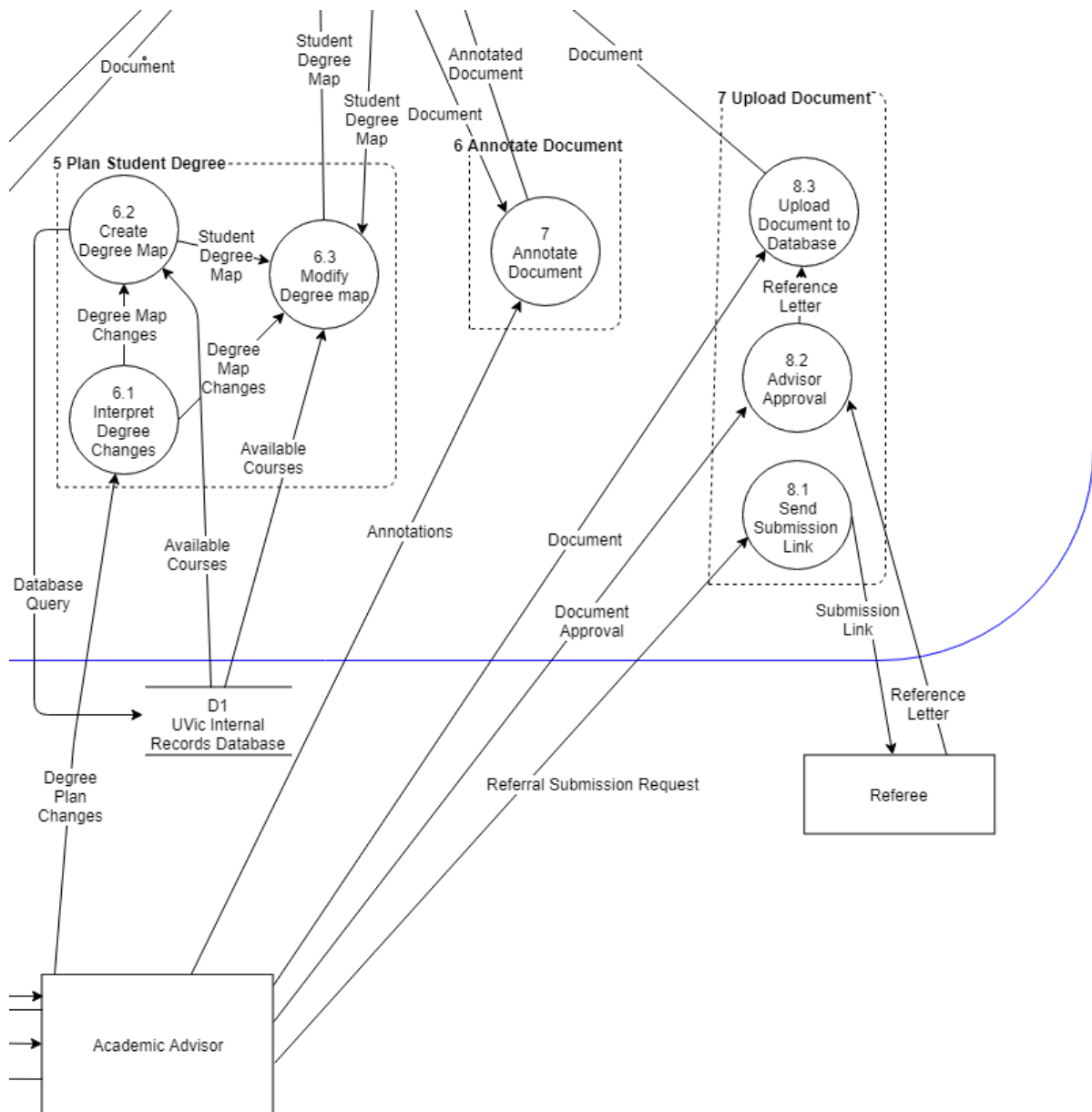


Figure 8: DFD Level 2: Bottom Right

## 7.2 Data Dictionary

Entity	ID	Attributes
Advisor-class	1	advisor-name:string advisor-netlink-id:string
Annotations	2	annotation-id:int
Degree Plan	3	degree-plan-id:int
Document	4	document-id:int
Reference Letter	5	reference-letter-id:int
Student-class	6	student-name:string student-netlink-id:string
Student Profile	7	student-profile-id:int
Student Record	8	student-record-id:int

Table 2: ERD Data Dictionary

## 7.3 Entity Relationship Diagram

The Entity Relationship Diagram shows the connections between the different entities involved in Eduvise.

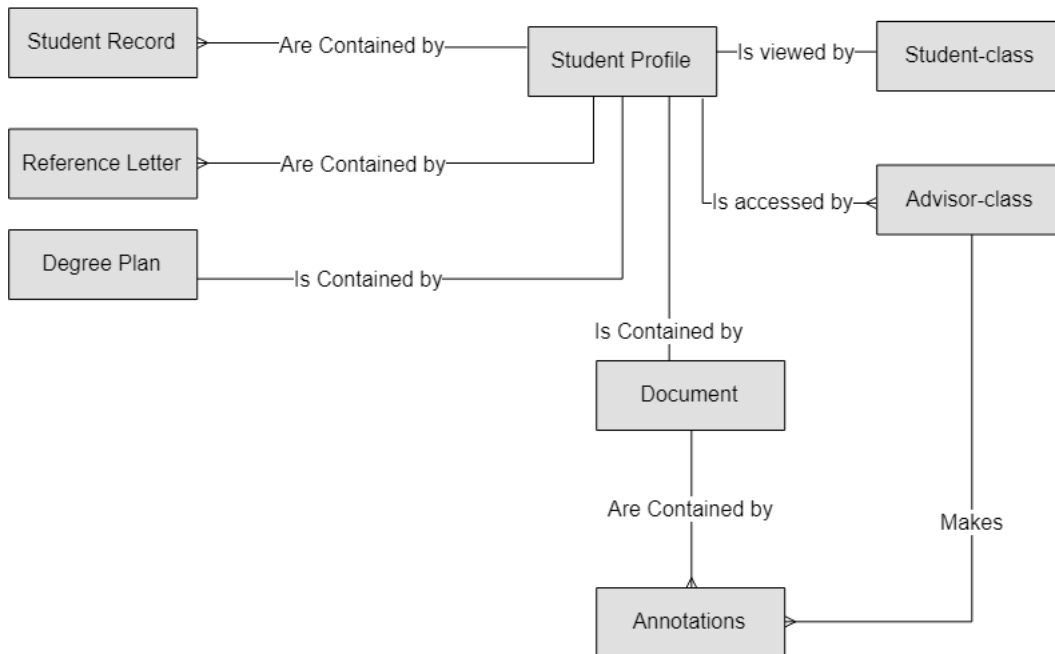


Figure 9: Entity Relationship Diagram

## 8 Use Cases and User Interface Models

This section presents the main use cases of Eduvise, as well as sequence diagrams and user interface models of sample scenarios of each use case. The "Logging In" use case (UC-8.1) is included after all of the main use cases are listed, in Section 8.16. All use cases which indicate that the user is logged in as a precondition include the "Logging In" use case.

### 8.1 Use Case Model

This use case model outlines all the use cases identified by Leap Designs for the three user classes of Eduvise. In the following sections, all the individual use cases will be analyzed in depth.

To simplify the use case model, Leap Designs omitted "includes" relationships between all use cases and the "Log In" use case with the exception of the "Submit Referral" use case. However, within our use cases there still exists the assumption that all the specified use cases do include the "Logging In" use case.

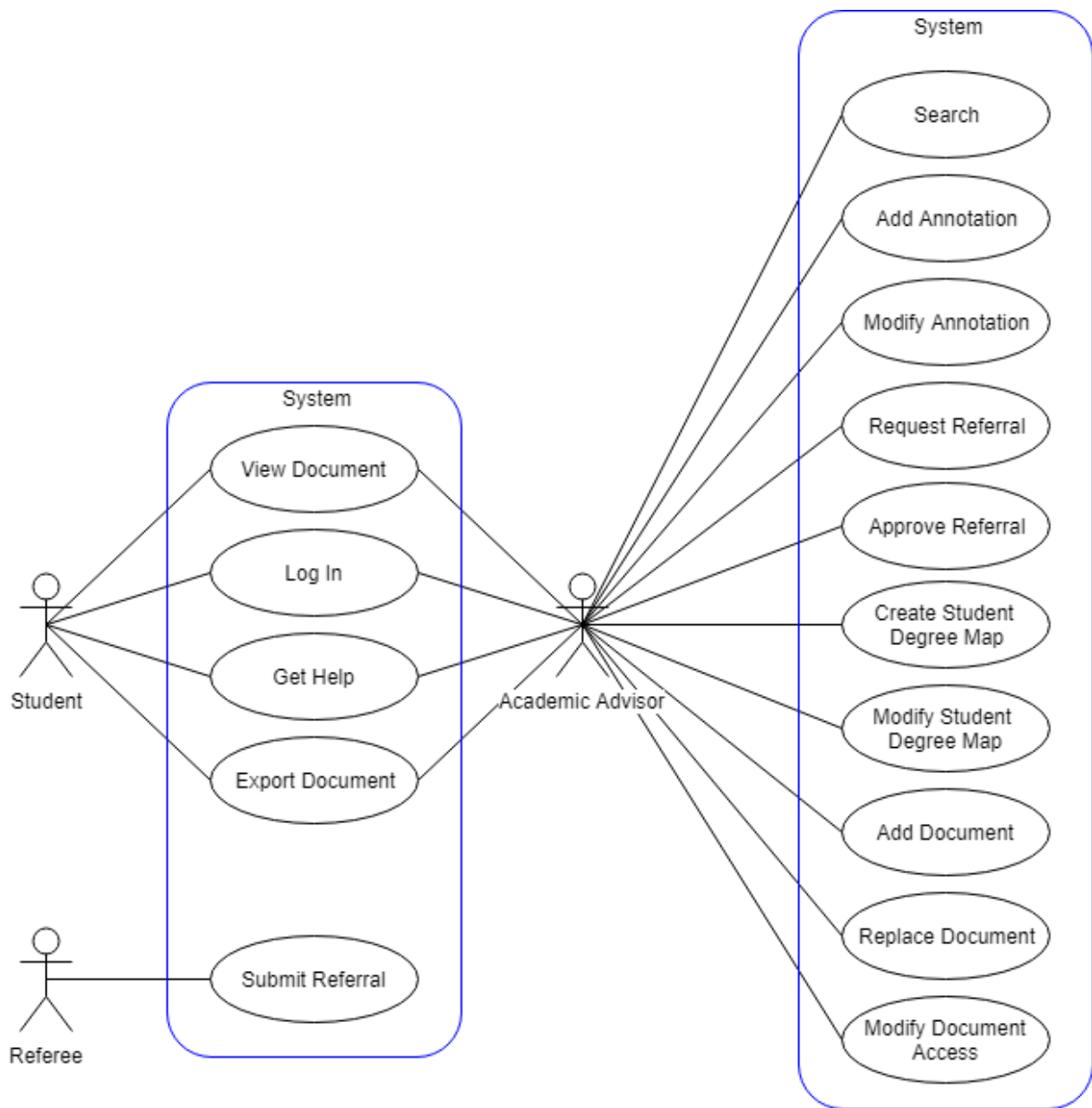


Figure 10: Use Case Model

## 8.2 UC-1.1: Searching

**Description** This use case outlines steps taken by an academic advisor to search for a student document located in the Eduvise database or a student record located in the UVic internal records database. This is done by searching for the student profile, professor profile, or class overview which is related to the desired record or document.

Actors	User (academic advisor)
Preconditions	<ul style="list-style-type: none"><li>• User is logged in to Eduvise</li><li>• User is viewing the search field</li></ul>
Steps	<ol style="list-style-type: none"><li>1 User inputs search query.</li><li>2 User may optionally select one or more filters to limit the search to some combination of student profiles, professor profiles, and course overviews.</li><li>3 User selects submit query option.</li><li>4 Query results of profiles and course overviews matching the search are listed on the page.</li><li>5 User selects desired query result from the list.</li><li>6 The selected item is expanded and displayed to the user.</li><li>7 The user inspects the profile or class overview for the desired document or record.</li></ol>
Success conditions	<ul style="list-style-type: none"><li>• Selected profile or class overview is presented to the user.</li></ul>
Alternate path	4.1 User does not find desired search results
Requirements	REQ-1.1-1.3,6.2

Table 3: Use Case 1.1 - Searching

## Sequence Diagram

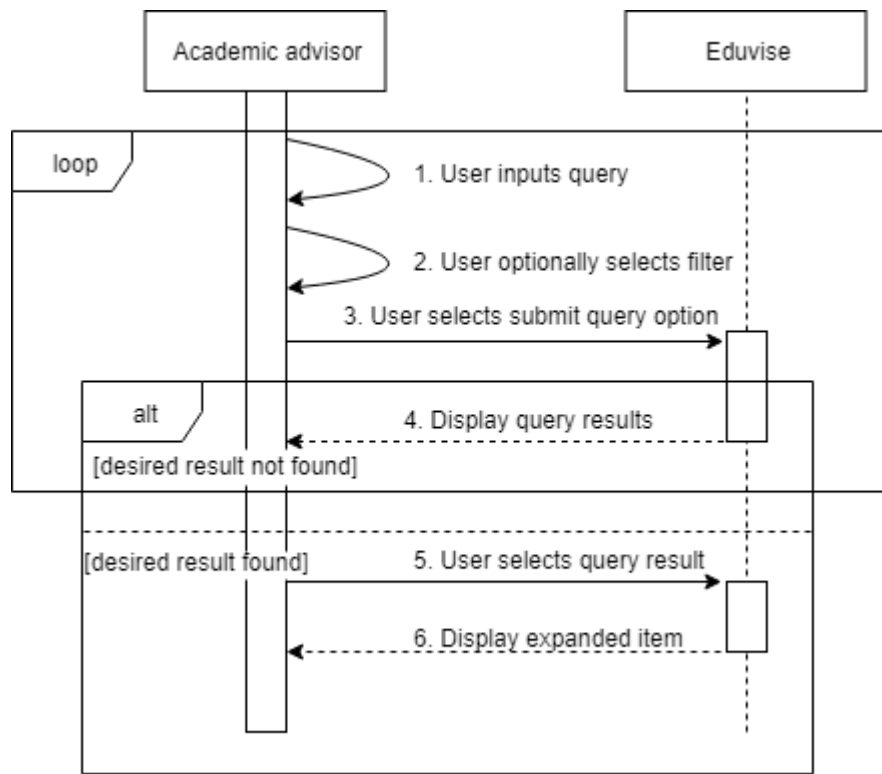
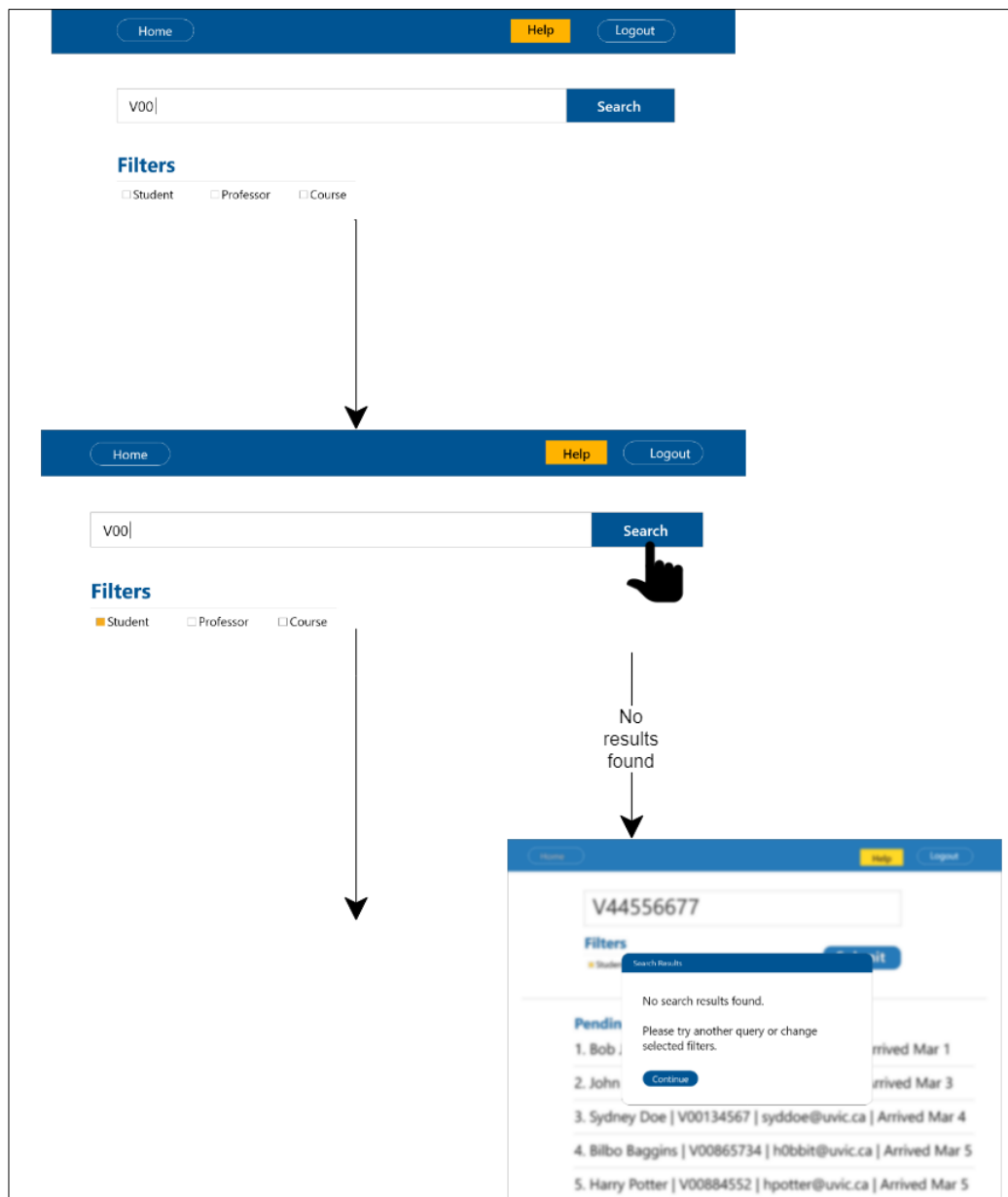


Figure 11: Sequence Diagram 1.1 - Searching

## User Interface Model

The scenario shown in UI Model 1.1 outlines the steps taken to search for a student profile. In this scenario, an academic advisor first selects the "student" check-box so that only the students are searched. Then the advisor inputs the student number of the student to be searched. If no student is found, then a message will be shown notifying the user. If the student is found, the user clicks on the corresponding result and is taken to the student's profile page.





