



COMP 3110 – FINAL PROJECT

School of Computer Science

Jameel Jiwani 104906763 | Bilal Malik 104435995 | Taqi Farooqui

104073892 | Muhammad Faraz Sohail 104986019 | Tara Das 104693373

| Alex Enache 104767870 | Keerthana Madhavan 104995097

Table of Contents

<i>Phase 1 Task 2: Problem Selection.....</i>	<i>2</i>
<i>Problem Description</i>	<i>2</i>
<i>Phase 1 Task 3: Business Reasons and Requirements</i>	<i>3</i>
<i>Phase 1 Task 4: Glossary</i>	<i>4</i>
<i>Phase 2 Task 1: Functional Requirements</i>	<i>7</i>
<i>Phase 2 Task 2 NON-FUNCTIONAL REQUIREMENTS</i>	<i>8</i>
<i>Phase 2 Task 2: Use Case Diagram and Description</i>	<i>10</i>
<i>Phase 2 Task 3 Domain Model</i>	<i>22</i>
<i>Phase 2 Task 4 Sequence Diagram</i>	<i>23</i>
<i>Phase 3 Task 1 Entity Relationship Diagram</i>	<i>34</i>
<i>Phase 3 Task 3 Software and Hardware Specification</i>	<i>35</i>
<i>Phase 3 Task 3a Design Class Diagram</i>	<i>36</i>
<i>Phase 3 Task 3b Design Sequence Diagram</i>	<i>37</i>
<i>Phase 3 Task 3c Three Layer Design Class Diagram.....</i>	<i>48</i>
<i>Phase 3 Task 3d Interface Design</i>	<i>49</i>
<i>Phase 3 Task 3e Table Diagram.....</i>	<i>54</i>
<i>Phase 3 Task 4 Demonstration Appointment Booking.....</i>	<i>55</i>

Phase 1 Task 2: Problem Selection

In our selection process, our team gathered different problems that people face in the world around us. We started with a brain storming session to collect ideas and mapped them out based on their complexity. We then spent the next week performing an in-depth analysis to determine how our skills can be used collectively to solve the problems we came up with. After we assembled the data found from our investigation, we narrowed the list down to just 3 problems that we felt the most confident in. Based on our skills and knowledge, we chose to work on a Course Sequence Generator for an undergraduate degree. Since there are many departments and majors in the university, we chose to create a system to help the students in the department of Computer Science.

Problem Description

The students in the computer science department at the university of Windsor face some very interesting problems that are very commonly overlooked. Some of the problem's students were having include issues related to courses that are only offered in particular semesters. Students will have to ensure that they are taking prerequisites in the correct semester to give themselves enough time to take the courses proceeding them. Students also need to consider how to distribute those courses, so their semesters are not too heavy. Some courses in the department rely heavily on assignment work and taking all of them concurrently can leave the student both burnt out and unmotivated. Most students try to manually plan out their entire degree, but it can be tricky if they don't know the rules of how they are scheduled. These rules cannot be found anywhere within the university's resources and are only passed on colloquially. In order to plan out their whole degree and ensure they will graduate within the time frame they expected, they need some sort of tool to guide them in this process. In order to tackle these problems, our system will produce a visual representation of a person's courses throughout their undergraduate career.

Phase 1 Task 3: Business Reasons and Requirements

Our system is designed to solve a unique and significant problem that students are facing at the University of Windsor. Students sometimes feel lost when posed with the challenge of picking their courses and ensuring their studies are on the right track to graduate on time. They are often unaware that certain courses are exclusively offered at specific times during the year and there isn't currently an implementation in place to find that information. We plan to solve this problem by designing a system that not only helps students plan their semester; but their entire degree. In terms of viability, there is a large audience that is willing to utilize this software, making this a tool that has the potential to be very high in demand. Utilizing different software engineering methods, we can minimize both the time, and cost of developing the software. Students will be able to select their major and specialization along with a timeline of how long they expect their degree to take. Based on the information provided, our system will generate a sequence that best fits their needs. Also, students will need a way to look back at the information (sequence) for reference in future when they are scheduling for their courses or reviewing their sequence. In order to do this, a functionality will be implemented that will allow users to create an account, save their sessions and log in later to view it. When the sequence is displayed, users can get more information about each course offered and see its description along with its prerequisites.

Phase 1 Task 4: Glossary

1. Semester - The set of 4 months in question. Denoted by Fall, Winter, and Summer.
2. Year - The year of study of the student or the year level of the course in question.
3. Course - The course ID and title.
4. Program - The degree program of study of the student in question.
5. Specialization - The program specialization of the student. (if applicable)
6. Availability - The availability of a course in question.
7. Conflict - A conflict between two or more course sections at the same or similar time slot.
8. Practical - A practical or laboratory section of a course.
9. Section - An allotted time for a course. This can be as a lecture or practical.
10. Lecture - An allotted time slot for teaching of a course by the professor.
11. Room - The assigned room for a given section.
12. Graduation Date - The expected graduation date for a student.
13. Enrollment Date - The enrollment date of a student.
14. Fall - The Fall semester, from September to December.
15. Winter - The Winter semester, from January to April.
16. Summer - The Summer semester, from June to August.
17. Intersession - The Intersession semester, from May to June.
18. Weeks - The amount of calendar weeks that a course will be in session.
19. Course Code - The course id.

- 20. Title - The course title.
- 21. Days - The days in a week that the section is in session.
- 22. Times - The times in the day that a section is in session.
- 23. Department - The department of the university that is offering the course in question, or the program that the student is enrolled in.
- 24. Prerequisites - The courses required to be completed in order for the student to enroll in a new course.
- 25. Status (available, conflict) - Availability of a course section.
- 26. Description - The course description.
- 27. Required Courses - Courses needed for the completion of the degree.
- 28. Core courses - Courses needed for the major
- 29. Non-Core Courses - Non-degree requirements for the program.

Phase 2 Task 1: Functional Requirements

Member: Everyone

ID	Description	Priority
RQ1	The system should allow the student to manage sequence	medium
RQ2	The system should allow the student to create sequence	High
RQ3	The system should allow the student to delete sequence	medium
RQ4	The system should allow the student to view sequence	High
RQ5	The system should allow the student to export/print sequence	Medium
RQ6	The system should allow the student to register account	High
RQ7	The system should allow the Administrator to manage majors	High
RQ8	The system should allow the Administrator to update majors	Medium
RQ9	The system should allow the Administrator to create majors	High
RQ10	The system should allow the Administrator to delete majors	Medium
RQ11	The system should allow the Administrator to add course	High

Phase 2 Task 2 NON-FUNCTIONAL REQUIREMENTS

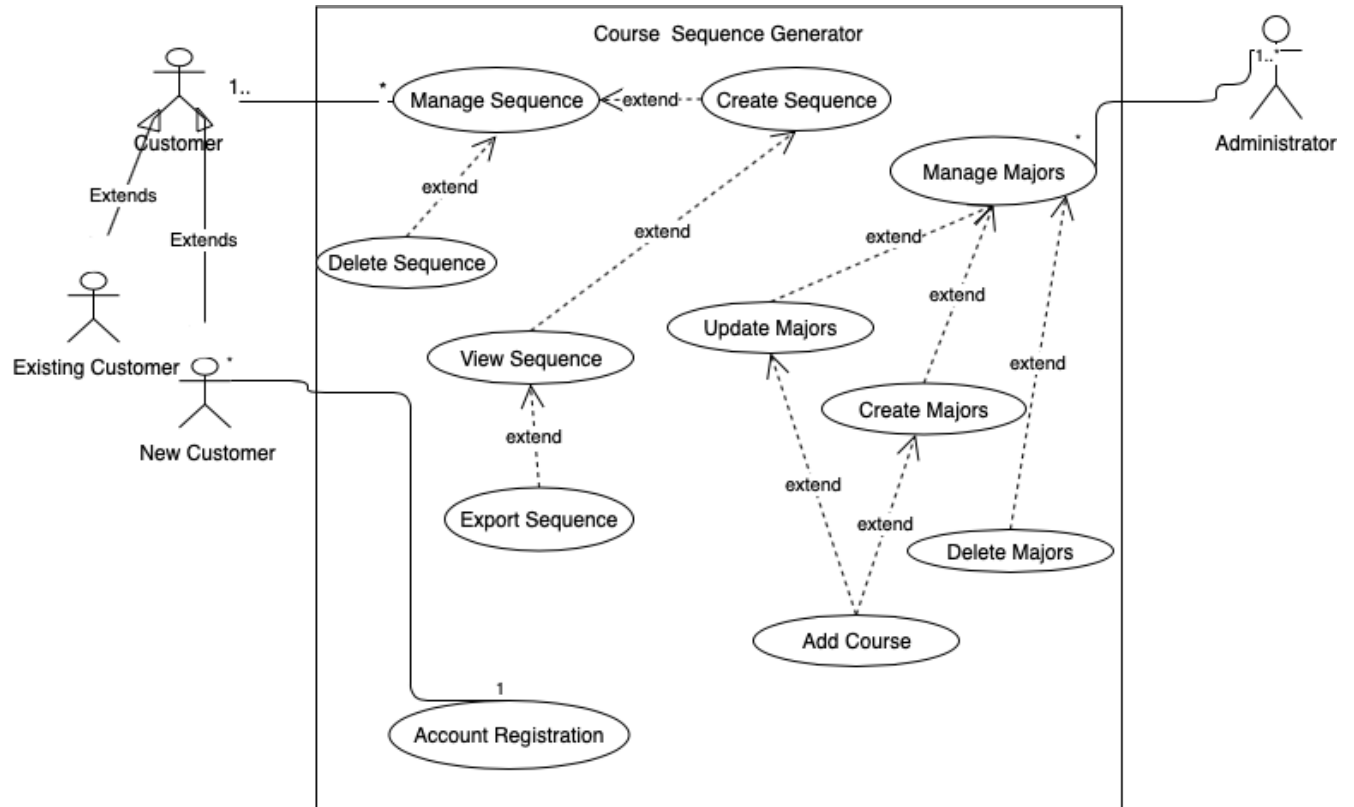
Member: Jameel Jiwani, Keerthana Madhavan

ID	Description	Priority
Operational Requirements		
RQ15	The system should be able to be work on a laptop, mobile or tablet	Low
RQ16	The system should be able to work on any web browser.	Medium
RQ17	The system should be able to operate in any OS environment.	High
RQ18	The system should be able to back up at the end of each day.	Low
RQ19	The system should be able to connect to printers (both wireless and wired).	Medium
RQ20	The system servers (both servers) and user must always have a connection to the internet.	High
RQ21	The system connection should be stable, and the system server should have one hundred percent uptime	Medium
Performance Requirements		
RQ22	Any interaction between the user and the system should not exceed 3 seconds.	High
RQ23	The system should receive updated data information to the server every 15 minutes. For each student to be reflected on their accounts.	Low
RQ24	The system should be available for use 24 hours per day, 365 days per year.	High
RQ25	The system should have the capacity to store all the majors, courses, and users.	High
RQ26	The system should be able to maintain data concurrency and data consistency.	High

Security		
RQ27	Only administrators can see personnel records of students and their schedules.	High
RQ28	Only the managers should be able to make changes like add majors, maintain databases, and maintain students account	High
RQ29	Students can see all their saved sequences from their account only	High
RQ30	Passwords of the students are encrypted in the database.	High
Cultural and Political Requirements		
RQ31	The system will only be available in English.	Medium
RQ32	The system will be a simple design for non-native speakers/readers.	Low

Phase 2 Task 2: Use Case Diagram and Description

Member: Keerthana Madhavan



Member: Keerthana Madhavan

Use Case Name: Create Sequence	ID: UC-1	Priority: High
Actor: Student		
Description: This use case describes a student who can create the sequence after inputting necessary information in the required form		
Trigger: the student wants to create a sequence (that lists all the courses that the student has to take in their 4-year undergraduate degree) for their major		
Type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal		
Preconditions:		
Normal Course: <ol style="list-style-type: none">1. After the student has logged in to his/her account. They are redirected to the Course Sequence Information Form.2. The Student can add majors by submitting this form with valid information.3. After entering all the required data field, the student submits the form to the system.4. Then the system communicates with our two servers to generate the sequence5. The sequence is generated broken into number years, semester and a list of course the student should take.		Information for Steps:
Alternative Courses:		
Postconditions: <ol style="list-style-type: none">1. The student can view the generated sequence that is in a pdf format.2. The sequence is created, and the sequence is saved in the sequence section of the website		
Exceptions: E1: Account is not valid (occurs at step 1) <ol style="list-style-type: none">1. System displays message that username/password is not valid2. System prompt the student to re-inter his/her account credentials E2: The information entered is not valid (occurs at step 3) <ol style="list-style-type: none">1. System displays message that the entered information is not valid2. System asks the student to reinter logical information.		

