Leap Designs

As requested by QED Technologies:
Eduvise

Authors:
Adam KWAN
Ahnaf AHMED
Austin SMITH
Darian SAMPARE
Hamzah MANSOUR
Jamie ST MARTIN
Sterling LAIRD

2019-03-20

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

Contents

Revision History 1

Table of Contents 2

1 Introduction 9
- 1.1 Purpose . . . 9
- 1.2 Project Scope . . . 9
- 1.3 Overview . . . 9

2 Overall Description 10
- 2.1 Product Perspective . . . 10
- 2.2 Product Features . . . 10
- 2.3 User Classes and Characteristics . . . 11
  - 2.3.1 Academic Advisor . . . 11
  - 2.3.2 Student . . . 11
  - 2.3.3 Referee . . . 11
- 2.4 Operating Environment . . . 11
- 2.5 Design and Implementation Constraints . . . 12
- 2.6 Assumption and Dependencies . . . 12

3 System Features 12
- 3.1 Searching . . . 12
  - 3.1.1 Description and Priority . . . 12
  - 3.1.2 Functional Requirements . . . 12
- 3.2 Adding Academic Advising Documents . . . 13
  - 3.2.1 Description and Priority . . . 13
  - 3.2.2 Functional Requirements . . . 13
- 3.3 Academic Advising Document Modification . . . 13
  - 3.3.1 Description and Priority . . . 13
  - 3.3.2 Functional Requirements . . . 13
- 3.4 Academic Advising Document Export . . . 14
  - 3.4.1 Description and Priority . . . 14
  - 3.4.2 Functional Requirements . . . 14
- 3.5 Academic Advising Document Viewing By Students . . . 14
  - 3.5.1 Description and Priority . . . 14

3.5.2 Functional Requirements . . . 14
3.6 Academic Advising Document Viewing By Academic Advisors . . . 14
3.6.1 Description and Priority . . . 14
3.6.2 Functional Requirements . . . 15
3.7 Student Degree Planning . . . 15
3.7.1 Description and Priority . . . 15
3.7.2 Functional Requirements . . . 15
3.8 Authentication . . . 15
3.8.1 Description and Priority . . . 15
3.8.2 Functional Requirements . . . 15

4 External Interface Requirements 15
4.1 User Interfaces . . . 16
4.2 Hardware Interfaces . . . 16
4.3 Software Interfaces . . . 16
4.4 Communications Interfaces . . . 17

5 Other Non-Functional Requirements 17
5.1 Performance Requirements . . . 17
5.1.1 Capacity . . . 17
5.1.2 Speed . . . 17
5.2 Safety Requirements . . . 17
5.3 Security Requirements . . . 17
5.4 Software Quality Attributes . . . 18
5.4.1 Usability . . . 18
5.4.2 Availability . . . 18
5.4.3 Testability . . . 18

6 Other Requirements 18

7 Diagrams 19
7.1 Data Dictionary . . . 19
7.2 Data Flow Diagrams . . . 21
7.2.1 Context Diagram . . . 22
7.2.2 DFD Level 1 . . . 23
7.2.3 DFD Level 2 Overview . . . 24

7.3 Entity Relationship Diagram 29

8 Use Cases **29**

8.1 Use Case Model 29
8.2 UC-1.1: Searching 31
8.3 UC-2.1: Adding Academic Advising Document 35
8.4 UC-2.2: Requesting Referral 39
8.5 UC-2.3: Submitting Referral 42
8.6 UC-2.4: Approving Referral 45
8.7 UC-3.1: Replacing Academic Advising Document 48
8.8 UC-3.2: Adding Annotation 51
8.9 UC-3.3: Modifying Annotation 56
8.10 UC-3.4: Modifying Document Access 60
8.11 UC-4.1: Document Exporting 62
8.12 UC-5.1: Document Viewing 66
8.13 UC-5.2: Getting Help 68
8.14 UC-7.1: Creating Student Degree Map 70
8.15 UC-7.2: Modifying Student Degree Map 76
8.16 UC-8.1: Logging in 80

List of Tables

1 Glossary 8
2 Data Dictionary 20
3 Use Case 1.1 - Searching 31
4 Use Case 2.1 - Adding Academic Advising Document 35
5 Use Case 2.2 - Requesting Referral 39
6 Use Case 2.3 - Submitting Referral 42
7 Use Case 2.4 - Approving Referral 45
8 Use Case 3.1 - Replacing Academic Advising Document 48
9 Use Case 3.2 - Adding Annotation 51
10 Use Case 3.3 - Modifying Annotation 56
11 Use Case 3.4 - Modifying Document Access 60
12 Use Case 4.1 - Document Exporting 63
13 Use Case 5.1 - Document Viewing 66
14 Use Case 5.2 - Getting Help 68
15 Use Case 7.1 - Creating Student Degree Map 71
16 Use Case 7.2 - Modifying Student Degree Map 76
17 Use Case 8.1 - Logging in 80

List of Figures

1 Data Flow Diagram Legend 21
2 Context Diagram 22
3 Data Flow Diagram Level 1 23
4 Data Flow Diagram Level 2 24
5 DFD Level 2: Top Left 25
6 DFD Level 2: Top Right 26

7 DFD Level 2: Bottom Left 27
8 DFD Level 2: Bottom Right 28
9 Entity Relationship Diagram 29
10 Use Case Model 30
11 Sequence Diagram 1.1 - Searching 32
12 UI Model 1.1 - Searching 34
13 Sequence Diagram 2.1 - Adding Academic Advising Document 36
14 UI Model 2.1 - Adding Academic Advising Document 38
15 Sequence Diagram 2.2 - Requesting Referral 40
16 UI Model 2.2 - Requesting Referral 41
17 Sequence Diagram 2.3 - Submitting Referral 43
18 UI Model 2.3 - Submitting Referral 44
19 Sequence Diagram 2.4 - Approving Referral 46
20 UI Model 2.4 - Approving Referral 47
21 Sequence Diagram 3.1 - Replacing Academic Advising Document 49
22 UI Model 3.1 - Replace Academic Advising Document 50
23 Sequence Diagram 3.2 - Adding Annotation 52
24 UI Model 3.2 - Adding Annotation 55
25 Sequence Diagram 3.3 - Modifying Annotation 57
26 UI Model 3.3 - Modifying Annotation 59
27 Sequence Diagram 3.4 - Modifying Document Access 61
28 UI Model 3.4 - Modifying Document Access 62
29 Sequence Diagram 4.1 - Document Exporting 64
30 UI Model 4.1 - Document Exporting 65
31 Sequence Diagram 5.1 - Document Viewing 66
32 UI Model 5.1 - Document Viewing 67
33 Sequence Diagram 5.2 - Getting Help 68

34 UI Model 5.2 - Getting Help 70
35 Sequence Diagram 7.1 - Creating Student Degree Map 72
36 UI Model 7.1 Creating Student Degree Map 75
37 Sequence Diagram 7.2 - Modifying Student Degree Map 77
38 UI Model 7.2 - Modifying Student Degree Map 79
39 Sequence Diagram 8.1 - Logging In 81
40 UI Model 8.1 Logging In 82

Glossary of Terms

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 file created by or uploaded to Eduvise each of which pertains to and is associated with a student.
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	Reference letter.
Student number	A unique numeric identifier given to each new student when enrolling at the University of Victoria in the form V00 followed by 6 other 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 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

This project'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 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, performance, and planning. 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, dependencies, and assumptions. 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 data dictionary, context diagram, data flow diagram level 1, data flow diagram level 2, and the entity relationship diagram. Section 8, "Use Cases", describes use

cases Eduvise and provides sequence diagrams and user interface models of the flow through each described use case.

2 Overall Description

2.1 Product Perspective

Eduvise is a software system that is currently being designed by LeapDesigns 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 progress towards 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

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

2.6 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 document 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.

REQ-1.2: An academic advisor must be able to search for each professor profile by name, faculty, or courses taught.
REQ-1.3: An academic advisor must be able to search for each course overview by course code, course number, or department.
REQ-1.4: An academic advisor must be able to expand or limit search results by using a search filter that restricts the search results to any combination of student profiles, professor profiles, and course overviews.

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

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 makes changes to documents, including completely replacing documents. These changes are 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.
REQ-3.2: An academic advisor must be able to add new annotations on an academic advising document.
REQ-3.3: An academic advisor must not be able to modify an academic advising document at the same time as any other academic advisor.

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

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 an easily printable or otherwise readable format.
REQ-4.2: A student must be able to export one or more documents that they have access to view in an easily printable or otherwise readable format.

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-3.4). 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 progress towards graduation, including specific goals for the student that have been set with an academic advisor.
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-3.4)
REQ-5.3: A student must not be able to view any document pertaining to a different student.
REQ-5.4: A student must be able to view each academic advising document accessible to that student without necessarily being within the UVic telecommunication network.

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.
REQ-6.2: An academic advisor must be able to view the average score achieved by all students in each class on a term-by-term basis, and be able to view the instructor(s) of each class on a term-by-term basis.
REQ-6.3: An academic advisor must be able to view each annotation added to each document by any academic advisor (see REQ-3.2).
REQ-6.4: An academic advisor must be able to view all correspondence between each student and each academic advisor.

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 add, view and edit progress towards specific goals for each student.
REQ-7.2: An academic advisor must be able to create and modify a personalized degree plan for each student.

3.8 Authentication

3.8.1 Description and Priority

The authentication feature will verify a student or an academic advisor's identity. 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 be granted access to the documents that they have permission to access only after authenticating their identity by providing their UVic NetLink credentials.

4 External Interface Requirements

4.1 User Interfaces

EIR-1.1: A user must always have access to any support information required to operate Eduvise.

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

4.3 Software Interfaces

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

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

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

- .doc
- .docx

- .odt
- .txt
- .pdf

4.4 Communications Interfaces

EIR-4.1: An academic advisor must be able to provide to a referee a method to submit a reference letter to Eduvise. This method must not provide access to any document stored by Eduvise. (see REQ-2.2)

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

5.1.2 Speed

NFR-1.3: A request for authentication of any user must be approved or declined within three seconds.
NFR-1.4: A request to view any academic advising document must be given a response within three seconds.
NFR-1.5: Changes made more than 10 seconds before an academic advising document is accessed must be reflected at the time of access.

5.2 Safety Requirements

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

5.3 Security Requirements

NFR-3.1: Eduvise must comply to BC's Freedom of Information and Protection of Privacy Act (FIPPA). [3]
NFR-3.2: Eduvise must comply to UVic's Protection of Privacy Policy. [4]
NFR-3.3: Eduvise must comply to UVic's Records Management Policy. [5]
NFR-3.4: No software system that is not Eduvise may have access to any academic advising document stored by Eduvise.

NFR-3.5: A student must not be able to access any academic advising document of any student other than themselves.

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.
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.
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 record that they have access to view.

5.4.2 Availability

NFR-4.5: 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.

5.4.3 Testability

NFR-4.6: Unit test coverage of Eduvise must be at least 70%.

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.

7 Diagrams

7.1 Data Dictionary

The Data Dictionary defines all data that is contained within the context diagram, DFD level 1, and DFD level 2, and Entity Relationship Diagram.

Annotated Document	Document from a student profile with attached annotations from an academic advisor.
Annotations	Text based notes attached to a document on a student's profile.
Approval Request	An email sent to an academic advisor notifying them a referee has submitted a reference letter.
Authorization	Response from the NetLink authentication server on the validity of a set of credentials.
Authorization Request	Request for credential authorization sent to the NetLink authentication server
Authorization Notification	UI update for an academic advisor following a successful login attempt.
Available Courses	A list of available courses for a selected term in a degree map
Course and Professor Search Results	Search results corresponding to a course or professor
Credentials	The NetLink ID and password of the user logging in.
Degree Map	The degree map includes classes to take by term, goals, and additional comments
Degree Map Changes	Text information into the student degree map template on a student's profile.
Document	Files that are associated with a specific student.
Document Approval	Approval of a reference letter sent by an academic advisor
Document Changes	Changes to a document including degree plan changes, annotations, and document approval
Export Request	A request to export a document from a specific student profile.
Non-existent Student Identification	Student Identification with no prior association to a student profile within Eduvise
Reference Letter	A document submitted by a referee
Referral Submission Request	A request sent by an academic advisor to tell Eduvise to send a submission link to a referee
Search Query	Keywords identifying a student, professor, or class. Examples would include names, NetLink IDs, student numbers, course codes, or faculty.

Search Results	A list of all students, professors, and classes that meet the search criteria.
Selected Student Profile	A student profile chosen by an academic advisor after a search
Student Degree Plan	A pdf document that contains a degree plan designed by an academic advisor in Eduvise
Student Identification	Student name and NetLink ID required to locate a student profile
Student Info	All student info corresponding to a specific student identification located in the Eduvise Internal Database (D2)
Student Profile	A UI Screen with a student's profile information.
Student Records	All student info corresponding to a specific student identification located in the UVic Internal Records Database (D1)
Student View Request	An authorized request from a student after successful login to view their own profile.
Submission Link	A link allowing a referee to submit a reference letter into Eduvise

Table 2: Data Dictionary

7.2 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. All data in the diagrams is defined above in 7.1 Data Dictionary. The following legend is used for the context diagram, data flow diagram (DFD) level 1, and DFD level 2.

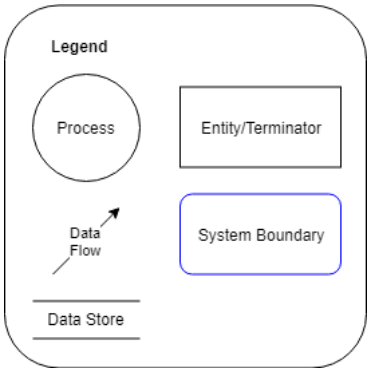


Figure 1: Data Flow Diagram Legend

7.2.1 Context Diagram

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

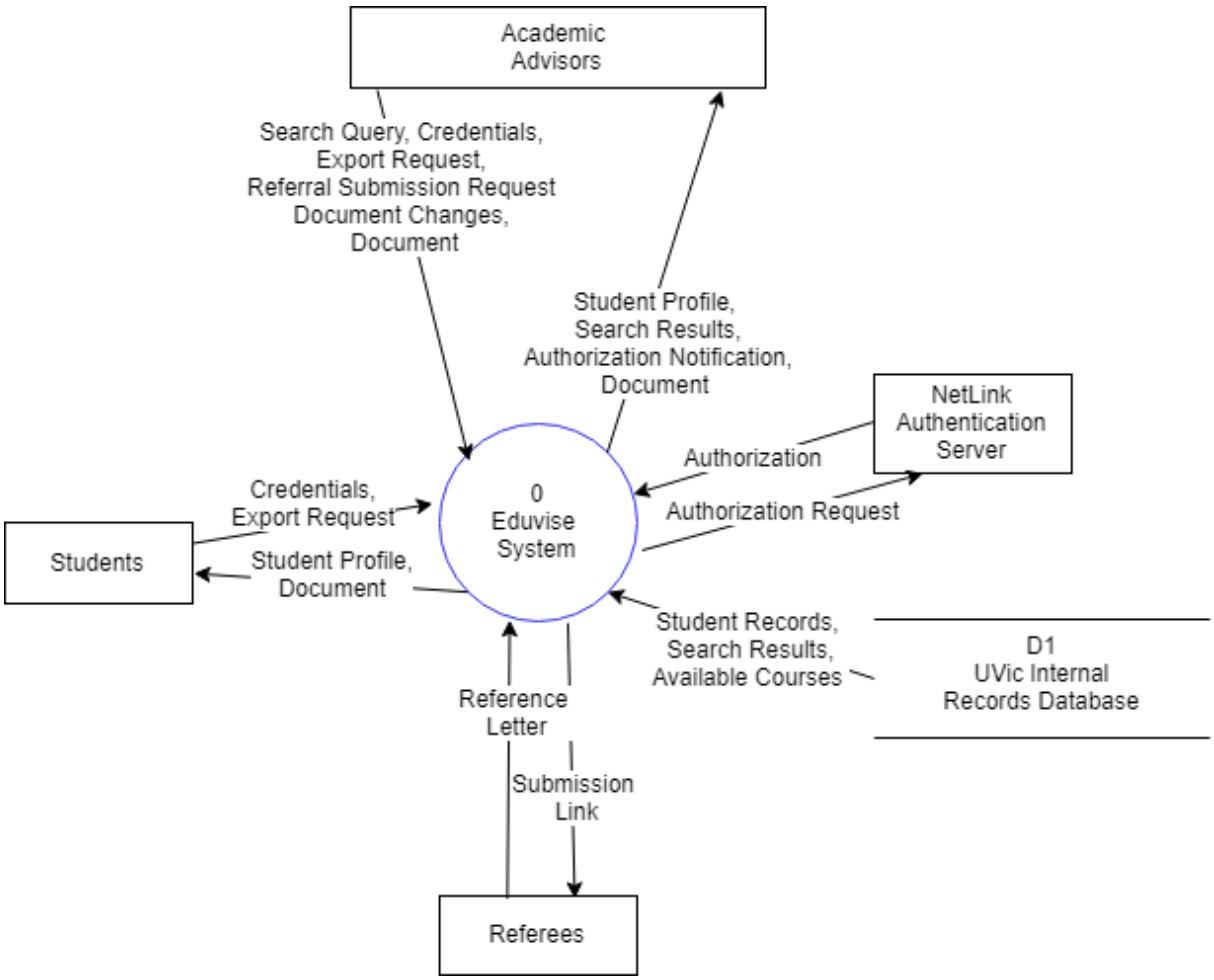


Figure 2: Context Diagram

7.2.2 DFD Level 1

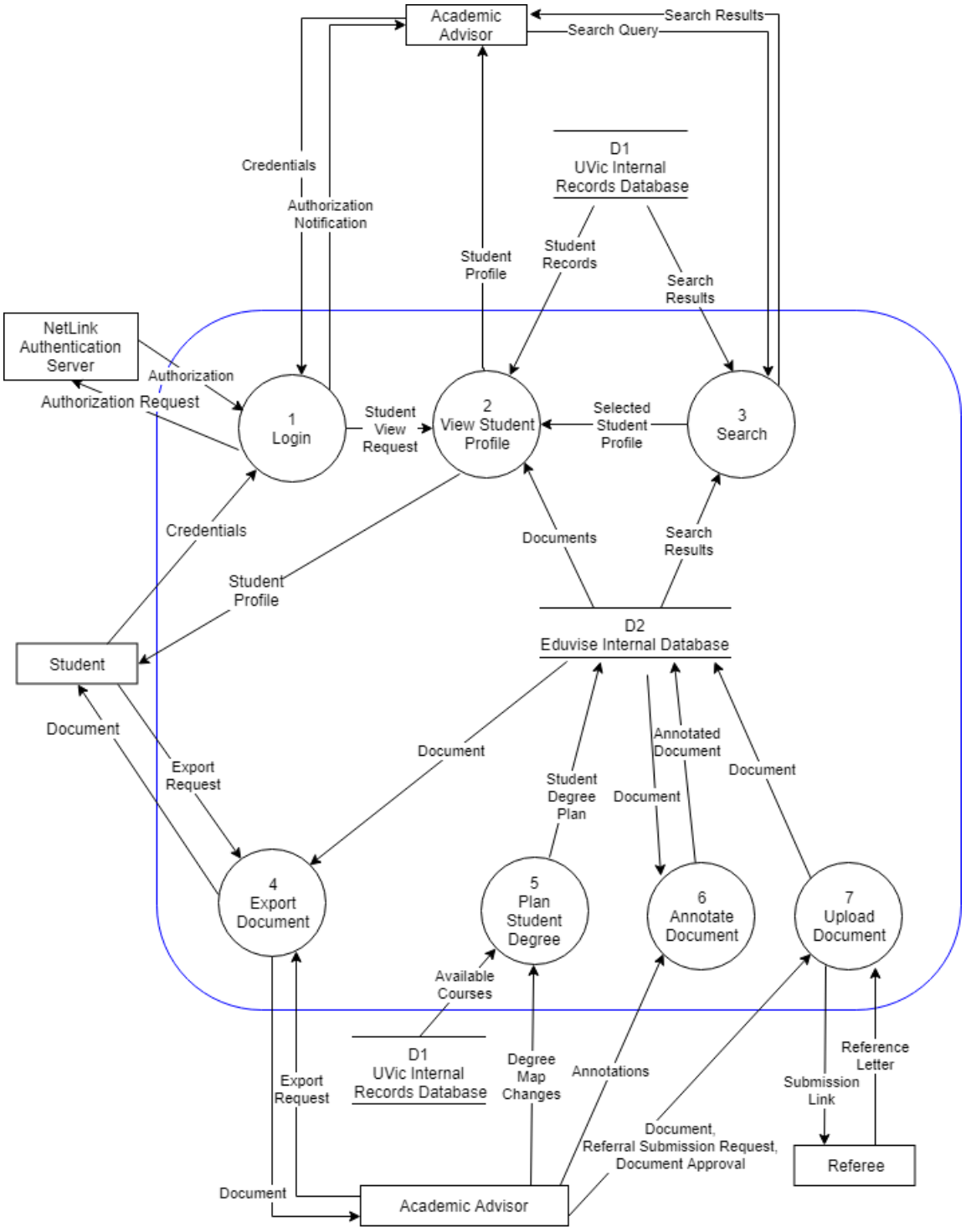


Figure 3: Data Flow Diagram Level 1

7.2.3 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.