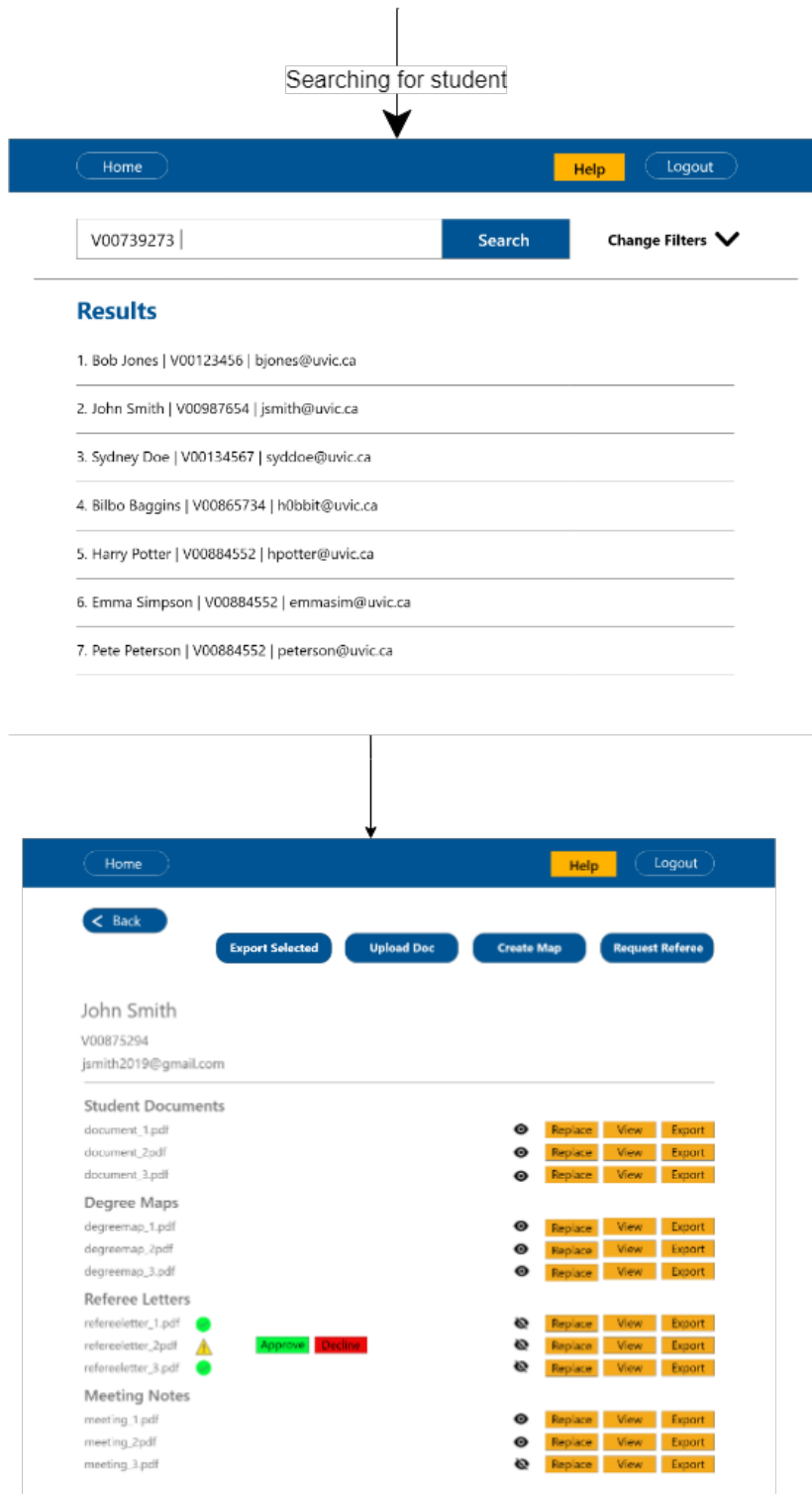


Figure 12: UI Model 1.1 - Searching

### 8.3 UC-2.1: Adding Academic Advising Document

**Description** This use case outlines the steps taken by an academic advisor to add an academic advising document to the profile of a student.

Actors	User (academic advisor)
Preconditions	<ul style="list-style-type: none"><li>• User is logged in to Eduvise</li><li>• User is viewing a student profile</li></ul>
Steps	<ol style="list-style-type: none"><li>1 User selects the option to add a new academic advising document.</li><li>2 User is prompted to select a document from the user's computer.</li><li>3 User selects new document's category, name, visibility, and location on their local computer to be added to Eduvise.</li><li>4 User selects option to submit document to Eduvise.</li><li>5 The document is displayed in the student's profile.</li></ol>
Success conditions	<ul style="list-style-type: none"><li>• The correct document has been added to the student's profile.</li></ul>
Alternate path	<ol style="list-style-type: none"><li>4.1 User selects option to cancel adding document</li><li>4.2 Selected document is discarded and no changes are made to the student's profile.</li><li>5.1 An unsupported file type is detected</li><li>5.2 Selected document is discarded and no changes are made to the student's profile.</li></ol>
Requirements	REQ-2.1

Table 4: Use Case 2.1 - Adding Academic Advising Document

## Sequence Diagram

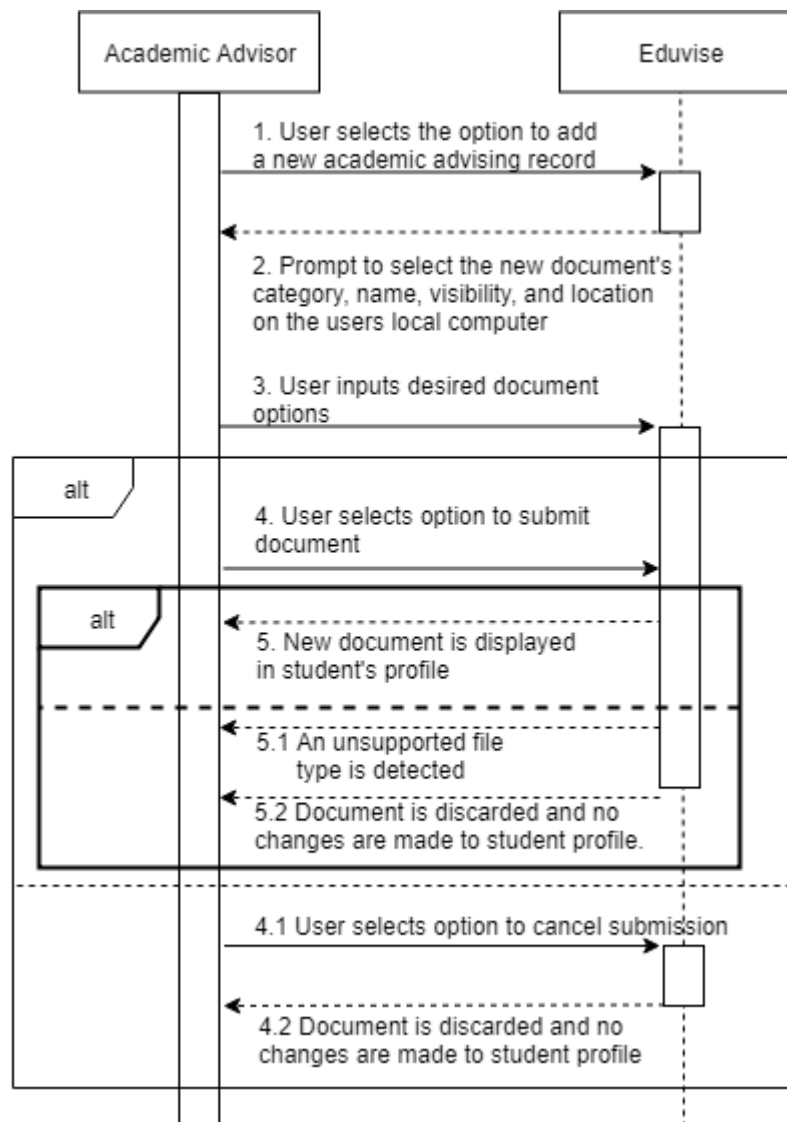


Figure 13: Sequence Diagram 2.1 - Adding Academic Advising Document

## User Interface Model

The scenario shown in UI Model 2.1 outlines the steps taken when an academic advisor uploads a document to a student profile. From the student profile advisor view page, the advisor clicks the "Upload Doc" button which pulls up a modal with a file path field, a document category check box, and a visibility setting. Once the advisor has input all information, they select upload and the file appears in the student's profile.

< Back

Export Selected

Upload Doc

Create Map

Request Referee

John Smith

V00875294

jsmith2019@gmail.com

### Student Documents

document\_1.pdf



Replace

View

Export

#### Document Upload

Document Path Browse

Select a category for the document

☒ General Documents ☐ Degree Maps

☐ Referee Letters ☐ Meeting Notes

☐ Make document visible to the student

Cancel Upload

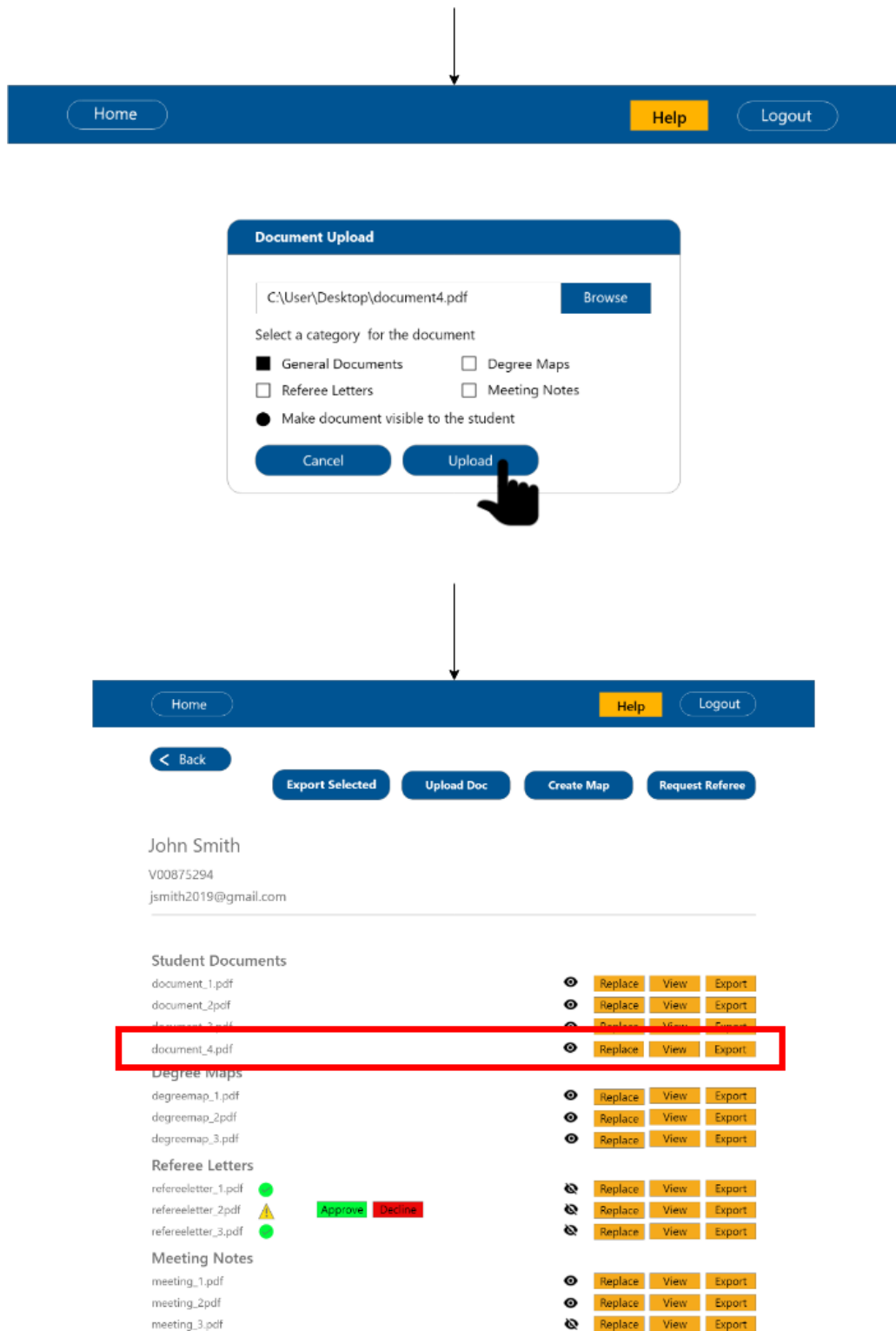


Figure 14: UI Model 2.1 - Adding Academic Advising Document

## 8.4 UC-2.2: Requesting Referral

**Description** This use case outlines the steps taken by an academic advisor to request a referral from a referee.

Actors	User (academic advisor)
Preconditions	<ul style="list-style-type: none"><li>• User is logged in to Eduvise</li><li>• User is viewing a student profile</li></ul>
Steps	<ol style="list-style-type: none"><li>1 User selects option to create a referral request.</li><li>2 User is prompted for email address of the referee.</li><li>3 User inputs the required referee email address.</li><li>4 User selects option to submit the referral request to the email address specified.</li><li>5 Eduvise sends a referral request to the referee by email.</li></ol>
Success conditions	<ul style="list-style-type: none"><li>• The referral request has been sent to the referee.</li></ul>
Alternate path	<ol style="list-style-type: none"><li>4.1 User selects option to cancel the referral request.</li><li>4.2 User is returned to the student profile.</li></ol>
Requirements	

Table 5: Use Case 2.2 - Requesting Referral

Sequence Diagram

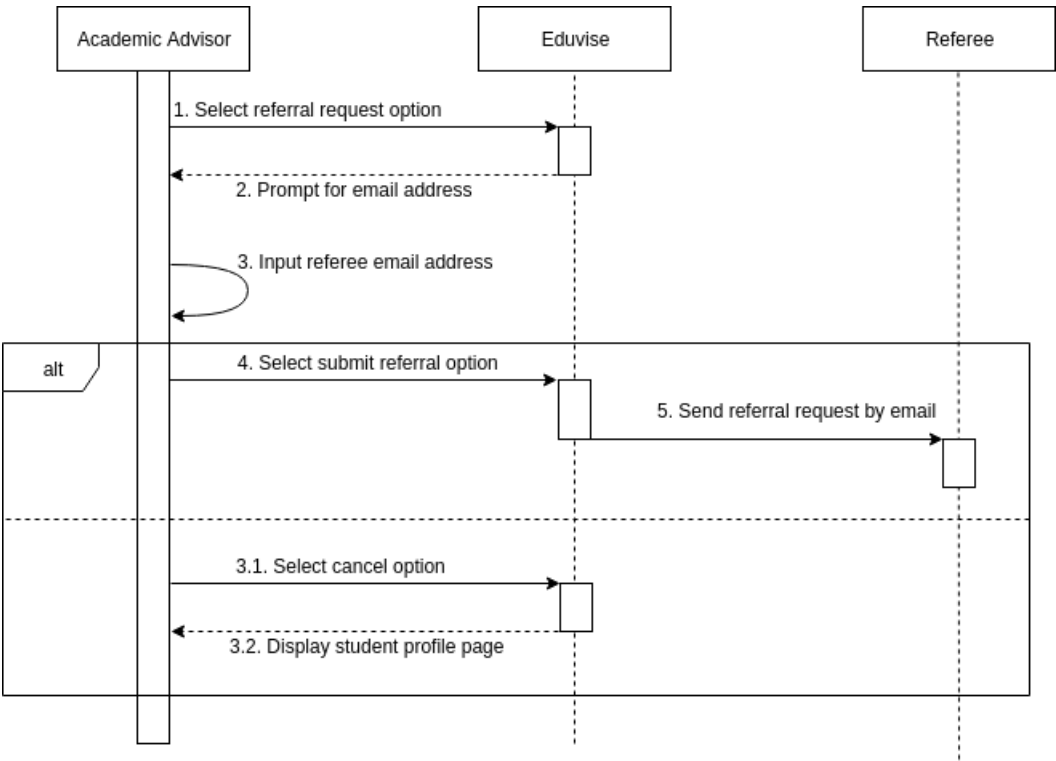


Figure 15: Sequence Diagram 2.2 - Requesting Referral



## User Interface Model

The scenario shown in UI Model 2.2 outlines the steps taken when an academic advisor requests a referral from a referee. From the student profile advisor view page the advisor clicks on the "Request Referral" button which pulls up a modal window with an email input field and message input field for the advisor to fill out. When the advisor clicks on the "Submit" button, the referral request is sent to the referee.

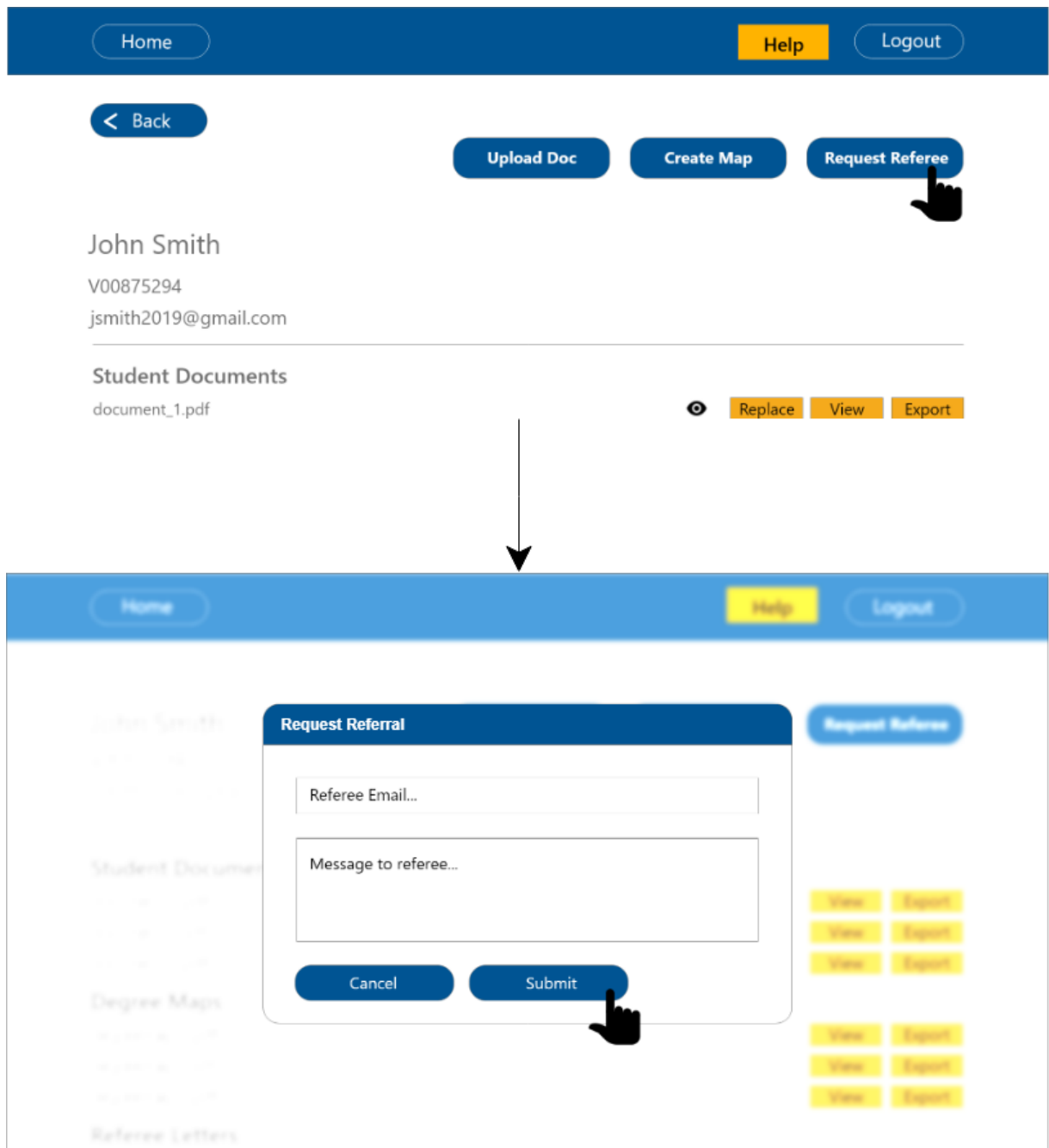


Figure 16: UI Model 2.2 - Requesting Referral

## 8.5 UC-2.3: Submitting Referral

**Description** This use case outlines the steps taken by a referee to submit a referral document in response to a referral request made by an academic advisor.