Use Case Name: Manage Sequence	ID: UC-2	Priority: Medium
Actor: Student		
Description: This use case describes a student who can manage their sequence that are stored in their sequences list of the website		
Trigger: the student wants to create, view, delete, print, or export a sequence (that lists all the courses that the student has to take in their 4-year undergraduate degree) for their major		
Type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal		
Preconditions:		
Normal Course: <div><div>1. After the student has logged in to his/her account. They are redirected to the home page</div><div>2. The Student can back click on a particular sequence.</div><div>3. Then they can choose from these options: delete, create, view, print or export sequence</div></div>	Information for Steps:	
Alternative Courses:		
Postconditions: <div><div>1. Based on what the student chooses, the system performs the operation.</div></div>		
Exceptions:		
<div>E1: Account is not valid (occurs at step 1)<div><div>1. System displays message that username/password is not valid</div><div>2. System prompt the student to re-inter his/her account credentials</div></div></div> <div>E2: The information chosen is not valid or does not exist (occurs at step 3)<div><div>1. System displays message that the entered information is not valid</div><div>2. System asks the student to re-choose an option</div></div></div>		

Use Case Name: View Sequence		ID: UC-3	Priority: High
Actor: Student			
Description: This use case describes a student who wants to view the sequence after it is generated			
Trigger: the student wants to view his/her course sequence from their sequence webpage section			
Type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal			
Preconditions: 1. There should be at least one generated course sequence saved in the student’s account			
Normal Course: 1. After the student has logged in to his/her account. They are redirected to the Course Sequence Information Form. (home page) 2. They navigate to the Sequence Pages that has all the saved course sequences 3. The student can now choose which sequences, view and click open to view the respective sequences		Information for Steps:	
Alternative Courses:			
Postconditions: 1. The student has their sequence popped up on the screen			
Exceptions: E1: Account is not valid (occurs at step 1) 1. System displays message that username/password is not valid 2. System prompt the student to re-enter his/her account credentials E2: Error opening Course Sequence Files (occurs at step 3) 1. System displays a message that the file cannot be opened 2. System prompt the user to regenerate the course sequence			

Use Case Name: Delete Sequence	ID: UC-4	Priority: Medium
Actor: Student		
Description: This use case describes a student who wants to delete the sequence after it is generated		
Trigger: the student wants to delete his/her course sequence from their sequence webpage section		
Type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal		
Preconditions: 2. There should be at least one generated course sequence saved in the student’s account		
Normal Course: 4. After the student has logged in to his/her account. They are redirected to the Course Sequence Information Form. (home page) 5. They navigate to the Sequence Pages that has all the saved course sequences 6. The student can now choose which sequences, click and delete the respective sequences	Information for Steps:	
Alternative Courses:		
Postconditions: 2. The student have their sequences deleted and off the list		
Exceptions: E1: Account is not valid (occurs at step 1) 3. System displays message that username/password is not valid 4. System prompt the student to re-enter his/her account credentials E2: Error opening Course Sequence Files (occurs at step 3) 3. System displays a message that the file cannot be opened 4. System prompt the user to refresh the screen and delete again		

Use Case Name: Export Sequence	ID: UC-5	Priority: Medium
Actor: Student		
Description: The use case describes a student who can save/export and print their sequence in the system both externally and within their account, after it is generated		
Trigger: the student wants to save his/her sequence to an external location		
Type: <input checked="" type="checkbox"/> External: Temporal:		
Preconditions: 1. The course sequence must be generated or must exist in the sequence page		
Normal Course: 1. After the student has logged in to his/her account. They are redirected to the Course Sequence Information Form. (home page) 2. They navigate to the Sequence Pages that has all the saved course sequences 3. The student can now choose which sequences, open and click to save or export to a desired location and print it	Information for Steps:	
Alternative Courses:		
Postconditions: 1. A saved copy of the course sequence in the student chosen location		
Exceptions: E1: Account is not valid (occurs at step 1) 5. System displays message that username/password is not valid 6. System prompt the student to re-enter his/her account credentials E2: Error saving Course Sequence Files (occurs at step 3) 5. System displays a message that the file cannot be saved 6. System prompts the student to refresh the page and try again 7. System prompt the user to regenerate the course sequence, if refreshing didn't work.		

Use Case Name: Manage Majors	ID: UC-6	Priority: High
Actor: Administrator		
Description: The use case describes an administrator who can manage the majors in the DAD Sequence account		
Trigger: When the administrator wants to manage the majors in the DAD Sequence System		
Type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal		
Preconditions: 1. Have at least one majors stored in their account		
Normal Course: 1. After the administrator has logged in to his/her account. They are redirected to the home page 2. They navigate to the settings page 3. They can manage majors in the Majors section, by adding, deleting, or updating majors that are present in the Computer Science	Information for Steps:	
Alternative Courses:		
Postconditions: 1. Updated Majors in the DAD Sequence system 1.1 Create Majors 1.2 Read Majors 1.3 Update Majors 1.4 Delete Majors 1.5 Add Majors 1.6 Edit Majors 2. When there is a change in the majors, the possible courses sequence for each major should be updated.		
Exceptions: E1: Error managing majors like problems updating the database (occurs at step 3) 1. System displays the to the administrator that there were technical glitches in the server while communication with the database serves to manage the majors 2. System can prompt the administrator to refresh the webpage and try again		

Use Case Name: Update Majors	ID: UC-7	Priority: Medium
Actor: Administrator		
Description: The use case describes an administrator who can update the majors in the DAD Sequence account with minor information		
Trigger: When the administrator wants to update the majors in the DAD Sequence System		
Type: <input checked="" type="checkbox"/> External Temporal		
Preconditions: 2. Have at least one majors stored in their account		
Normal Course: 4. After the administrator has logged in to his/her account. They are redirected to the home page 5. They navigate to the settings page 6. They can update majors in the Majors section, by adding, deleting, or updating majors or making minor changes that are present in the Computer Science	Information for Steps:	
Alternative Courses:		
Postconditions: 3. Updated Majors in the DAD Sequence system 4. When there is a change in the majors, the possible courses sequence for each major should be updated.		
Exceptions: E1: Error managing majors like problems updating the database (occurs at step 3) 3. System displays the to the administrator that there were technical glitches in the server while communication with the database serves to manage the majors 4. System can prompt the administrator to refresh the webpage and try again		

Use Case Name: Create Majors	ID: UC-8	Priority: High
Actor: Administrator		
Description: The use case describes an administrator who can create the majors in the DAD Sequence account with minor information		
Trigger: When the administrator. wants to create new majors in the DAD Sequence System		
Type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal		
Preconditions:		
Normal Course: <div>7. After the administrator has logged in to his/her account. They are redirected to the home page 8. They navigate to the settings page 9. They can create new majors</div>	Information for Steps:	
Alternative Courses:		
Postconditions: <div>1. The new majors’ sequences have to be updated in the database servers with a new tuple of information for the newly added major 2. When there is a change in the majors, the possible courses sequence for each major should be updated.</div>		
Exceptions: E1: Error managing majors like problems updating the database (occurs at step 3) <div>1. System displays the to the administrator that there were technical glitches in the server while communication with the database serves to manage the majors 2. System can prompt the administrator to refresh the webpage and try again</div>		

Use Case Name: Delete Majors	ID: UC-9	Priority: Medium
Actor: Administrator		
Description: The use case describes an administrator who can delete the majors in the DAD Sequence system		
Trigger: When the administrator wants to create delete a major in the DAD Sequence System		
Type: <input checked="" type="checkbox"/> External Temporal		
Preconditions: 1. Have at least one majors stored in their system		
Normal Course: 10. After the administrator has logged in to his/her account. They are redirected to the home page 11. They navigate to the settings page 12. They can choose from the list of majors 13. They can delete the chosen majors	Information for Steps:	
Alternative Courses:		
Postconditions: 3. The new majors' sequences have to be updated in the database servers with a new tuple of information for the newly added major 4. When there is a change in the majors, the possible courses sequence for each major should be updated.		
Exceptions: E1: Error managing majors like problems updating the database (occurs at step 3) 3. System displays the to the administrator that there were technical glitches in the server while communication with the database serves to manage the majors 4. System can prompt the administrator to refresh the webpage and try again		

Use Case Name: Add Courses	ID: UC-10	Priority: High
Actor: Administrator		
Description: The use case describes that the administrator can add new courses when new majors are added		
Trigger: When the administrator wants to add extra courses for the major that changes were made to by the department		
Type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal		
Preconditions: <div><div>1. A generate course sequence for any major should exist before the student is allowed to add any courses</div><div>2. Or the administrator can add a completely new major with its courses to the system</div></div>		
Normal Course: <div><div>1. After the administrator has logged in to his/her account. They are redirected to the Course Sequence Information Form. (home page)</div><div>2. They navigate to the manage majors' section</div><div>3. The administrator can choose a major and add courses to that database</div><div>4. The major database is then updated.</div></div>	Information for Steps:	
Alternative Courses:		
Postconditions: <div><div>1. Updated courses for the chosen course sequence that the administrator added for a particular course</div></div>		
Exceptions: E1: Error adding course, like problems updating the database (occurs at step 4) <div><div>1. System displays the to the student that there were technical glitches in the server while communication with the database serves to manage the majors</div><div>2. System can prompt the student to refresh the webpage and try again by entering a valid course.</div></div>		

Use Case Name: Account Registration	ID: UC-11	Priority: High
Actor: Student		
Description: The use case describes that a new student can register to be part of the DAD Sequence system		
Trigger: When a new student wants to use the DAD Course Sequence services.		
Type: <input checked="" type="checkbox"/> External Temporal		
Preconditions:		
Normal Course: 1. The student visits our website 2. The system requires the student to log in or register for an account to access its services. 3. The student enters the necessary information for registration and clicks submit	Information for Steps:	
Alternative Courses:		
Postconditions: 1. The student now has account with the DAD Course Sequence Generator.		
Exceptions: E1: Account Registration Information is not valid (occurs at step 3) 1. System displays message that username/password is not valid 2. System prompt the student to re-enter his/her account credentials		

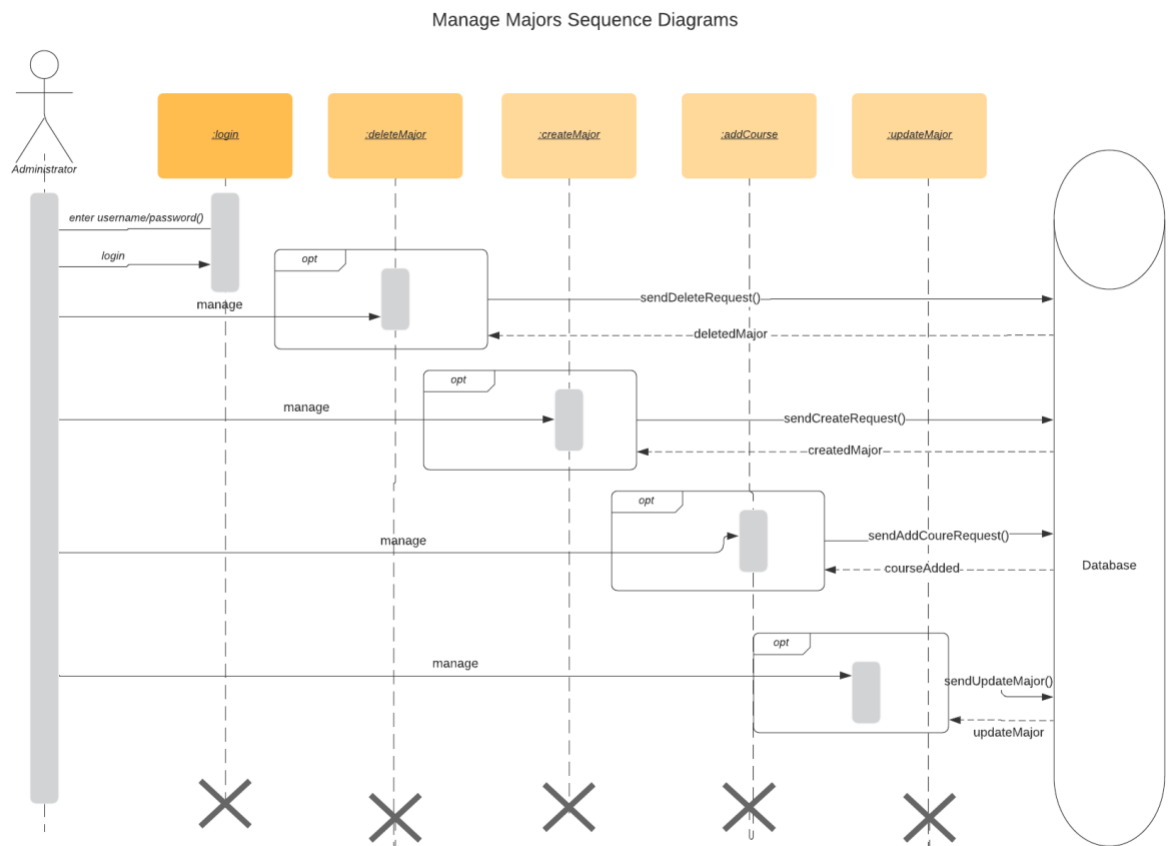
Phase 2 Task 3 Domain Model

Our system is not Object Oriented. So, Domain Model will not be required.