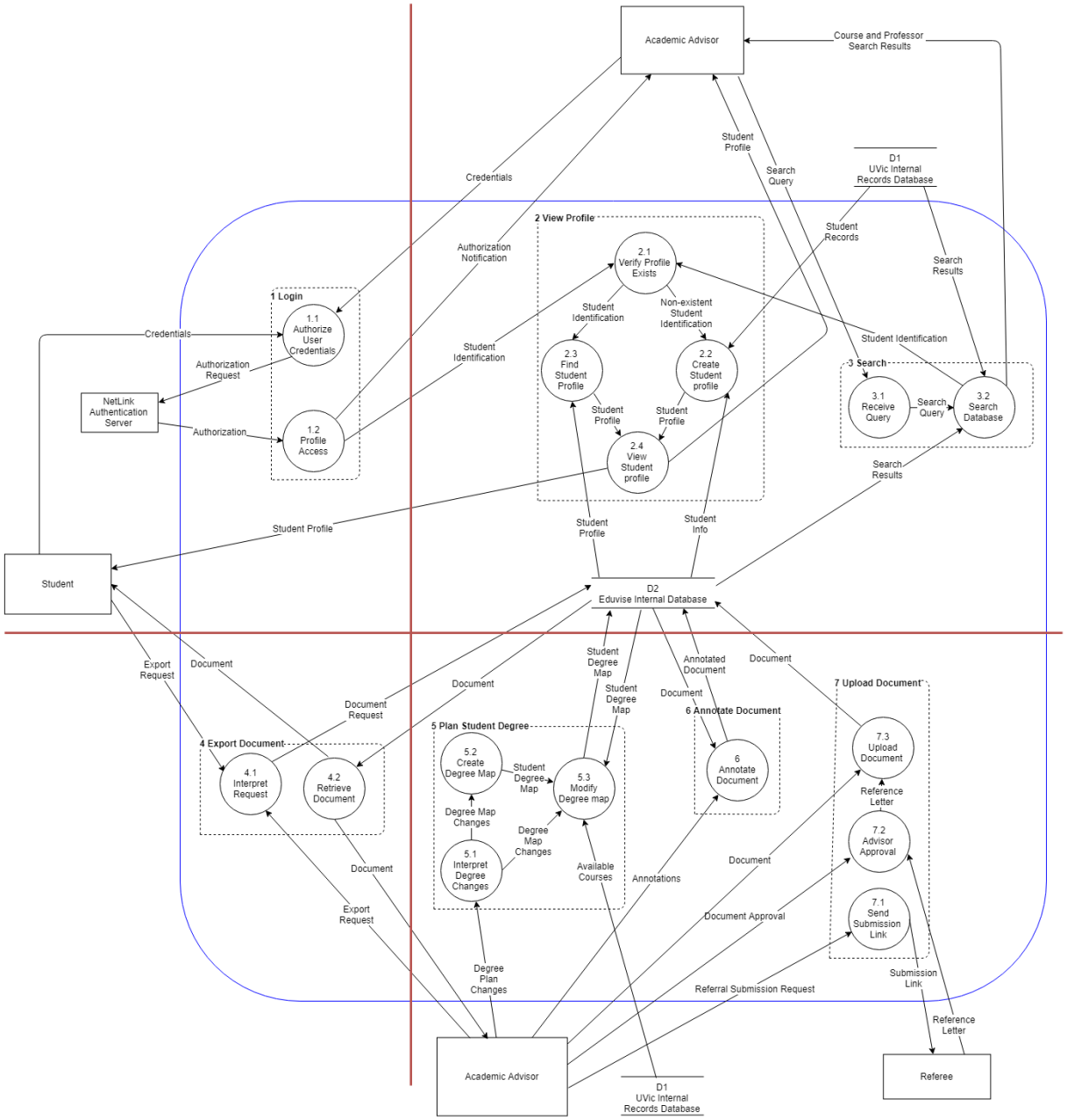


Figure 4: Data Flow Diagram Level 2

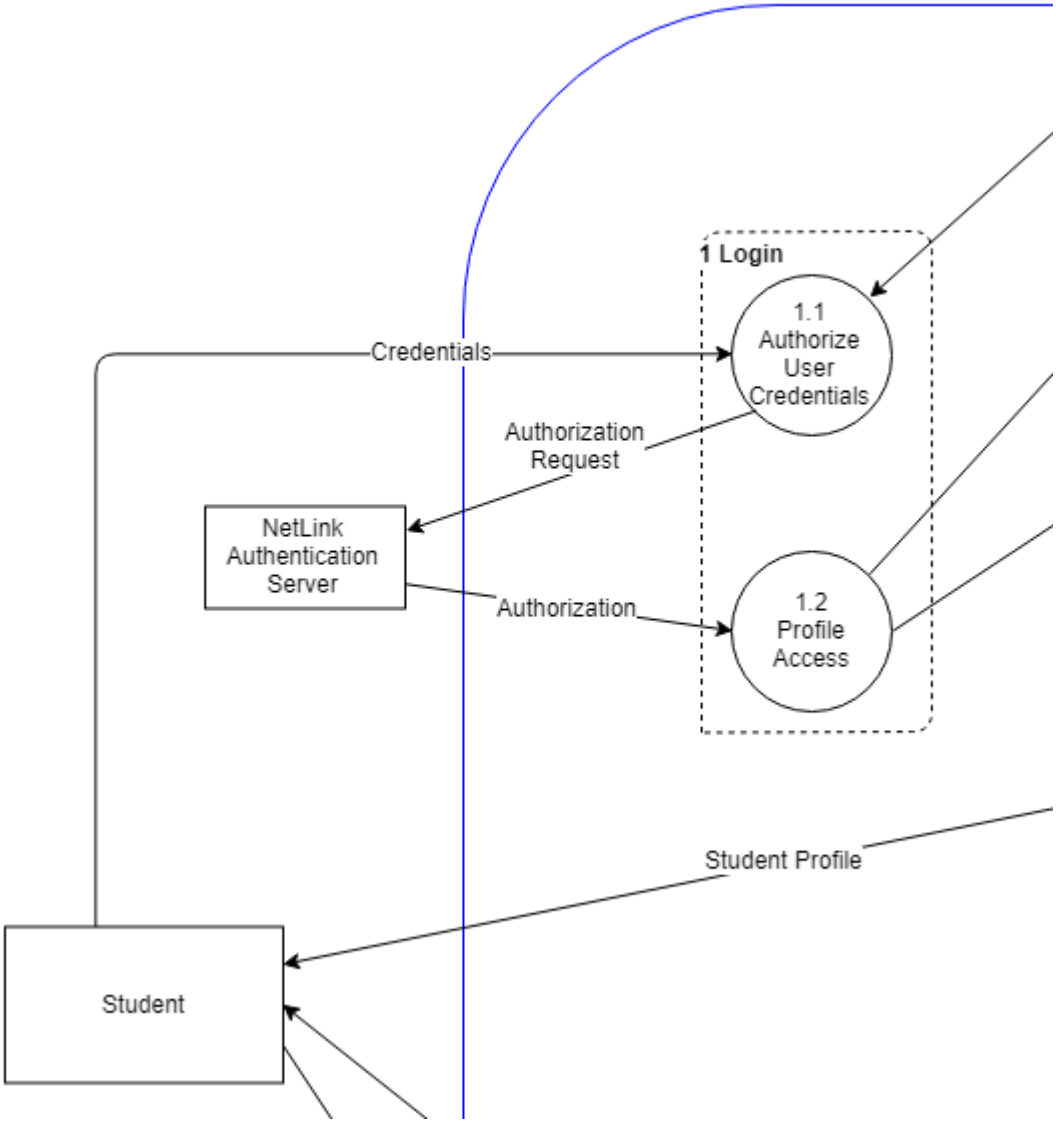


Figure 5: DFD Level 2: Top Left

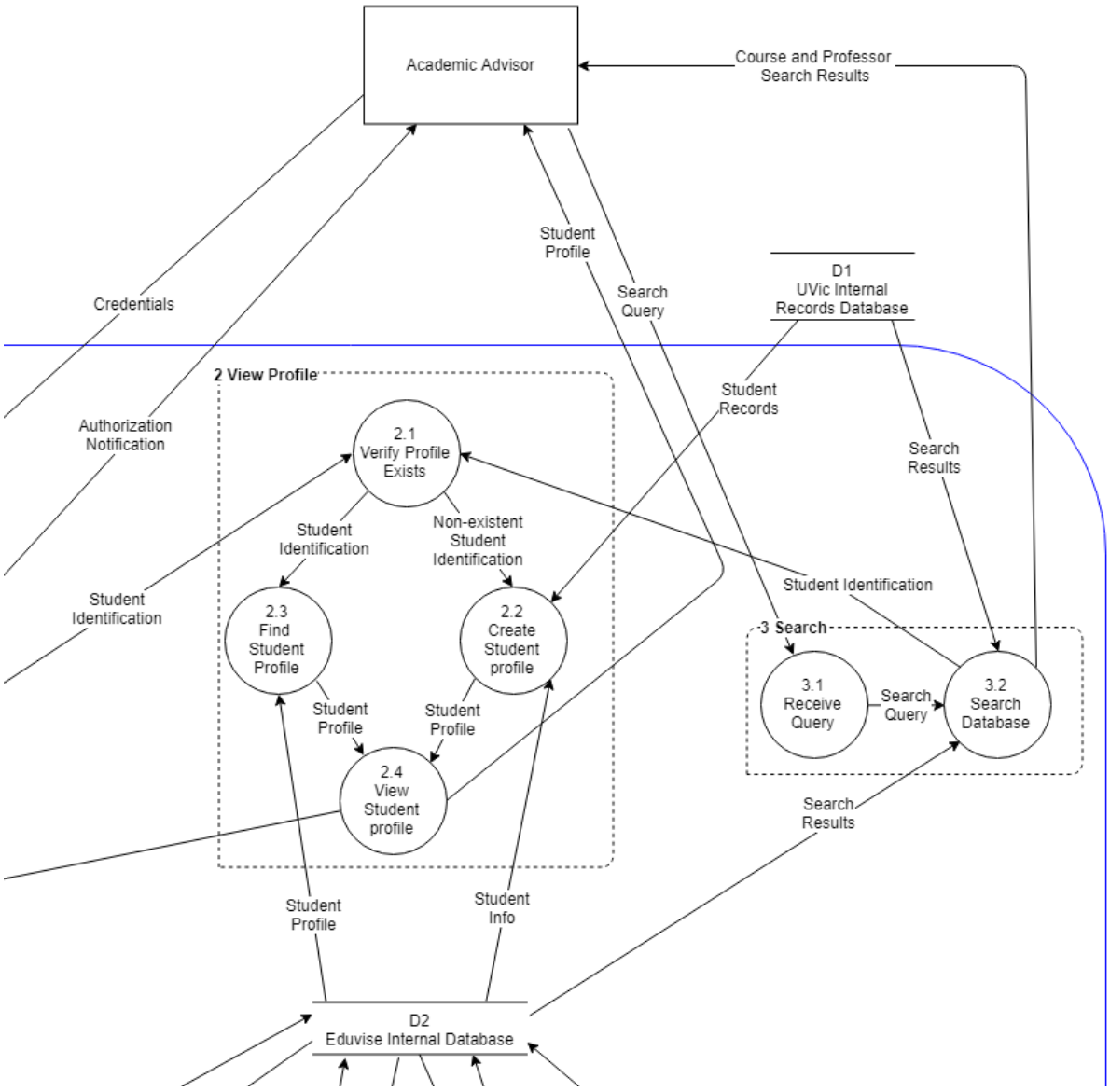


Figure 6: DFD Level 2: Top Right

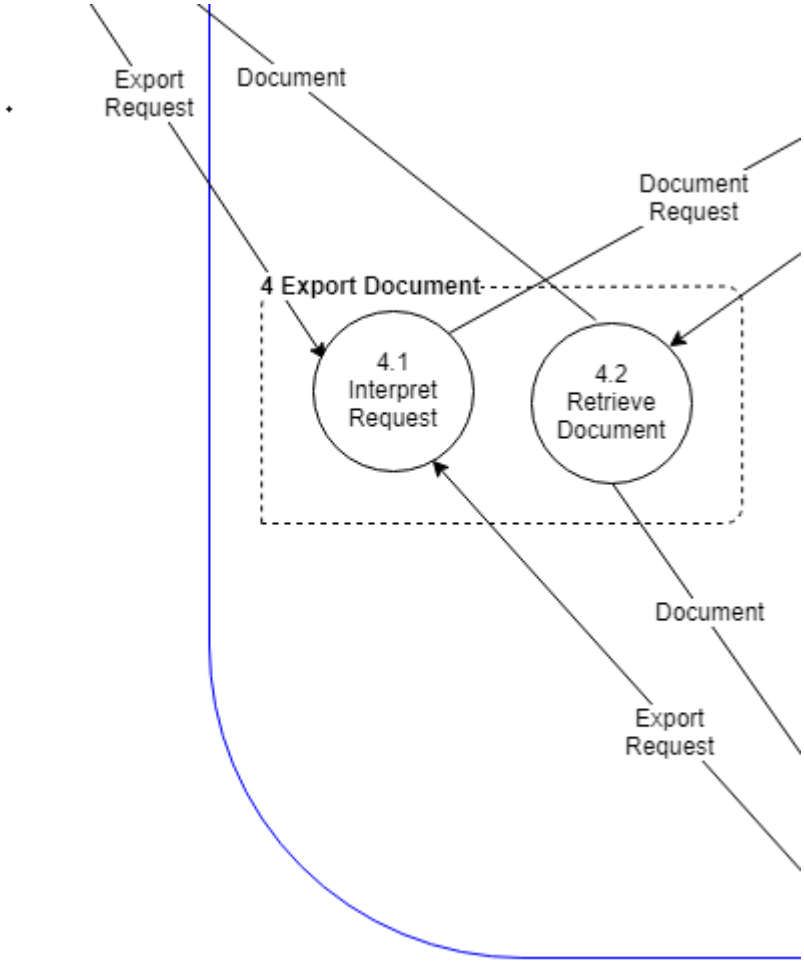


Figure 7: DFD Level 2: Bottom Left

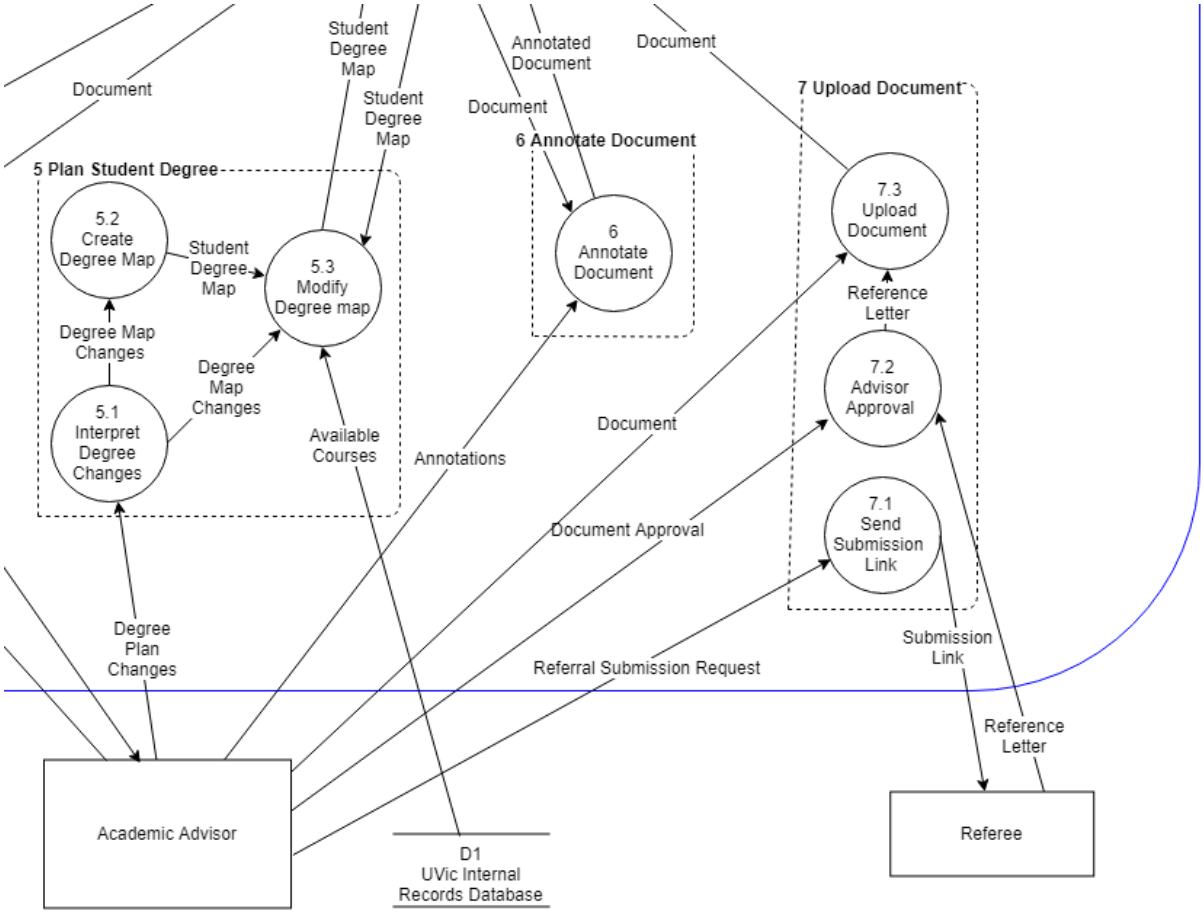


Figure 8: DFD Level 2: Bottom Right

7.3 Entity Relationship Diagram

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

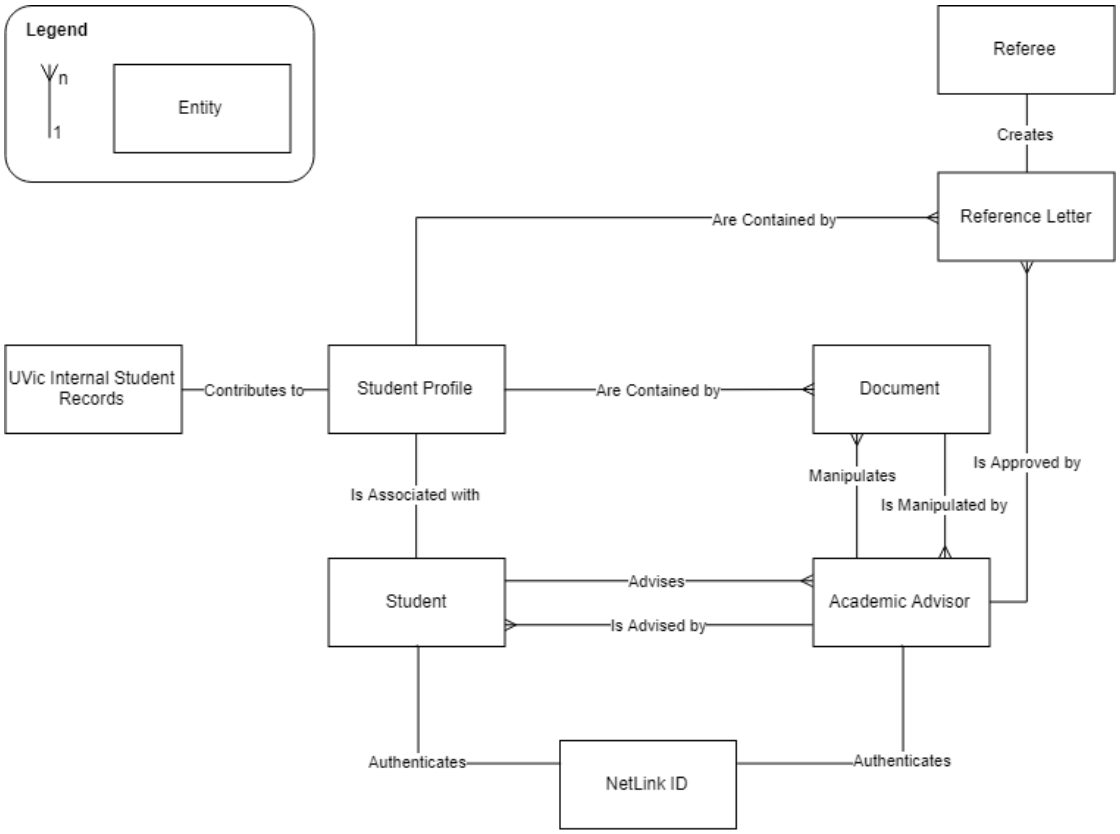


Figure 9: Entity Relationship Diagram

8 Use Cases

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.