Actors	User (referee)
Preconditions	<ul style="list-style-type: none"><li>• User has received a referral request email</li><li>• User has a referral document on their local computer</li></ul>
Steps	<ol style="list-style-type: none"><li>1 User follows the link in the referral request to the referral submission page.</li><li>2 User is shown the referral submission page.</li><li>3 User selects the option to submit a referral document.</li><li>4 User is prompted to select a document.</li><li>5 User selects the document to submit as a referral.</li><li>6 User is prompted to confirm their document submission.</li><li>7 User selects the confirm submission option to send the referral document to Eduvise.</li><li>8 User receives a referral submission confirmation email.</li></ol>
Success conditions	<ul style="list-style-type: none"><li>• The referral document has been submitted to Eduvise for approval and the academic advisor that requested the referral has been notified via email.</li></ul>
Alternate path	7.1 User cancels submitting the document
Requirements	

Table 6: Use Case 2.3 - Submitting Referral

Sequence Diagram

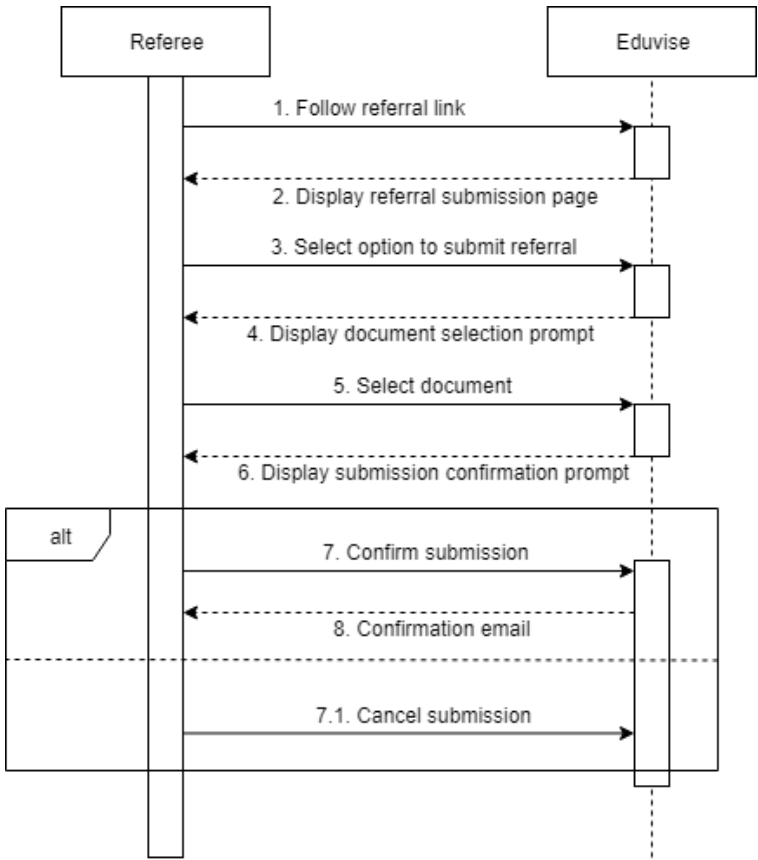


Figure 17: Sequence Diagram 2.3 - Submitting Referral

### User Interface Model

The scenario shown in UI Model 2.3 outlines the steps taken when a referee submits a reference letter and the approval process. First, the referee opens the submission link sent by the advisor and this brings the referee to the Eduvise system. From here they can send a document and write a message back to the advisor. It should also be noted that this is not the same as upload document because this is used specifically for a referee to submit a reference letter

The diagram illustrates the process of submitting a referral. At the top, a label "From Submission Link" has an arrow pointing down to a solid blue horizontal bar. Below this bar is a form titled "Referre Response". The form contains a "Document Name" input field with a "Browse" button to its right. Below the input field is a text area labeled "Message to advisor...". At the bottom of the form are two buttons: "Cancel" and "Submit". A hand icon is shown clicking the "Submit" button.

Figure 18: UI Model 2.3 - Submitting Referral

## 8.6 UC-2.4: Approving Referral

**Description** This use case outlines the steps taken by an academic advisor to approve a referral.

Actors	User (academic advisor)
Preconditions	<ul style="list-style-type: none"><li>• User is logged in to Eduvise</li><li>• User has requested a referral from a referee</li><li>• Referee has submitted a referral document</li></ul>
Steps	<ol style="list-style-type: none"><li>1 User that requested the referral is notified via UVic email when the response is received.</li><li>2 User selects a link to the referral document found in the email notification.</li><li>3 User is redirected to the referral document in Eduvise.</li><li>4 User examines the referral letter from the referee and selects option to approve the referral document.</li><li>5 The referral document is added to the student's profile.</li></ol>
Success conditions	<ul style="list-style-type: none"><li>• The referral document has been correctly added to Eduvise.</li></ul>
Alternate path	<ol style="list-style-type: none"><li>4.1 User selects option to decline the referral document.</li><li>4.2 Eduvise displays prompt to decline the referral, with the option to resend the referral request.</li><li>4.3 User selects option to either resend or not resend the referral request.</li><li>4.4 User optionally inputs a response to the referee in the prompt.</li><li>4.5 User selects option to confirm the referral rejection.</li><li>4.6 Eduvise notifies the referee of the referral rejection by email.</li></ol>
Requirements	

Table 7: Use Case 2.4 - Approving Referral

## Sequence Diagram

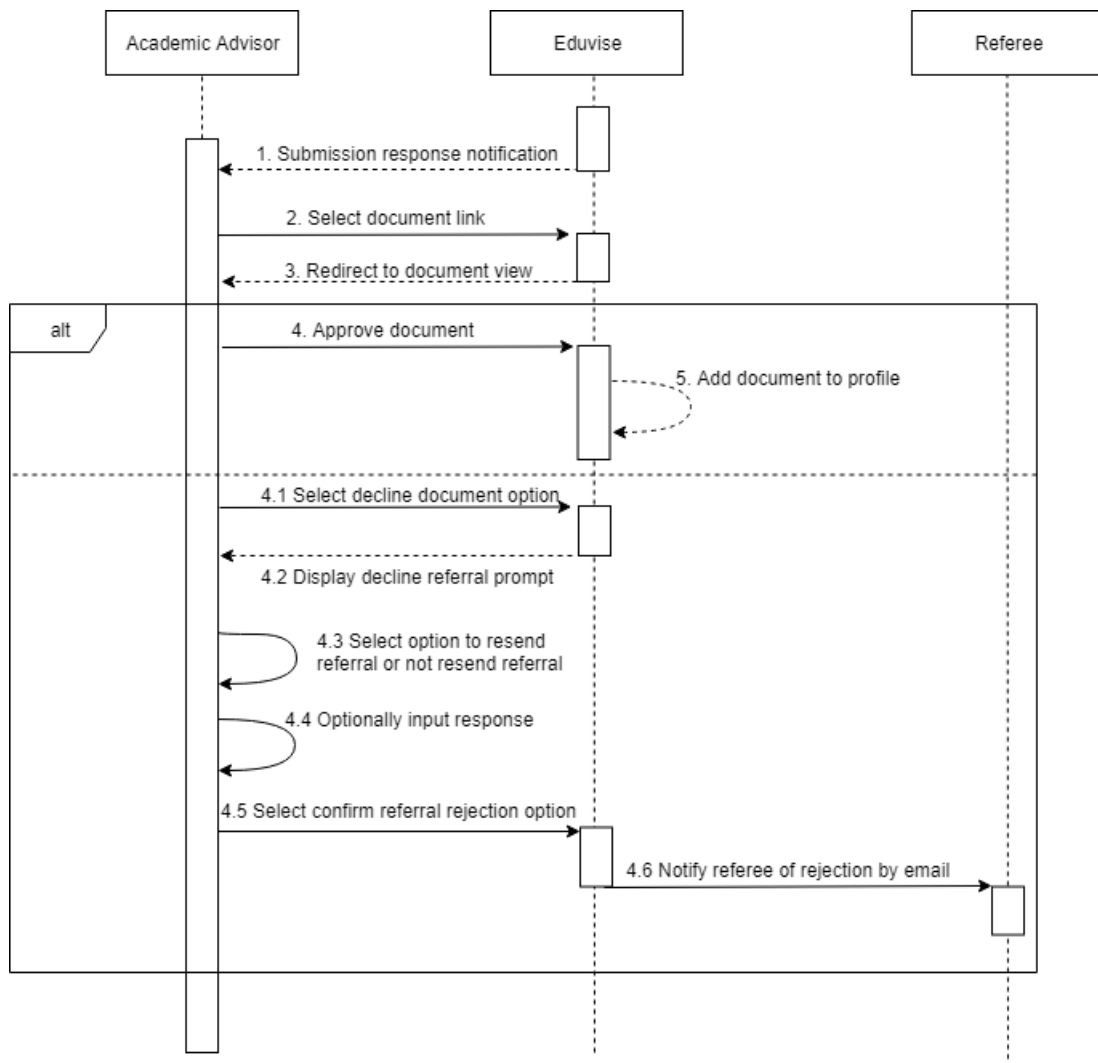


Figure 19: Sequence Diagram 2.4 - Approving Referral

### **User Interface Model**

The scenario shown in UI Model 2.4 outlines the steps taken when an academic advisor receives a notification from Eduvise that a referral letter has been submitted by a referee. From the student profile page, the advisor can click on "Approve" or "Decline" to approve or decline a referral document. This pulls up a modal window that asks the advisor to cancel or confirm this action. When declining, the academic advisor has the option to include a message to the referee and chooses between sending or not sending another referral request to the referee.

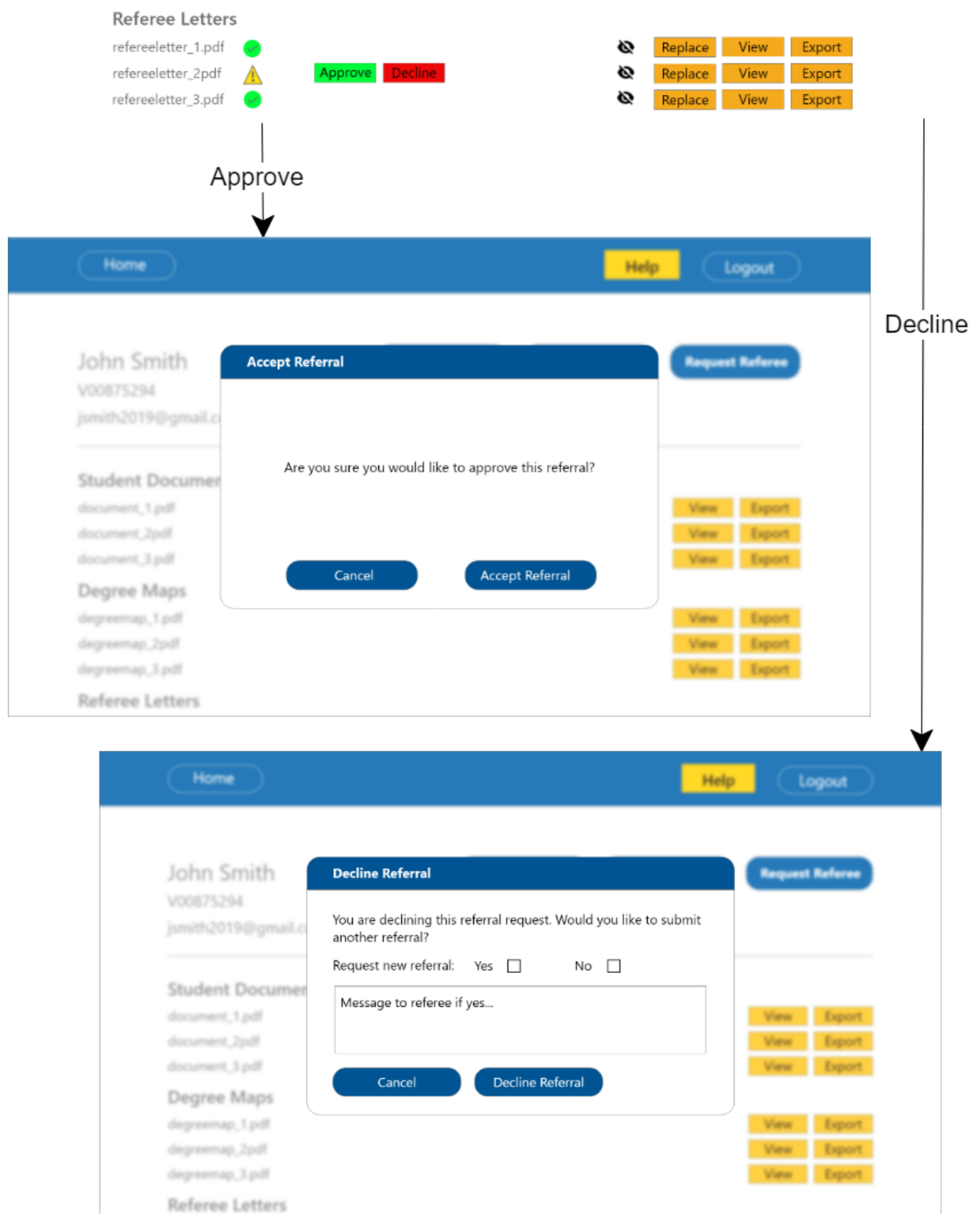


Figure 20: UI Model 2.4 - Approving Referral



## 8.7 UC-3.1: Replacing Academic Advising Document

**Description** This use case outline the steps taken by an academic advisor when replacing an academic advising document with an updated version of the same academic advising document.

Actors	User (academic advisor)
Preconditions	<ul style="list-style-type: none"><li>• User is logged in to Eduvise.</li><li>• User is viewing the profile of a student.</li><li>• Academic Advising Document to be replaced exists.</li></ul>
Steps	<ol style="list-style-type: none"><li>1 User selects option to replace an academic advising document.</li><li>2 File upload screen is brought up.</li><li>3 User locates and selects document on their local computer.</li><li>4 User submits document to Eduvise.</li><li>5 Eduvise prompts user if they are sure they want to replace the selected document with the uploaded one.</li><li>6 User selects option to confirm.</li><li>7 User is brought to student's profile where the old document is replaced with the newly submitted document.</li></ol>
Success conditions	<ul style="list-style-type: none"><li>• The old document is no longer visible and the new document is added to the student's profile.</li></ul>
Alternate path	<ol style="list-style-type: none"><li>5.1 User selects cancel option</li><li>5.2 Uploaded document is discarded and selected document remains.</li></ol>
Requirements	REQ-3.1

Table 8: Use Case 3.1 - Replacing Academic Advising Document

Sequence Diagram

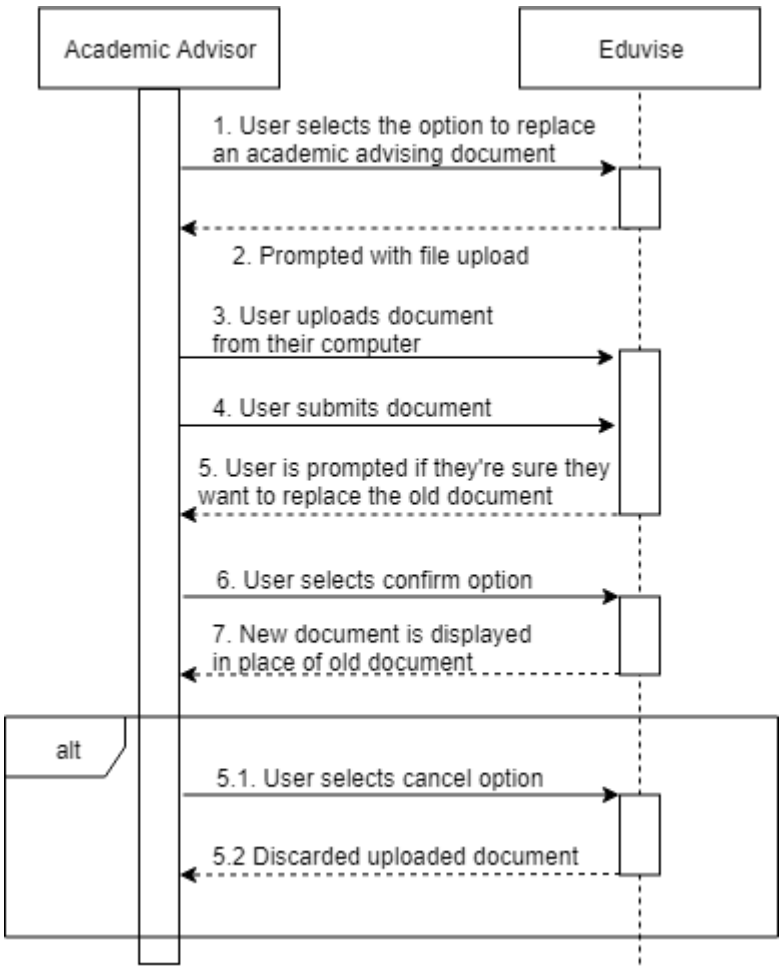


Figure 21: Sequence Diagram 3.1 - Replacing Academic Advising Document

User Interface Model

The scenario shown in UI Model 3.1 outlines the steps taken when a advisor replaces a document on a student’s profile. From the student profile screen the advisor can click the "replace" button on a document . This action brings up a modal to browse for the new document to replace the old document.