Phase 2 Task 4 Sequence Diagram

Member: Everyone

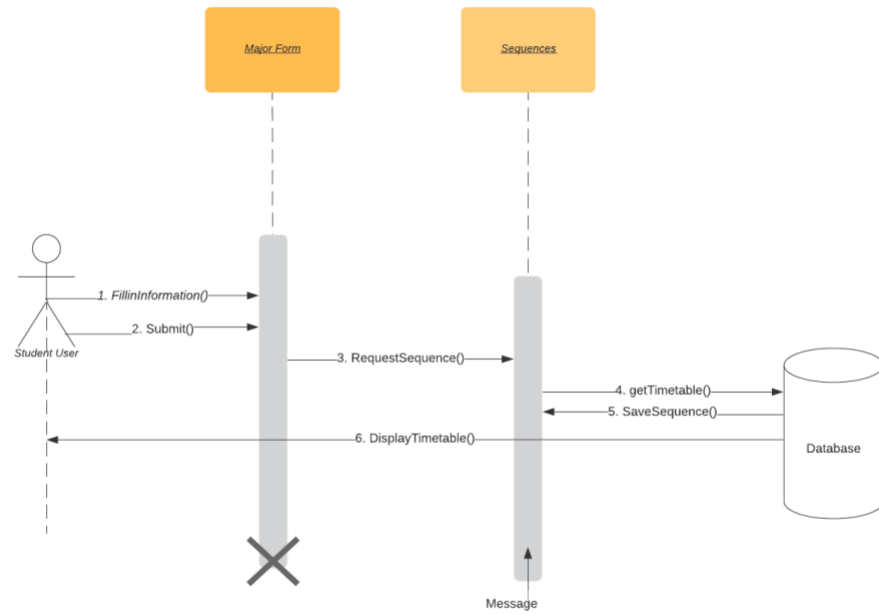
UC-6: Manage Majors



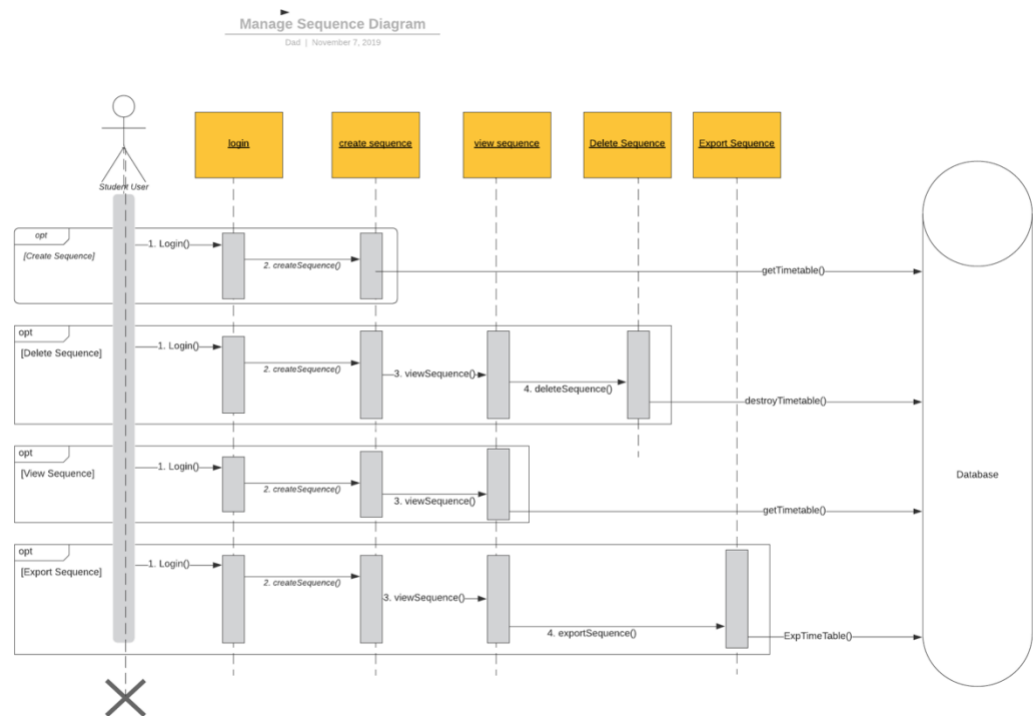
UC-1 : Create Sequence

Create Course Sequence Diagram

Bilal Malik | November 7, 2019



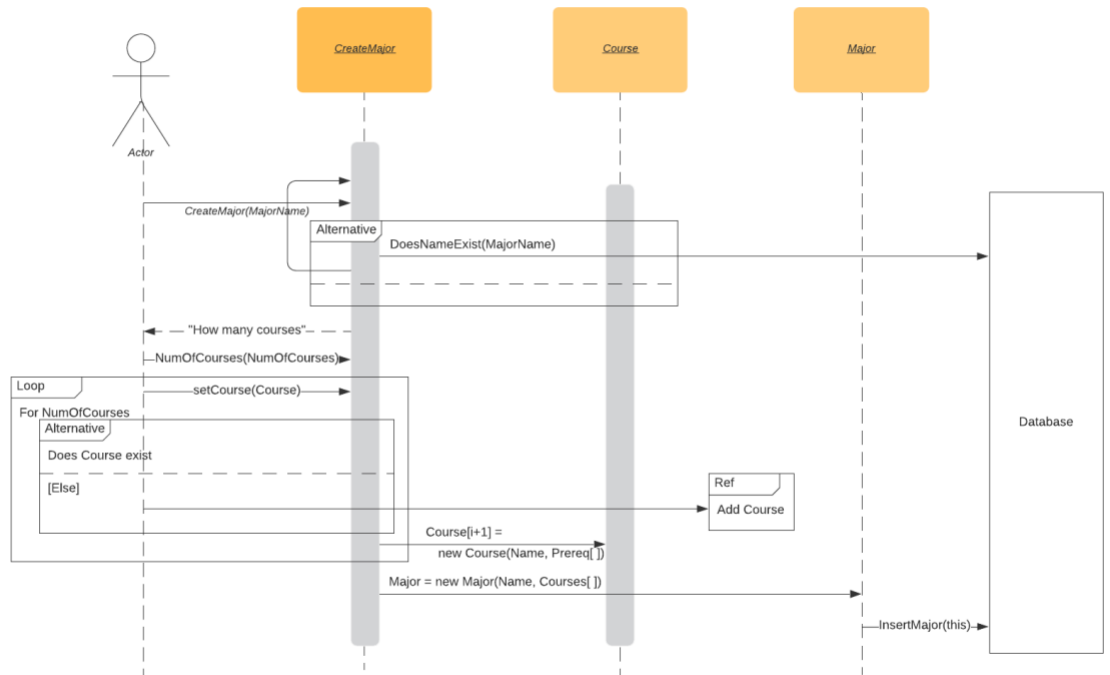
UC-2: Manage Sequence



UC-8: Create Major

Create Major

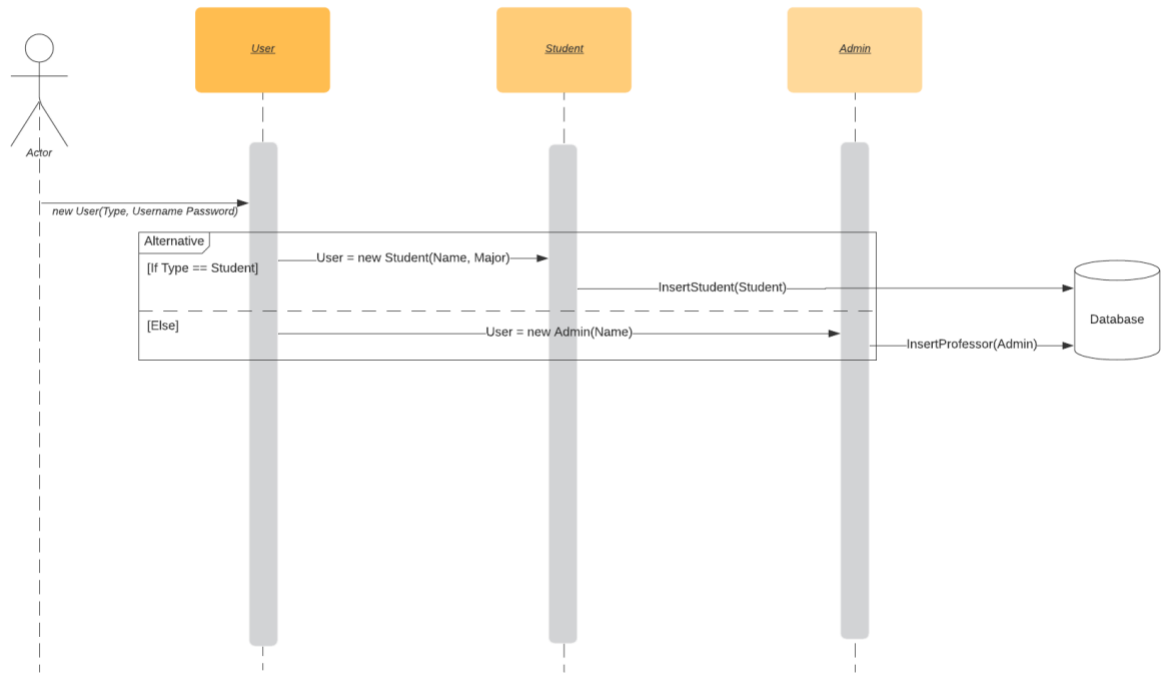
Alexander Enache | November 7, 2019



UC-11: Account Registration

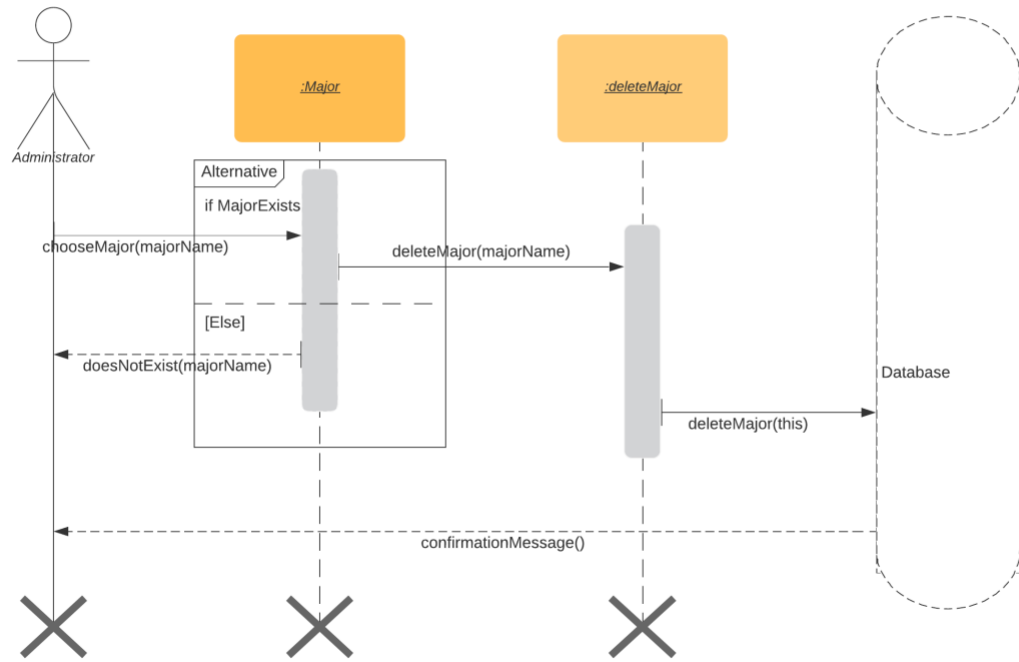
Account Registration

Alexander Enache | November 7, 2019

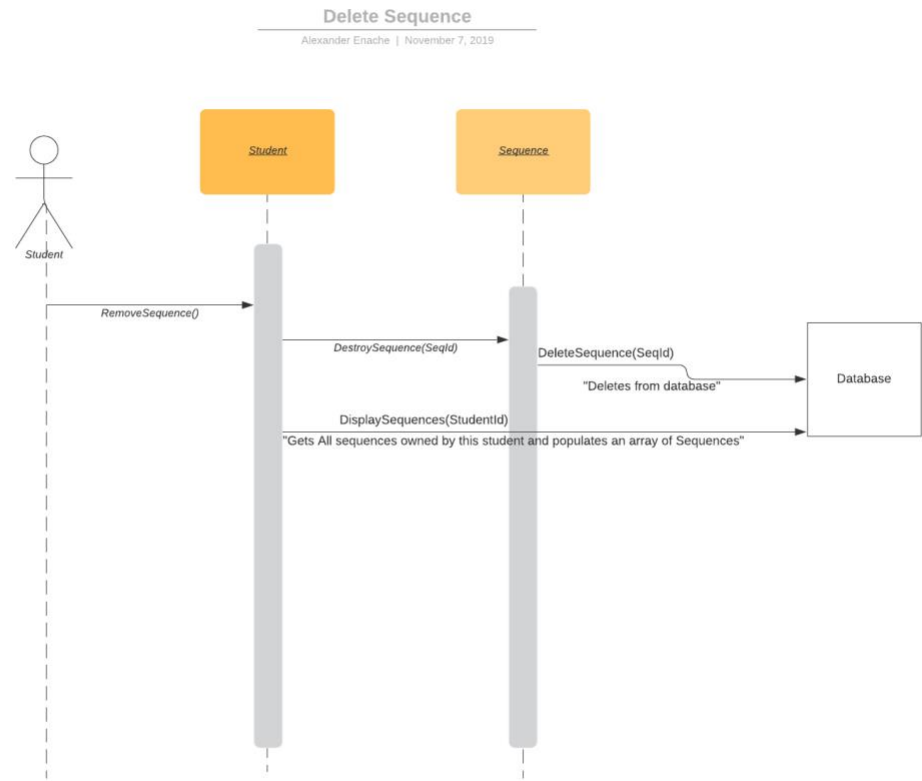


UC-9: Delete Majors

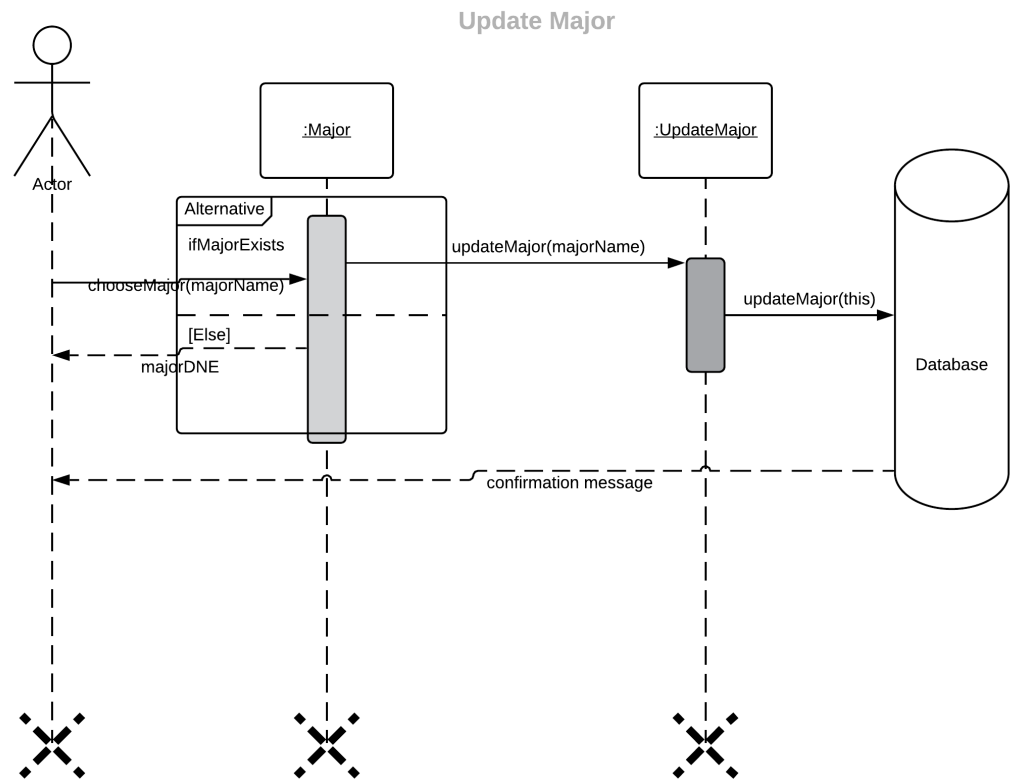
Delete Major



UC-4: Delete Sequence



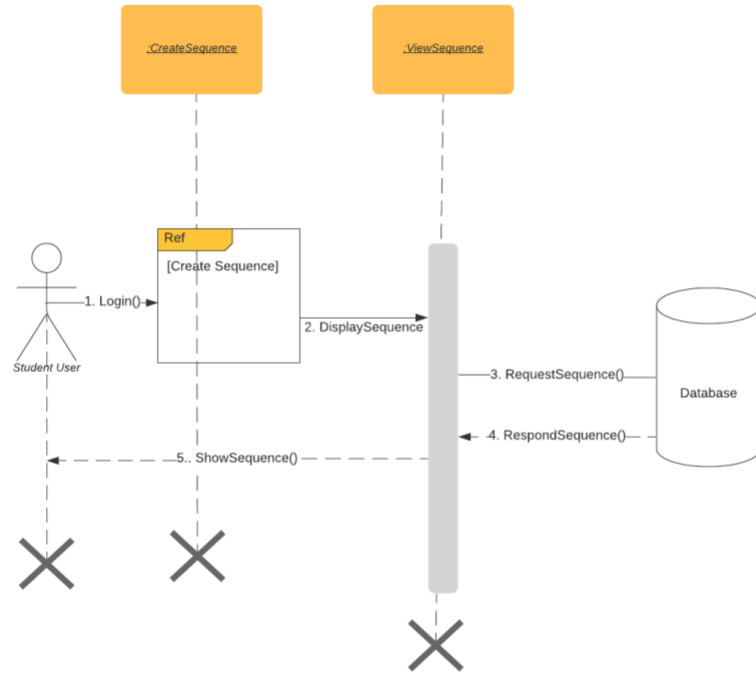
UC-7: Update Majors



UC-3: View Sequence

View Course Sequence Diagram

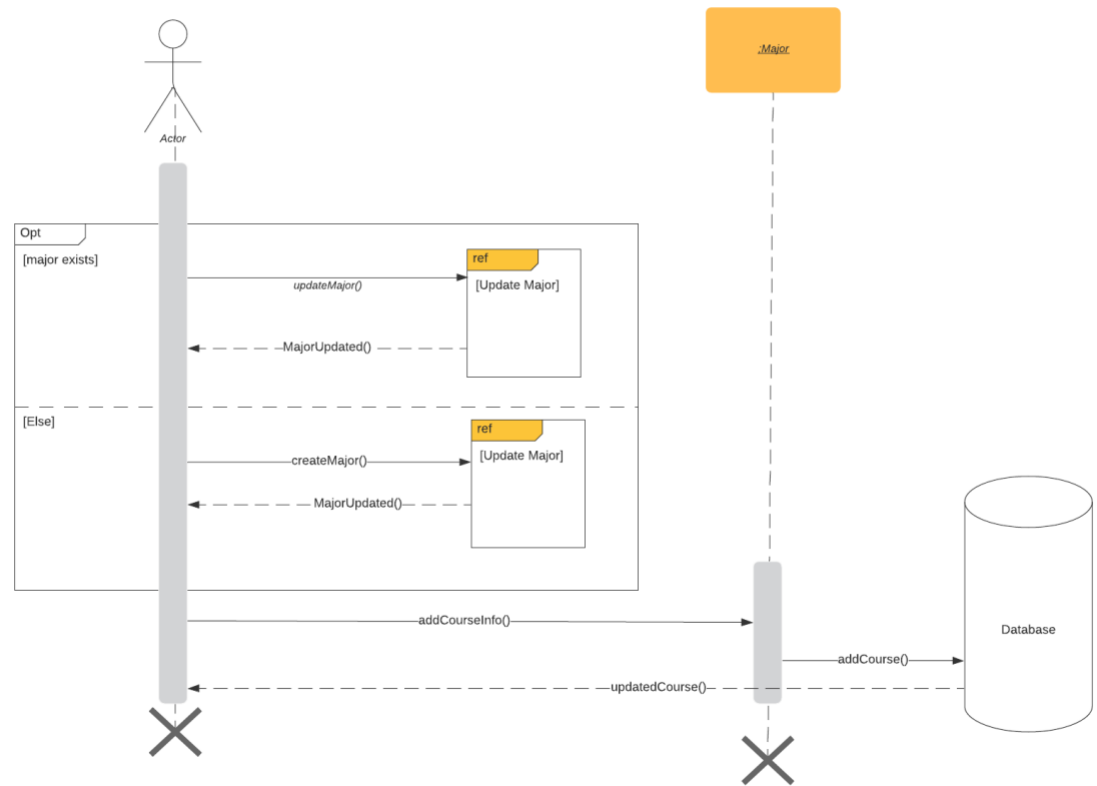
Bilal Malik | November 7, 2019



UC-10: Add Course

Add Course Diagram

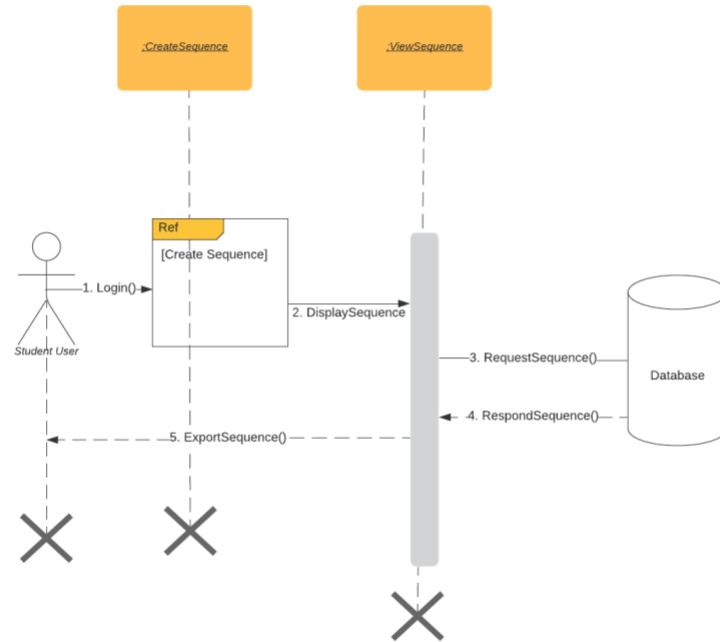
Bilal Malik | November 7, 2019



UC-5: Export Sequence

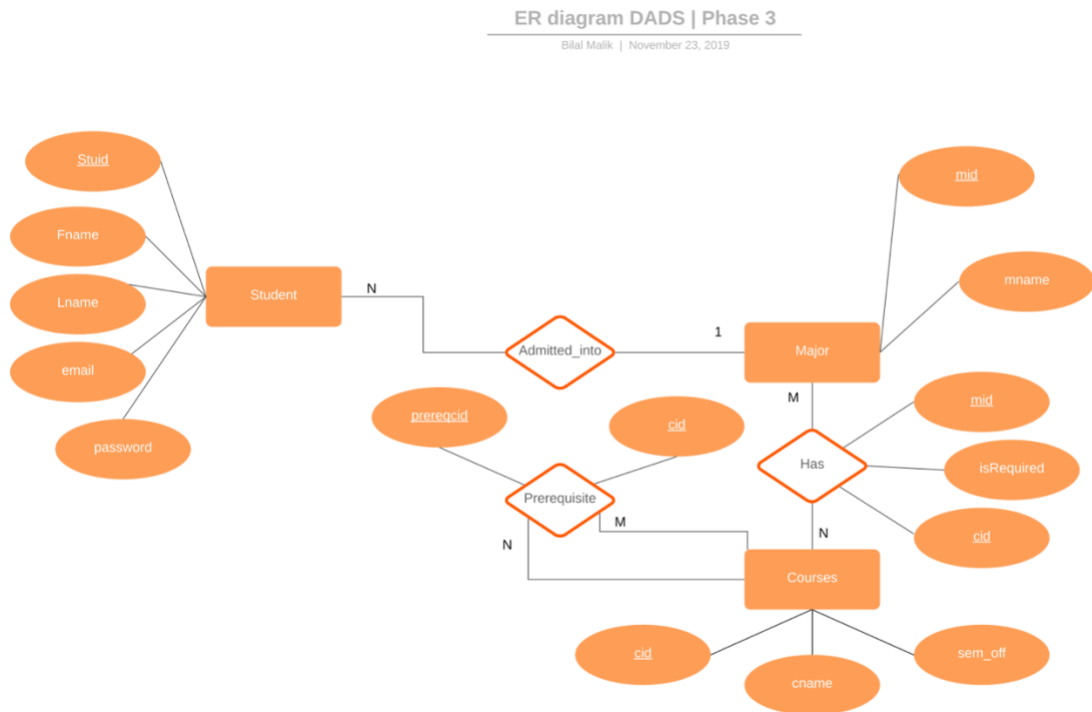
Export Course Sequence Diagram

Bilal Malik | November 7, 2019



Phase 3 Task 1 Entity Relationship Diagram

Member: Bilal Malik



Phase 3 Task 3 Software and Hardware Specification

Member: Jameel Jawani

Since we are using a 3—tiered architecture design, we need 2 servers. 1 will host our API (business logic and data layer) and the other will host our front-end. (view layer)