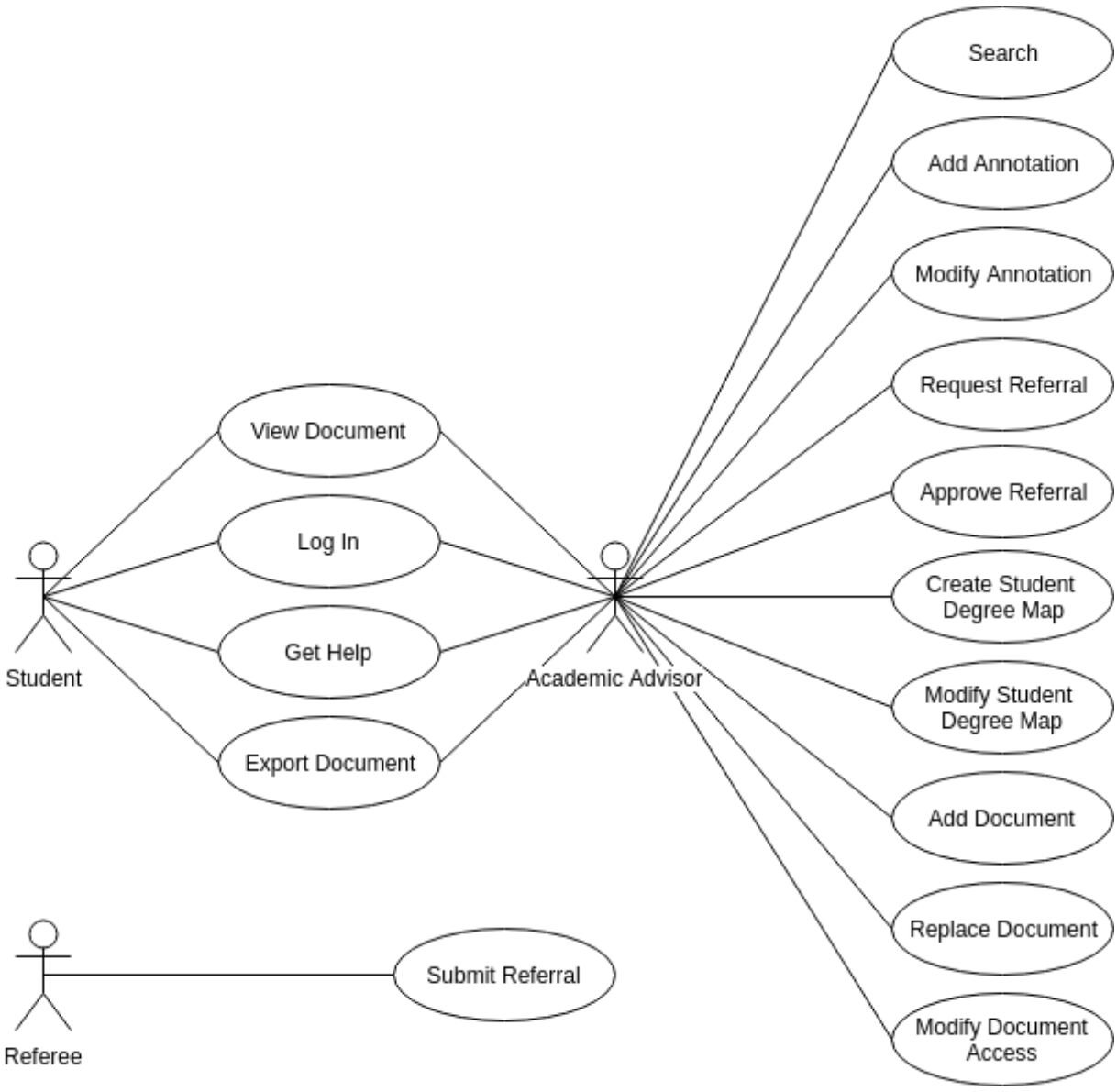


Figure 10: Use Case Model

8.2 UC-1.1: Searching

Description This use case outlines steps taken by an academic advisor to search for a document located in the Eduvise database or a student record located in the UVic internal records database.

Actors	User (academic advisor)
Preconditions	• User is logged in to Eduvise • User is on the home page
Steps	1 User inputs search query. 2 User may optionally select one or more filters to limit the search to some combination of student profiles, professor profiles, and course overviews. 3 User selects submit query option. 4 Query results are listed on the page. 5 User selects desired query result from the list. 6 The selected item is expanded and displayed on the page.
Success conditions	• Selected item is presented to the user
Alternate path	4.1 User does not find desired search results 4.2 Use case starts from step 1 with the desired record type and a different search query.
Requirements	REQ-1.1-1.3,6.2

Table 3: Use Case 1.1 - Searching

Sequence Diagram

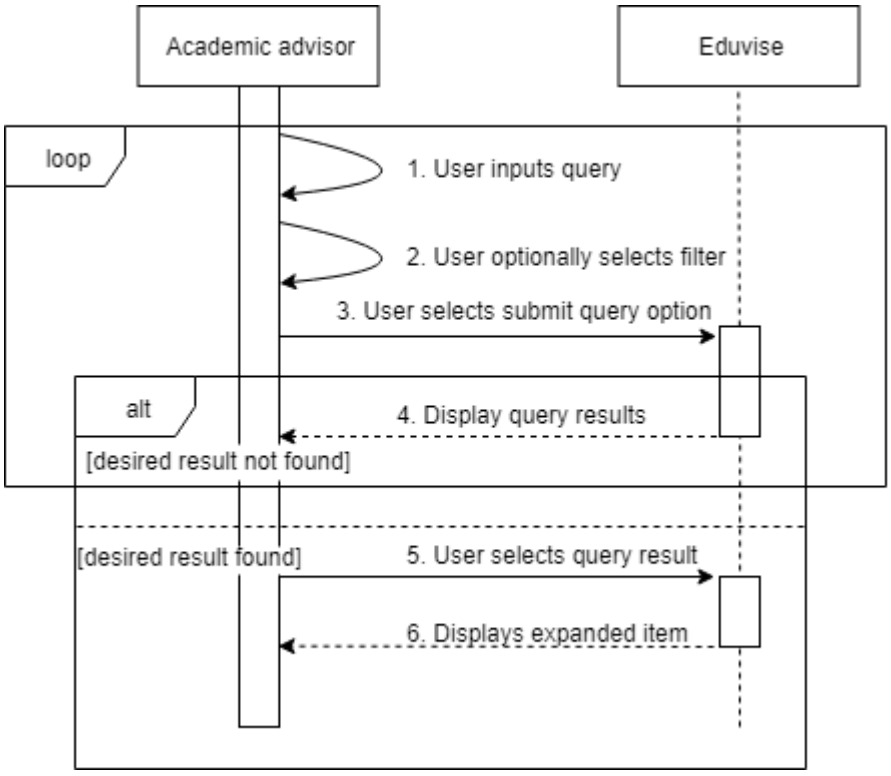
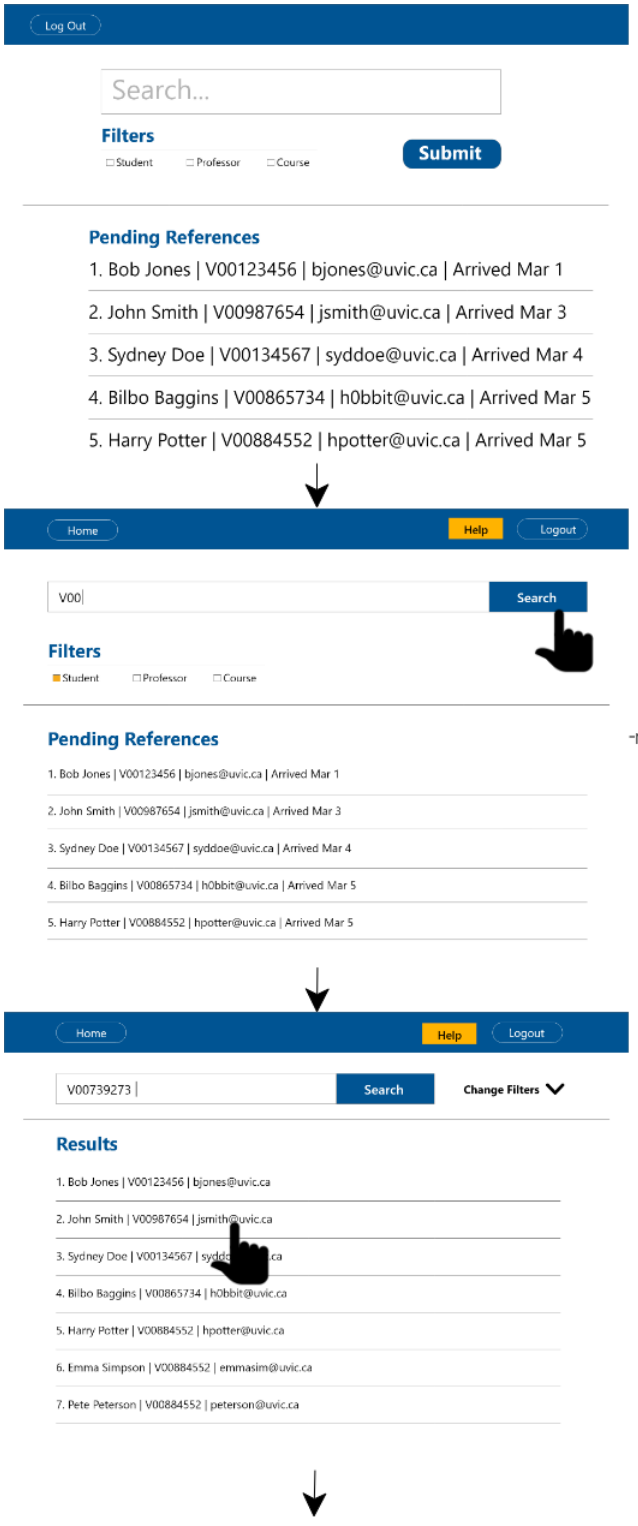


Figure 11: Sequence Diagram 1.1 - Searching

User Interface Model

This scenario 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 the 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. As shown, there are 3 final possible screens, depending on if the advisor searches for a student (as done in the remainder of the model screens), a professor, or a course.


Log Out
Search...
Filters
Student
Professor
Course
Submit
Pending References
1. Bob Jones | V00123456 | bjones@uvic.ca | Arrived Mar 1
2. John Smith | V00987654 | jsmith@uvic.ca | Arrived Mar 3
3. Sydney Doe | V00134567 | syddoe@uvic.ca | Arrived Mar 4
4. Bilbo Baggins | V00865734 | h0bbit@uvic.ca | Arrived Mar 5
5. Harry Potter | V00884552 | hpotter@uvic.ca | Arrived Mar 5
Home
Help
Logout
V00
Search
Filters
Student
Professor
Course
Pending References
1. Bob Jones | V00123456 | bjones@uvic.ca | Arrived Mar 1
2. John Smith | V00987654 | jsmith@uvic.ca | Arrived Mar 3
3. Sydney Doe | V00134567 | syddoe@uvic.ca | Arrived Mar 4
4. Bilbo Baggins | V00865734 | h0bbit@uvic.ca | Arrived Mar 5
5. Harry Potter | V00884552 | hpotter@uvic.ca | Arrived Mar 5
Home
Help
Logout
V00739273
Search
Change Filters
Results
1. Bob Jones | V00123456 | bjones@uvic.ca
2. John Smith | V00987654 | jsmith@uvic.ca
3. Sydney Doe | V00134567 | syd
4. Bilbo Baggins | V00865734 | h0bbit@uvic.ca
5. Harry Potter | V00884552 | hpotter@uvic.ca
6. Emma Simpson | V00884552 | emmasim@uvic.ca
7. Pete Peterson | V00884552 | peterson@uvic.ca
No results found
V44556677
Filters
Search Results
No search results found.
Please try another query or change selected filters.
Continue
1. Bob
2. John
3. Sydney
4. Bilbo Baggins | V00865734 | h0bbit@uvic.ca | Arrived Mar 5
5. Harry Potter | V00884552 | hpotter@uvic.ca | Arrived Mar 5

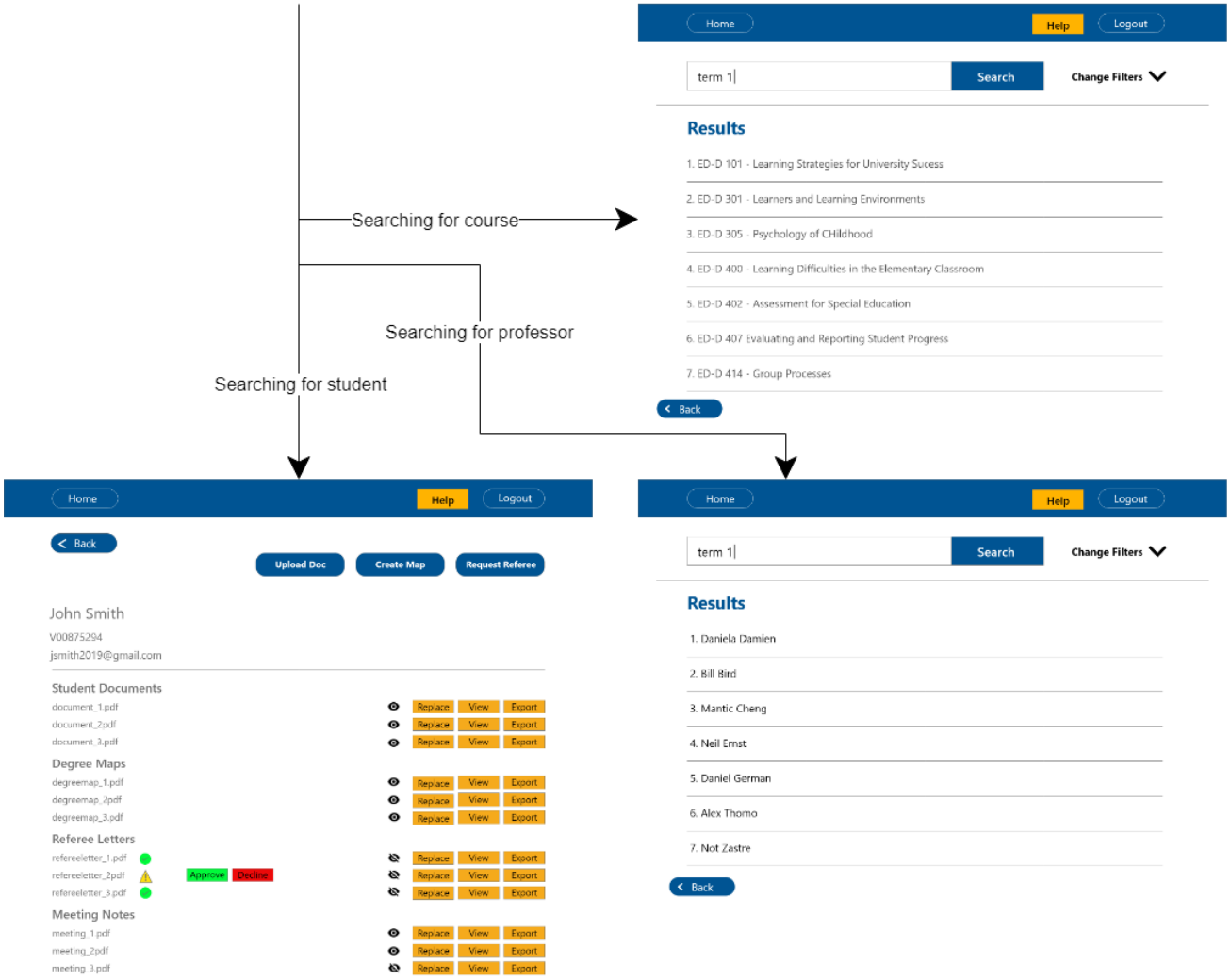


Figure 12: UI Model 1.1 - Searching

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	• User is logged in to Eduvise • User is viewing a student profile
Steps	1 User selects the option to add a new academic advising document. 2 User is prompted to select a document from the user's computer. 3 User selects document from their computer to be added to Eduvise. 4 User selects the document category and whether the student will have access to view the document. 5 User selects option to submit document to Eduvise. 6 The document is displayed in the student's profile.
Success conditions	• The correct document has been added to the student's profile.
Alternate path	5.1 User selects option to cancel adding document 5.2 Selected document is discarded and no changes are made to the student's profile.
Requirements	REQ-2.1