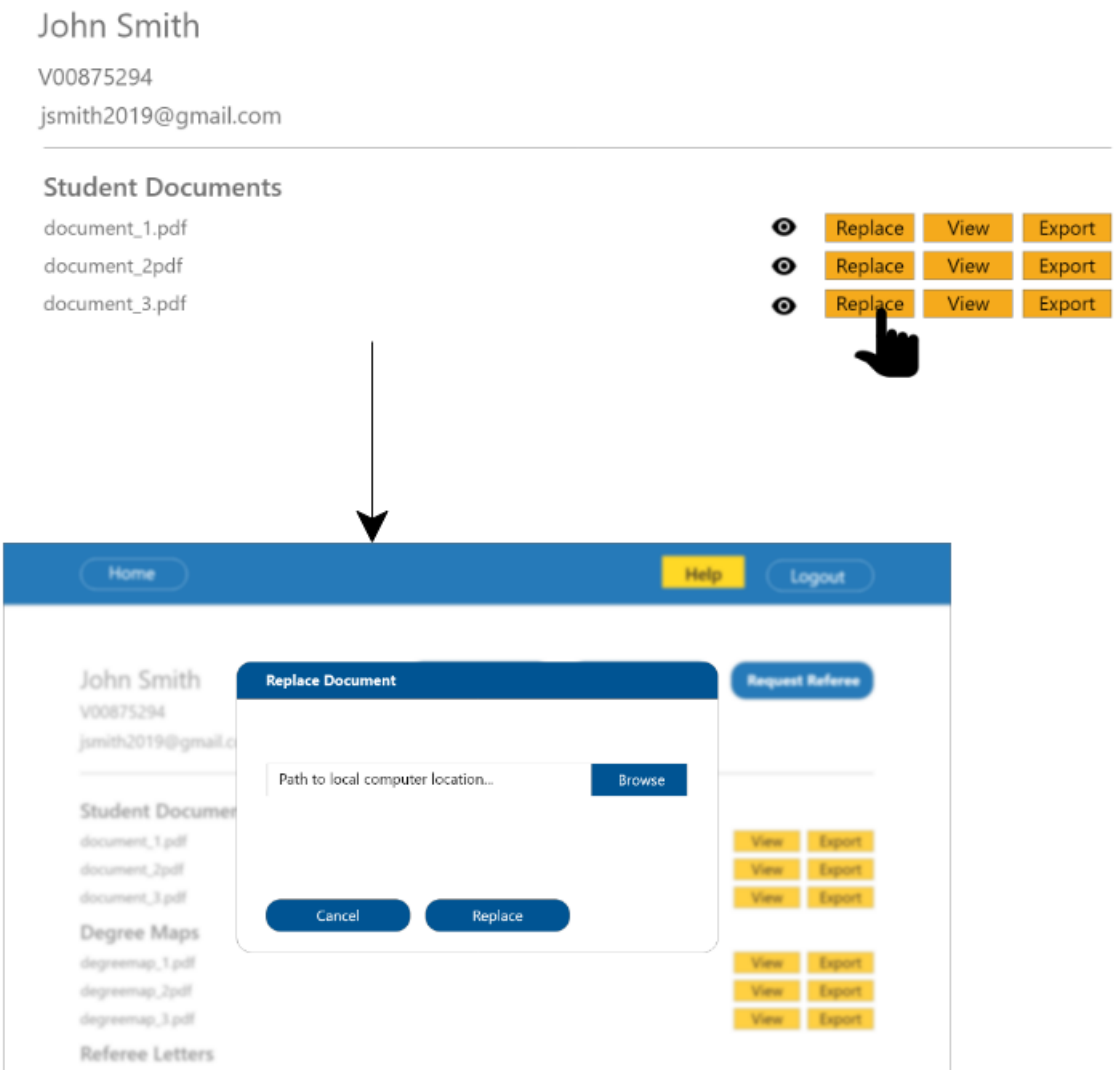


Figure 22: UI Model 3.1 - Replace Academic Advising Document

## 8.8 UC-3.2: Adding Annotation

**Description** This use case outlines the steps taken by an academic advisor to add an annotation to an academic advising document.

Actors	User (academic advisor)
Preconditions	<ul style="list-style-type: none"><li>• User is logged in to Eduvise.</li><li>• User is viewing an academic advising document on a student's profile.</li></ul>
Steps	<ol style="list-style-type: none"><li>1 User selects the "view" option for the desired document.</li><li>2 User selects the "edit" option to enter annotation editing mode.</li><li>3 Eduvise displays an empty text box.</li><li>4 User enters the annotation into the text box.</li><li>5 User clicks submit and the annotation is added.</li><li>6 User is taken to the student's profile.</li></ol>
Success conditions	<ul style="list-style-type: none"><li>• The annotation has been added to the document and is viewable by any academic advisor viewing the document.</li></ul>
Alternate path	<ol style="list-style-type: none"><li>5.1 User selects cancel option</li><li>5.2 Document is displayed and old annotation is unchanged.</li></ol>
Requirements	REQ-3.2

Table 9: Use Case 3.2 - Adding Annotation

## Sequence Diagram

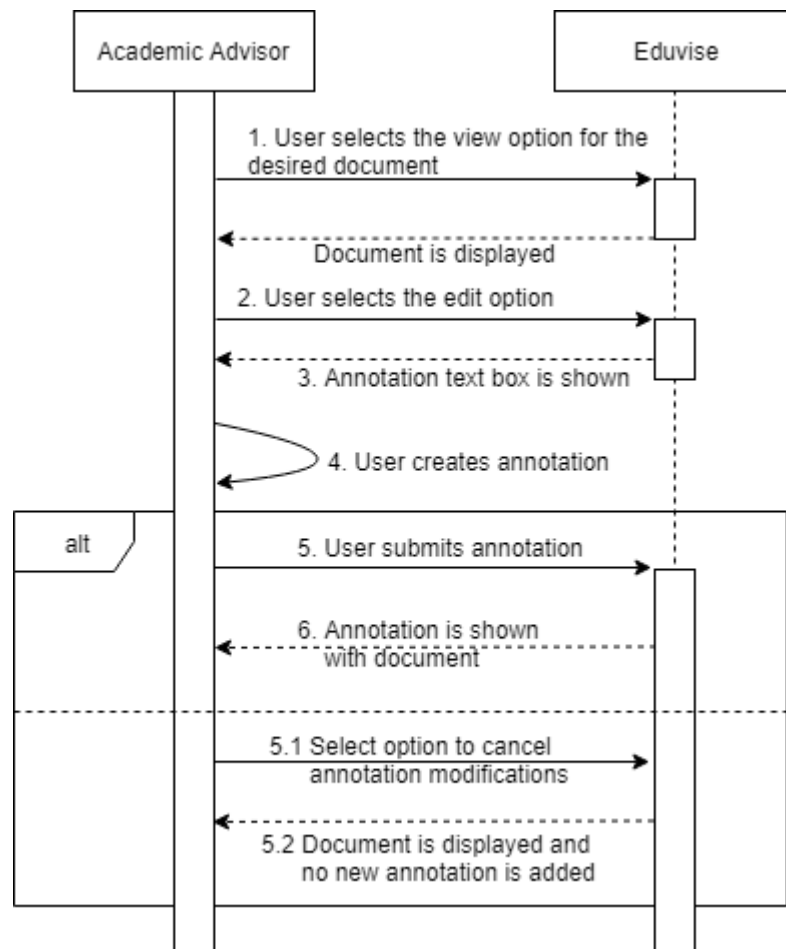


Figure 23: Sequence Diagram 3.2 - Adding Annotation

### **User Interface Model**

The scenario shown in UI Model 3.2 outlines the steps taken to add an annotation to a student's academic advising document. In this scenario, an academic advisor is already a student's profile page, so they first locate the document on the student's profile that they want to annotate, and click the "View" button. When viewing the document, the advisor then clicks the "Edit" button in the "Document Notes" section to edit the document's related annotations. The advisor then enters the new annotation into the text box. Finally, the advisor clicks either the cancel or save button to accept or reject to changes made to the document notes. The user is then taken back to the student's profile page when either option is selected.

jsmith2019@gmail.com

## document\_3.pdf



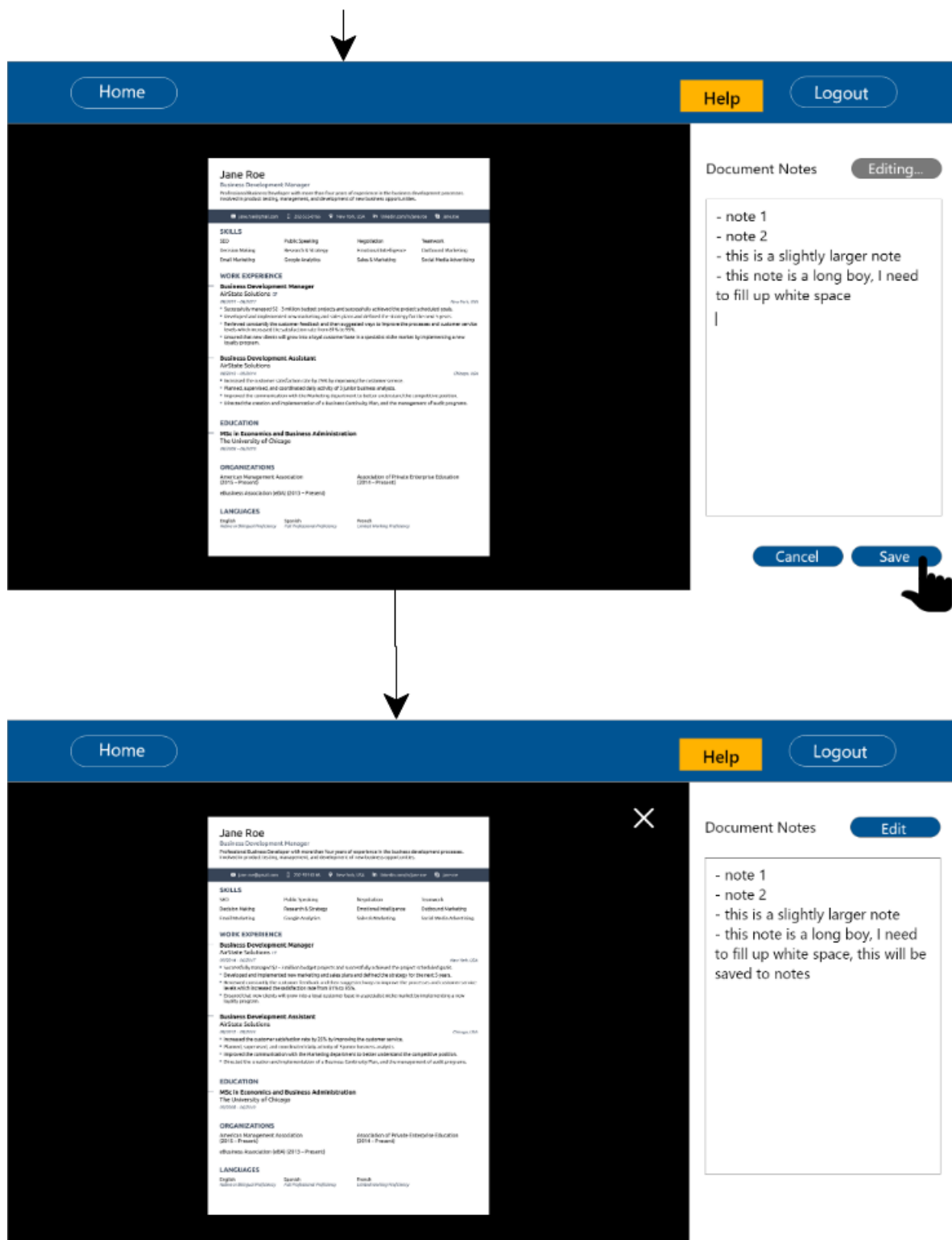


Figure 24: UI Model 3.2 - Adding Annotation



## 8.9 UC-3.3: Modifying Annotation

**Description** This use case outlines the steps taken by an academic advisor to modify an existing annotation on an academic advising document.

Actors	User (academic advisor)
Preconditions	<ul style="list-style-type: none"><li>• User is logged in to Eduvise.</li><li>• User is viewing an academic advising document on a student's profile.</li></ul>
Steps	<ol style="list-style-type: none"><li>1 User requests to view all annotations on the document.</li><li>2 The document is shown in annotation mode.</li><li>3 User selects the annotation to be modified.</li><li>4 The annotation turns into a text box.</li><li>5 User inputs annotation modifications.</li><li>6 User selects option to submit annotation modifications.</li><li>7 Document is reloaded with new annotation.</li></ol>
Success conditions	<ul style="list-style-type: none"><li>• The new annotation is correctly visible on the document.</li></ul>
Alternate path	<ol style="list-style-type: none"><li>6.1 User selects cancel option</li><li>6.2 Document is displayed and old annotation is unchanged.</li></ol>
Requirements	REQ-3.4

Table 10: Use Case 3.3 - Modifying Annotation

## Sequence Diagram

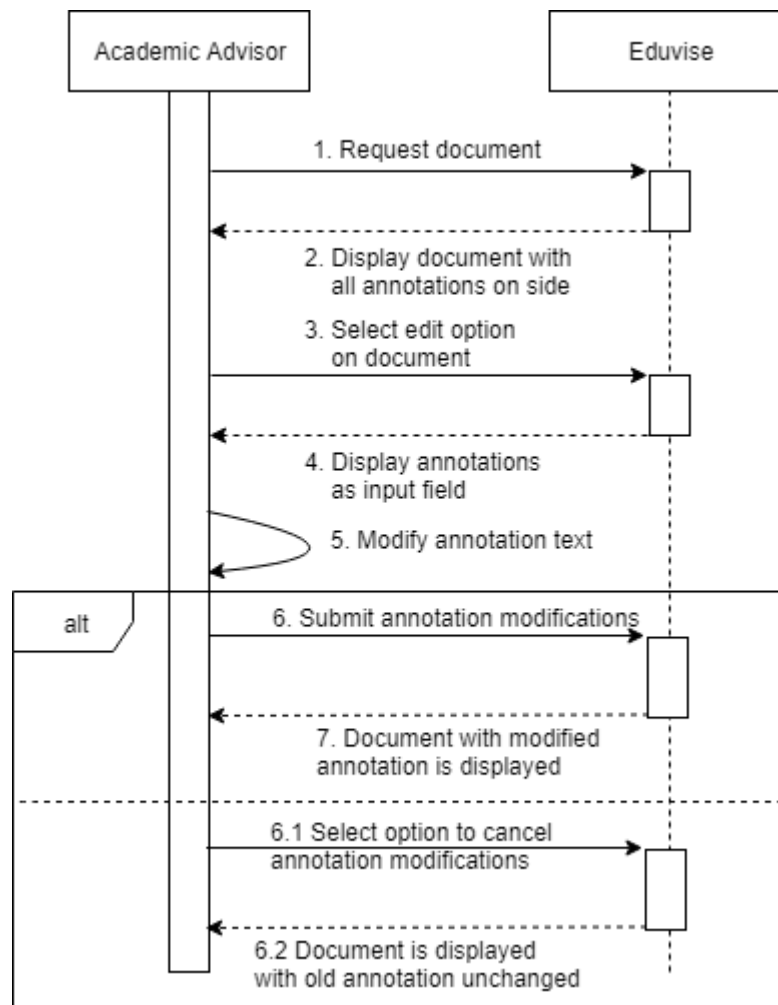


Figure 25: Sequence Diagram 3.3 - Modifying Annotation

## User Interface Model

The scenario shown in UI Model 3.3 outlines the steps taken by an advisor to modify an already existing annotation. From a student's profile screen, an advisor can click view on any document they wish to annotate. From the document viewing screen the advisor can click on the edit button to bring up the note editor. Once modification of annotations is made the advisor can save the annotations associated with the document.



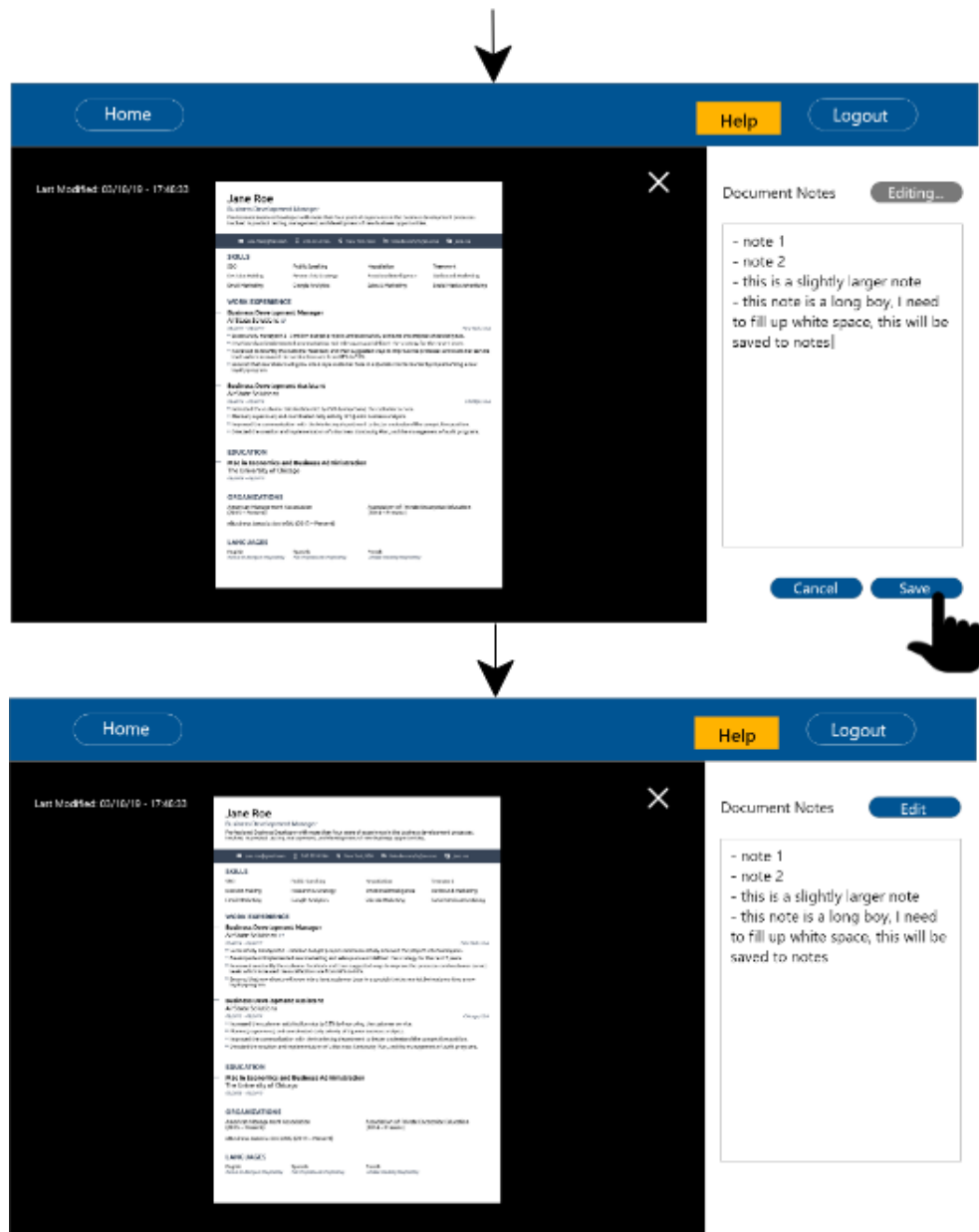


Figure 26: UI Model 3.3 - Modifying Annotation

## 8.10 UC-3.4: Modifying Document Access

**Description** Some documents in Eduvise may not be permitted to be viewed by the student the document pertains to. The visibility of a document must be able to be changed at any time, to rectify a mistake in the initial visibility permissions or for any other reason. This use case outlines the steps taken by an academic advisor to change document access.