For our API (business logic and data layer) we decided to use an Amazon Web Service T2.Micro EC2 (Elastic Cloud Compute). The T2 Micro server we chose has 1 CPU and 1 GB of RAM and is currently free to use for 1 year which is why we chose this. The EC2 Instance is connected by a 100Gbps connection and is based off of a cloud infrastructure or CDN. (Content Distribution Network) This means that it will provide users a connection to the instance of our server that is close by regardless of their geographical location.

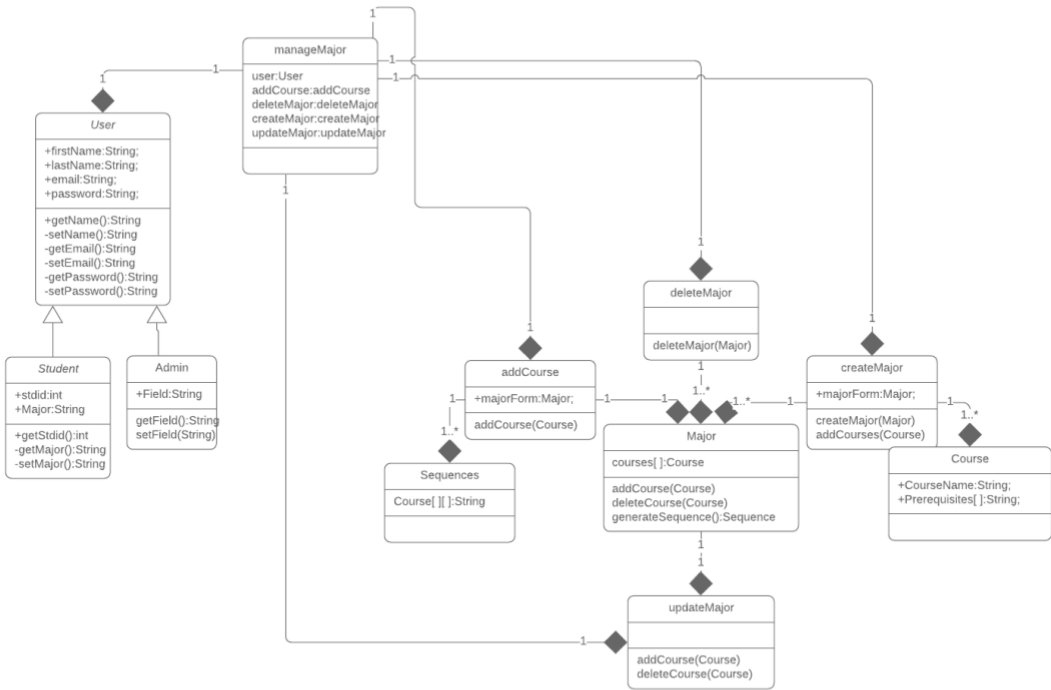
We are using the Fedora operating system which NodeJS running on it to host our API and a MySQL database on the same server. This allows a fast connection between the applications because they don't require a network transmission and stay connected via the application layer. Process communication is faster and more reliable than network communication so we thought this would be the best choice rather than to abstract the logic of the data layer.

For our front-end (view layer) we decided to use GitHub pages. The GitHub Pages website does not provide any details on the hardware or software specifications of their web hosting solution. This solution not only handles the domain name for us, but it's secure, reliable and most of all, free. The setup is extremely simple, and it doesn't require us to have any understanding of the server that hosts our application.

We used ReactJS to create the UI. For business and data logic, we used the HTTP protocol to send requests to our API (see above) which responds with the necessary data we need to run our application based on our design diagrams.