Table 4: Use Case 2.1 - Adding Academic Advising Document

Sequence Diagram

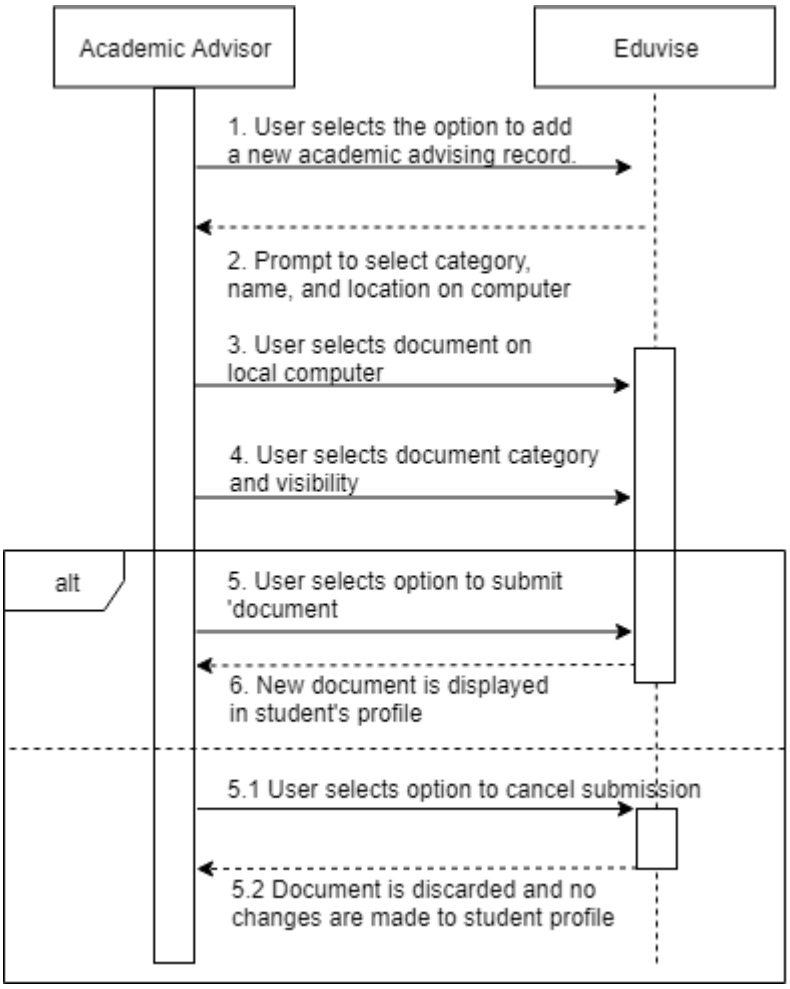
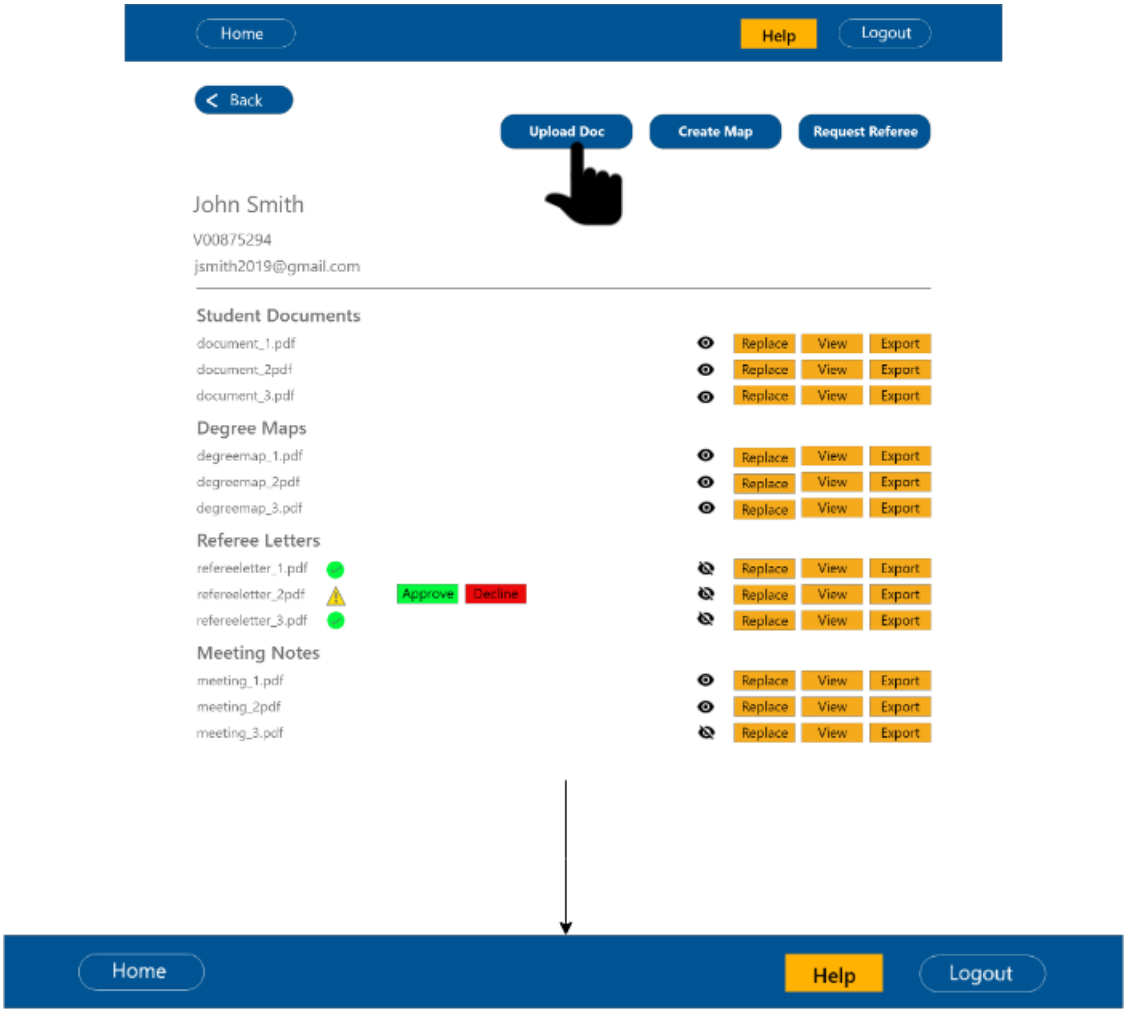
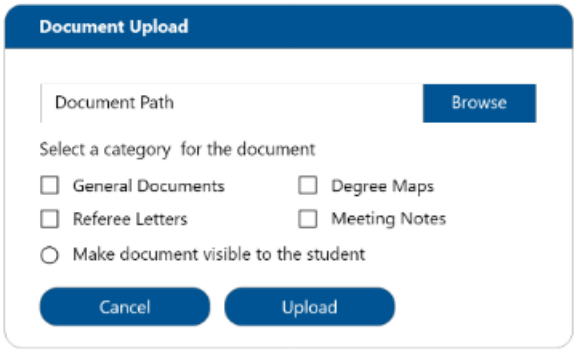


Figure 13: Sequence Diagram 2.1 - Adding Academic Advising Document

User Interface Model

This scenario 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 file 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.


Home
Help
Logout
Back
Upload Doc
Create Map
Request Referee
John Smith
V00875294
jsmith2019@gmail.com
Student Documents
document_1.pdf
document_2pdf
document_3.pdf
Degree Maps
degreemap_1.pdf
degreemap_2pdf
degreemap_3.pdf
Referee Letters
refereeletter_1.pdf
refereeletter_2pdf
refereeletter_3.pdf
Approve
Decline
Meeting Notes
meeting_1.pdf
meeting_2pdf
meeting_3.pdf
Replace
View
Export
Home
Help
Logout


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

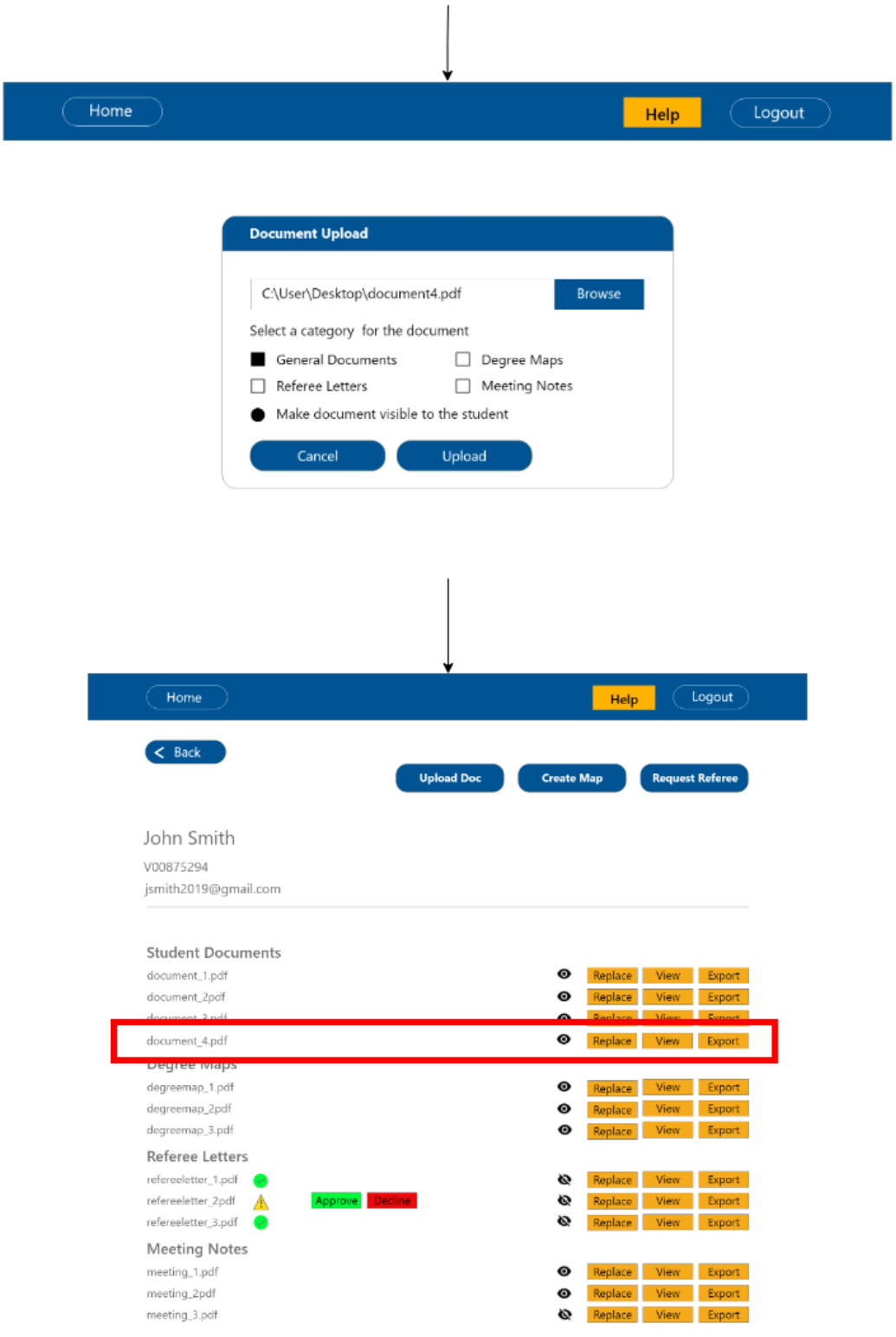


Figure 14: UI Model 2.1 - Adding Academic Advising Document

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	• User is logged in to Eduvise • User is viewing a student profile
Steps	1 User selects option to create a referral request. 2 User is prompted for email address of the referee. 3 User inputs the required referee email address. 4 User selects option to submit the referral request to the email address specified. 5 Eduvise sends a referral request to the referee by email.
Success conditions	• The referral request has been sent to the referee.
Alternate path	3.1 User selects option to cancel the referral request. 3.2 User is returned to the student profile.
Requirements	EIR-4.1

Table 5: Use Case 2.2 - Requesting Referral

Sequence Diagram

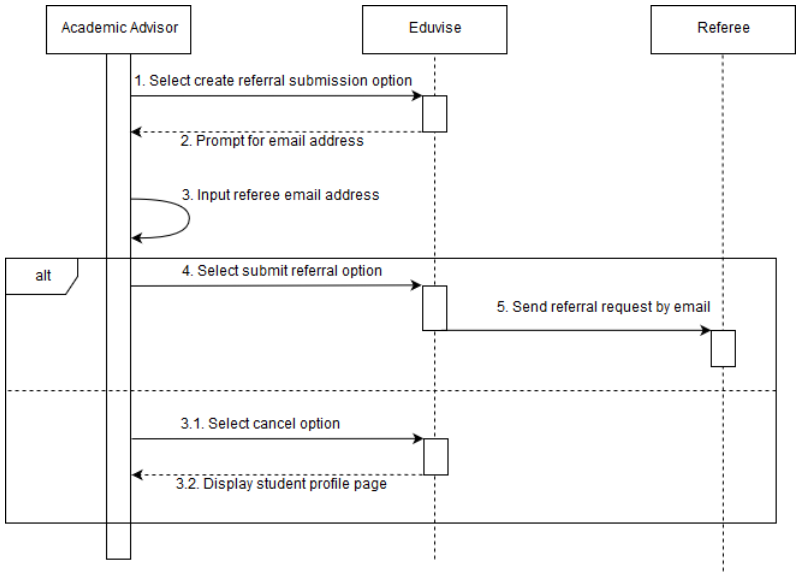


Figure 15: Sequence Diagram 2.2 - Requesting Referral

User Interface Model

This scenario outlines the steps taken when an academic advisor requests a reference letter from 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.

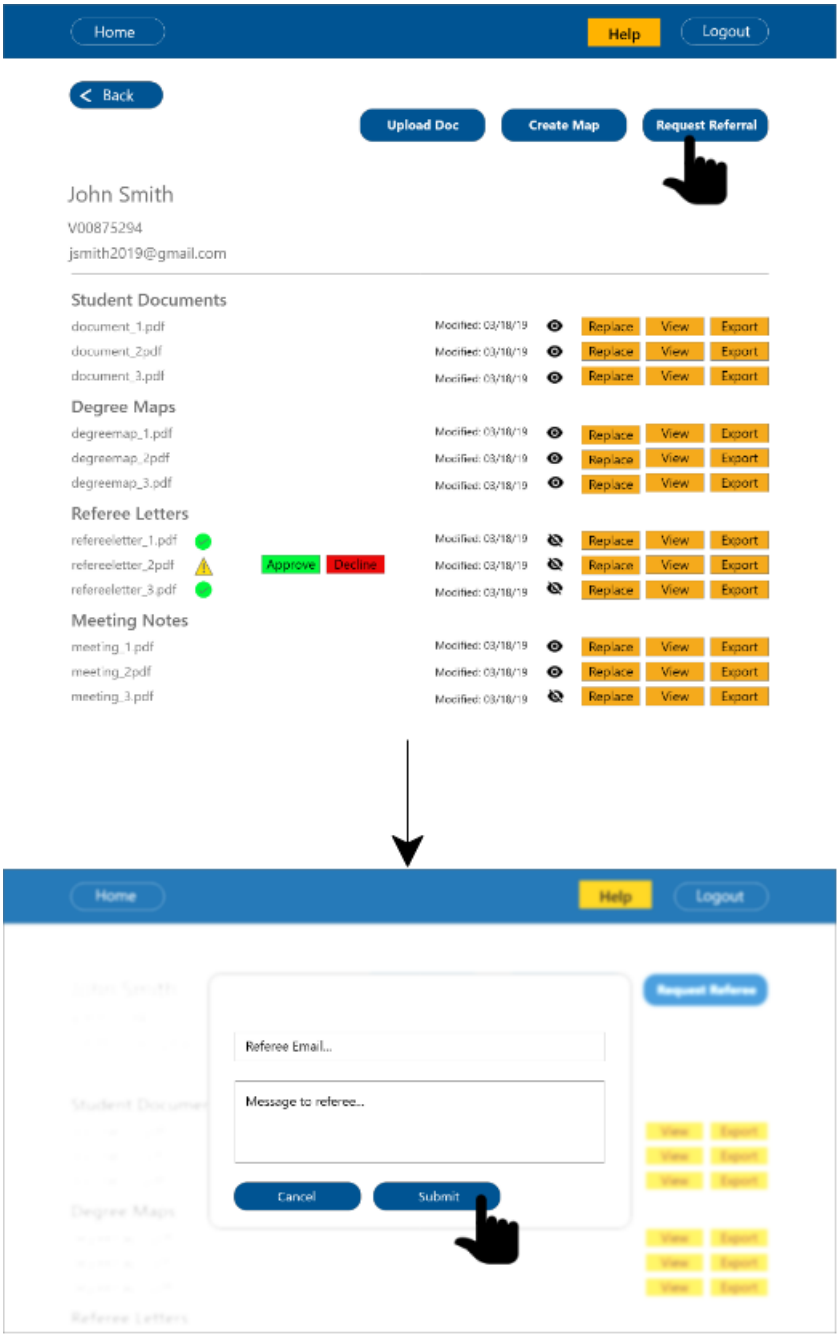


Figure 16: UI Model 2.2 - Requesting Referral

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	• User has received a referral request email • User has a referral document on their local computer
Steps	1 User follows the link in the referral request to the referral submission page. 2 User is shown the referral submission page. 3 User selects the option to submit a referral document. 4 User is prompted to select a document. 5 User selects the document to submit as a referral. 6 User is prompted to confirm their document submission. 7 User selects the confirm submission option to send the referral document to Eduvise. 8 User receives a referral submission confirmation email.
Success conditions	• The referral document has been submitted to Eduvise for approval and the designated academic advisor has been notified via email.
Alternate path	7.1 User cancels submitting the document 7.2 Use case continues at step 2.
Requirements	EIR-4.1

Table 6: Use Case 2.3 - Submitting Referral

Sequence Diagram

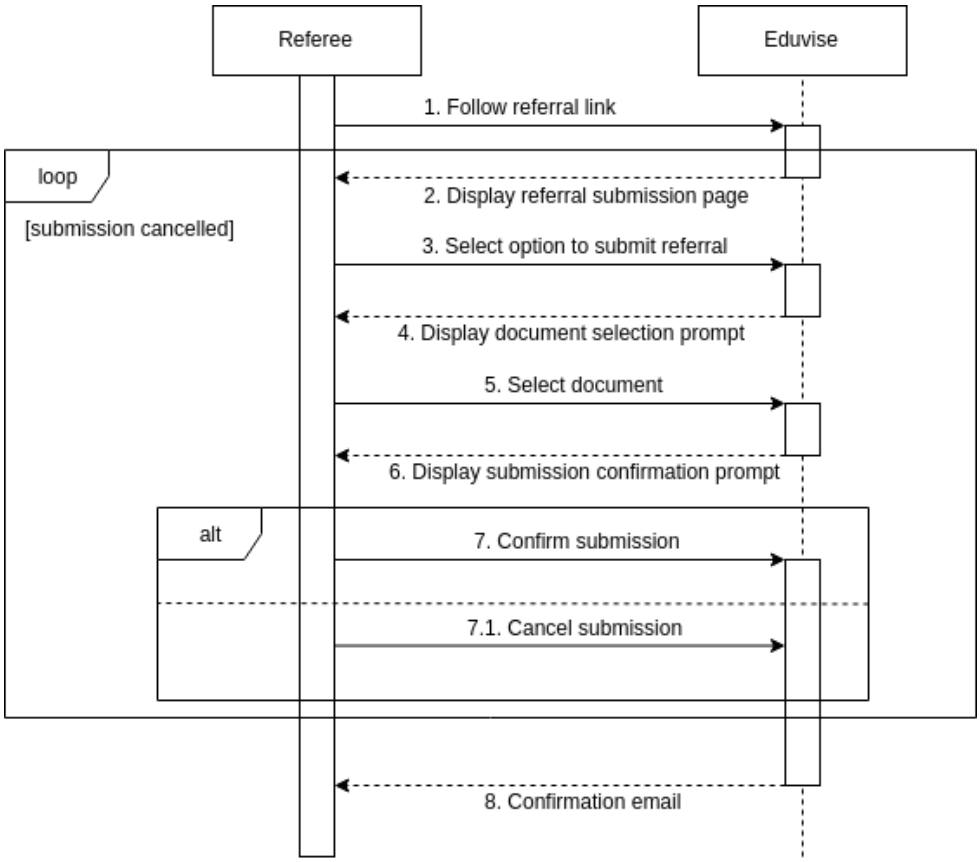


Figure 17: Sequence Diagram 2.3 - Submitting Referral

User Interface Model

This scenario 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.