Actors	User (academic advisor)
Preconditions	<ul style="list-style-type: none"><li>• User is logged in to Eduvise.</li><li>• User is viewing the profile of a student.</li></ul>
Steps	<ol style="list-style-type: none"><li>1 User selects the option to change document visibility which is located next to the desired document.</li><li>2 The user is prompted to confirm their selection.</li><li>3 User selects option to commit the change to document visibility.</li><li>4 User is taken back to student profile.</li></ol>
Success conditions	<ul style="list-style-type: none"><li>• The correct document has the requested visibility option.</li></ul>
Alternate path	<ol style="list-style-type: none"><li>3.1 User selects cancel option</li><li>3.2 User is taken back to student profile with no changes made to the document visibility.</li></ol>
Requirements	REQ-3.5

Table 11: Use Case 3.4 - Modifying Document Access

Sequence Diagram

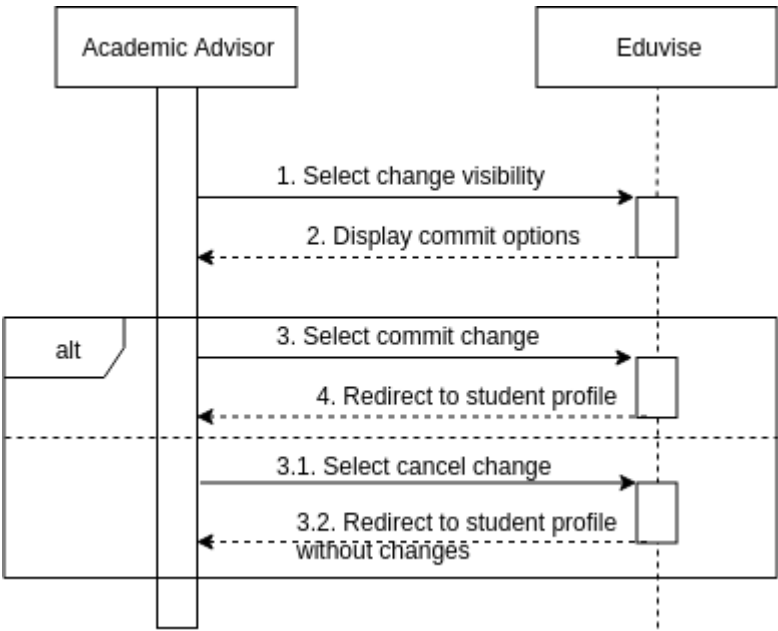


Figure 27: Sequence Diagram 3.4 - Modifying Document Access

User Interface Model

The scenario shown in UI Model 3.4 outlines the steps taken by an advisor to modify the visibility of a document. From a student’s profile screen, an advisor can click the "eye" symbol on any document they wish to make invisible to a student. The advisor will be prompted with a confirm dialog box and after confirming the chosen document will no longer be visible to the student and the "eye" icon will be crossed out.

John Smith  
V00875294  
jsmith2019@gmail.com

Student Documents

document\_1.pdf  
document\_2pdf  
document\_3.pdf



Replace

View

Export

Replace

View

Export

Replace

View

Export

Home

Help

Logout

John Smith  
V00875294  
jsmith2019@gmail.c

Student Documen  
document\_1.pdf  
document\_2pdf  
document\_3.pdf  
Degree Maps  
degreemap\_1.pdf  
degreemap\_2pdf  
degreemap\_3.pdf  
Referee Letters

Request Referee

View

Export

View

Export

View

Export

View

Export

View

Export

Disable Student Visibility

Are you sure you would like to disable the student from viewing this document?

Cancel

Disable Visibilty

John Smith  
V00875294  
jsmith2019@gmail.com

Student Documents

document\_1.pdf  
document\_2pdf  
document\_3.pdf



Replace

View

Export

Replace

View

Export

Replace

View

Export

Figure 28: UI Model 3.4 - Modifying Document Access

## 8.11 UC-4.1: Document Exporting

**Description** This use case outlines the steps taken by an academic advisor or a student to export one or more academic advising documents to their local computer. A user can only export academic advising documents that are visible to that user.

Actors	User (academic advisor, student)
Preconditions	<ul style="list-style-type: none"><li>• User is logged in to Eduvise.</li><li>• User is viewing the profile of a student.</li></ul>
Steps	<ol style="list-style-type: none"><li>1 User selects option to export a document.</li><li>2 The user is prompted to select where the document(s) will be saved on their local computer.</li><li>3 User selects the desired location where the document(s) are to be exported and saved.</li><li>4 User submits the desired selection.</li><li>5 The document(s) are exported to the chosen location as a single file.</li></ol>
Success conditions	<ul style="list-style-type: none"><li>• The correct document(s) have been exported to their local computer.</li></ul>
Alternate path	<ol style="list-style-type: none"><li>1.1 User selects option to export multiple documents</li><li>1.2 User is prompted to select the documents to be exported.</li><li>1.3 User selects multiple documents to be exported.</li><li>1.4 User selects continue option.</li><li>1.5 User continues at step 2.</li></ol>
Requirements	REQ-4.1,4.2

Table 12: Use Case 4.1 - Document Exporting



## Sequence Diagram

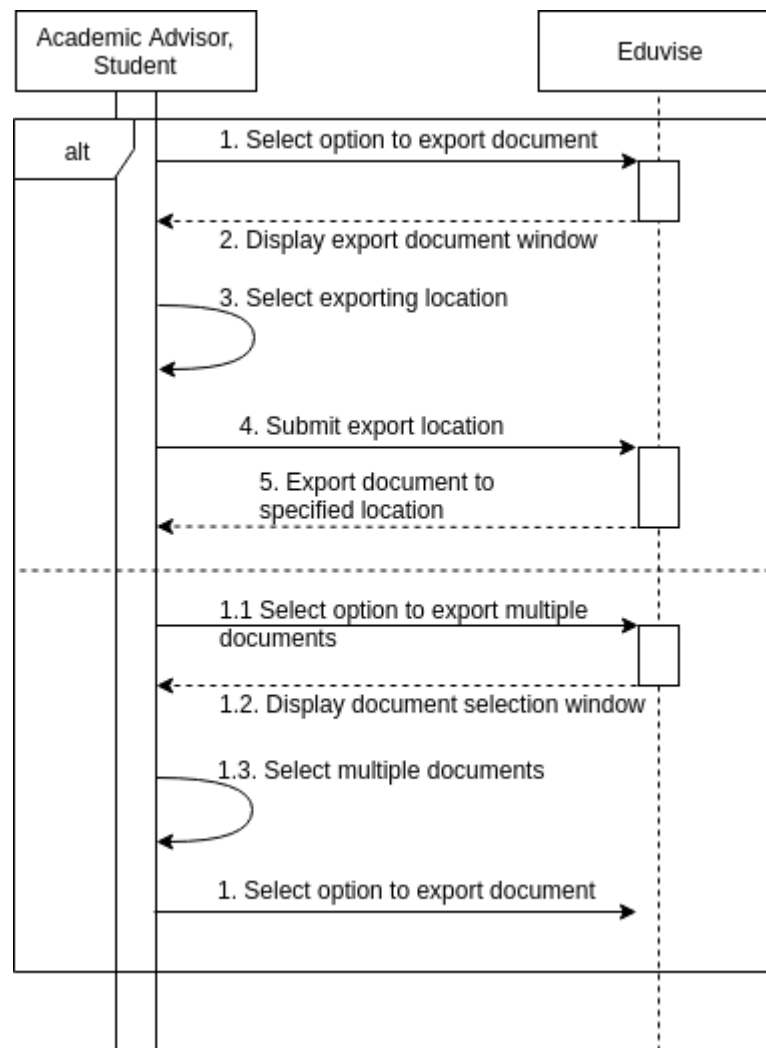


Figure 29: Sequence Diagram 4.1 - Document Exporting

## User Interface Model

The scenario shown in UI Model 4.1 shows the steps needed for a user to export documents to their computer. The user can either be an academic advisor or student, with the steps being functionally the same. On a student's profile page the advisor or student simply clicks export and a menu will pop up allowing the user to select a file path on their computer. Upon clicking browse the file system of the user's computer will open allowing the user to navigate to a location. After selecting a location the user clicks submit and the document will be saved to their chosen location. The process required to export multiple documents begins with instead choosing the export multiple documents option, selecting all documents desired to be exported, then continuing through the same steps to select a location and export the documents.

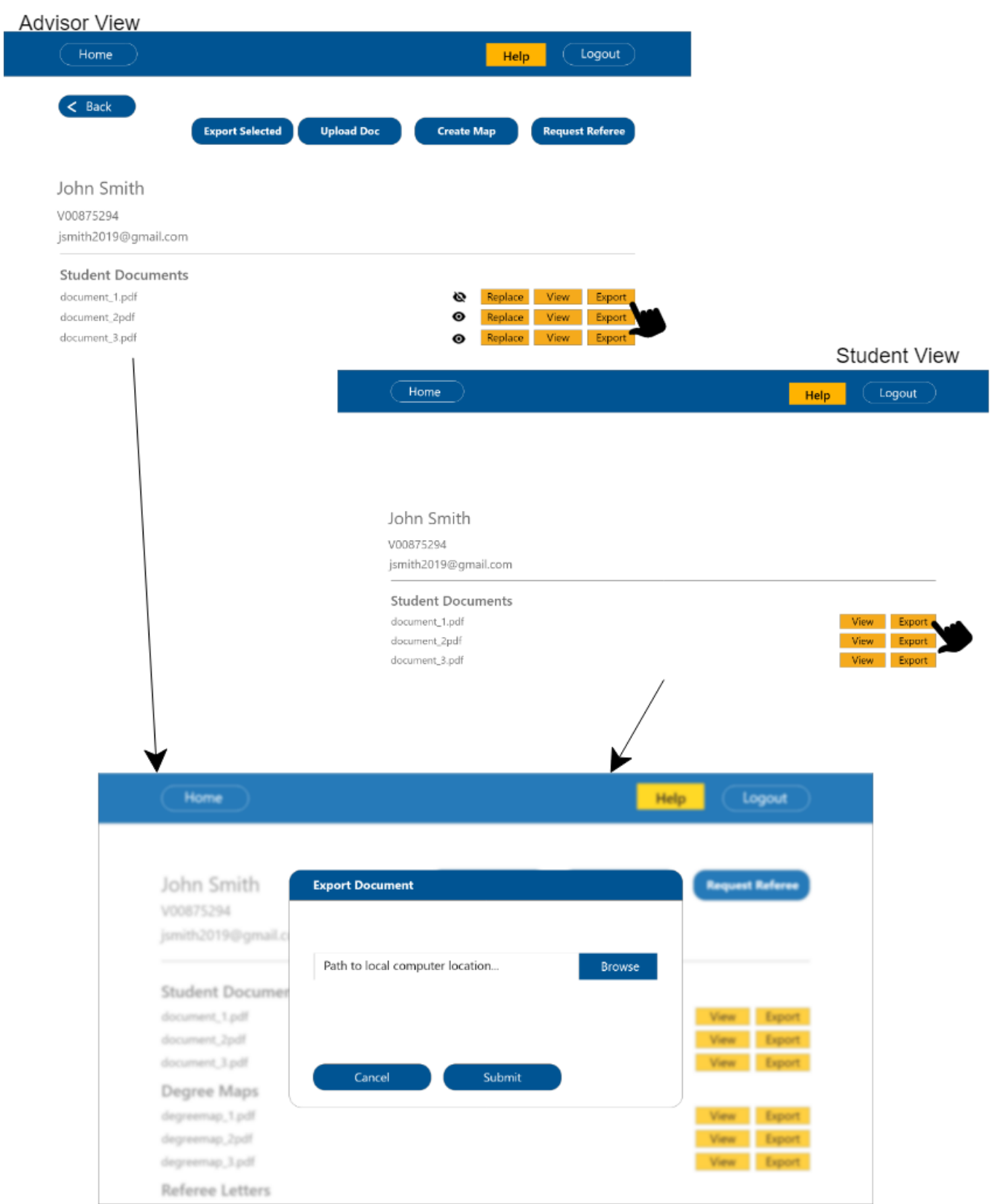


Figure 30: UI Model 4.1 - Document Exporting

## 8.12 UC-5.1: Document Viewing

**Description** This use case outlines the steps taken by an academic advisor or a student to view an academic advising document. A user can only view an academic advising document that is visible to that user.

Actors	User (academic advisor, student)
Preconditions	<ul style="list-style-type: none"><li>• User is logged in to Eduvise.</li><li>• User is viewing the profile of a student.</li></ul>
Steps	<ol style="list-style-type: none"><li>1 User selects a document from the student profile.</li><li>2 The selected document is displayed in the browser.</li></ol>
Success conditions	<ul style="list-style-type: none"><li>• The user has viewed the correct document.</li></ul>
Alternate path	None
Requirements	REQ-5.2,6.3

Table 13: Use Case 5.1 - Document Viewing

### Sequence Diagram

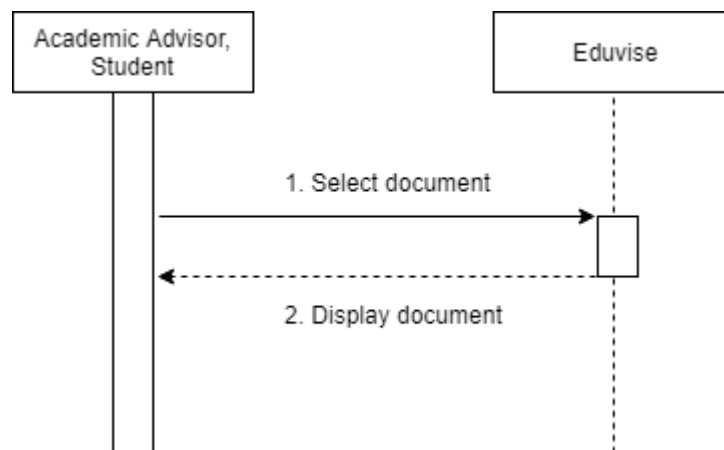


Figure 31: Sequence Diagram 5.1 - Document Viewing

### User Interface Model

The scenario shown in UI Model 5.1 outlines the steps taken to view an academic advising document related to a student's profile. In this scenario the user is already on a student's profile, either their own profile if the user is a student, or some student as a result of a search query. The user then clicks on the "View" button of the desired document and is then taken to the document viewing page.