Phase 3 Task 3a Design Class Diagram

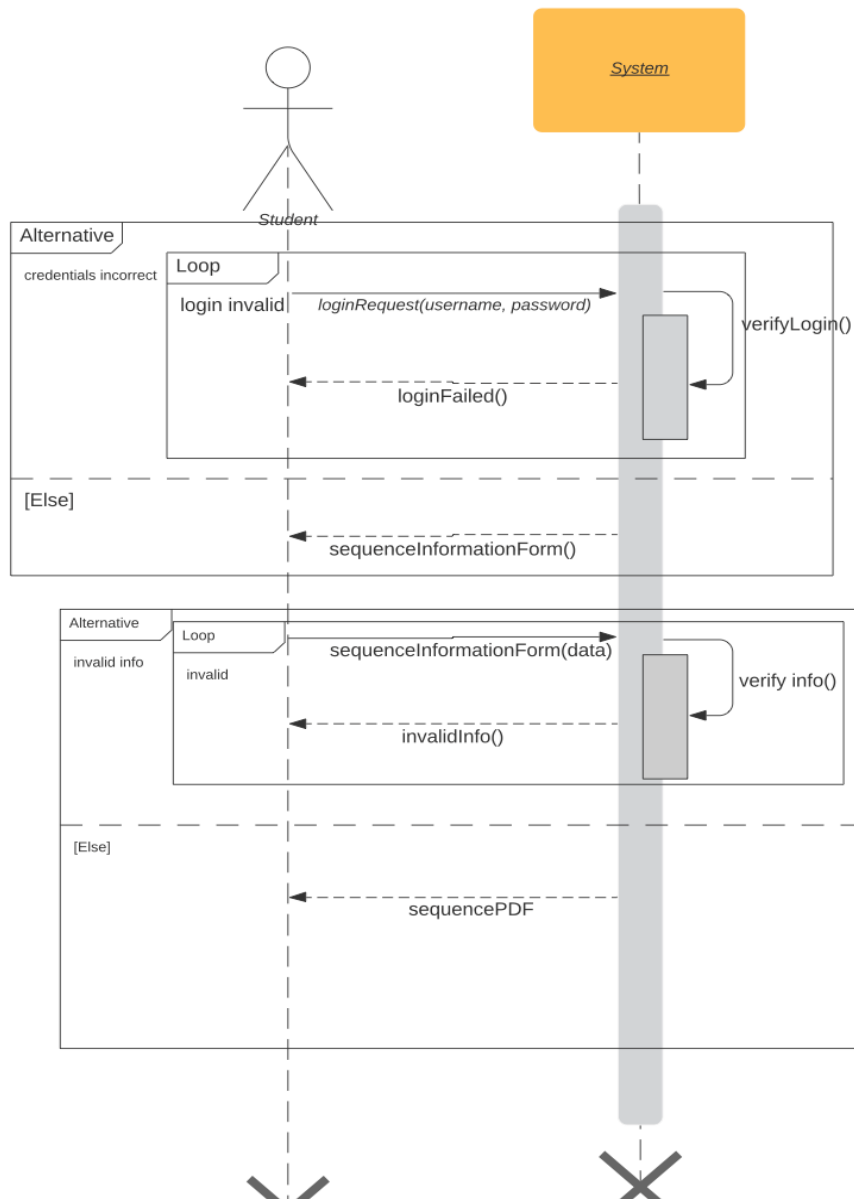
Member: Alexander Enache



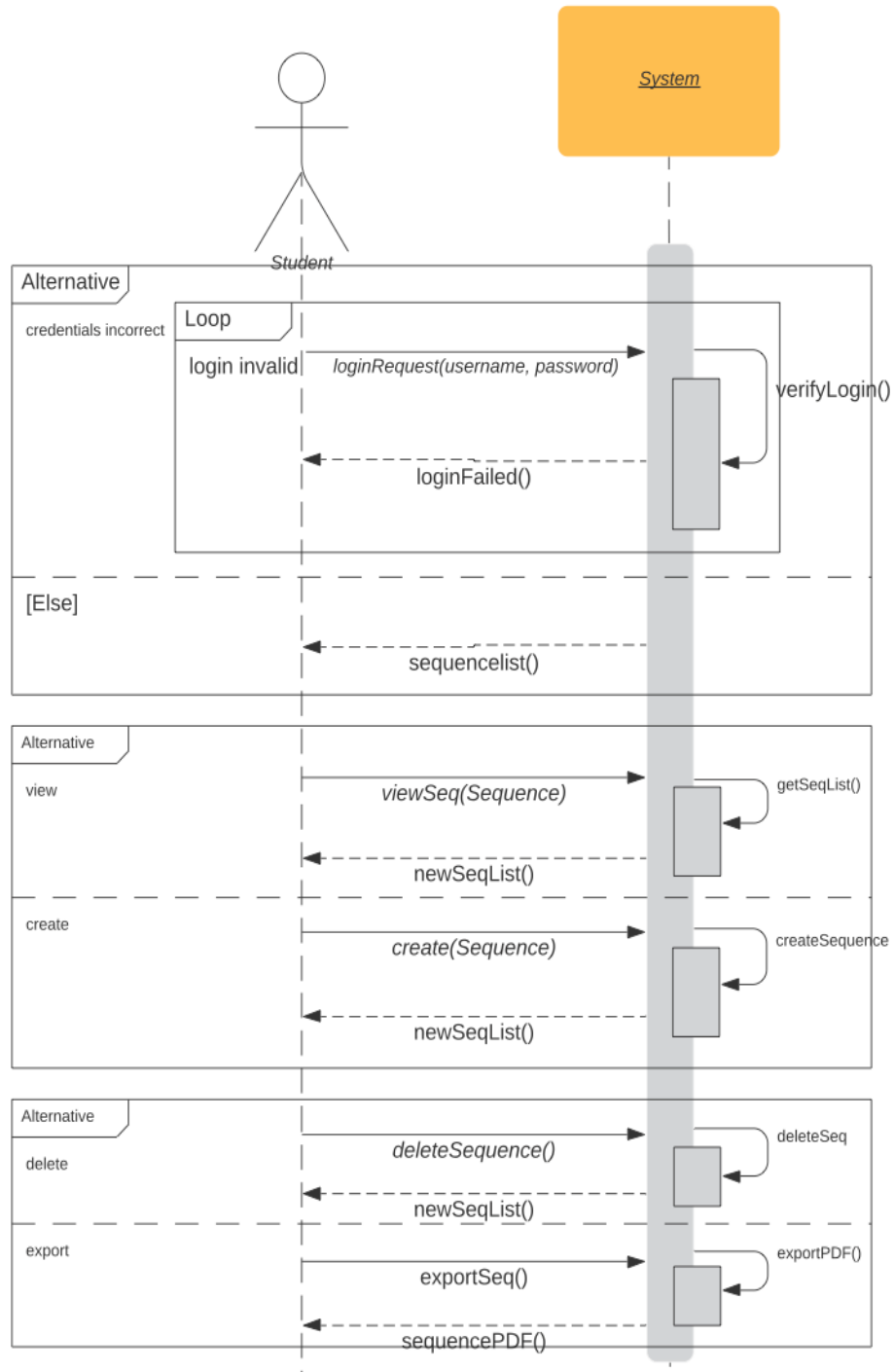
Phase 3 Task 3b Design Sequence Diagram

Member: Tara Das

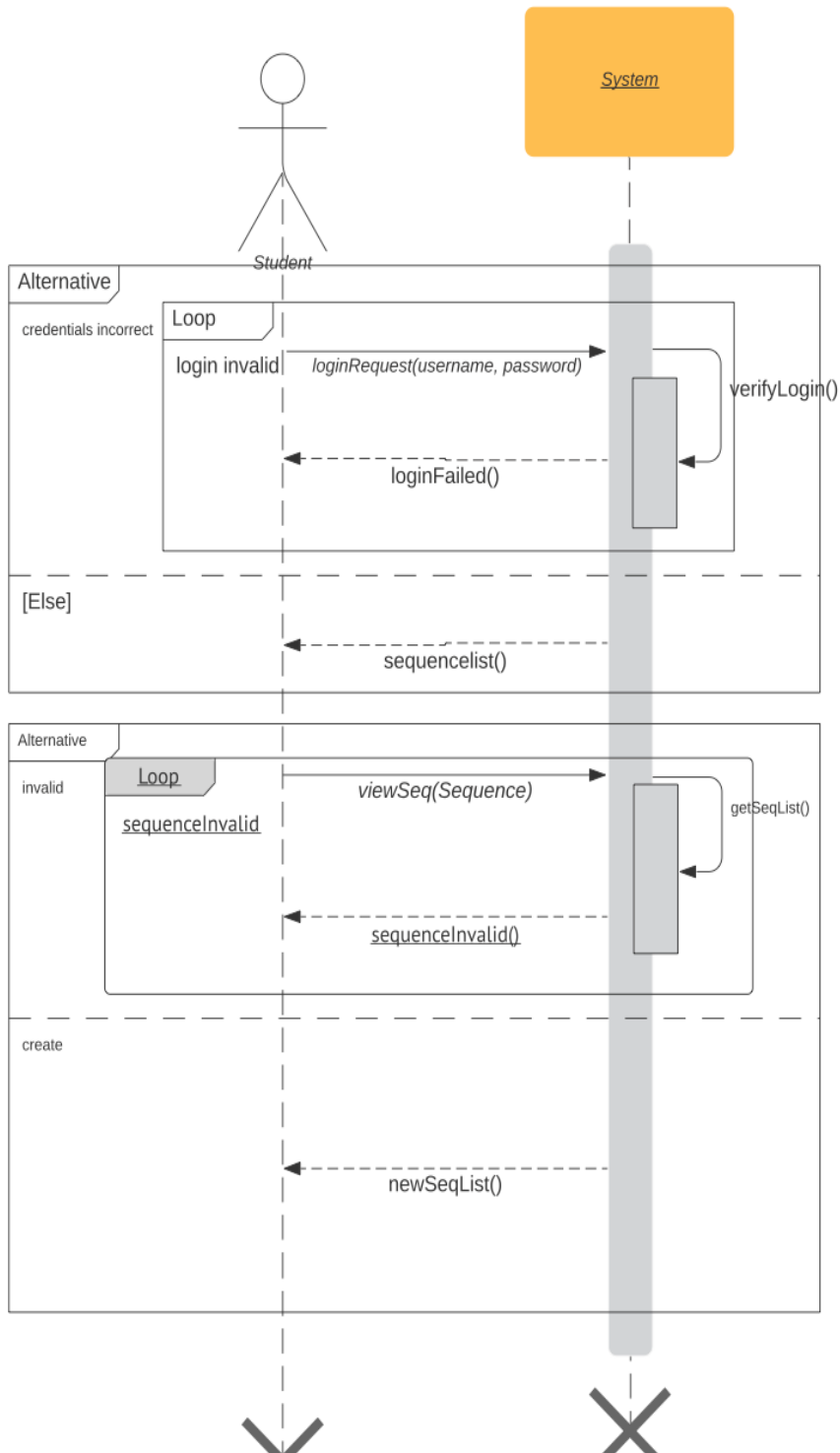
UC-1: Create Sequence



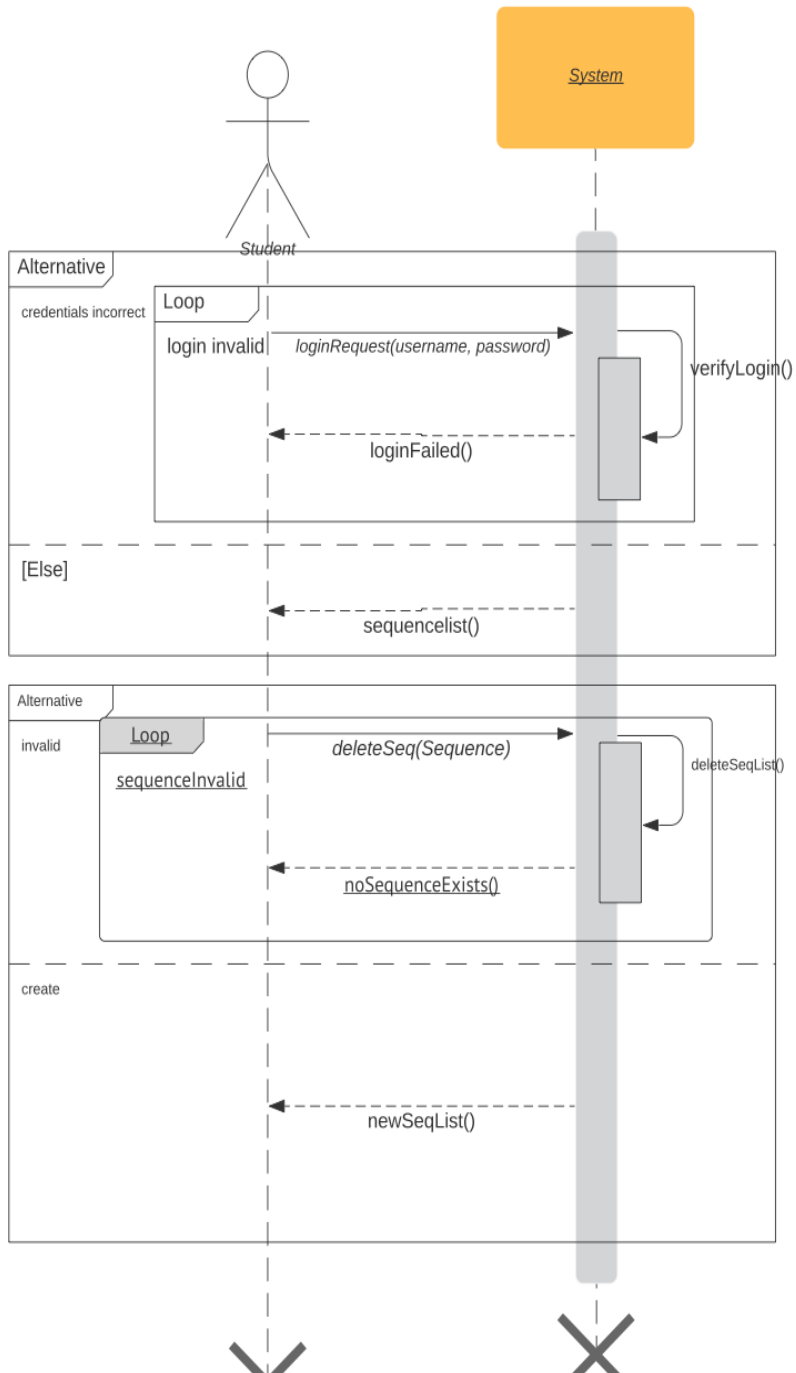
UC-2: Manage Sequence



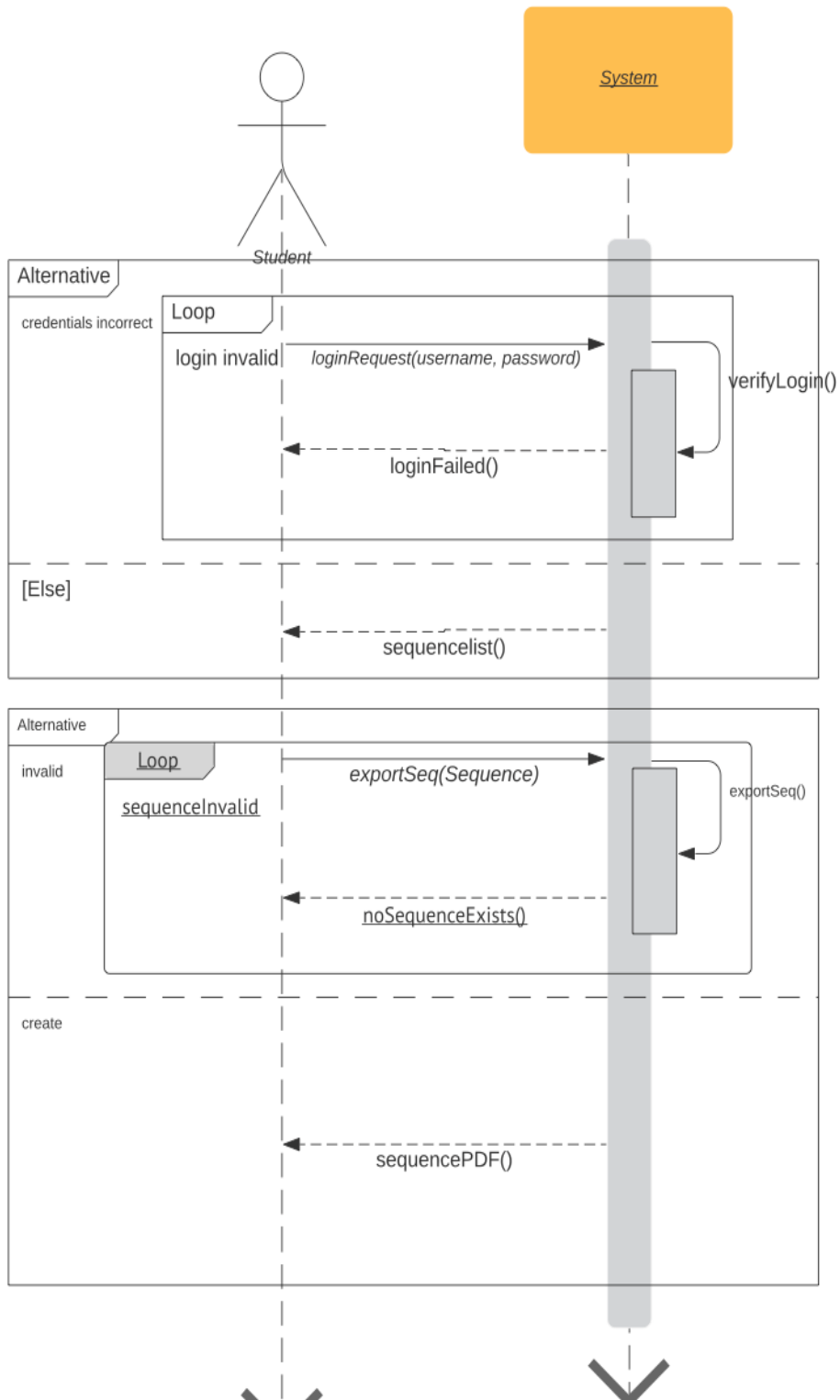
UC-3: View Sequence



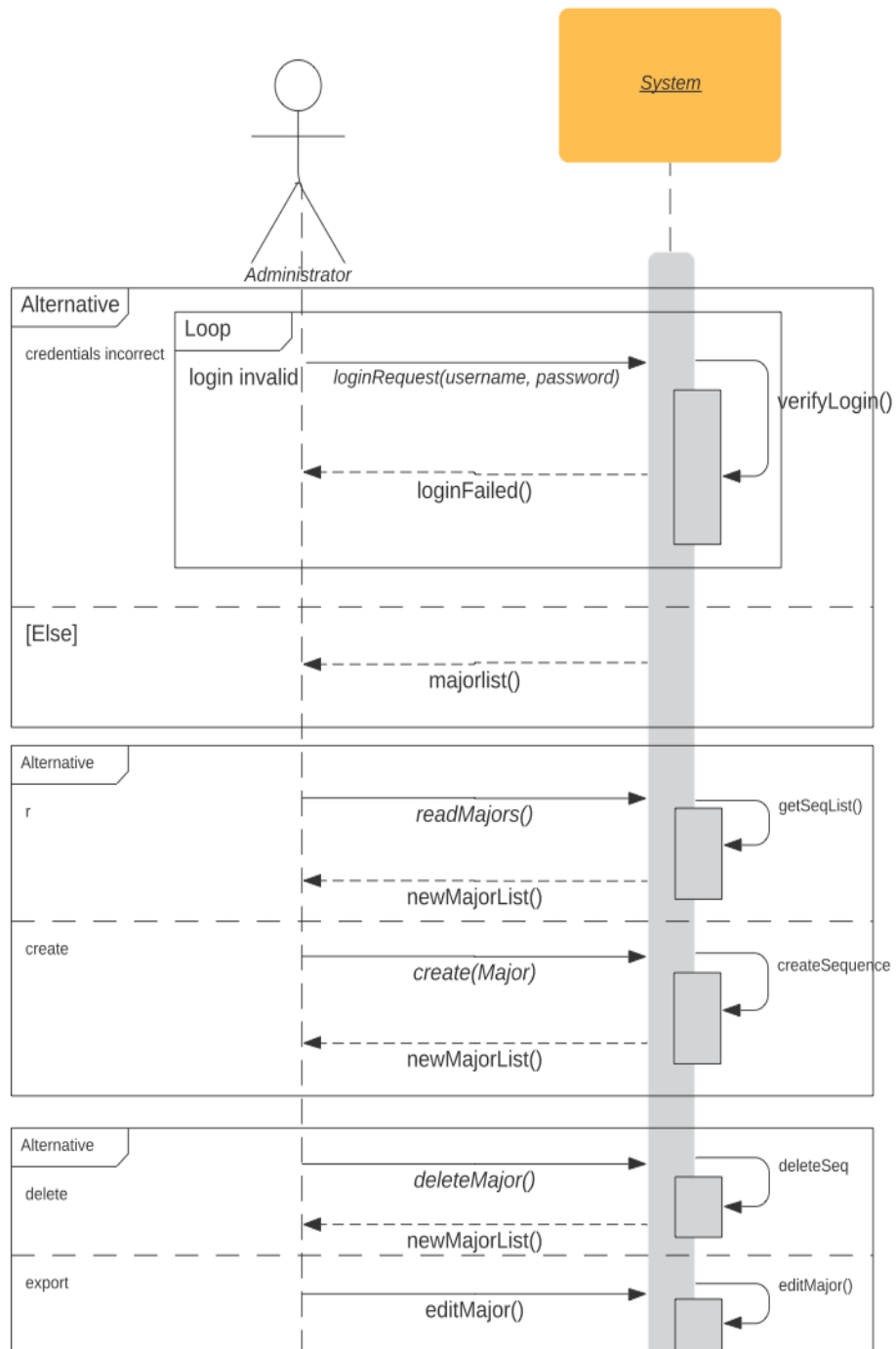
UC-4: Delete Sequence



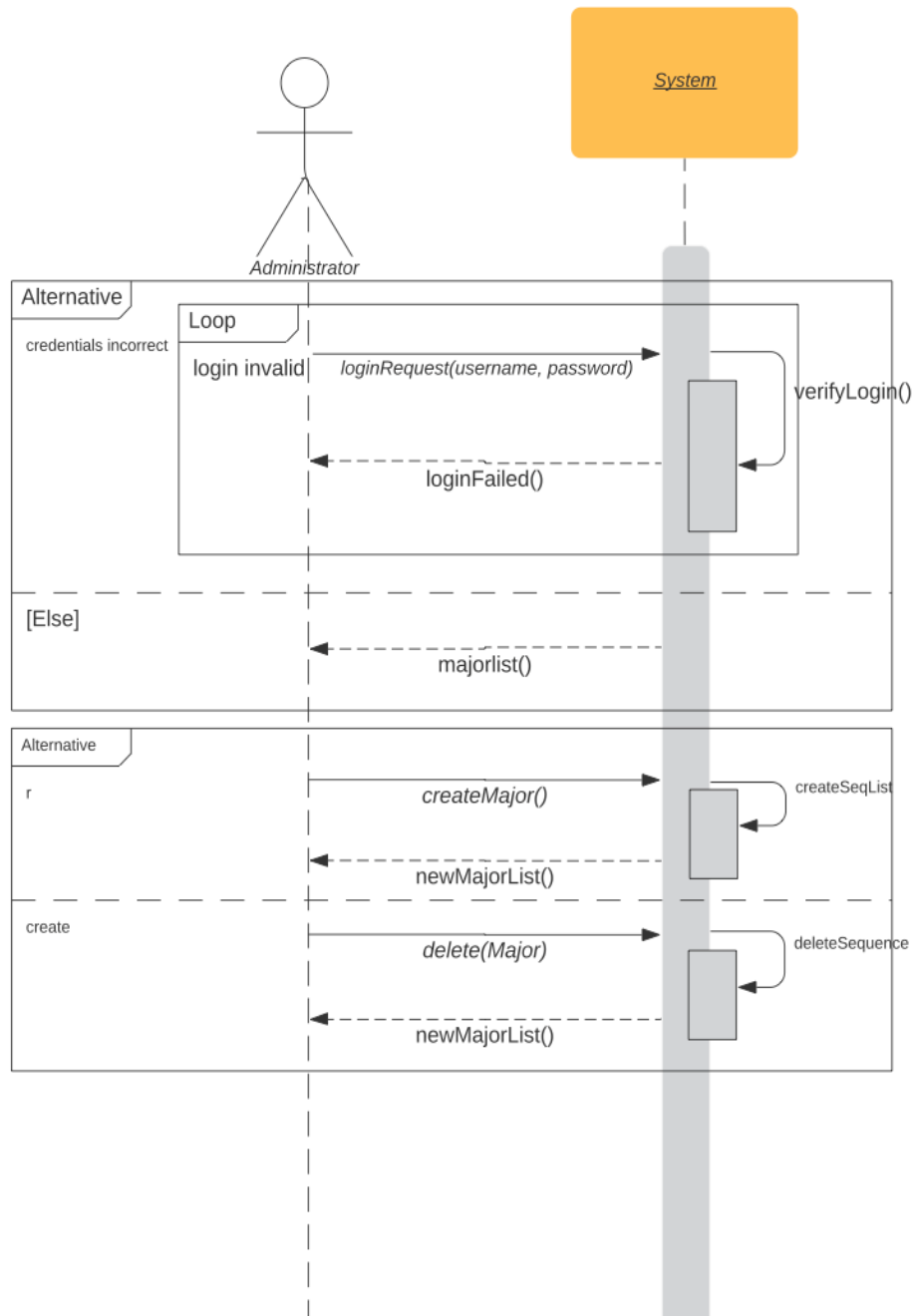
UC-5: Export Sequence



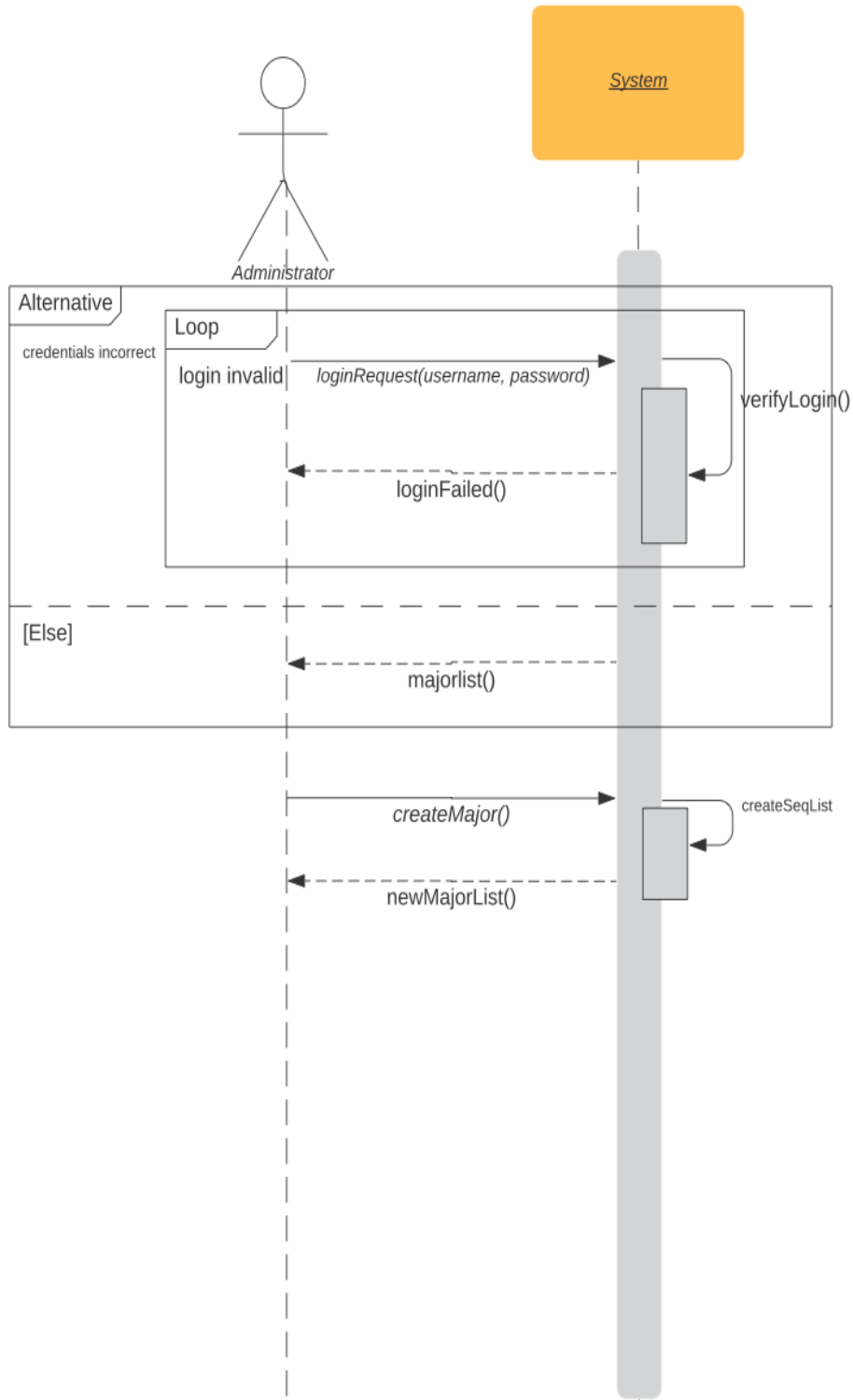
UC-6: Manage Majors



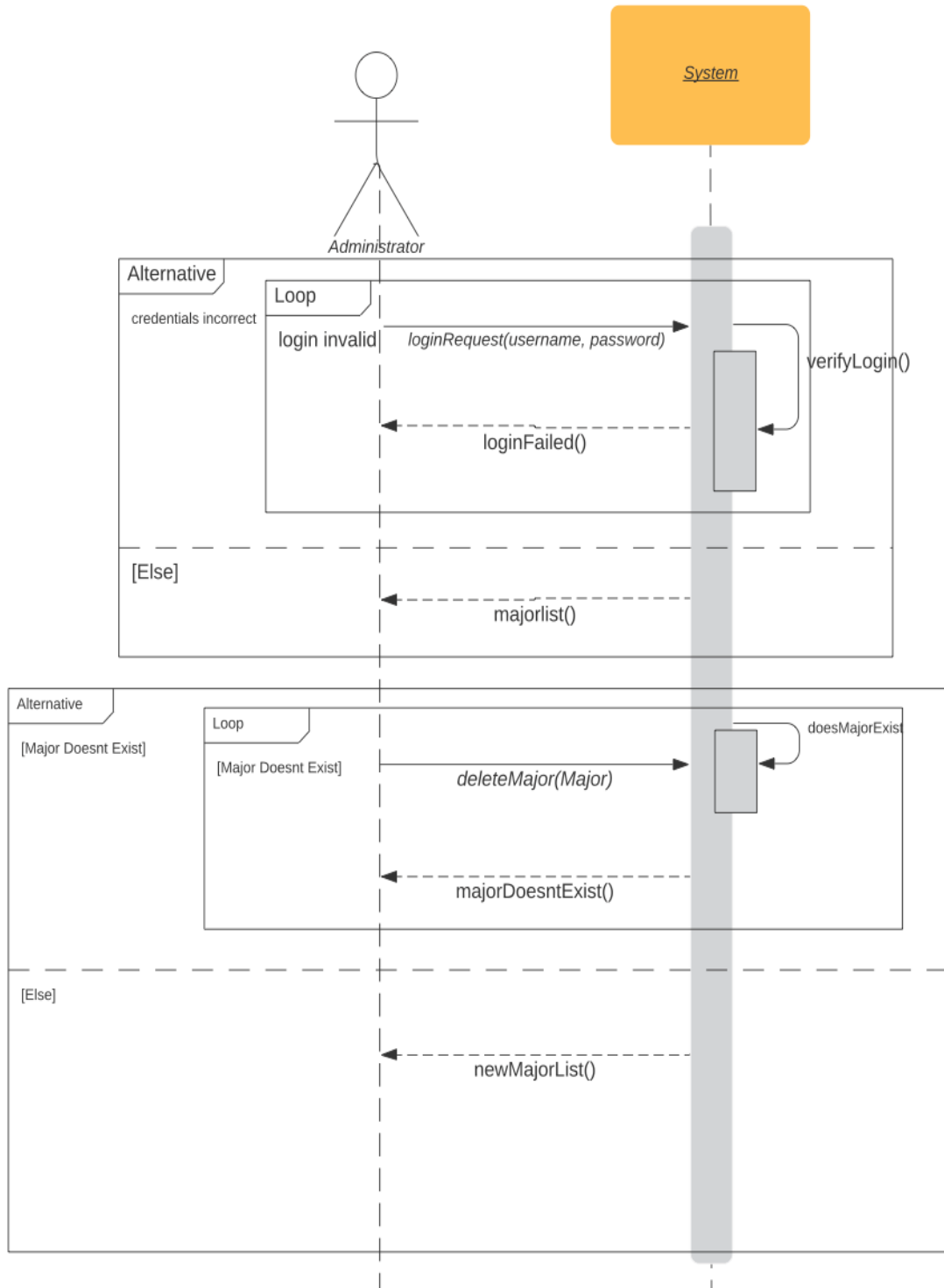
UC-6: Update Majors



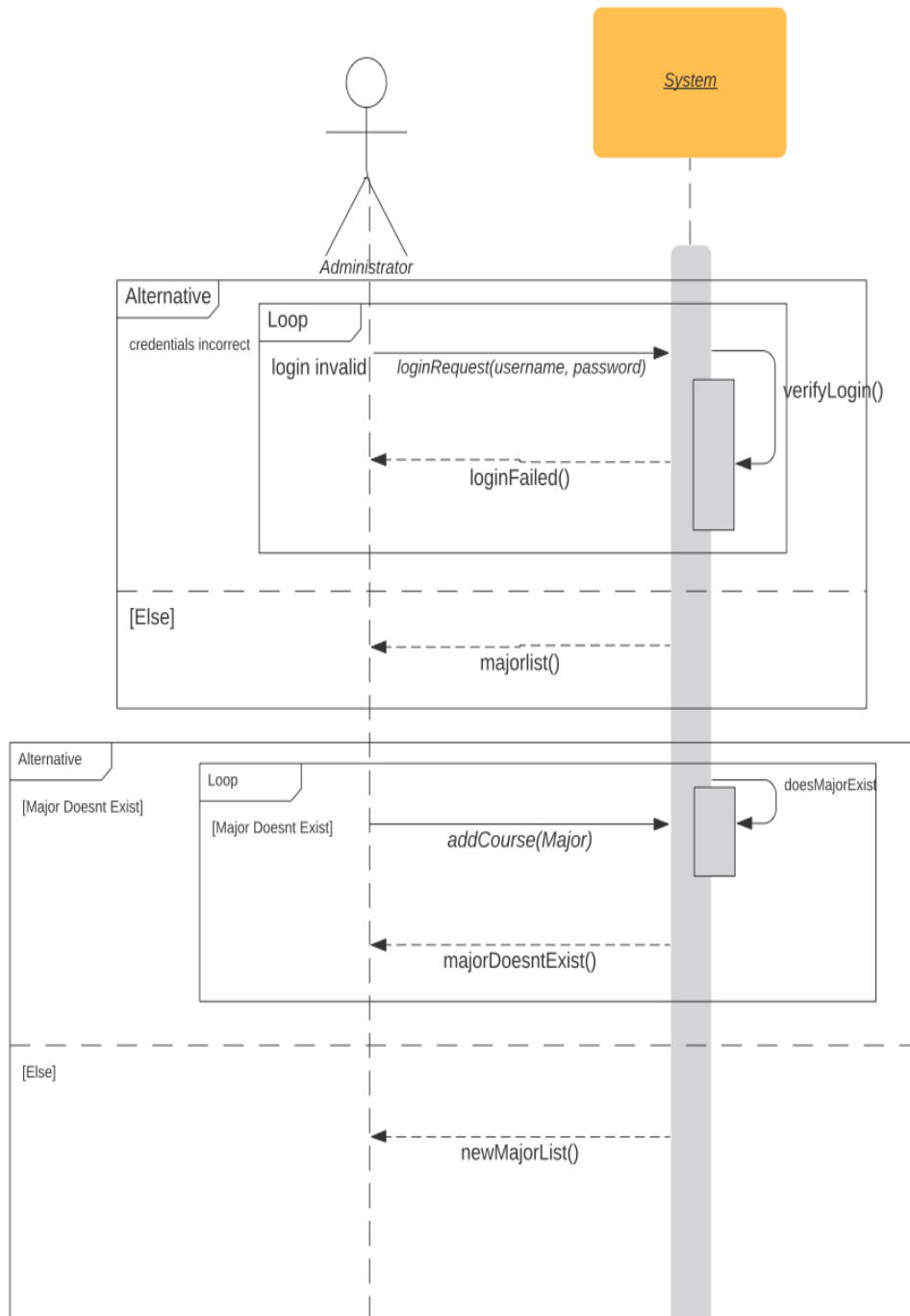
UC-8: Create Majors



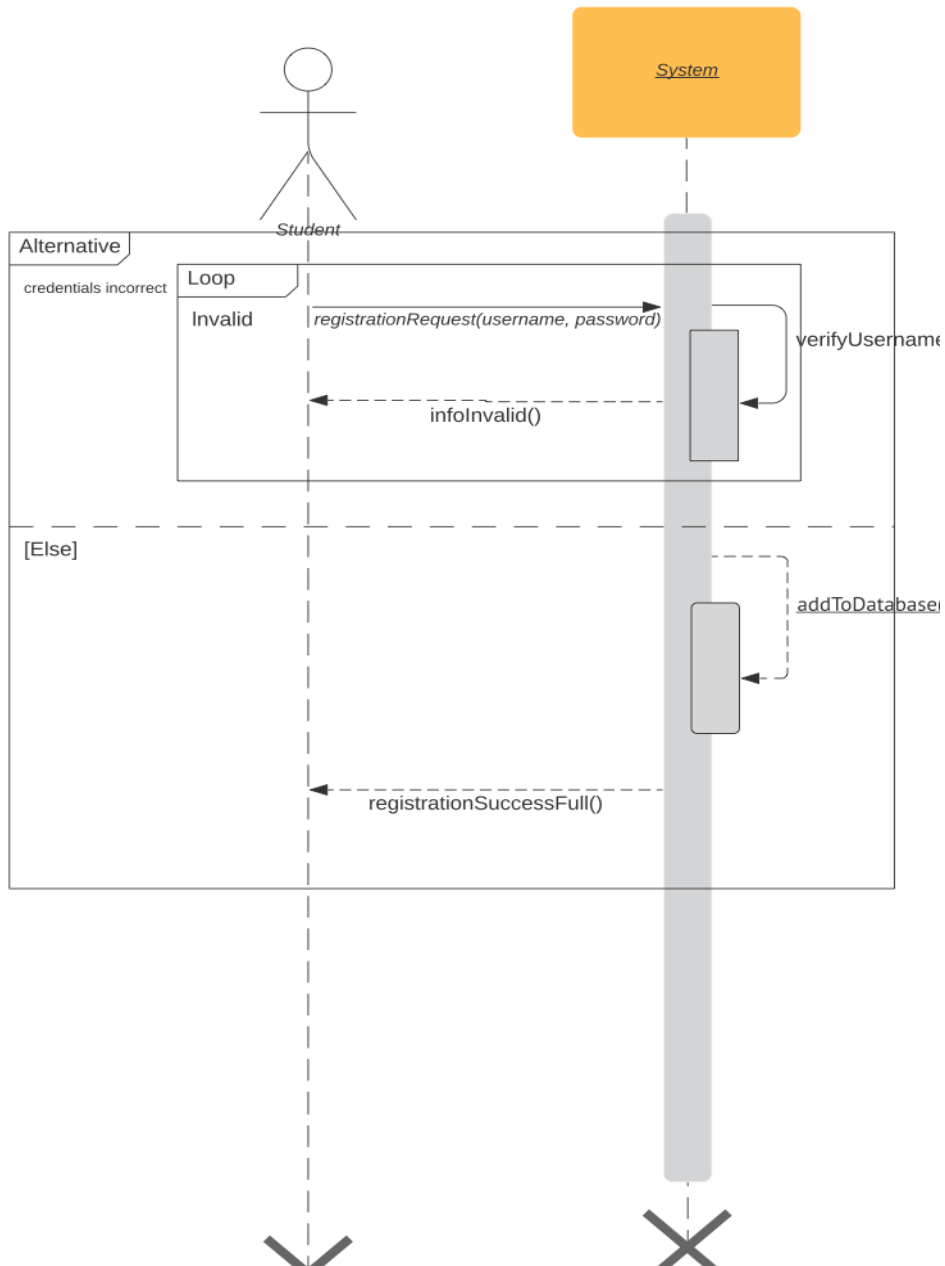
UC-9: Delete Majors



UC-10: Add Course

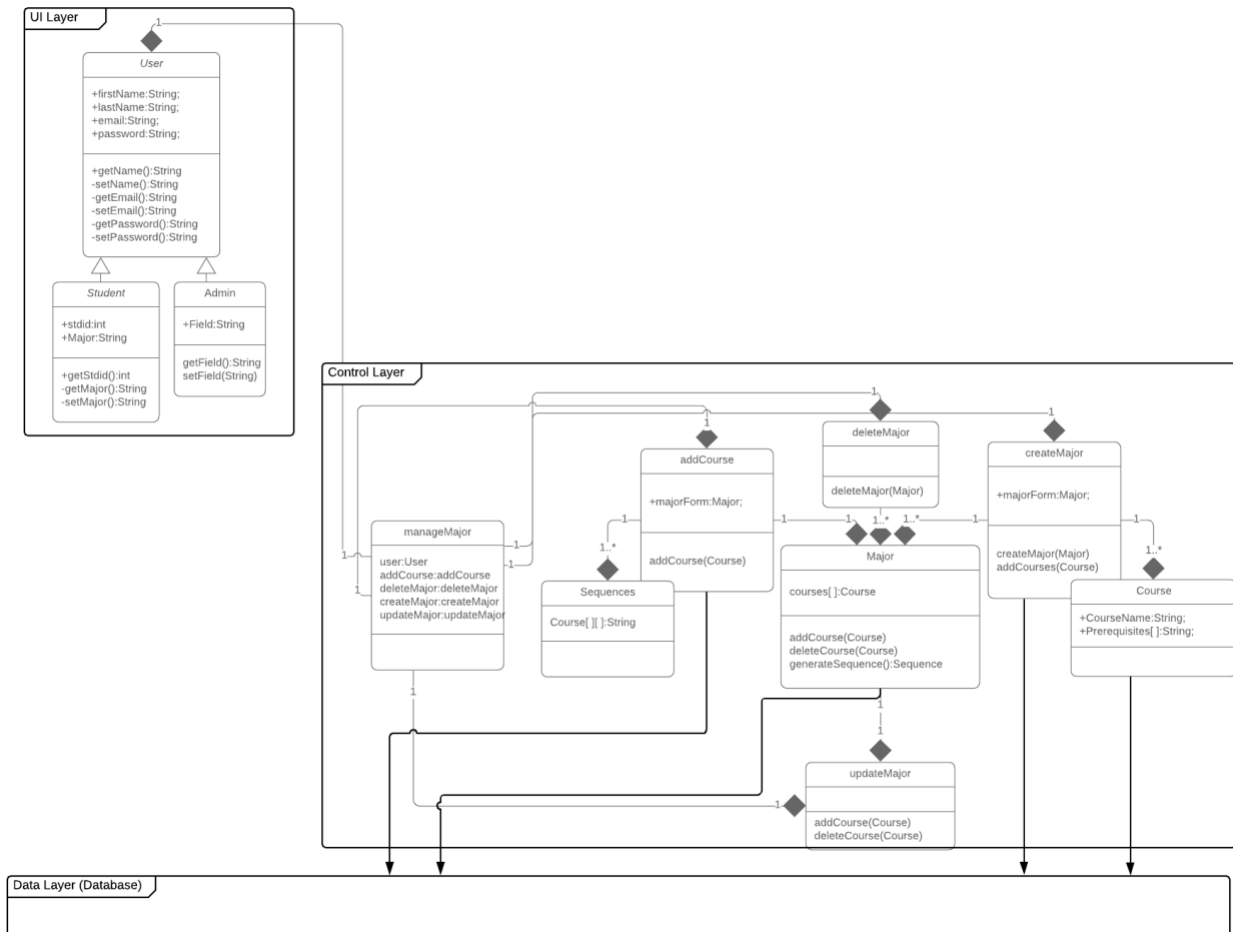


UC-11: Account Registration



Phase 3 Task 3c Three Layer Design Class Diagram

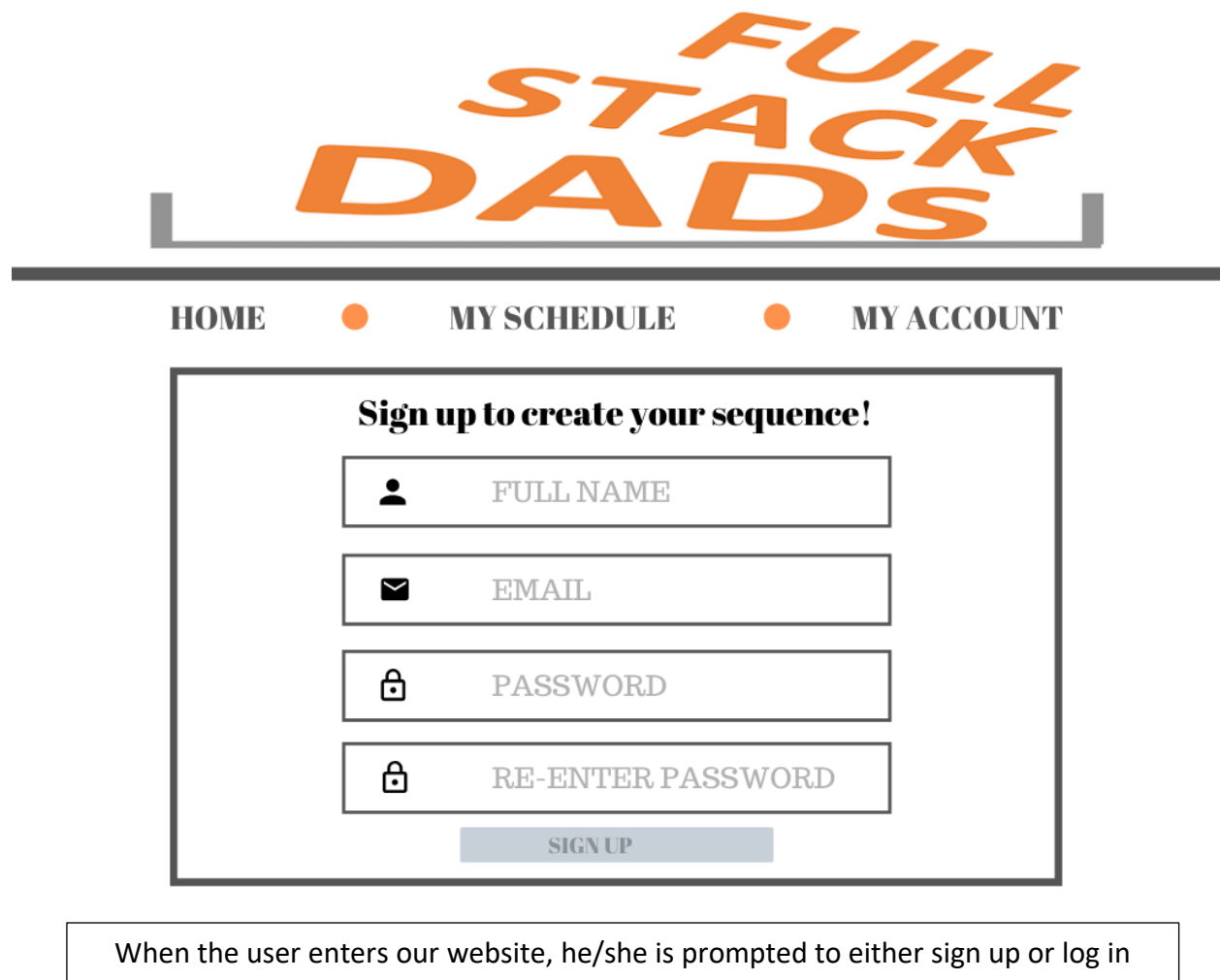
Member: Alexander Enache



Phase 3 Task 3d Interface Design

Member: Keerthana Madhavan

Landing Page