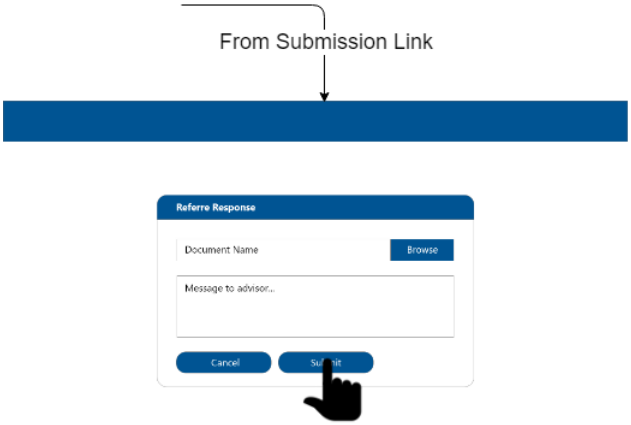


Figure 18: UI Model 2.3 - Submitting Referral

8.6 UC-2.4: Approving Referral

Description This use case outlines the steps taken by an academic advisor to approve a referral. The UC-2.3 for the advisor to approve the referral.

Actors	User (academic advisor)
Preconditions	• User is logged in to Eduvise • User has requested a referral from a referee • Referee has submitted a referral document
Steps	1 User is notified via email when the response is received. 2 User selects a link to the referral document found in the email notification. 3 User is redirected to the referral document in Eduvise. 4 User examines the response from the referee and selects option to approve the referral document. 5 The referral document is added to the student's profile.
Success conditions	• The referral document has been correctly added to Eduvise.
Alternate path	3.1 User selects option to decline the referral document. 3.2 Eduvise displays prompt to decline the referral, with the option to resend the referral request. 3.3 User selects option to either resend or not resend the referral request. 3.4 User optionally inputs a response to the referee in the prompt. 3.5 User selects option to confirm the referral rejection. 3.6 Eduvise notifies the referee of the referral rejection by email.
Requirements	EIR-4.1

Table 7: Use Case 2.4 - Approving Referral

Sequence Diagram

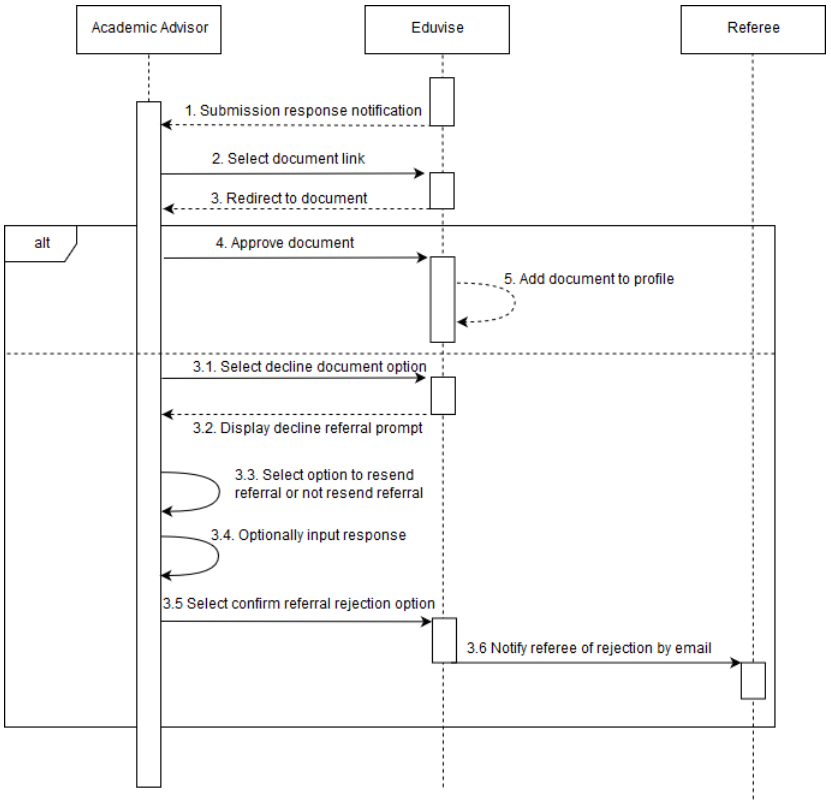


Figure 19: Sequence Diagram 2.4 - Approving Referral

User Interface Model

This scenario 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 pull 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.

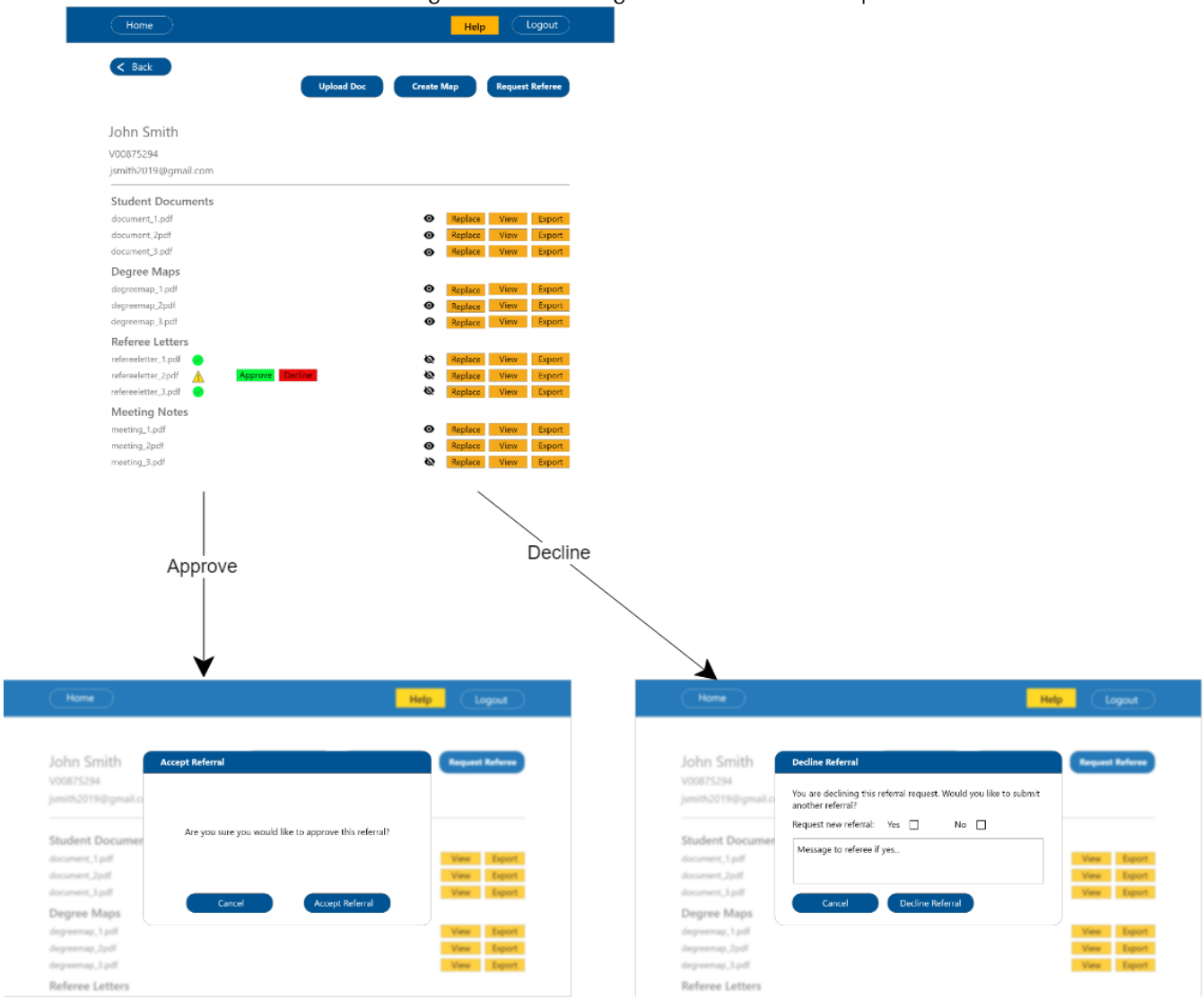


Figure 20: UI Model 2.4 - Approving Referral

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	• User is logged in to Eduvise. • User is viewing the profile of a student.
Steps	1 User selects option to replace an academic advising document. 2 User selects which academic advising document to replace. 3 File upload screen is brought up. 4 User locates and selects document on their local computer. 5 User submits document to Eduvise. 6 Eduvise prompts user if they are sure they want to replace the selected document with the uploaded one. 7 User selects option to confirm. 8 User is brought to student's profile where the old document is replaced with the newly submitted document.
Success conditions	• The old document is no longer visible and the new document is added to the student's profile.
Alternate path	6.1 User selects cancel option 6.2 Uploaded document is discarded and selected document remains.
Requirements	REQ-3.1

Table 8: Use Case 3.1 - Replacing Academic Advising Document

Sequence Diagram

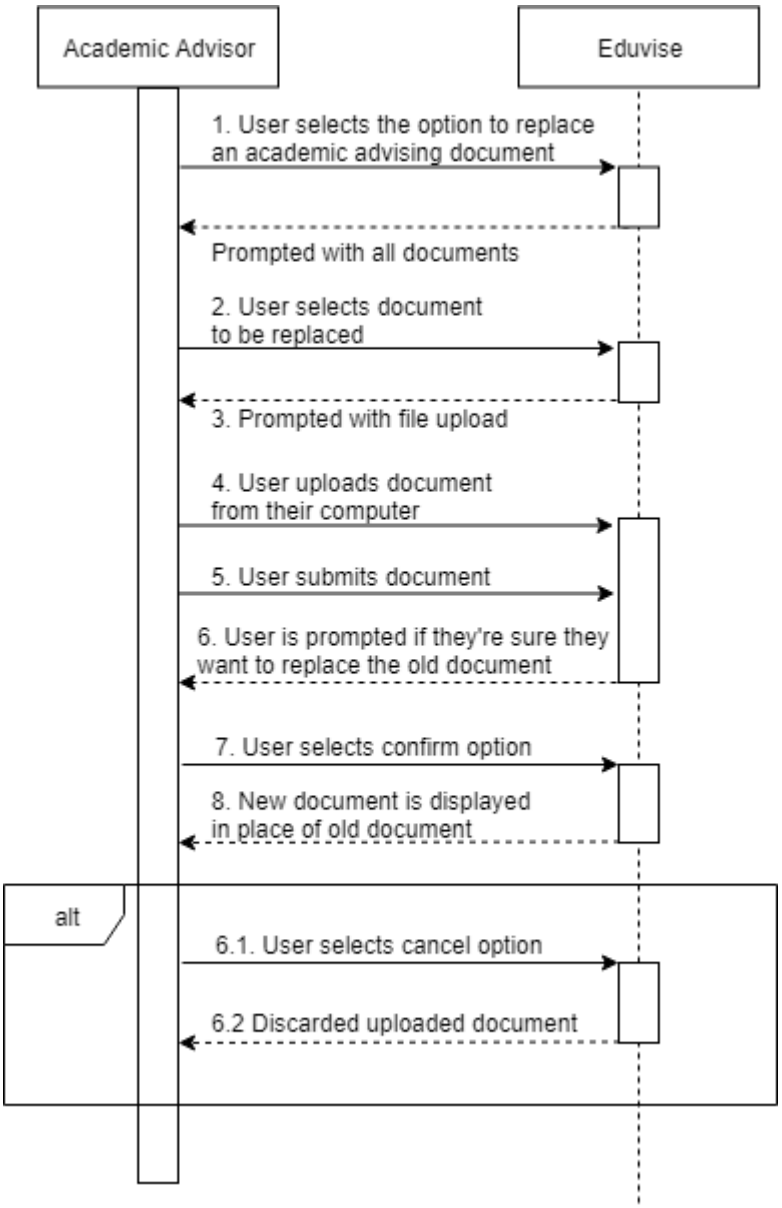


Figure 21: Sequence Diagram 3.1 - Replacing Academic Advising Document

User Interface Model

This scenario 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.

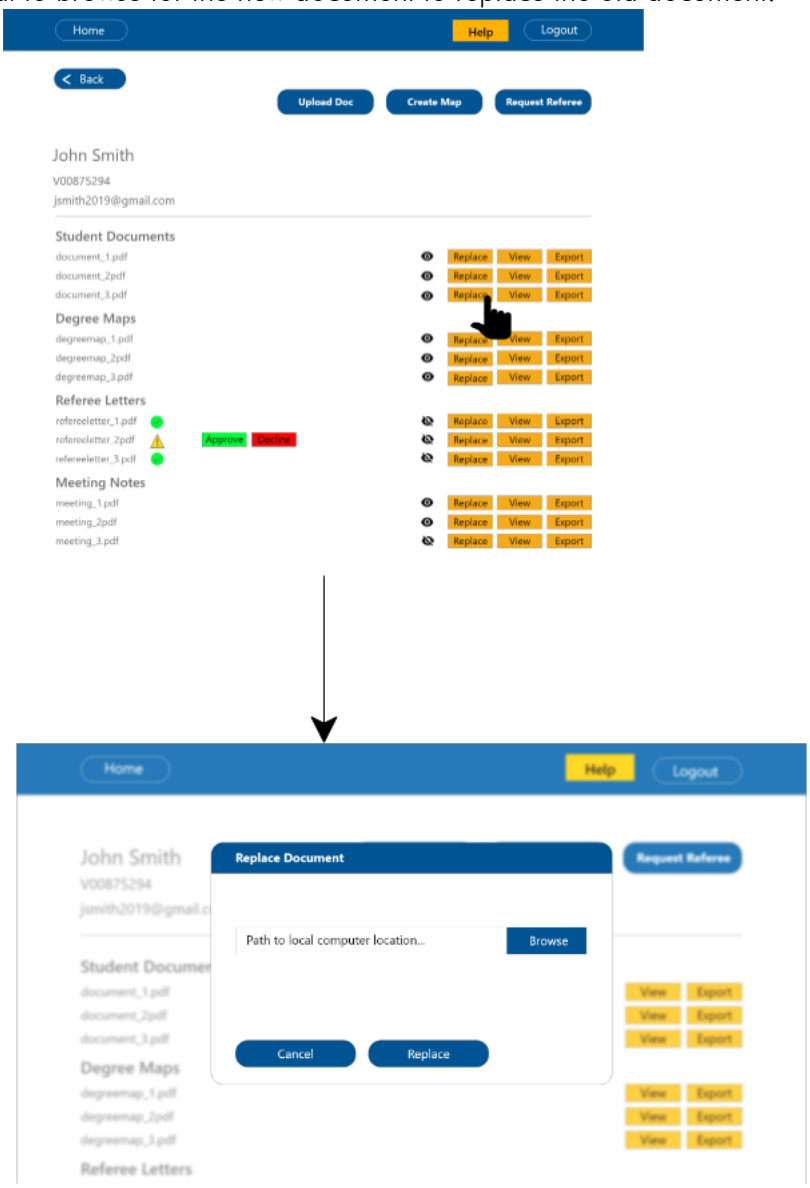


Figure 22: UI Model 3.1 - Replace Academic Advising Document

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	• User is logged in to Eduvise. • User is viewing an academic advising document on a student's profile.
Steps	1 User selects the "add annotation" option. 2 Eduvise displays an empty text box. 3 User enters the annotation into the text box. 4 User clicks submit and the annotation is added. 5 Document is reloaded and displayed.
Success conditions	• The annotation has been correctly added to the document.
Alternate path	None
Requirements	REQ-3.2

Table 9: Use Case 3.2 - Adding Annotation

Sequence Diagram

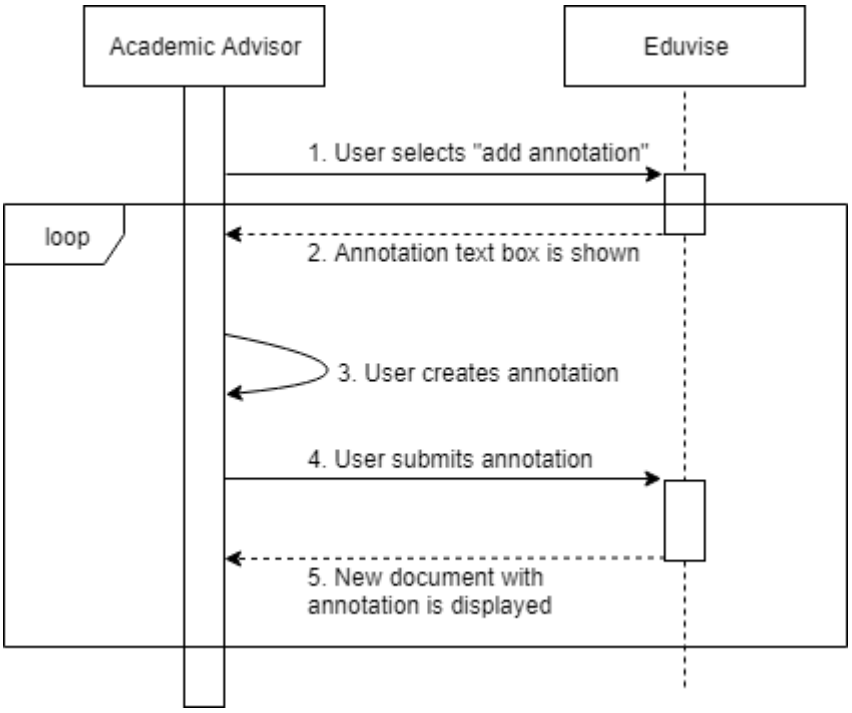
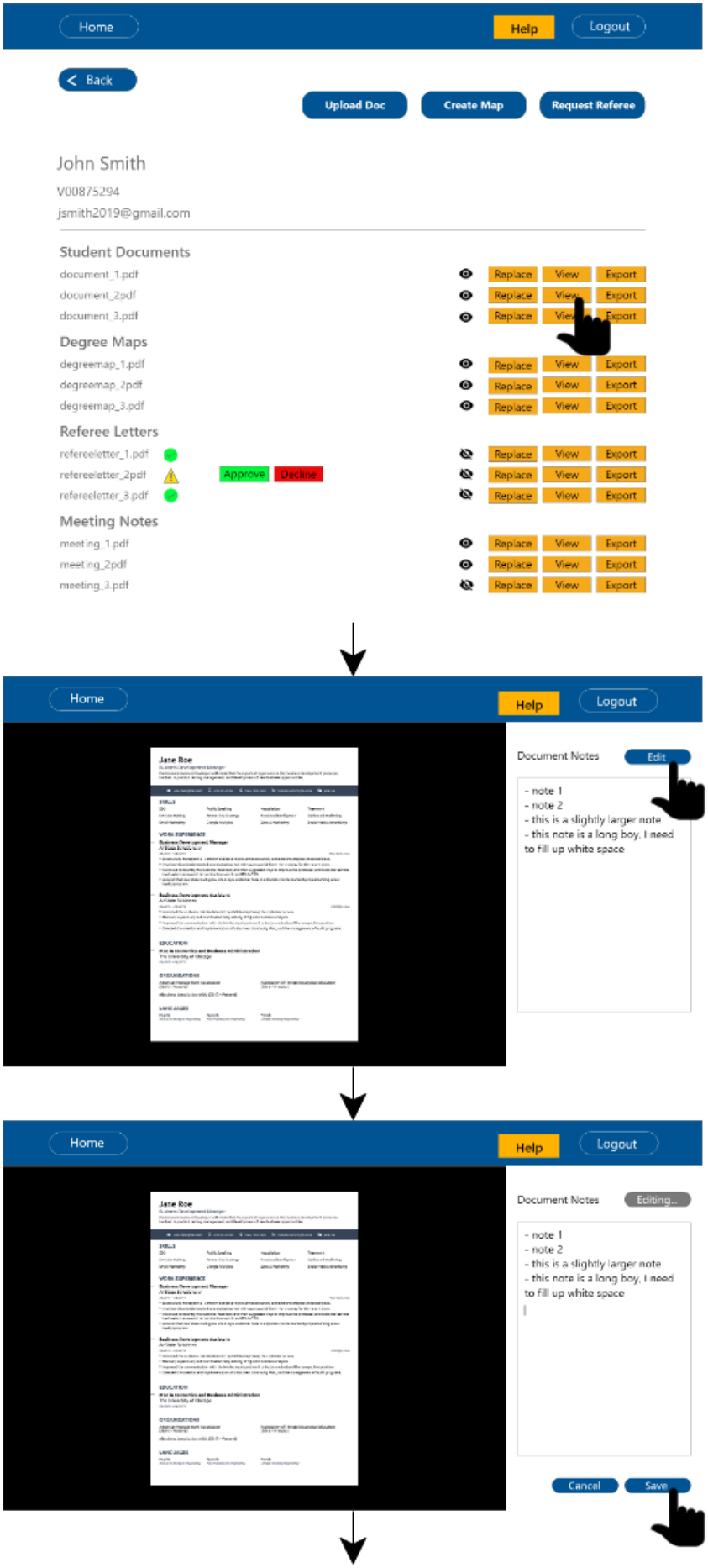


Figure 23: Sequence Diagram 3.2 - Adding Annotation

User Interface Model

This scenario 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 types in 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.