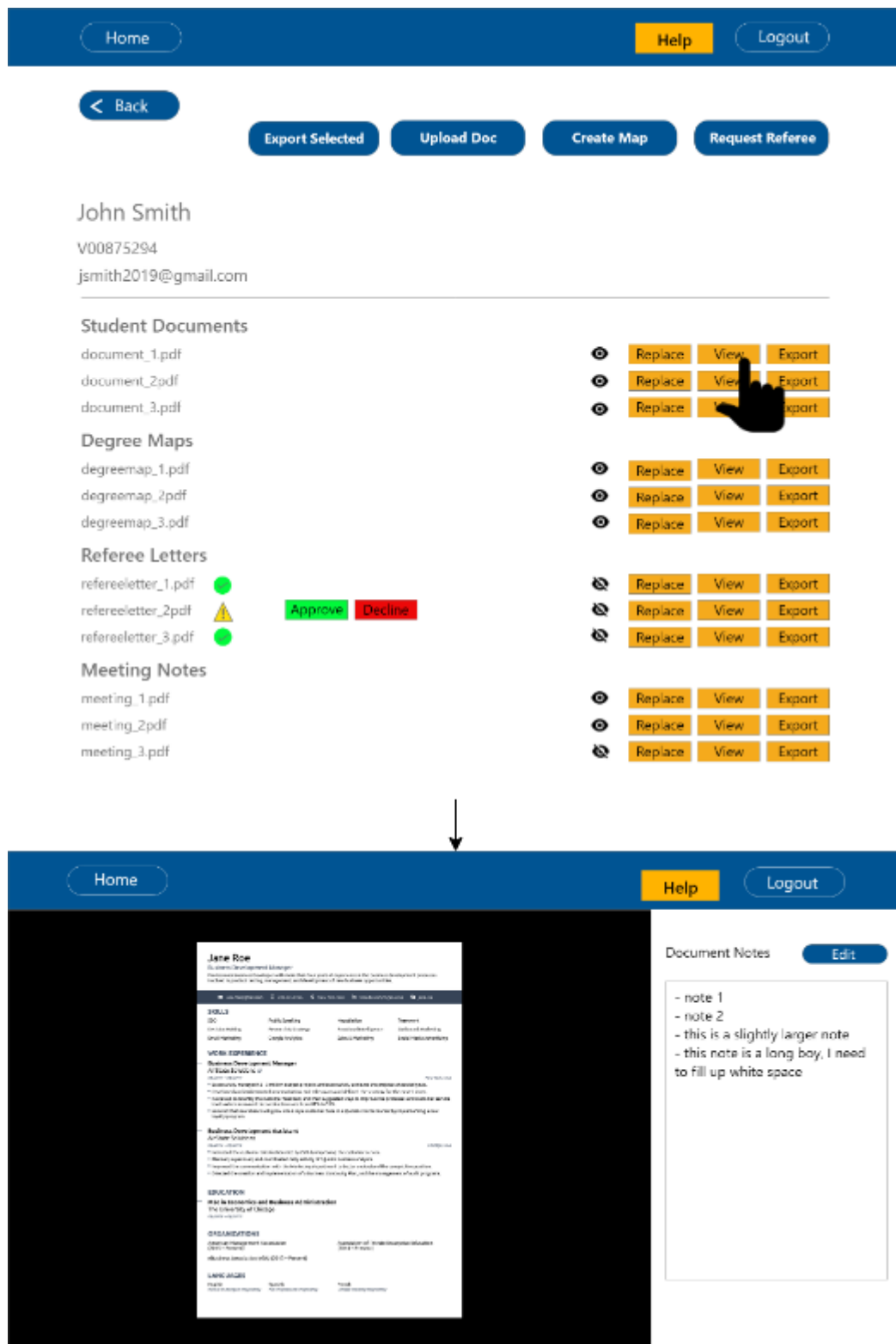


Figure 32: UI Model 5.1 - Document Viewing

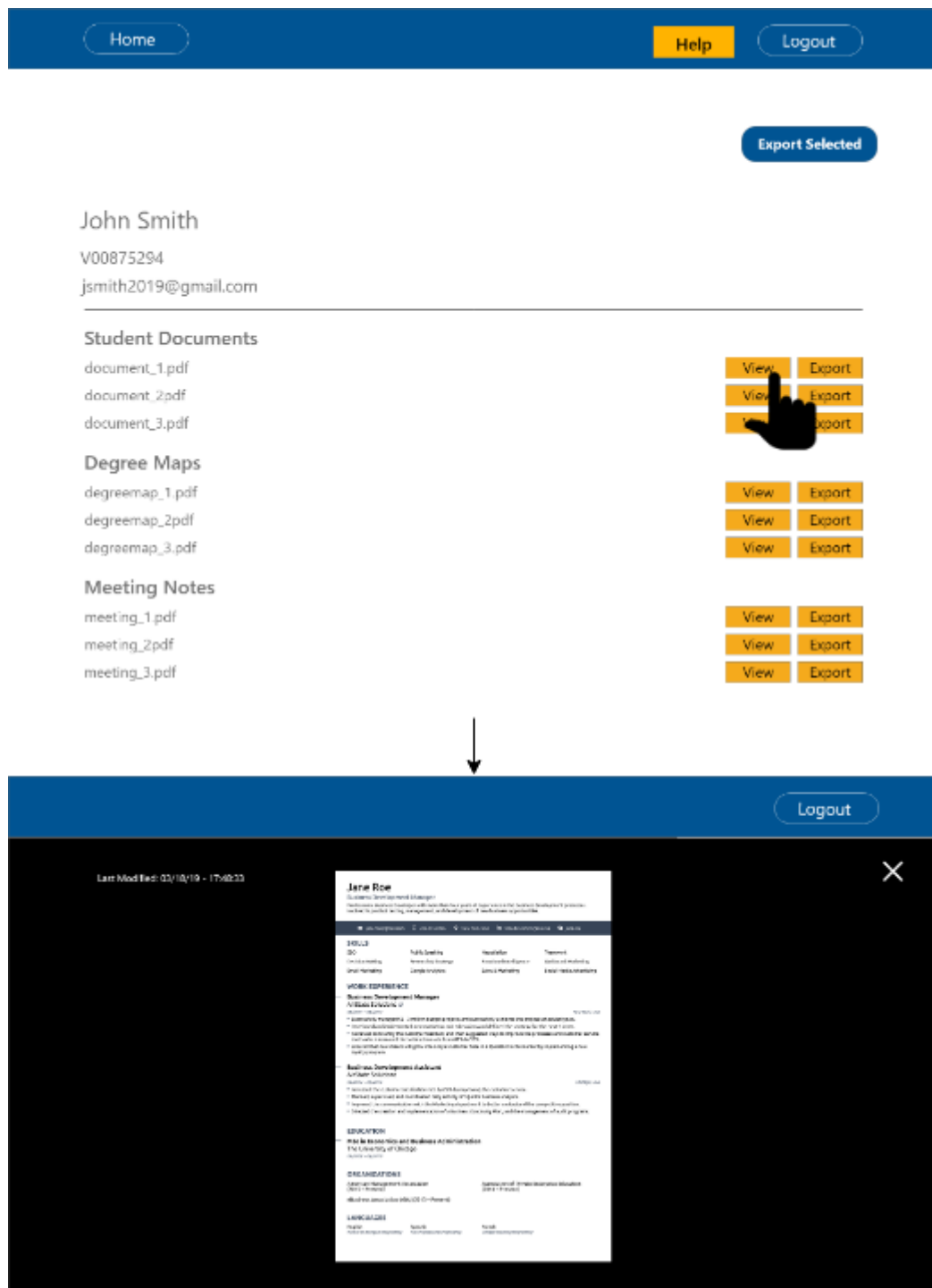


Figure 33: UI Model 5.1.2 - Document Viewing By Student

### 8.13 UC-6.1: Getting Help

**Description** This use case outlines the steps taken by an academic advisor or a student to receive help about correctly using Eduvise.

Actors	User (academic advisor, student)
Preconditions	<ul style="list-style-type: none"><li>• User is logged in to Eduvise.</li></ul>
Steps	<ol style="list-style-type: none"><li>1 User selects the help button.</li><li>2 Help window is displayed on the user's page.</li><li>3 User navigates help window to locate section related to the user's current query and selects the desired selection.</li><li>4 The detailed info of the selected section is displayed.</li></ol>
Success conditions	<ul style="list-style-type: none"><li>• The user receives correct information about their current query.</li></ul>
Requirements	EIR-1.1, NFR-4.2

Table 14: Use Case 6.1 - Getting Help

#### Sequence Diagram

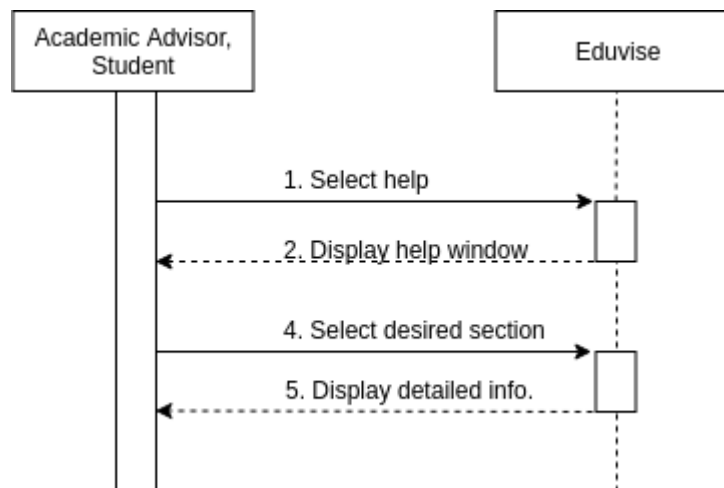
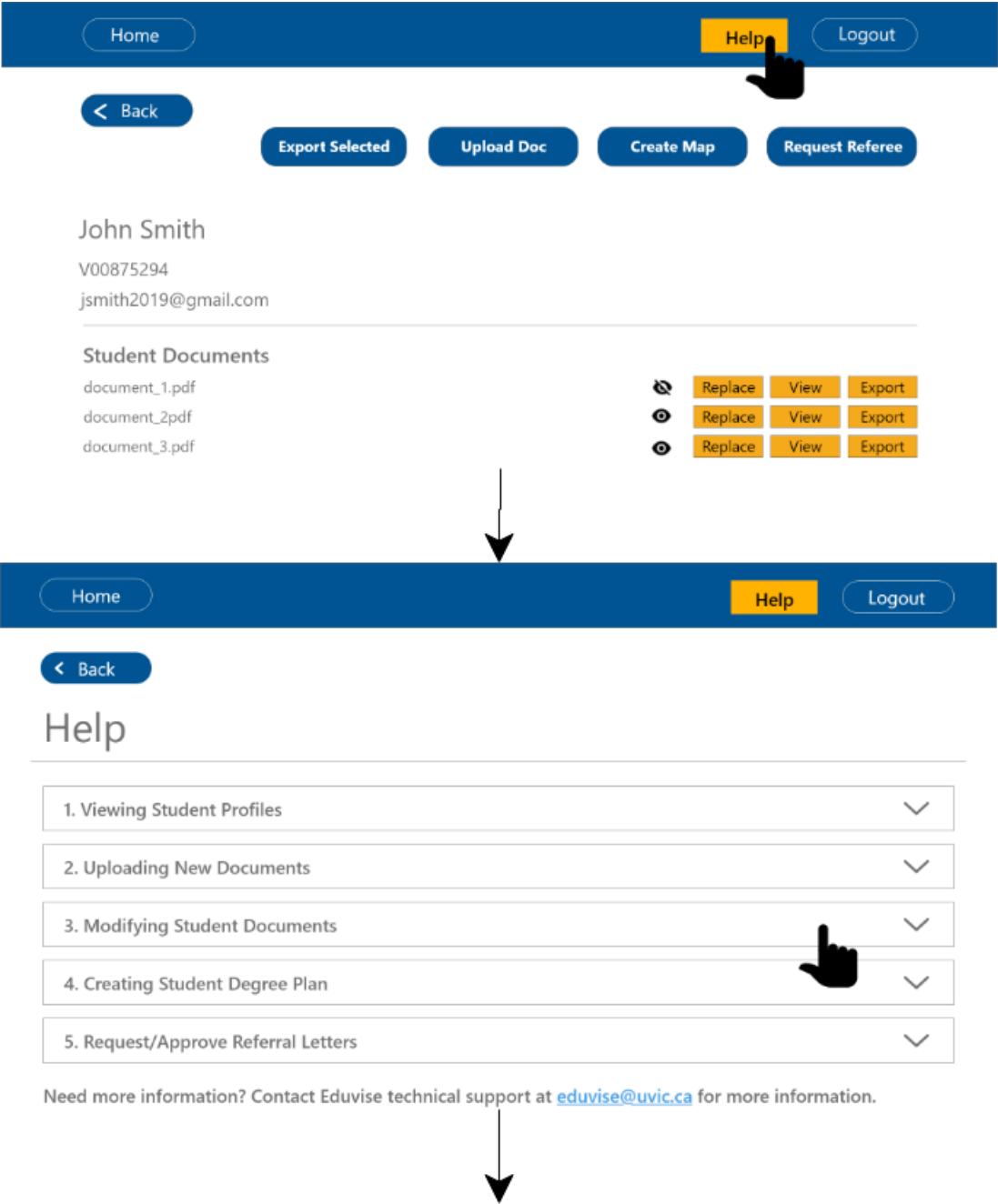


Figure 34: Sequence Diagram 6.1 - Getting Help

#### User Interface Model

The scenario shown in UI Model 6.1 outlines the steps taken by an advisor getting help. From the student profile screen there is a help button shown which the advisor can click. On click the advisor is brought to the help screen with the options of help within the following five categories: Viewing Student Profiles, Uploading Documents, Modifying Student Documents, Creating Student Degree

Map, and Sending Referral Letters Requests and Accepting Referral Letters. By clicking on one of these five options a drop down of steps becomes visible. If these options are unable to help the student an email address for additional help is available.



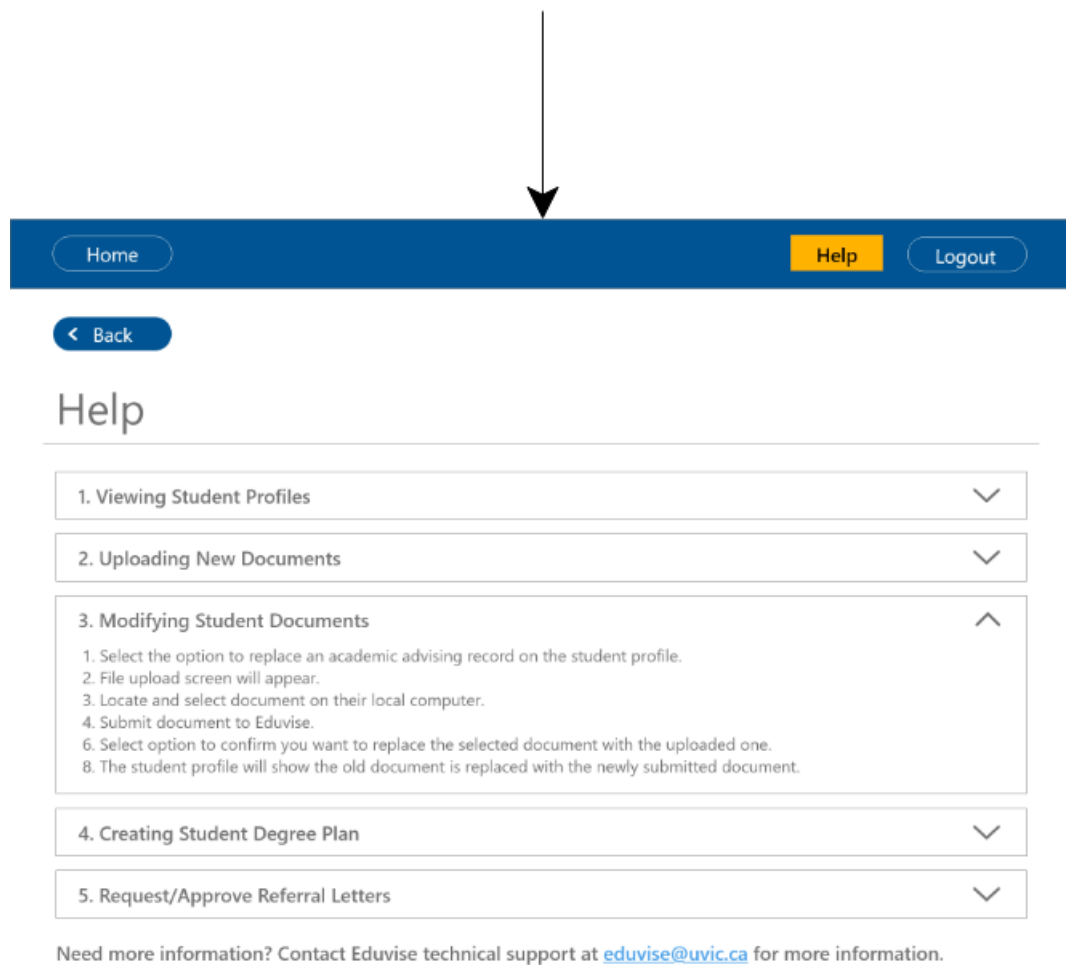


Figure 35: UI Model 6.1 - Getting Help

### 8.14 UC-7.1: Creating Student Degree Map

**Description** This use case outlines the steps taken by an academic advisor to create a new degree map on a student's profile for degree planning with that student.



Actors	User (academic advisor)
Preconditions	<ul style="list-style-type: none"> <li>• User is logged in to Eduvise.</li> <li>• User is viewing the profile of a student.</li> </ul>
Steps	<ol style="list-style-type: none"> <li>1 User selects the option to create a new student degree map.</li> <li>2 An empty degree map is produced for the user.</li> <li>3 User selects the desired term from the map.</li> <li>4 User selects from a list of classes to add to the term.</li> <li>5 User repeats step 3 and 4 for each desired term.</li> <li>6 User selects option to submit student degree map.</li> <li>7 User is shown a degree map on the the student's profile</li> </ol>
Success conditions	<ul style="list-style-type: none"> <li>• The desired degree map is produced and saved to the student's profile.</li> </ul>
Alternate path	<ol style="list-style-type: none"> <li>5.1 User cancels creation of degree map.</li> <li>5.2 New student degree map is discarded and no changes are made to the student profile.</li> </ol>
Requirements	REQ-7.4, REQ-7.5

Table 15: Use Case 7.1 - Creating Student Degree Map

## Sequence Diagram

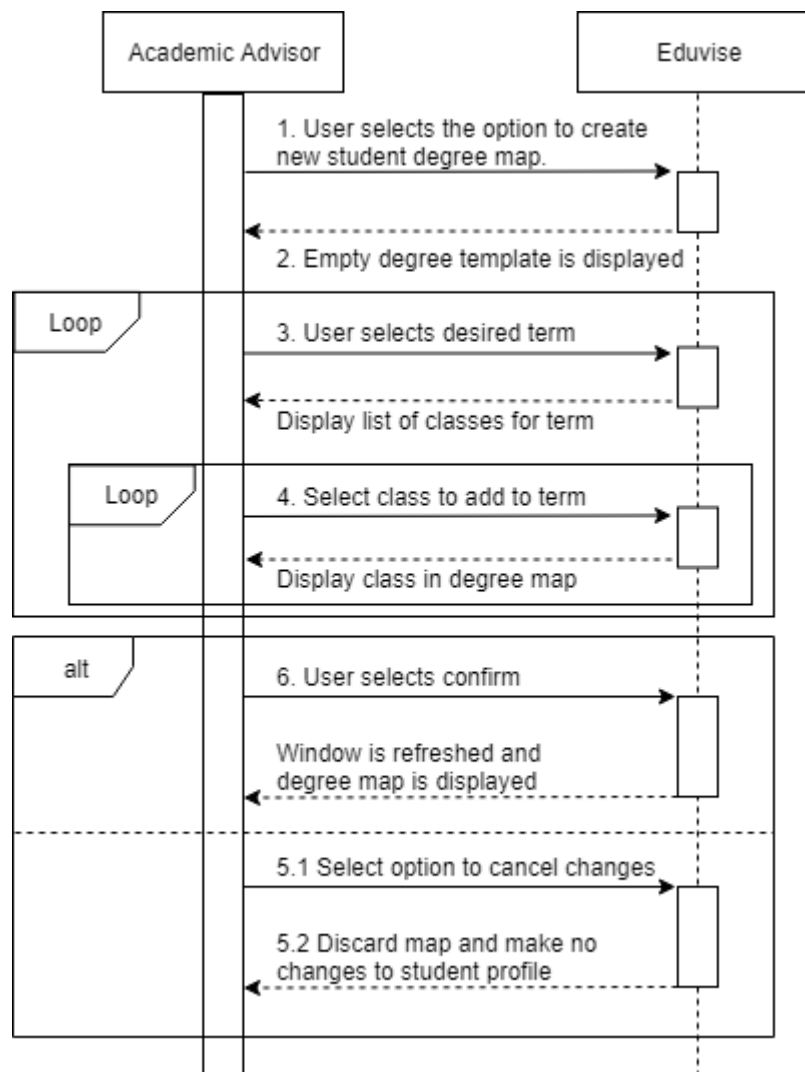


Figure 36: Sequence Diagram 7.1 - Creating Student Degree Map

### User Interface Model

The scenario shown in UI Model 7.1 outlines the steps taken by an advisor when creating a new student degree map. In this scenario, the advisor is already on a student's profile, and the first step is to click the "Create Map" button. When the advisor is taken to the "Create Degree Map" screen with an newly created, empty degree map, the advisor will click the "Add Courses" button and is taken to a screen to select the term to add courses for. After the advisor selects a term, the "Submit" button is pressed to continue to the screen to add classes to the selected term. The advisor then goes through a list of classes for that term and selects the classes for the degree map during that term. The advisor then clicks the "Submit" button to submit the chosen classes for the selected term and is taken back to the "Create Degree Map" page. The user repeats this process for all the required terms for the student. The user is able to click on each course in order to delete or replace a single course. When all terms have been completed the advisor clicks the "Save Map" button to save the new degree map in Eduvise.