The landing page features a large, stylized orange logo for 'FULL STACK DADS' at the top. Below the logo is a horizontal navigation bar with three items: 'HOME', 'MY SCHEDULE', and 'MY ACCOUNT', each preceded by an orange circle. The main content area is a white box with a dark border containing a sign-up form. The form has the heading 'Sign up to create your sequence!' and four input fields: 'FULL NAME' (with a person icon), 'EMAIL' (with an envelope icon), 'PASSWORD' (with a lock icon), and 'RE-ENTER PASSWORD' (with a lock icon). A 'SIGN UP' button is at the bottom of the form. A text box at the bottom of the page states: 'When the user enters our website, he/she is prompted to either sign up or log in'.


FULL STACK DADS


HOME ● MY SCHEDULE ● MY ACCOUNT

Sign up to create your sequence!

 FULL NAME

 EMAIL

 PASSWORD

 RE-ENTER PASSWORD

SIGN UP

When the user enters our website, he/she is prompted to either sign up or log in

Home Page

FULL STACK DADS

HOME

MY SEQUENCE

MY ACCOUNT

Answer the following question to generate your sequence!

University Start Year. :	<div>2018</div> <div>2019</div> <div>2020</div>
Department :	<div>Computer Science</div> <div>Business</div> <div>Economics</div>
Major :	<div>Hon Comp Sci with Soft</div> <div>General Comp Sci</div> <div>Comp Sci AI</div>
Epected Graduate Date :	<div>2022</div> <div>2023</div> <div>2024</div>