Home
Help
Logout
Back
Upload Doc
Create Map
Request Referee
John Smith
V00875294
jsmith2019@gmail.com
Student Documents
document_1.pdf
document_2.pdf
document_3.pdf
Degree Maps
degreemap_1.pdf
degreemap_2.pdf
degreemap_3.pdf
Referee Letters
refereeletter_1.pdf
refereeletter_2.pdf
refereeletter_3.pdf
Approve
Decline
Meeting Notes
meeting_1.pdf
meeting_2.pdf
meeting_3.pdf
Replace
View
Export
Home
Help
Logout
Jane Roe
Document Notes
Edit
- note 1
- note 2
- this is a slightly larger note
- this note is a long boy, I need to fill up white space
Home
Help
Logout
Jane Roe
Document Notes
Editing...
- note 1
- note 2
- this is a slightly larger note
- this note is a long boy, I need to fill up white space
Cancel
Save

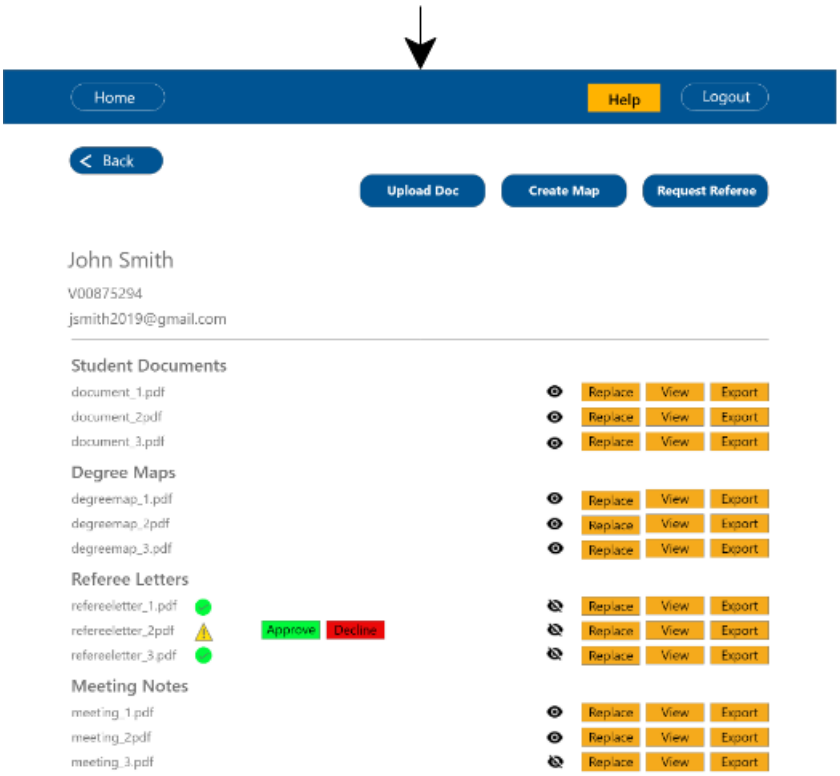


Figure 24: UI Model 3.2 - Adding Annotation

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	• User is logged in to Eduvise. • User is viewing an academic advising document on a student's profile.
Steps	1 User requests to view all annotations on the document. 2 The document is shown in annotation mode. 3 User selects the annotation to be modified. 4 The annotation turns into a text box. 5 User inputs annotation modifications. 6 User selects option to submit annotation modifications. 7 Document is reloaded with new annotation.
Success conditions	• The new annotation is correctly visible on the document.
Alternate path	5.1 User selects cancel option 5.2 Document is displayed and old annotation is unchanged.
Requirements	REQ-3.4

Table 10: Use Case 3.3 - Modifying Annotation

Sequence Diagram

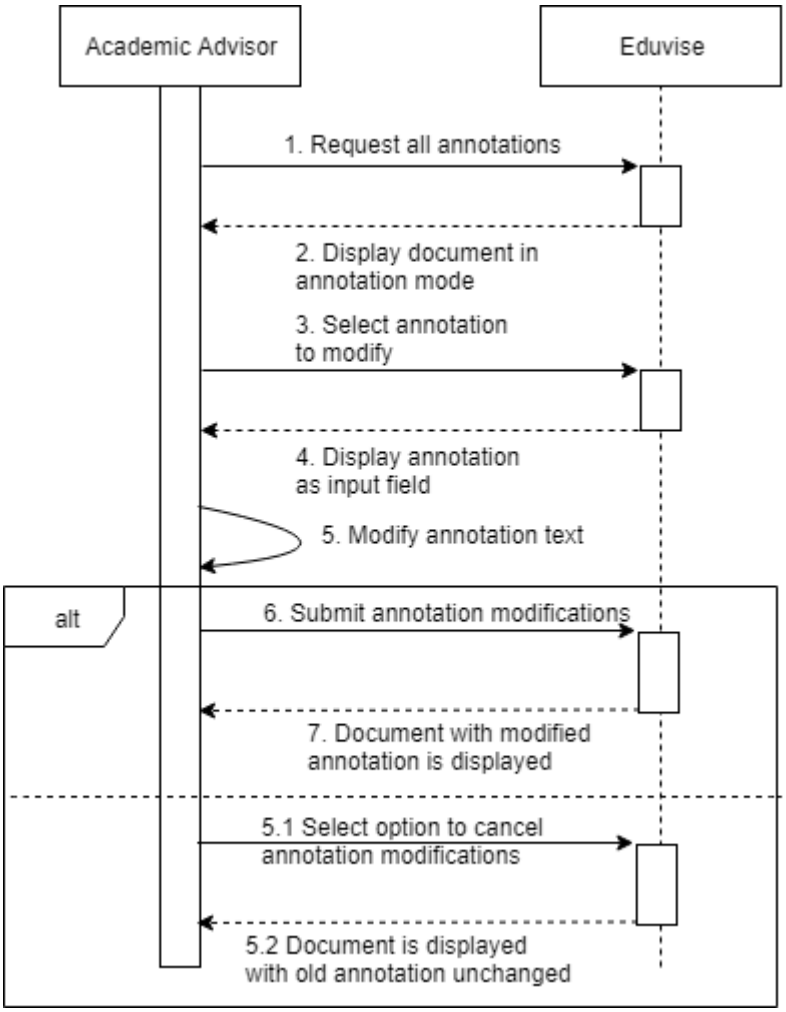


Figure 25: Sequence Diagram 3.3 - Modifying Annotation

User Interface Model

This scenario 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.

Home
Help
Logout
Back
Upload Doc
Create Map
Request Referee
John Smith
V00875294
jsmith2019@gmail.com
Student Documents
document_1.pdf
document_2pdf
document_3.pdf
Degree Maps
degreemap_1.pdf
degreemap_2pdf
degreemap_3.pdf
Referee Letters
refereeletter_1.pdf
refereeletter_2pdf
refereeletter_3.pdf
Approve
Decline
Meeting Notes
meeting_1.pdf
meeting_2pdf
meeting_3.pdf
Replace
View
Export
Jane Roe
Document Notes
Edit
- note 1
- note 2
- this is a slightly larger note
- this note is a long boy, I need to fill up white space
Editing...
Cancel
Save

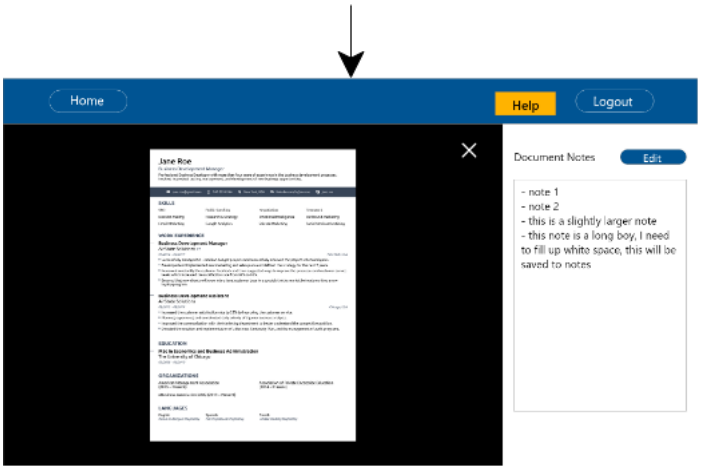


Figure 26: UI Model 3.3 - Modifying Annotation

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	• User is logged in to Eduvise. • User is viewing the profile of a student.
Steps	1 User selects the change document visibility option next to the desired document. 2 User selects option to commit change to document visibility. 3 User is taken back to student profile.
Success conditions	• The correct document has the requested visibility option.
Alternate path	2.1 User selects cancel option 2.2 No changes are made to the document visibility. 2.3 User is taken back to student profile.
Requirements	REQ-3.5

Table 11: Use Case 3.4 - Modifying Document Access

Sequence Diagram

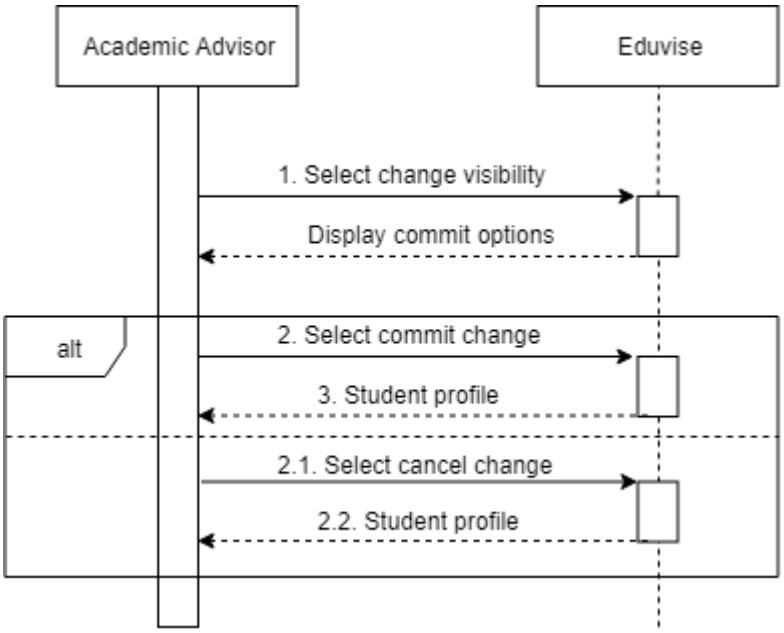


Figure 27: Sequence Diagram 3.4 - Modifying Document Access

User Interface Model

This scenario 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.

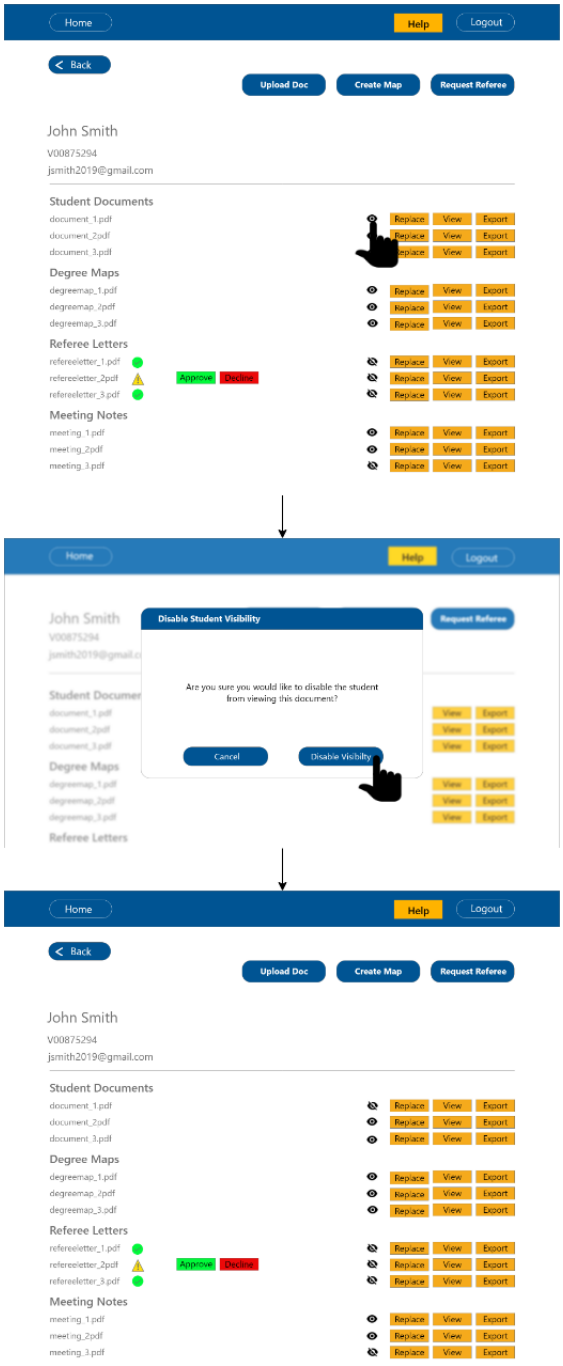


Figure 28: UI Model 3.4 - Modifying Document Access

8.11 UC-4.1: Document Exporting

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

Actors	User (academic advisor, student)
Preconditions	• User is logged in to Eduvise. • User is viewing the profile of a student.
Steps	1 User select option to export a document. 2 A window appears for the user to navigate in order to specify where the document will be saved to their local computer. 3 User selects the desired location where the document is to be exported. 4 User submits desired export location. 5 The document is exported to the chosen location.
Success conditions	• The correct document has been exported to their local computer.
Alternate path	None
Requirements	REQ-4.1,4.2

Table 12: Use Case 4.1 - Document Exporting

Sequence Diagram

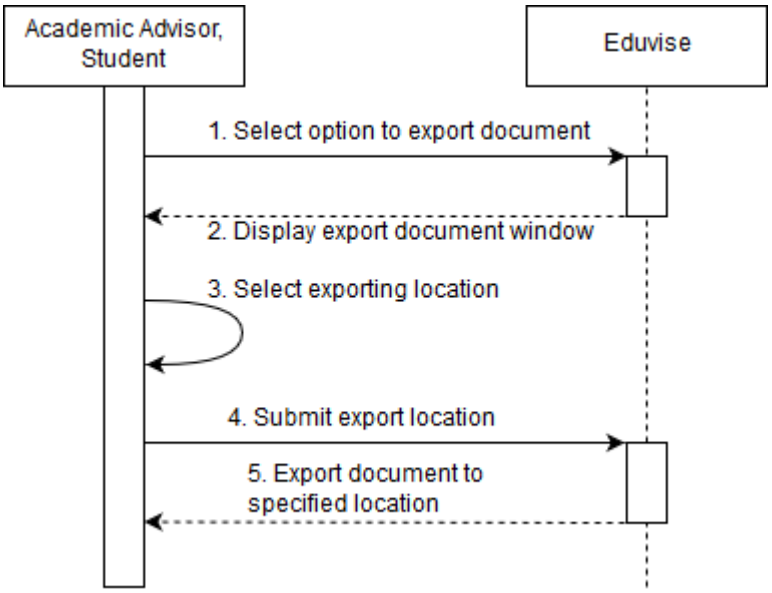


Figure 29: Sequence Diagram 4.1 - Document Exporting

User Interface Model

This scenario shows the steps needed for a user to export a document 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.

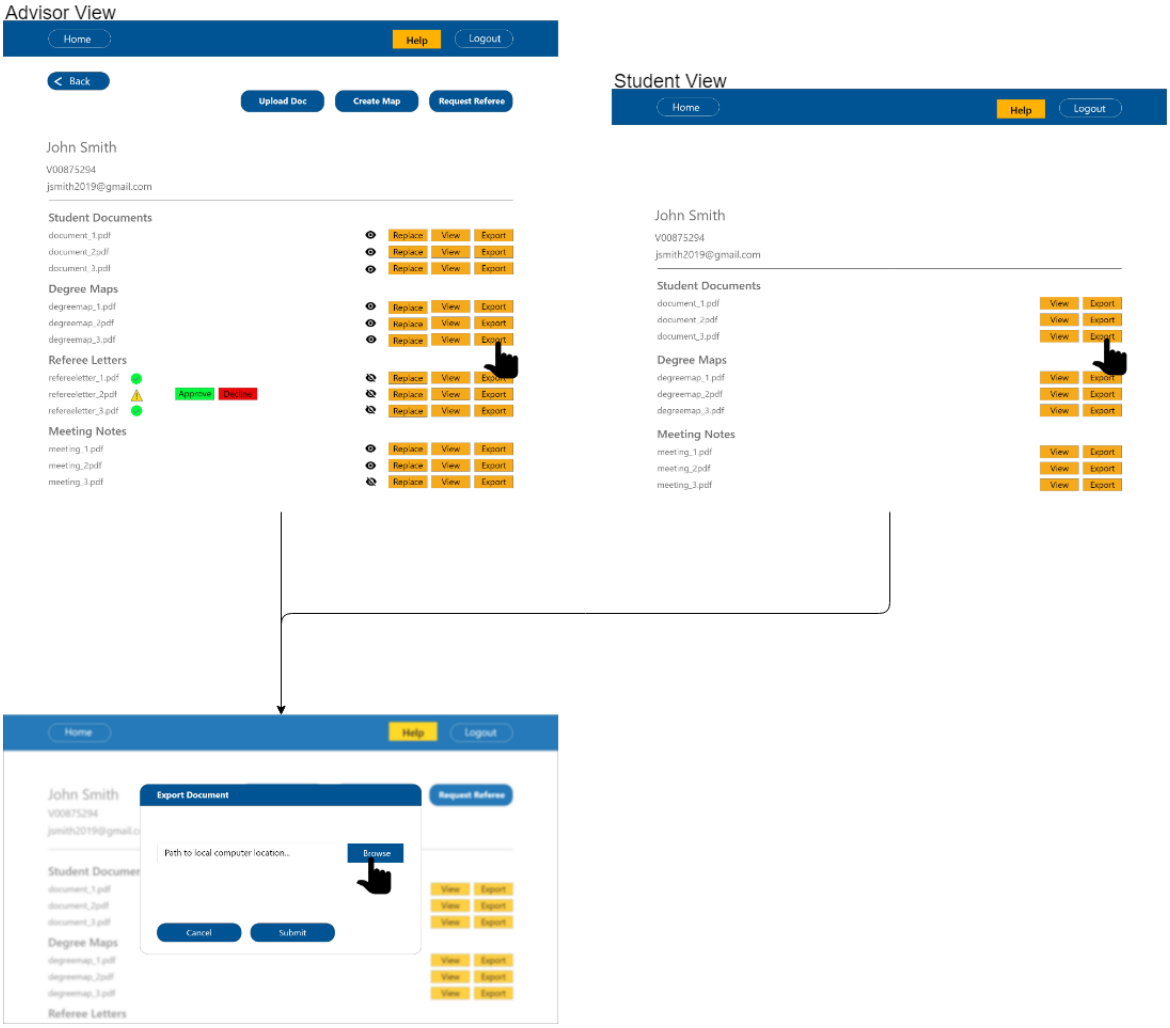


Figure 30: UI Model 4.1 - Document Exporting

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	• User is logged in to Eduvise. • User is viewing the profile of a student.
Steps	1 User selects a document from the student profile. 2 The selected document is displayed in the browser.
Success conditions	• The user has viewed the correct document.
Alternate path	None
Requirements	REQ-5.2,6.3

Table 13: Use Case 5.1 - Document Viewing

Sequence Diagram

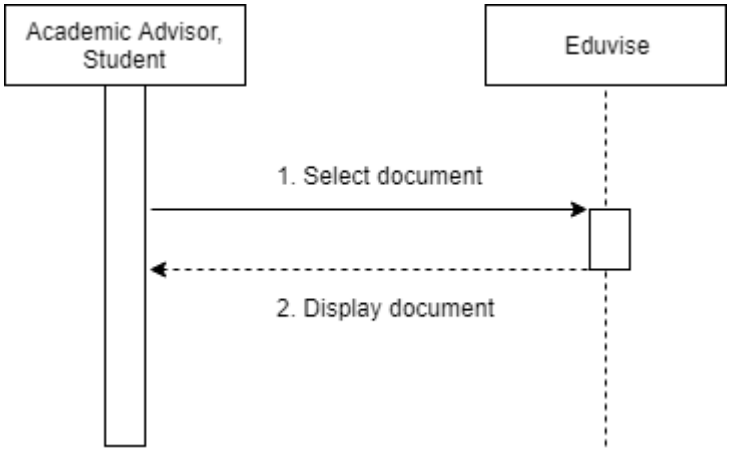


Figure 31: Sequence Diagram 5.1 - Document Viewing

User Interface Model

This scenario 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.