[< Back](#)

[Export Selected](#)
[Upload Doc](#)
[Create Map](#)
[Request Referee](#)

John Smith

V00875294

jsmith2019@gmail.com

### Student Documents

document_1.pdf		<a href="#">Replace</a>	<a href="#">View</a>	<a href="#">Export</a>
document_2.pdf		<a href="#">Replace</a>	<a href="#">View</a>	<a href="#">Export</a>
document_3.pdf		<a href="#">Replace</a>	<a href="#">View</a>	<a href="#">Export</a>



[< Back](#)

[Add Courses](#)

## Degree Map

Year 1			Year 2		
Fall	Spring	Summer	Fall	Spring	Summer
None	None	None	None	None	None
None	None	None	None	None	None
None	None	None	None	None	None
None	None	None	None	None	None
None	None	None	None	None	None
None	None	None	None	None	None
None	None	None	None	None	None

Year 3			Year 4		
Fall	Spring	Summer	Fall	Spring	Summer
None	None	None	None	None	None
None	None	None	None	None	None
None	None	None	None	None	None
None	None	None	None	None	None
None	None	None	None	None	None
None	None	None	None	None	None
None	None	None	None	None	None

### Academic Goals

[Edit Goals](#)

- 
- 
- 
- 

### Additional Comments

- 
- 
- 
- 



**Degree Map Editor**

Select a Year:

☐ Year 1

☐ Year 2

☐ Year 3

☐ Year 4

Select a term:

☐ Fall

☐ Spring

☐ Summer

**Degree Map Editor**

Select a Year:

☒ Year 1

☐ Year 2

☐ Year 3

☐ Year 4

Select a term:

☒ Fall

☐ Spring

☐ Summer

**Degree Map Editor**

Select classes to add:

Available Class 1

Available Class 2

Available Class 3

Available Class 4

Available Class 5

☐

☐

☐

☐

☐

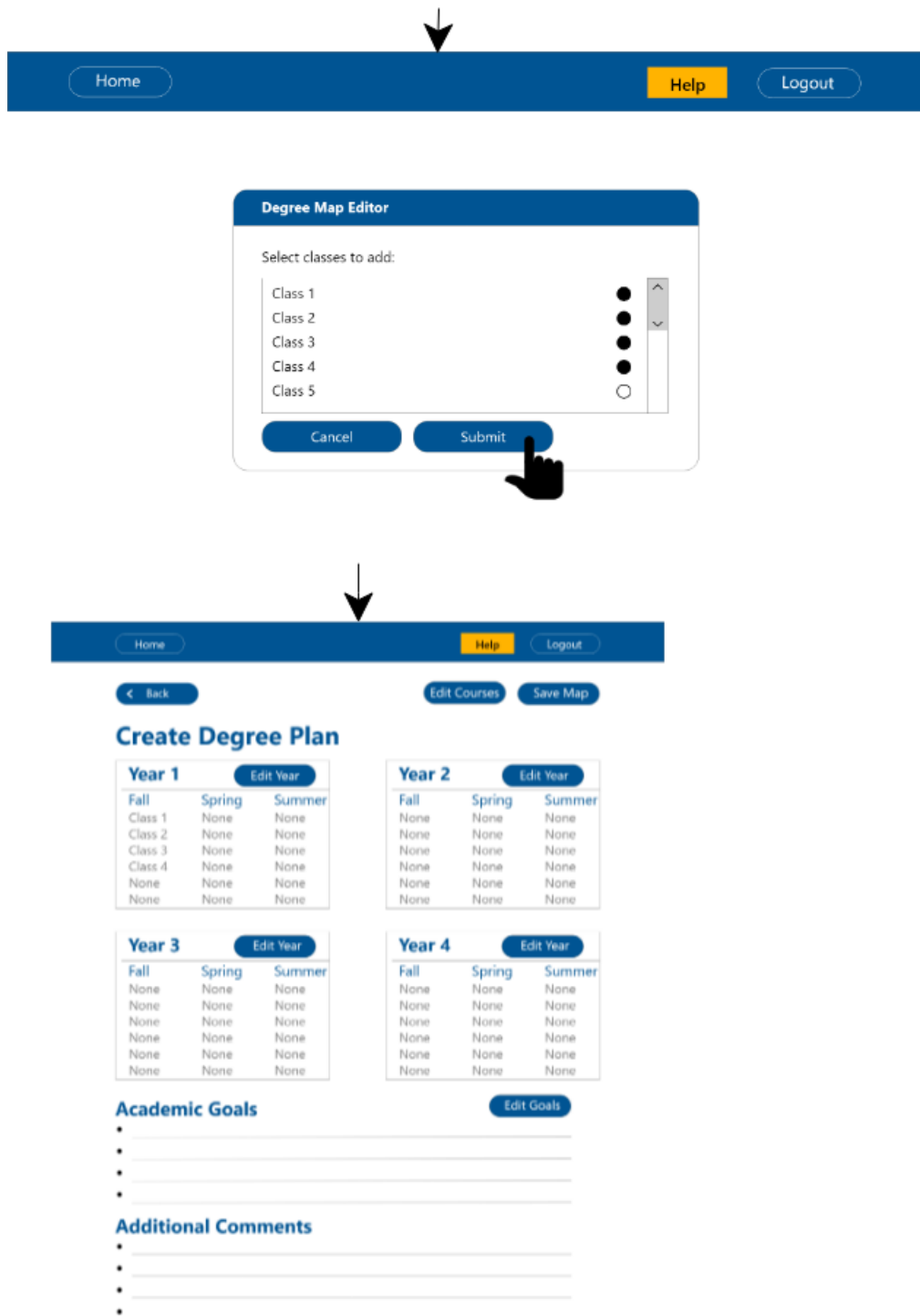


Figure 37: UI Model 7.1 Creating Student Degree Map

### 8.15 UC-7.2: Modifying Student Degree Map

**Description** This use case outlines the steps taken by an academic advisor to modify an student's existing degree map.

Actors	User (academic advisor)
Preconditions	<ul style="list-style-type: none"><li>• User is logged in to Eduvise.</li><li>• User is viewing a student's degree map</li></ul>
Steps	<ol style="list-style-type: none"><li>1 User selects the class from the degree map that they wish to change.</li><li>2 User changes or deletes the selected class.</li><li>3 User selects option to save changes to the degree map.</li><li>4 The degree map on the student's profile is updated.</li></ol>
Success conditions	<ul style="list-style-type: none"><li>• The student's degree map has been updated correctly in Eduvise.</li></ul>
Alternate path	<ol style="list-style-type: none"><li>4.1 User selects option to cancel changes to degree map.</li><li>4.2 The degree map reverts to its original state.</li></ol>
Requirements	REQ-7.1, REQ-7.2, REQ-7.3, EIR-4.1

Table 16: Use Case 7.2 - Modifying Student Degree Map

## Sequence Diagram

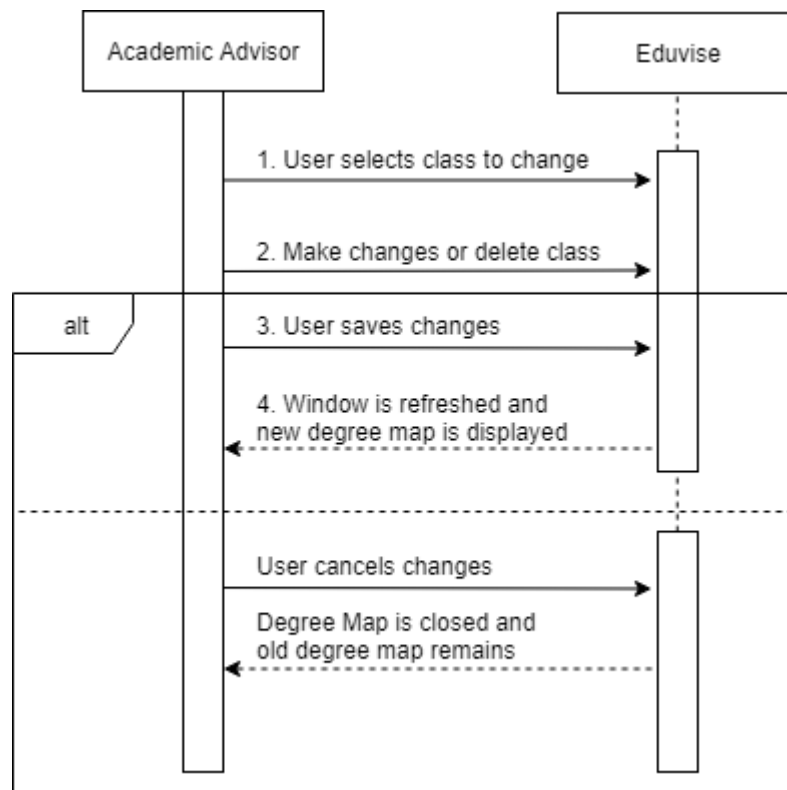


Figure 38: Sequence Diagram 7.2 - Modifying Student Degree Map

## User Interface Model

The scenario shown in UI Model 7.2 outlines the steps taken by an advisor when editing an existing student degree map. In this scenario, the advisor is already viewing the student degree map that they wish to edit. The advisor selects one of the courses and is prompted with a modal dialog asking if they would like to delete or replace that course. If they choose to replace a course, they are provided with a list of courses available for that term to replace the course. The advisor selects one of the courses, and the course is updated on the student degree map. If they choose to delete a course, they are notified that the course has been successfully deleted from that student degree map. The user is then redirected to the student degree map, now without the deleted course.

[Home](#)
[Help](#)
[Logout](#)

[< Back](#)
[Add Courses](#)

### Degree Map

**Year 1**

Fall	Spring	Summer
Class 1	None	None
Class 2	None	None
Class 3	None	None
Class 4	None	None
Class 5	None	None
None	None	None

**Year 2**

Fall	Spring	Summer
None	None	None
None	None	None
None	None	None
None	None	None
None	None	None
None	None	None

**Year 3**

Fall	Spring	Summer
None	None	None
None	None	None
None	None	None
None	None	None
None	None	None
None	None	None

**Year 4**

Fall	Spring	Summer
None	None	None
None	None	None
None	None	None
None	None	None
None	None	None
None	None	None

**Academic Goals**

Edit Goals

**Additional Comments**

[Home](#)
[Help](#)
[Logout](#)

[< Back](#)
[Add Courses](#)

### Degree

**Year 1**

Fall	Spring	Summer
Class 1	None	None
Class 2	None	None
Class 3	None	None
Class 4	None	None
Class 5	None	None
None	None	None

**Year 3**

Fall	Spring	Summer
None	None	None
None	None	None
None	None	None
None	None	None
None	None	None
None	None	None

**Year 4**

Fall	Spring	Summer
None	None	None
None	None	None
None	None	None
None	None	None
None	None	None
None	None	None

**Academic Goals**

Edit Goals

**Additional Comments**

Delete Course →

[Home](#)
[Help](#)
[Logout](#)

[< Back](#)
[Add Courses](#)

### Degree

**Year 1**

Fall	Spring	Summer
None	None	None
Class 2	None	None
Class 3	None	None
Class 4	None	None
Class 5	None	None
None	None	None

**Year 3**

Fall	Spring	Summer
None	None	None
None	None	None
None	None	None
None	None	None
None	None	None
None	None	None

**Year 4**

Fall	Spring	Summer
None	None	None
None	None	None
None	None	None
None	None	None
None	None	None
None	None	None

**Academic Goals**

Edit Goals

**Additional Comments**

Replace Course



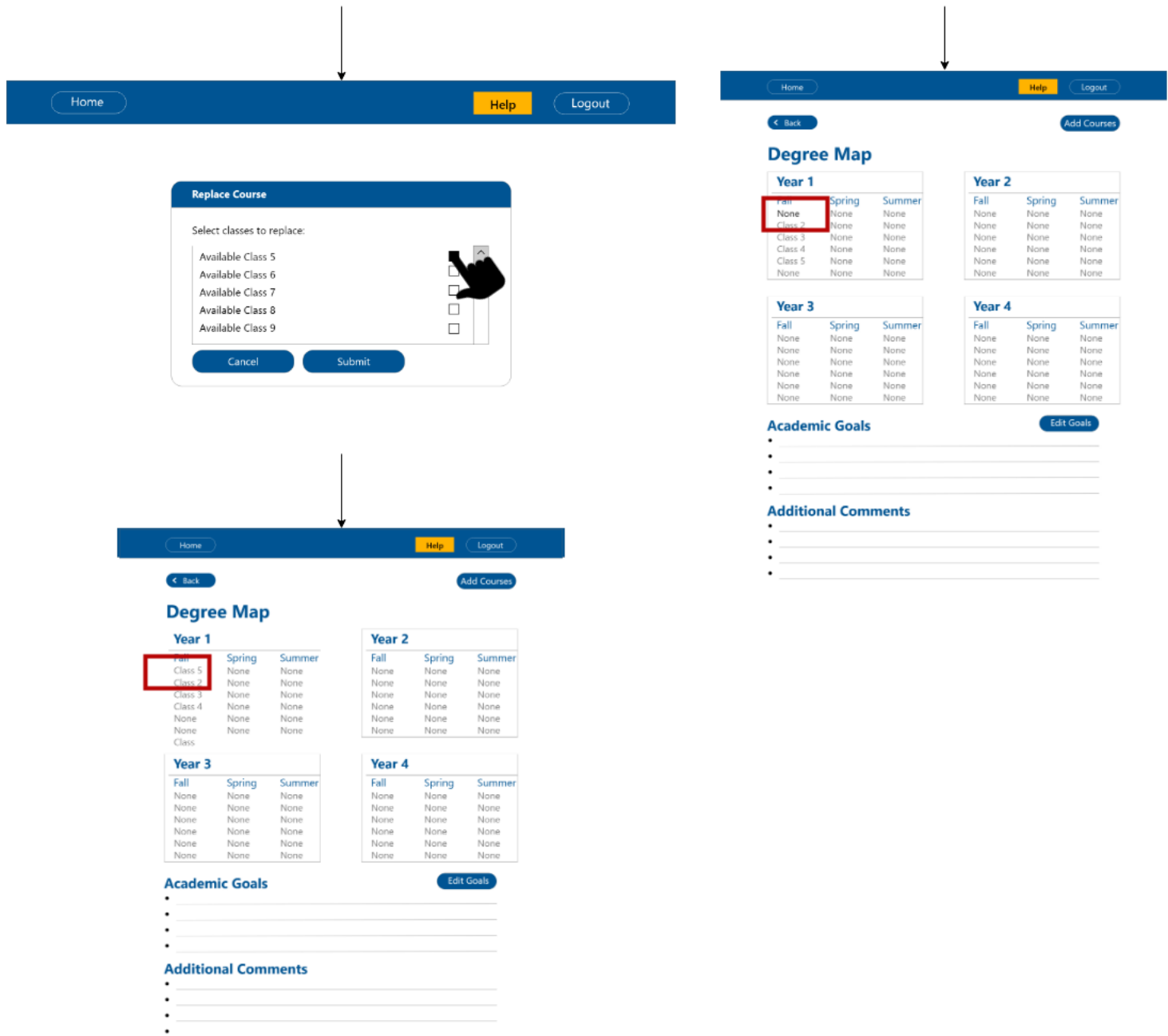


Figure 39: UI Model 7.2 - Modifying Student Degree Map

## 8.16 UC-8.1: Logging in

**Description** To ensure data integrity and confidentiality, as well as ensure authentic usage of the service, users of Eduvise must first verify their authenticity using login functionality to gain access to Eduvise. This login feature will remain consistent with other UVic services by using the user's existing NetLink ID and password to access Eduvise. An academic advisor or student must use this feature for access, but referees can receive limited access to Eduvise through the process outlined in UC-2.3.

Actors	User (academic advisor, student)
Preconditions	<ul style="list-style-type: none"><li>• User has a valid NetLink ID (account)</li></ul>
Steps	<ol style="list-style-type: none"><li>1 User navigates to the Eduvise login.</li><li>2 User inputs NetLink ID and NetLink password.</li><li>3 User submits credentials for verification.</li><li>4 User is taken to the user's Eduvise home page.</li></ol>
Success conditions	<ul style="list-style-type: none"><li>• The user credentials are successfully verified and the user's home page is displayed to the user.</li></ul>
Alternate path	<ol style="list-style-type: none"><li>3.1 User credentials are not verified</li><li>3.2 User is notified that credentials have not been verified.</li><li>3.3 The use case restarts at step 1.</li></ol>
Requirements	None

Table 17: Use Case 8.1 - Logging in

## Sequence Diagram

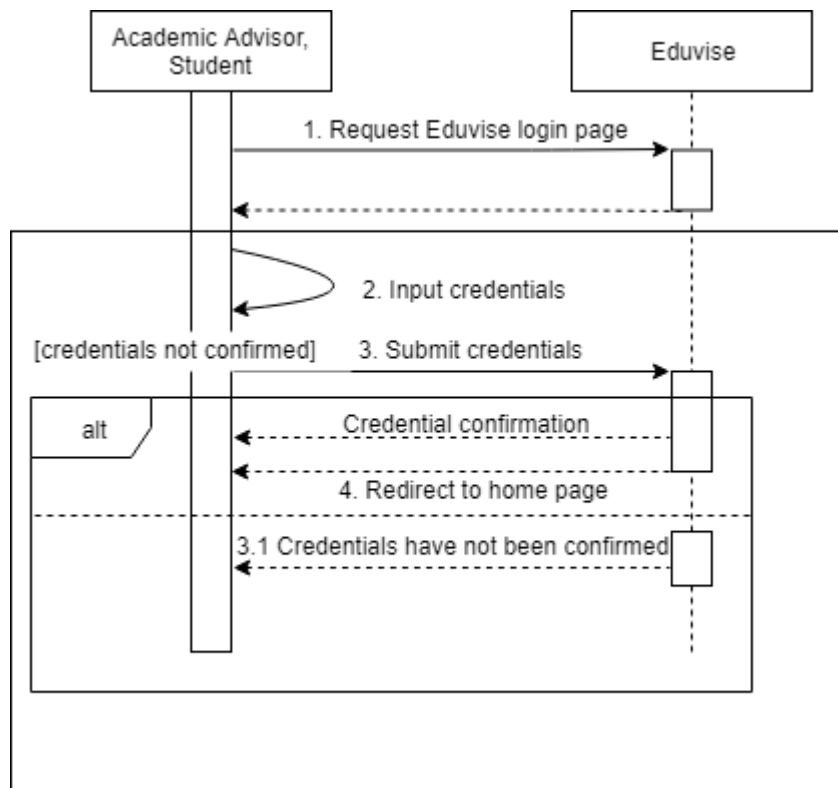


Figure 40: Sequence Diagram 8.1 - Logging In

## User Interface Model

The scenario shown in UI Model 8.1 outlines the steps taken to log into the Eduvise system. In this scenario the advisor or student first enters their Netlink ID and password into the input form, and then presses the submit button. There are two end states, on the left is the academic advisor home page, and on the right is the student home page.