	<div>SUBMIT</div>

After logging in the user is redirected to the home page where he/she will enter the necessary information to generate a course sequence for their major.

My Sequence Page

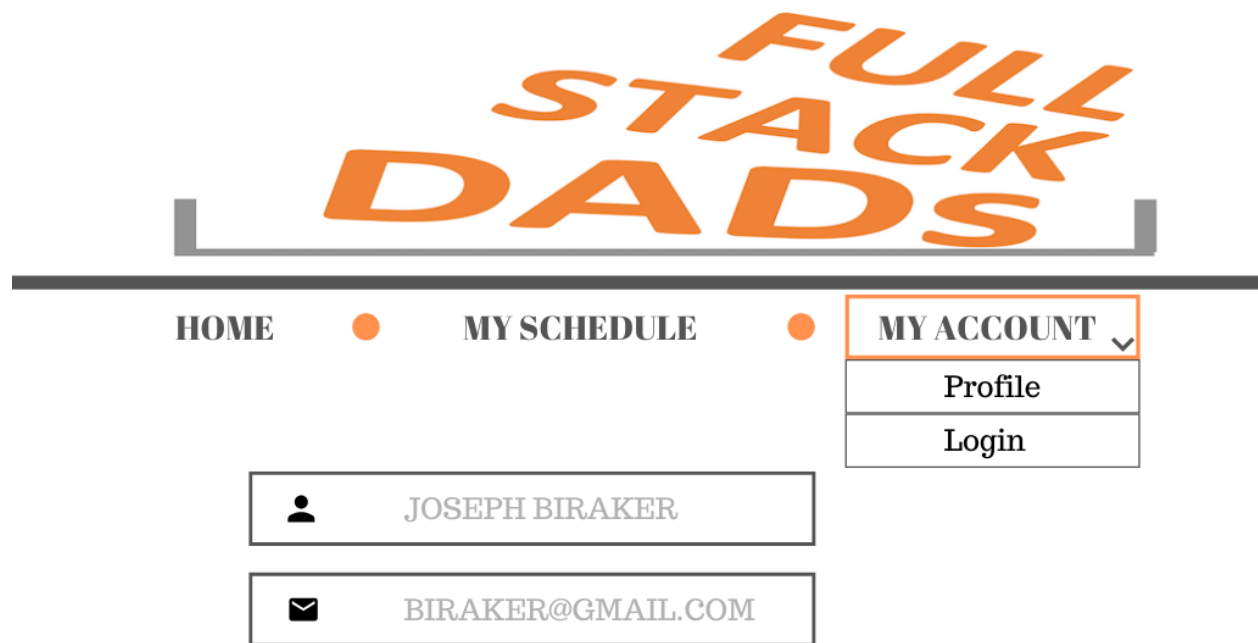
FULL STACK DADS

[HOME](#)[MY SEQUENCE](#)[MY ACCOUNT](#)

Year 1:	<table><tr><th>FALL</th><th>Winter</th><th>Summer</th></tr><tr><td>COMP_1000 COMP_1400 MATH_1200 MATH_1250 MATH_1250</td><td>COMP_1410 MATH_1730 MATH_1020</td><td></td></tr></table>	FALL	Winter	Summer	COMP_1000 COMP_1400 MATH_1200 MATH_1250 MATH_1250	COMP_1410 MATH_1730 MATH_1020		<div>CREATE SEQUENCE</div> <div>MANAGE SEQUENCE</div>
FALL	Winter	Summer						
COMP_1000 COMP_1400 MATH_1200 MATH_1250 MATH_1250	COMP_1410 MATH_1730 MATH_1020							
Year 2:	<table><tr><th>FALL</th><th>Winter</th><th>Summer</th></tr><tr><td>COMP_2120 COMP_2540 COMP_2560</td><td>COMP_2140 COMP_2800</td><td>CO-OP</td></tr></table>	FALL	Winter	Summer	COMP_2120 COMP_2540 COMP_2560	COMP_2140 COMP_2800	CO-OP	<div>VIEW SEQUENCE</div> <div>DELETE SEQUENCE</div>
FALL	Winter	Summer						
COMP_2120 COMP_2540 COMP_2560	COMP_2140 COMP_2800	CO-OP						