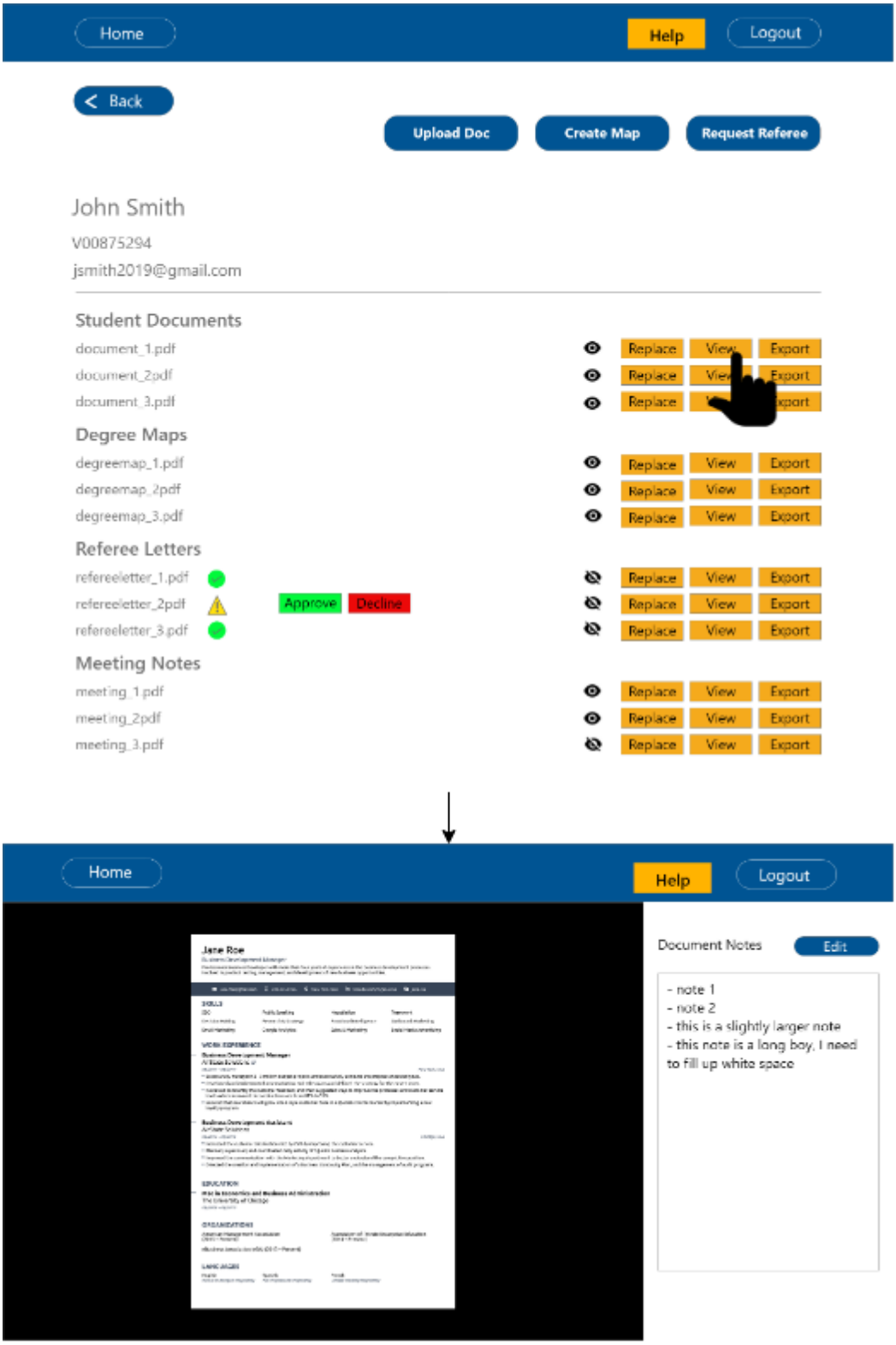


Figure 32: UI Model 5.1 - Document Viewing

8.13 UC-5.2: 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	• User is logged in to Eduvise.
Steps	1 User selects the help button. 2 Help window is displayed on the user's page. 3 User navigates help window to locate section related to the user's current query. 4 User selects the desired section from the help window. 5 The detailed info of the selected section is displayed.
Success conditions	• The user receives correct information about their current query.
Requirements	EIR-1.1, NFR-4.2

Table 14: Use Case 5.2 - Getting Help

Sequence Diagram

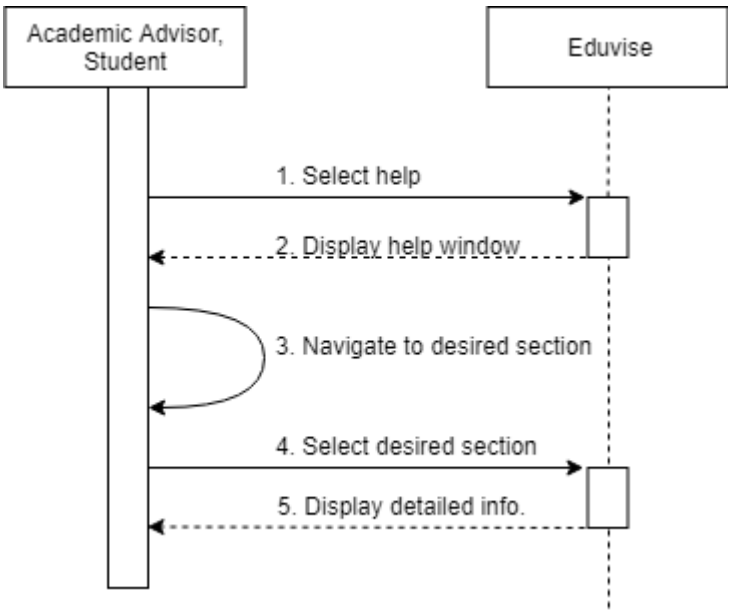


Figure 33: Sequence Diagram 5.2 - Getting Help

User Interface Model

This scenario 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/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.

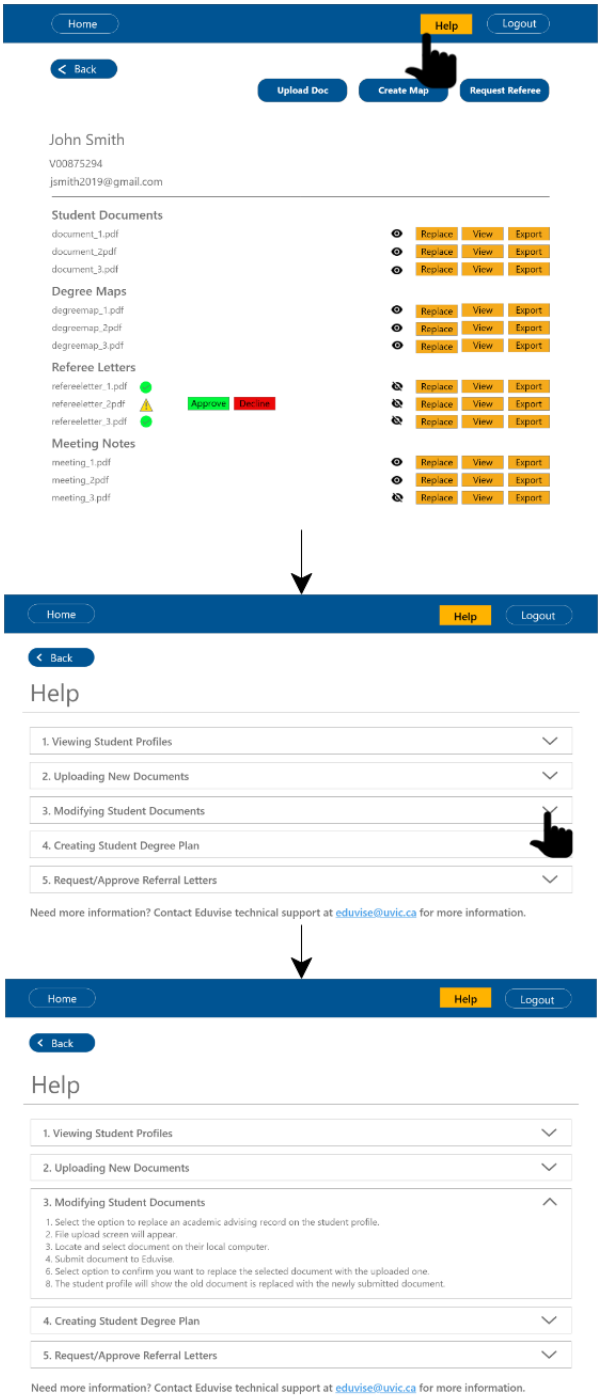


Figure 34: UI Model 5.2 - Getting Help

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	• User is logged in to Eduvise. • User is viewing the profile of a student.
Steps	1 User selects the option to create a new student degree map. 2 An empty degree template is produced for the user. 3 User selects the desired term from the template. 4 User selects from a list of classes to add to the term. 5 User repeats step 3 and 4 for each desired term. 6 User selects option to submit student degree map. 7 User is shown a degree map on the the student's profile
Success conditions	• The desired degree map is produced and saved to the student's profile.
Alternate path	5.1 User cancels creation of degree map. 5.2 New student degree map is discarded and no changes are made to the student profile.
Requirements	REQ-7.2

Table 15: Use Case 7.1 - Creating Student Degree Map

Sequence Diagram

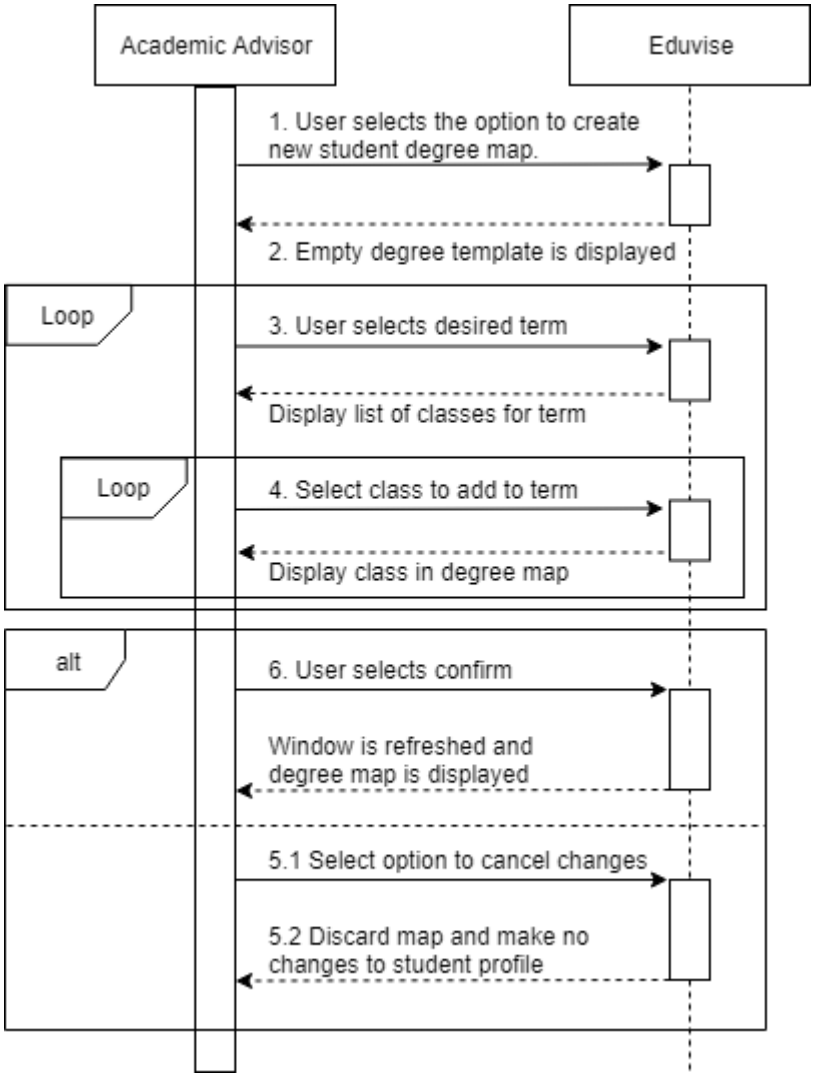
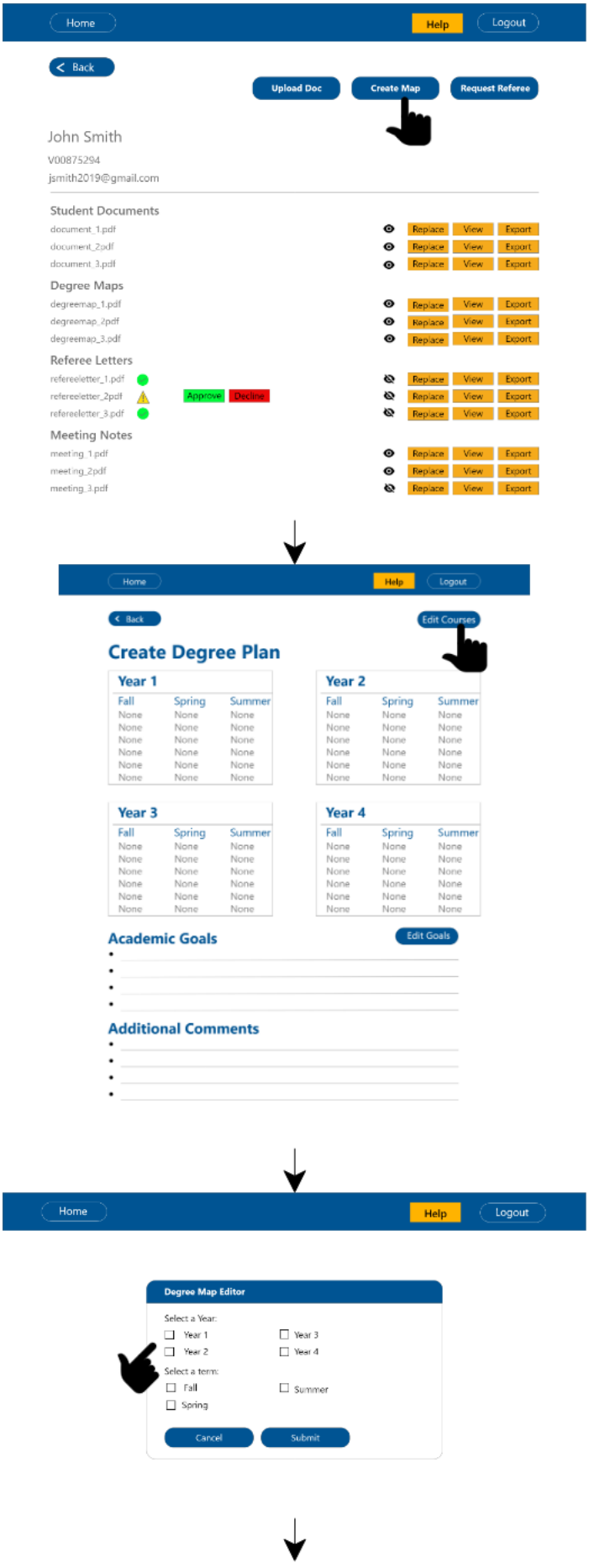


Figure 35: Sequence Diagram 7.1 - Creating Student Degree Map

User Interface Model

This scenario 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 "Edit 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. When all terms have been completed the advisor clicks the "Save Map" button to submit the new degree map to Eduvise.