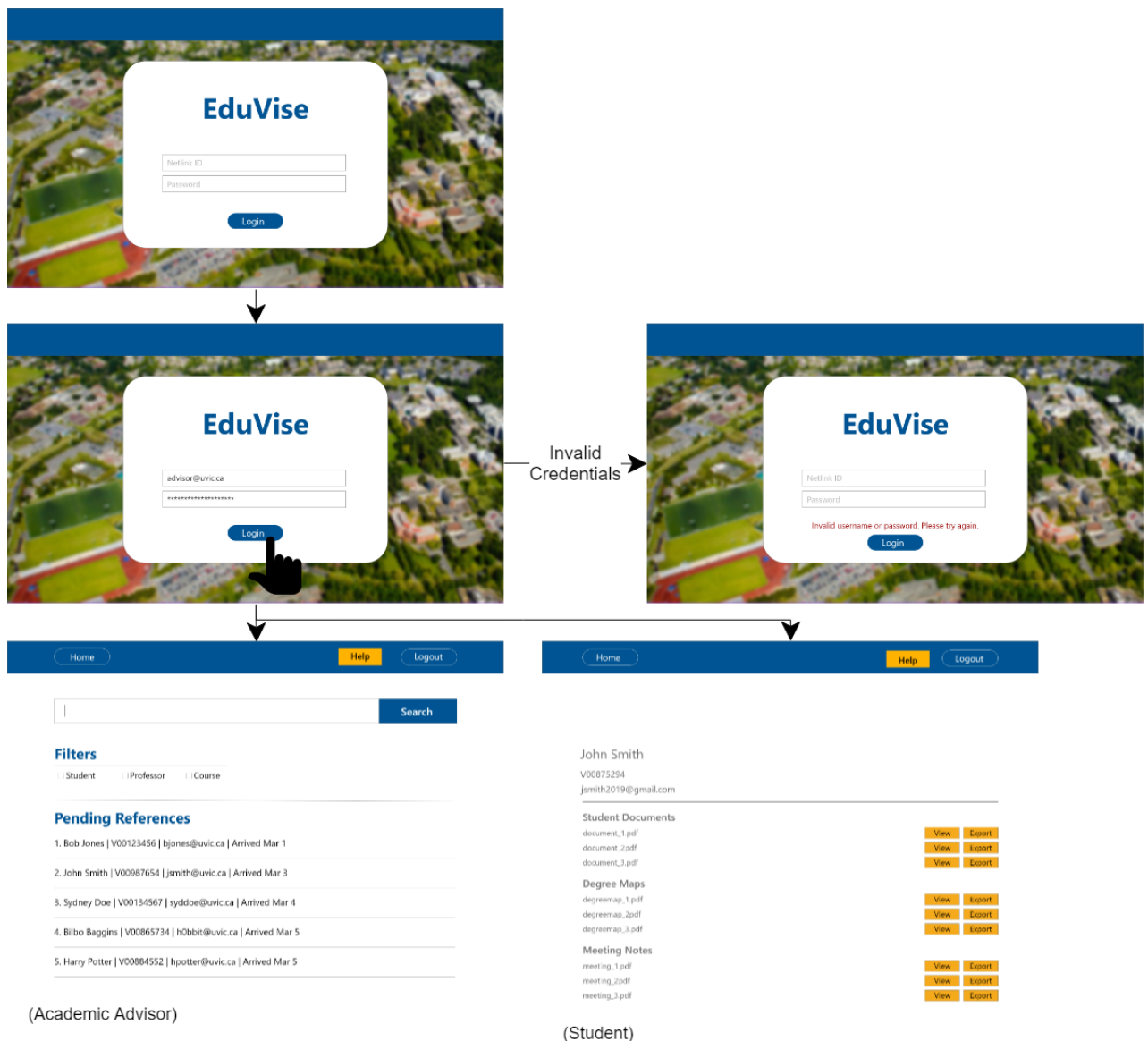


Figure 41: UI Model 8.1 Logging In

## 9 Forward Traceability

This section presents test cases that are sufficient to verify the correctness of an implementation of the requirements of Eduvise. A traceability matrix is also provided for each test suite, to allow forward traceability of these requirements in further documents or an implementation of Eduvise.

The following section contains the test cases needed for the verification of an implementation of Eduvise. The test cases are organized into test suites of test cases that test related functionality. Additionally we provide a test matrix for each test suite to display the dependency between requirements and tests.

In the following section, we start many test cases with the phrase "as an academic advisor" or "as a student". These two phrases are used to denote that the test case should be performed while the

tester is logged in to the system with the NetLink ID of an academic advisor or a Faculty of Education student, respectively.

## **9.1 Test Suite 1 - Searching**

This test suite covers tests related to the system feature for searching for information in Eduvise.

**TC-1.1:** As an academic advisor, search for a student by name. Verify that the expected student is returned by the search functionality.

**TC-1.2:** As an academic advisor, search for a student by NetLink ID. Verify that the expected student is returned by the search functionality.

**TC-1.3:** As an academic advisor, search for a student by student number. Verify that the expected student is returned by the search functionality.

**TC-1.4:** As an academic advisor, search for a professor by name. Verify that the expected professor is returned by the search functionality.

**TC-1.5:** As an academic advisor, search for professors by faculty. Verify that every professor returned by the search functionality is in the faculty.

**TC-1.6:** As an academic advisor, search for professors by courses taught. Verify that the professors returned by the search functionality include each current professor who has taught the course.

**TC-1.7:** As an academic advisor, search for a course by course code. Verify that the expected course is returned by the search functionality.

**TC-1.8:** As an academic advisor, search for courses by department. Verify that the courses returned by the search functionality matches the list of all courses offered by the department.

System Feature: Searching	Priority	TC-1.1	TC-1.2	TC-1.3	TC-1.4	TC-1.5	TC-1.6	TC-1.7	TC-1.8
REQ-1.1	High	X	X	X					
REQ-1.2	High				X	X	X		
REQ-1.3	High							X	X

Figure 42: Searching Test Matrix

## 9.2 Test Suite 2 - Adding Documents

This test suite covers tests related to the system feature for adding academic advising documents to Eduvise.

**TC-2.1:** As an academic advisor, upload a document to a student's profile, with the option selected for the file to not be visible to the student. Access the student's profile. Verify that the document has been added to the profile. Verify that the document can be viewed. As the student, access the student's profile. Verify that the document is not visible.

**TC-2.2:** As an academic advisor, request a referral for a student from an accessible email address. Access the email account. Verify that the email has been received. Follow the link in the email. Verify that the referral page is reached. Upload a document to the referral page. Submit the referral. As the academic advisor, verify that the referral is visible but not accepted. Accept the referral. Verify that the correct referral document has been added to the student's profile. As the student, access the student's profile. Verify that the referral document is not visible.

**TC-2.3:** As an academic advisor, upload a document to a student's profile, with the option selected for the file to be visible to the student. Access the student's profile. Verify that the document has been added to the profile. Verify that the document can be viewed. As the student, access the student's profile. Verify that the document can be viewed.

System Feature: Adding Academic Advising Documents	Priority	TC-2.1	TC-2.2	TC-2.3
REQ-2.1	High	X		
REQ-2.2	High		X	
REQ-2.3	High			X

Figure 43: Adding Documents Test Matrix

### 9.3 Test Suite 3 - Document Modification

This test suite covers tests related to the system feature for modifying academic advising documents in Eduvise.

**TC-3.1:** As an academic advisor, use the replace functionality to replace an existing academic advising document with a different document. View the document. Verify that the document has been replaced with the new document.

**TC-3.2:** As an academic advisor, add an annotation on an existing academic advising document. Save the changes. Close the document. View the document. Verify that the annotation exists.

**TC-3.3:** As an academic advisor, enter edit mode on an annotation on an existing academic advising document. At the same time, as a different academic advisor, attempt to enter edit mode on an annotation on the same document. Verify that the second academic advisor is unable to enter edit mode.

**TC-3.4:** As an academic advisor, enter edit mode on an annotation on an existing academic advising document. Edit the annotation. Save the changes. Close the document. View the document. Verify that the annotation is now the edited annotation.

**TC-3.5:** As an academic advisor, access a student profile with an existing academic advising document. Set the visibility of a document so that the document can be viewed by the student. As the student, access the student's profile. Verify that the document can be viewed.

**TC-3.6:** As an academic advisor, access a student profile with an existing academic advising document. Set the visibility of a document so that the document cannot be viewed by the student. As

the student, access the student's profile. Verify that the document is not visible.

System Feature: Academic Advising Document Modification	Priority	TC-3.1	TC-3.2	TC-3.3	TC-3.4	TC-3.5	TC-3.6
REQ-3.1	High	X					
REQ-3.2	High		X				
REQ-3.3	High			X			
REQ-3.4	High				X		
REQ-3.4	High					X	X

Figure 44: Document Modification Test Matrix

#### 9.4 Test Suite 4 - Document Export

This test suite covers tests related to the system feature for exporting academic advising documents from Eduvise.

**TC-4.1:** As an academic advisor, export a document pertaining to a student. Verify that the exported document matches the document in Eduvise. Verify that the exported document is able to be printed.

**TC-4.2:** As a student, export a document pertaining to the student. Verify that the exported document matches the document in Eduvise. Verify that the exported document is able to be printed.



System Feature: Academic Advising Document Export	Priority	TC-4.1	TC-4.2
REQ-4.1	Medium	X	
REQ-4.2	Medium		X

Figure 45: Document Export Test Matrix

### 9.5 Test Suite 5 - Document Viewing By Students

This test suite covers tests related to the system feature for viewing academic advising documents as a student.

**TC-5.1:** As a student, view the student's degree map. Verify that the degree map can be viewed.

**TC-5.2:** Perform TC-3.6.

**TC-5.3:** As a student, attempt to access a different student's profile. Verify that the other student's profile is not accessible.

System Feature: Academic Advising Document Viewing By Student	Priority	TC-5.1	TC-5.2	TC-5.3
REQ-5.1	Low	X		
REQ-5.2	Low		X	
REQ-5.3	Low			X

Figure 46: Student Document Viewing Test Matrix

## 9.6 Test Suite 6 - Document Viewing By Academic Advisors

This test suite covers tests related to the system feature for viewing academic advising documents as an academic advisor.

**TC-6.1:** As an academic advisor, attempt to view a student's degree map. Verify that the academic advisor can view the degree map.

**TC-6.2:** As an academic advisor, add an annotation on an existing academic advising document. Save the changes. As a different academic advisor, view the same document. Verify that the annotations on the document match the annotations added by the first academic advisor.

**TC-6.3:** As an academic advisor, view an academic advising document on a student's profile. Verify that the document can be viewed.

System Feature: Academic Advsing Document Viewing By Academic Advisors	Priority	TC-6.1	TC-6.2	TC-6.3
REQ-6.1	High	X		
REQ-6.2	High		X	
REQ-6.3	High			X

Figure 47: Advisor Document Viewing Test Matrix

## 9.7 Test Suite 7 - Student Degree Planning

This test suite covers tests related to the system feature for planning a student's degree using Eduvise.

**TC-7.1:** As an academic advisor, view a student's degree map. Verify that the student's goals can be viewed.

**TC-7.2:** As an academic advisor, view a student's degree map. Add a goal to the degree map for that student. Save the changes. Close the degree map. As a different academic advisor, view the

student's degree map. Verify that the added goal is visible.

**TC-7.3:** As an academic advisor, view a student's degree map. Edit an existing goal on the degree map for that student. Save the changes. Close the degree map. View the student's degree map. Verify that the edited goal is visible.

**TC-7.4:** As an academic advisor, create a new degree map for a student. Add a course to a term. Change the course to a different course. Add a course to a different term. Save the changes. Close the degree map. View the degree map and verify that the viewed degree map matches the created degree map.

**TC-7.5:** As an academic advisor, view a degree map for a student. Change a course to a different course. Remove a course. Add a course. Save the changes. Close the degree map. View the degree map. Verify that the content of the viewed degree map matches the changes made to the degree map.

System Feature: Student Degree Planning	Priority	TC-7.1	TC-7.2	TC-7.3	TC-7.4	TC-7.5
REQ-7.1	Low	X				
REQ-7.2	Low		X			
REQ-7.3	Low			X		
REQ-7.4	Low				X	
REQ-7.5	Low					X

Figure 48: Degree Planning Test Matrix

## 9.8 Test Suite 8 - Authentication

This test suite covers tests related to the system feature for authenticating users of Eduvise.

**TC-8.1:** Navigate to the Eduvise login screen. Input an academic advisor's NetLink credentials. Verify that the academic advisor is authenticated and taken to the academic advising home page.

**TC-8.2:** Navigate to the Eduvise login screen. Input a student's NetLink credentials. Verify that the

student is authenticated and taken to the student's profile.

**TC-8.3:** Navigate to the Eduvise login screen. Input credentials that do not match any user's NetLink credentials. Verify that the user is not authenticated. Verify that the user is prompted to input correct credentials.

System Feature: Authentication	Priority	TC-8.1	TC-8.2	TC-8.3
REQ-8.1	High	X	X	X

Figure 49: Authentication Test Matrix

## 9.9 Test Suite 9 - External Interfaces

This test suite covers tests related to external interfaces that Eduvise requires for correct operation.

**TC-9.1:** As an academic advisor, verify that each screen has access to a "help" button. View the help menu. Verify that the help menu is related to the current screen content.

**TC-9.2:** Access Eduvise as an academic advisor using each of the devices required by the Faculty of Education's academic advising office (2009 20" iMac, 2011 21.5" iMac, 2011 27" iMac, 2013 27" iMac). Verify that Eduvise operates without issue on each of the devices.

**TC-9.3:** Perform TC-8.1. Perform TC-8.2. Perform TC-8.3.

**TC-9.4:** As an academic advisor, access Eduvise using each of the operating systems required by the Faculty of Education's academic advising office (Windows 7, Windows 10, OS X 10.4, macOS 10.12). Verify that Eduvise operates without issue on each of the operating systems.

**TC-9.5:** As an academic advisor, submit a document of each of the required file types (.doc, .docx, .odt, .txt, .pdf) to a student's profile. For each document, do the following:

- i. View the document.
- ii. Verify the viewed document matches the submitted document.

iii. Export the document.

iv. Verify that the exported document matches the submitted document.

**TC-9.6:** As an academic advisor, request a referral from an accessible email address. Verify that the email address receives a referral request as an email.

**TC-9.7:** As an academic advisor, navigate to a student's profile. Verify that each of the required pieces of student information, as specified in EIR-3.2, are accessible.

External Interface Requirements	Priority	TC-9.1	TC-9.2	TC-9.3	TC-9.4	TC-9.5	TC-9.6	TC-9.7
EIR-1.1	Low	X						
EIR-2.1	High		X					
EIR-3.1	High			X				
EIR-3.2	Medium							X
EIR-3.3	High				X			
EIR-3.4	High					X		
EIR-4.1	Medium						X	

Figure 50: External Requirements Test Matrix

## 9.10 Test Suite 10 - Non-Functional Requirements

This test suite covers tests that verify satisfaction of non-functional requirements.

**TC-10.1:** Create a student profile for each student in the Faculty of Education. Add every academic advising documents pertaining to each student. Verify that every academic advising document is accessible in Eduvise. Verify that performance is not compromised, as per TC-10.2, TC-10.3, and TC-10.4.

**TC-10.2:** Simulate 200 users concurrently authenticating themselves to Eduvise. Verify that no user's authentication takes more than 3 seconds to complete.

**TC-10.3:** Simulate 200 users each concurrently accessing a different document in Eduvise. Verify

that no document access takes more than 3 seconds to access.

**TC-10.4:** Simulate 200 users concurrently accessing Eduvise. Modify an academic advising document. Save the changes. Close the document. 10 seconds later, view the same academic advising document. Verify that the viewed document matches the modified document.

**TC-10.5:** Verify that a backup of all Eduvise data exists. Verify that all data in Eduvise can be restored from this backup.

**TC-10.6:** Perform an internal or external audit of Eduvise. Verify that Eduvise complies with BC's Freedom of Information and Protection of Privacy Act.

**TC-10.7:** Perform an internal or external audit of Eduvise. Verify that Eduvise complies with UVic's Protection of Privacy Policy.

**TC-10.8:** Perform an internal or external audit of Eduvise. Verify that Eduvise complies with UVic's Records Management Policy.

**TC-10.9:** Train a new academic advisor for 2 hours in how to use Eduvise. Verify that the new academic advisor is able to complete authentication, search, document upload, document viewing, annotation, document replacing, and document exporting operations in Eduvise.

**TC-10.10:** Record the amount of technical support each academic advisor requires when using Eduvise. Verify that the average amount of support is no more than two hours per month for each academic advisor.

**TC-10.11:** Observe a student using Eduvise and instruct them to access and export an academic advising document. Verify that they are able to perform each task within 30 seconds without any assistance.

**TC-10.12:** Record Eduvise up-time. Verify that Eduvise is accessible during working hours with 99.9% confidence.

**TC-10.13:** Record the number of lines of code that Eduvise unit tests cover. Verify that the number of lines of code covered by unit tests is no less than 70% of the total number of lines of code in the Eduvise implementation.

**TC-10.14:** Perform a penetration test on Eduvise. Verify that no documents were accessed in the penetration test.

Non-functional Requirements	Priority	TC-10.1	TC-10.2	TC-10.3	TC-10.4	TC-10.5	TC-10.6	TC-10.7	TC-10.8	TC-10.9	TC-10.10	TC-10.11	TC-10.12	TC-10.13	TC-10.14
NFR-1.1	High	X													
NFR-1.2	Low		X	X	X										
NFR-1.3	Low		X												
NFR-1.4	Low			X											
NFR-1.5	Medium				X										
NFR-2.1	High					X									
NFR-3.1	High						X								
NFR-3.2	High							X							
NFR-3.3	High								X						
NFR-3.4	High														X
NFR-4.1	Medium									X					
NFR-4.2	Medium										X				
NFR-4.3	Low											X			
NFR-4.4	High												X		
NFR-4.5	Low													X	

Figure 51: Non-Functional Requirements Test Matrix

## 9.11 Test Suite 11 - Other

This test suite covers tests that are not covered in other test suites.

**TC-11.1:** Perform an internal or external audit of Eduvise. Verify that a document cannot be removed from Eduvise until a minimum of 3 years after the student that the document pertains to has left UVic.

**TC-11.2:** Disconnect from the UVic telecommunications network. Connect to a different telecommunications network. Login to Eduvise as a student. Verify that the student is able to access their home page. Attempt to view an academic advising document. Verify that the academic advising document is able to be viewed.

Other Requirements	Priority	TC-11.1	TC-11.2
OREQ-1	High	X	
OREQ-2	Low		X

Figure 52: Other Requirements Test Matrix



## References

- [1] University of Victoria, "Protection of privacy and access to information - university of victoria." <https://www.uvic.ca/vpfo/departments/privacy/policies-and-procedures/index.php>, 2019. [Online; accessed 05-February-2019].
- [2] University of Victoria, "Statistics." <https://web.uvic.ca/calendar2019-01/cal/uvic/statistics.html>, 2019. [Online; accessed 05-February-2019].
- [3] BC Legislature, "Freedom of information and protection of privacy act." [http://www.bclaws.ca/Recon/document/ID/freeside/96165\\_00](http://www.bclaws.ca/Recon/document/ID/freeside/96165_00), 1996. [Online; accessed 05-February-2019].
- [4] University of Victoria, "Protection of privacy policy." <https://www.uvic.ca/universitysecretary/assets/docs/policies/GV0235.pdf>, 2019. [Online; accessed 05-February-2019].
- [5] University of Victoria, "Records management policy." <https://www.uvic.ca/universitysecretary/assets/docs/policies/IM7700.pdf>, 2019. [Online; accessed 05-February-2019].