Year 3:	<table><tr><th>FALL</th><th>Winter</th><th>Summer</th></tr><tr><td>COMP_3110 COMP_3150 COMP_3220</td><td>COMP_3300 COMP_3400</td><td></td></tr></table>	FALL	Winter	Summer	COMP_3110 COMP_3150 COMP_3220	COMP_3300 COMP_3400		<div>EXPORT SEQUENCE</div>
FALL	Winter	Summer						
COMP_3110 COMP_3150 COMP_3220	COMP_3300 COMP_3400							
Year 4::								

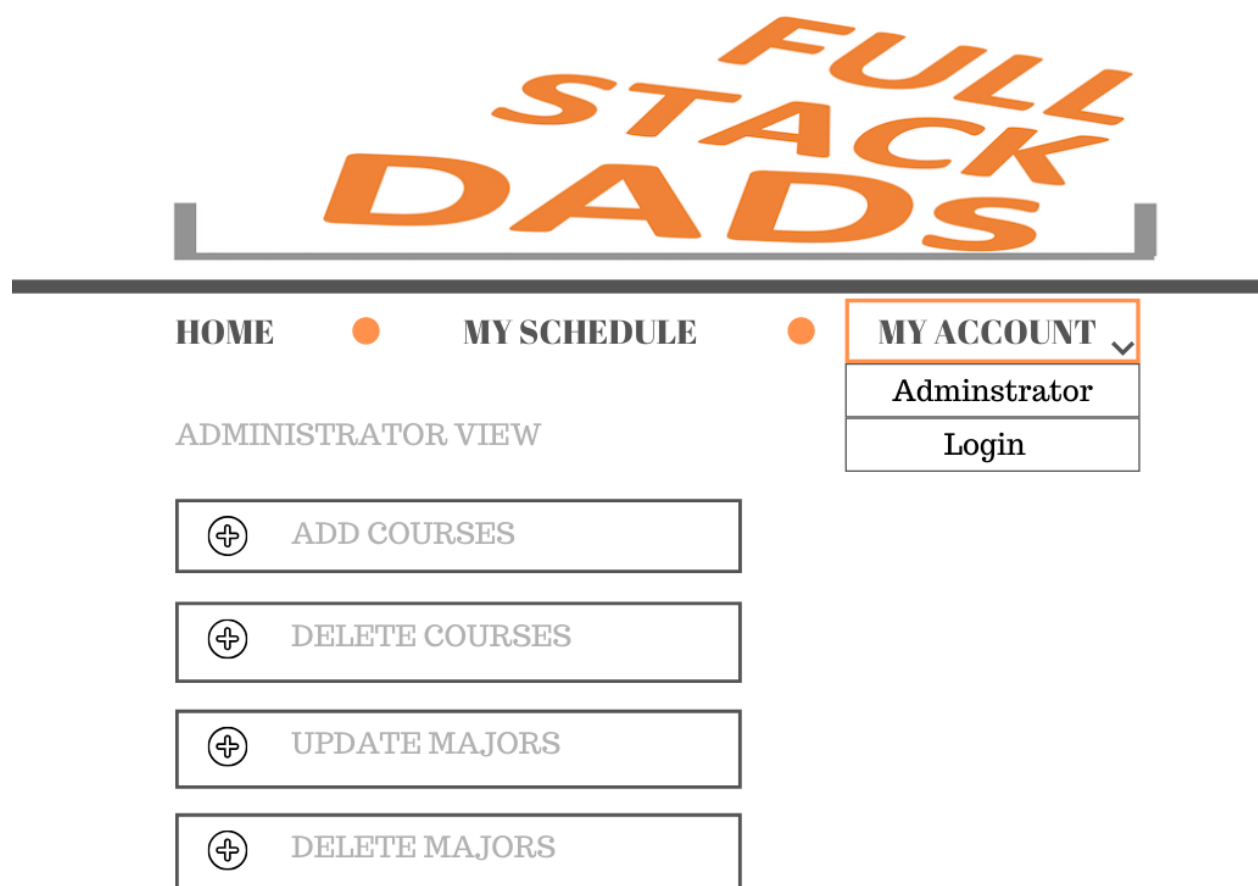
After submitting a request to generate a sequence, the user is redirected to my sequence webpage where it will display a table of sequences. The user will also have various functionality on the right side of the web page such as create, manage, view, delete, and export sequence.

My Account Page



In this webpage, the user can manage his or her profile like access such information or modify their profile. In this section, he or she can also login and logoff.

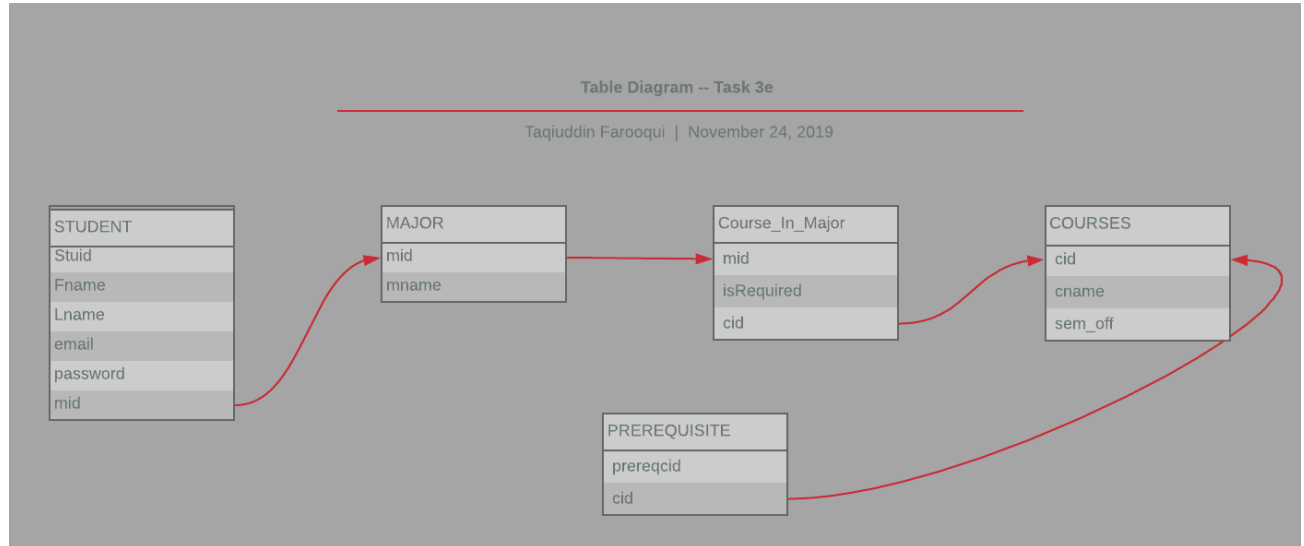
Administrator Page



In this webpage, the administrator can login to his or her account using the landing page. Then he or she can use the following functionalities such as add, delete, update and delete courses.

Phase 3 Task 3e Table Diagram

Member: Taqi Farooqui and Muhammad Faraz Sohail



Phase 3 Task 4 Demonstration Appointment Booking

Our appointment has been booked for the project demonstration on Wednesday November 27th 2019 at 5:00 pm