Home
Help
Logout
Back
Upload Doc
Create Map
Request Referee
John Smith
V00875294
jsmith2019@gmail.com
Student Documents
document_1.pdf
document_2.pdf
document_3.pdf
Degree Maps
degreemap_1.pdf
degreemap_2pdf
degreemap_3.pdf
Referee Letters
refereeletter_1.pdf
refereeletter_2pdf
refereeletter_3.pdf
Approve
Decline
Meeting Notes
meeting_1.pdf
meeting_2.pdf
meeting_3.pdf
Replace
View
Export
Edit Courses
Create Degree Plan
Year 1
Year 2
Year 3
Year 4
Fall
Spring
Summer
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
Cancel
Submit

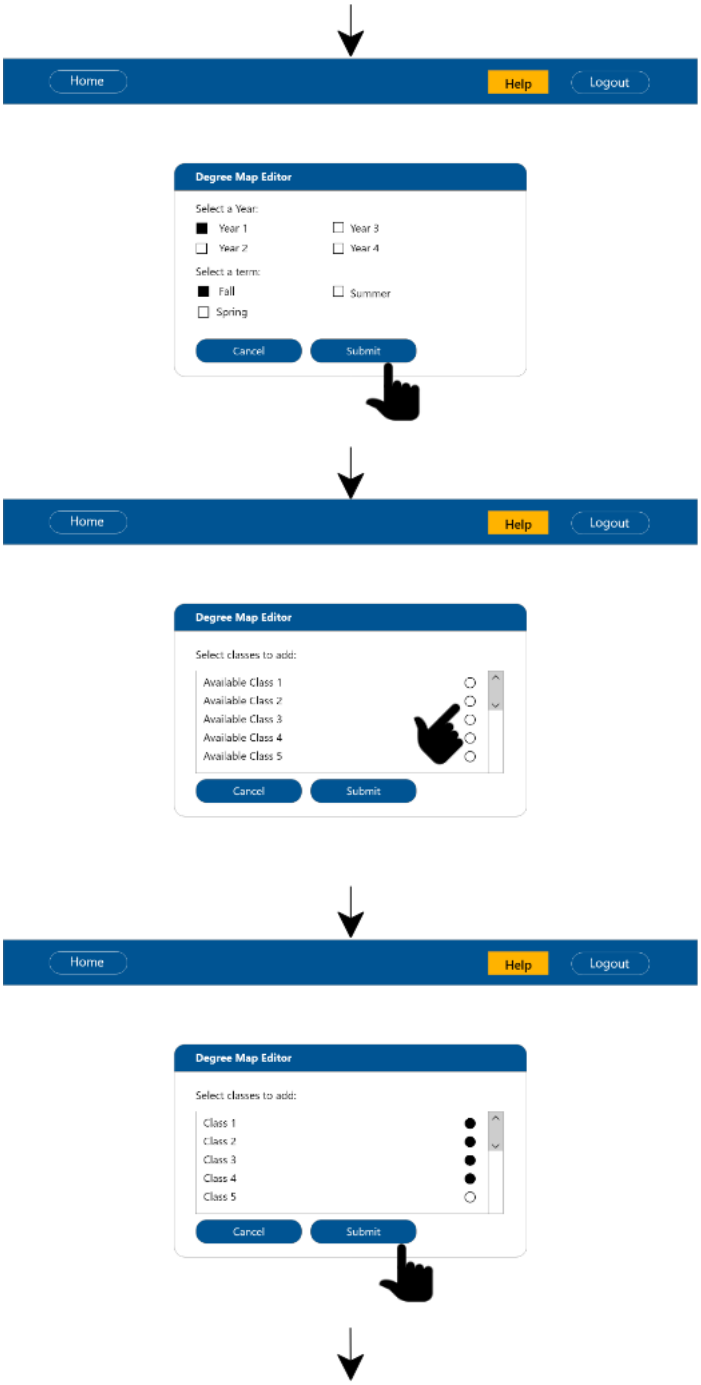

Home
Help
Logout
Degree Map Editor
Select a Year:
Year 1
Year 2
Year 3
Year 4
Select a term:
Fall
Spring
Summer
Cancel
Submit
Home
Help
Logout
Degree Map Editor
Select classes to add:
Available Class 1
Available Class 2
Available Class 3
Available Class 4
Available Class 5
Cancel
Submit
Home
Help
Logout
Degree Map Editor
Select classes to add:
Class 1
Class 2
Class 3
Class 4
Class 5
Cancel
Submit

Figure 36: 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	• User is logged in to Eduvise. • User is viewing a student's degree map
Steps	1 User selects the class from the degree map that they wish to change. 2 User changes or deletes the selected class. 3 User selects option to save changes to the degree map. 4 The degree map on the student's profile is updated.
Success conditions	• The student's degree map has been updated correctly in Eduvise.
Alternate path	4.1 User selects option to cancel changes to degree map. 4.2 The degree map reverts to its original state.
Requirements	REQ-7.1 EIR-4.1

Table 16: Use Case 7.2 - Modifying Student Degree Map

Sequence Diagram

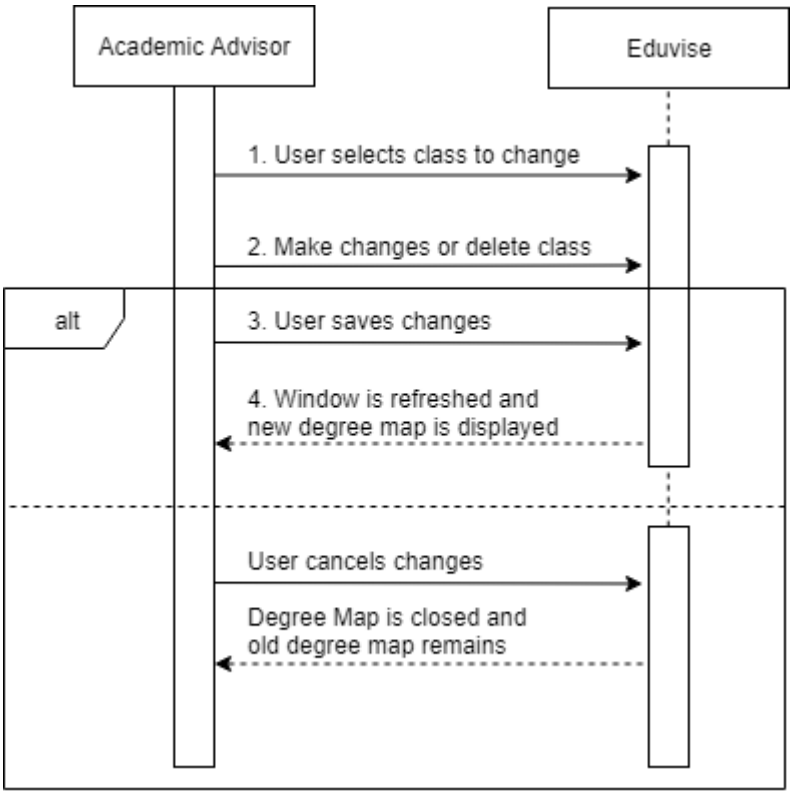
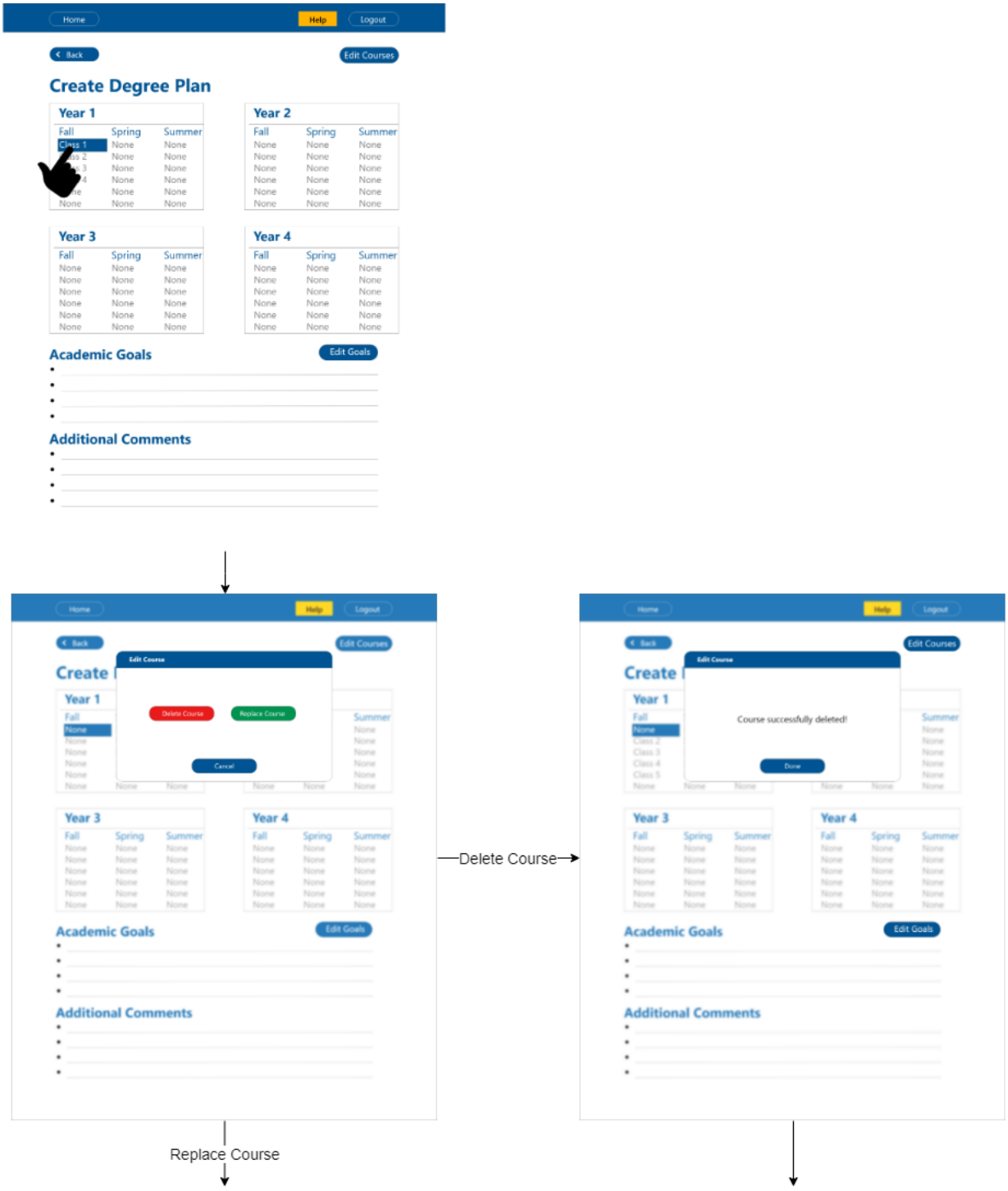


Figure 37: Sequence Diagram 7.2 - Modifying Student Degree Map

User Interface Model

This scenario outlines the steps taken by an advisor when editing an existing student degree map. In this scenario, the advisor is already viewing the document they wish to edit. The advisor selects one of the courses and is prompted with a modal dialog asking if they will delete the course or replace it. If they select replace, they are provided with a list of courses available for that term. The advisor selects one of the courses, and the course is updated on the student degree map.


Home
Help
Logout
Back
Edit Courses
Create Degree Plan
Year 1
Fall
Spring
Summer
Year 2
Year 3
Year 4
Academic Goals
Edit Goals
Additional Comments
Edit Course
Delete Course
Replace Course
Cancel
Course successfully deleted!
Done
Delete Course
Replace Course

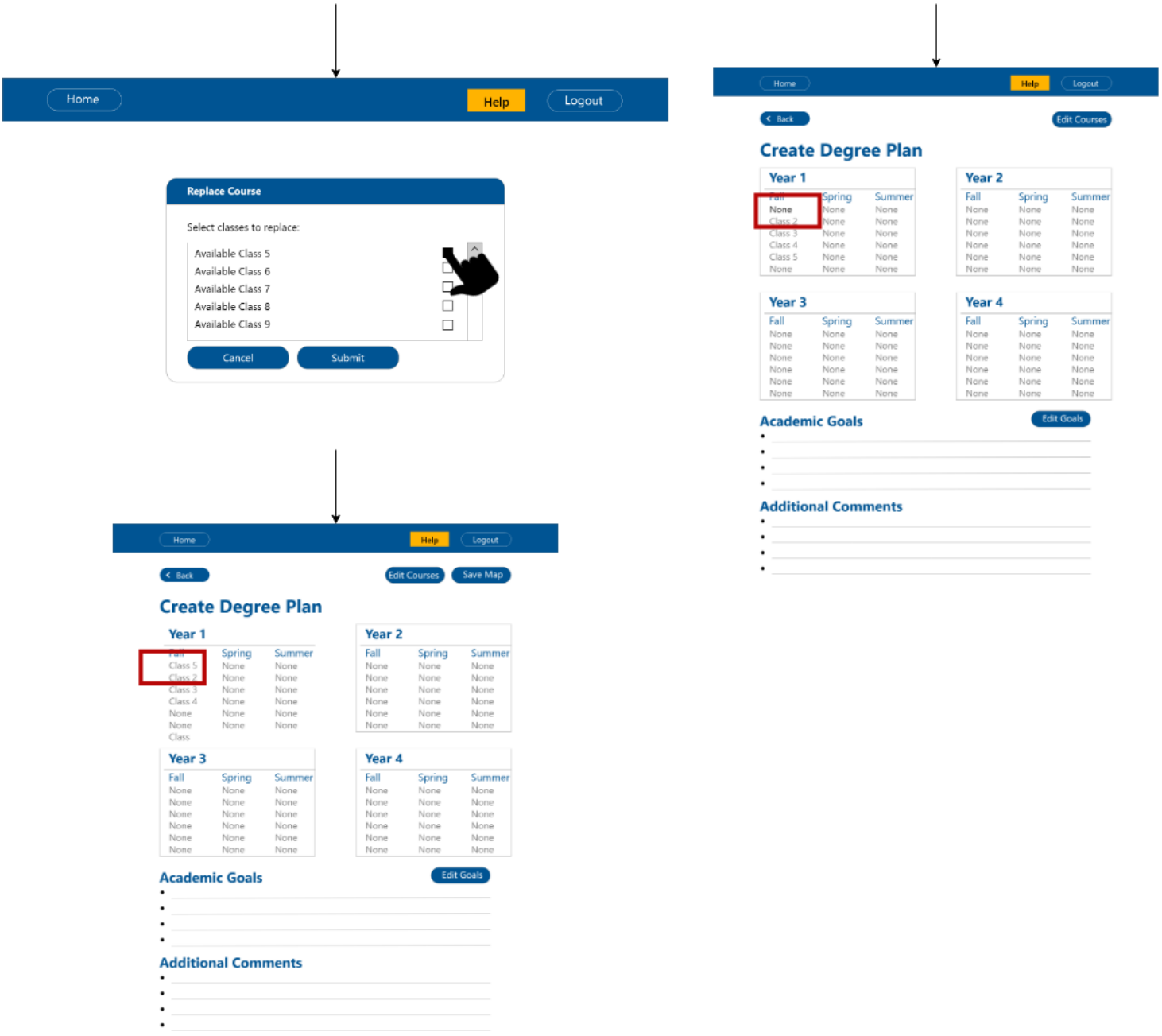


Figure 38: UI Model 7.2 - Modifying Student Degree Map

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 use 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	• User has a valid NetLink ID (account)
Steps	1 User navigates to the Eduvise login page. 2 User inputs NetLink ID and NetLink password. 3 User submits credentials for verification. 4 User is taken to the user's Eduvise home page.
Success conditions	• The user credentials are successfully verified and the user's home page is displayed to the user.
Alternate path	3.1 User credentials are not verified 3.2 User is notified that credentials have not been verified. 3.3 The use case restarts at step 1.
Requirements	None

Table 17: Use Case 8.1 - Logging in

Sequence Diagram

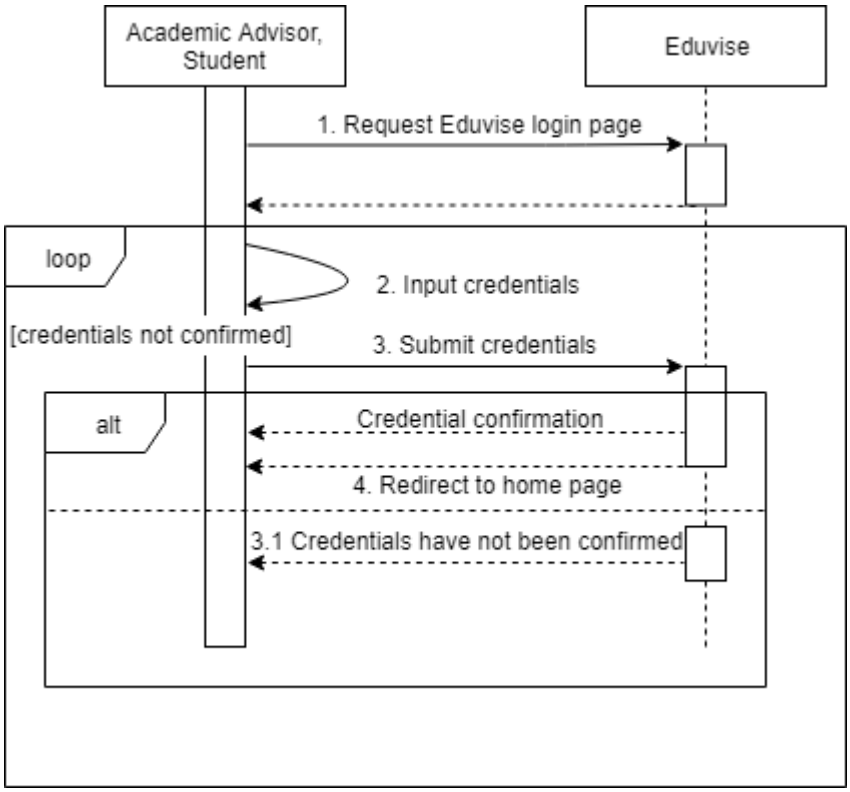


Figure 39: Sequence Diagram 8.1 - Logging In

User Interface Model

This scenario 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.

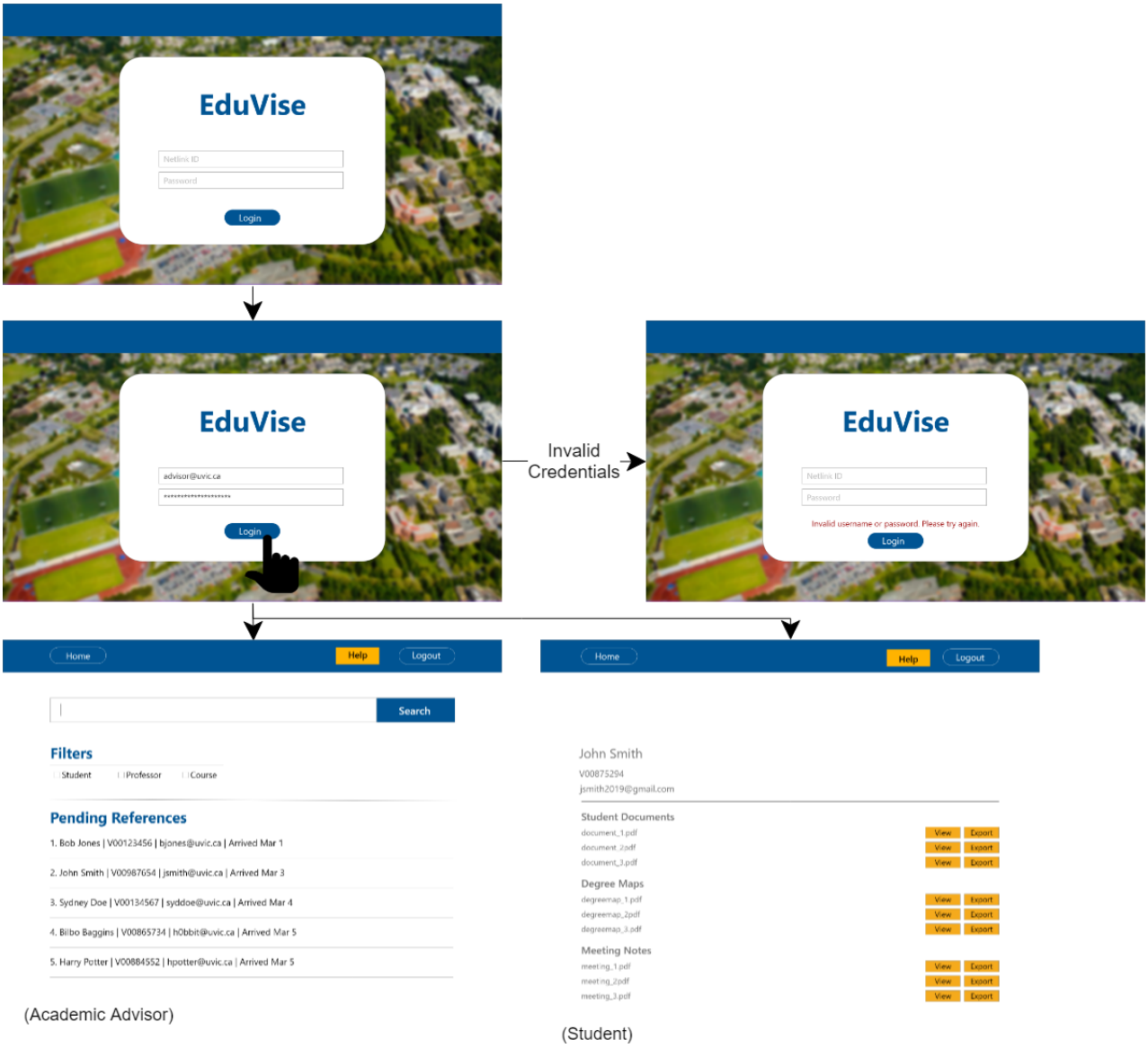


Figure 40: UI Model 8.1 Logging In

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, 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].