

SOEN 6481 Summer 2021

# **SRS E-CONCORDIA DRIVE**

Tavtej Singh Lehri  
StudentID - 40121745

August 17,2021

# Contents

<b>1</b>	<b>Use Case Model</b>	<b>3</b>
1.1	Introduction . . . . .	3
1.2	Actor-Goal List . . . . .	3
1.3	Use Case Model . . . . .	4
1.3.1	Use Case ID: UC001 . . . . .	4
1.3.2	Use Case ID: UC002 . . . . .	5
1.3.3	Use Case ID: UC003 . . . . .	6
1.3.4	Use Case ID: UC004 . . . . .	7
1.3.5	Use Case ID: UC005 . . . . .	8
1.3.6	Use Case ID: UC006 . . . . .	9
1.3.7	Use Case ID: UC007 . . . . .	10
1.3.8	Use Case ID: UC008 . . . . .	11
1.3.9	Use Case ID: UC009 . . . . .	12
1.3.10	Use Case ID: UC010 . . . . .	13
1.3.11	Use Case ID: UC011 . . . . .	14
1.3.12	Use Case ID: UC012 . . . . .	15
1.3.13	Use Case ID: UC013 . . . . .	17
1.3.14	Use Case ID: UC014 . . . . .	17
1.3.15	Use Case ID: UC015 . . . . .	18
1.4	UML Diagrams . . . . .	20
1.4.1	UML Use Case Diagram . . . . .	20
1.4.2	UML Sequence Diagram . . . . .	20
1.4.3	UML Activity Diagram . . . . .	21
1.5	Appendix . . . . .	22
<b>2</b>	<b>Supplementary Specification</b>	<b>23</b>
2.1	Introduction . . . . .	23
2.1.1	Purpose . . . . .	23
2.1.2	Scope . . . . .	23
2.1.3	Definitions, Acronyms, Abbreviations . . . . .	23
2.1.4	References . . . . .	24
2.2	Functionality . . . . .	24
2.2.1	Error Logs . . . . .	24
2.2.2	Remote Access . . . . .	25

2.3	Usability . . . . .	25
2.3.1	Training Time . . . . .	25
2.3.2	Industry Standards . . . . .	25
2.3.3	Chat Bots . . . . .	25
2.3.4	Multiple Systems/Web-Browser Support . . . . .	25
2.4	Reliability . . . . .	25
2.4.1	Availability, MTBF, MTTR, MDR . . . . .	25
2.5	Performance . . . . .	26
2.5.1	Response Time . . . . .	26
2.5.2	Throughput . . . . .	26
2.5.3	Capacity . . . . .	26
2.5.4	Database/Server response time . . . . .	26
2.5.5	Resource Utilization: . . . . .	26
2.6	Supportability . . . . .	26
2.7	Design Constraints . . . . .	26
2.7.1	Languages Supported: . . . . .	27
2.7.2	Development tools: . . . . .	27
2.7.3	Management tools: . . . . .	27
2.7.4	Architectural Constraints: . . . . .	27
2.8	Online User Documentation and Help System Requirement . . . . .	27
2.9	Glossary . . . . .	27
2.9.1	Definitions . . . . .	27
2.9.2	Acronyms and Abbreviations . . . . .	28
2.10	Appendix . . . . .	29

# Chapter 1

## Use Case Model

### 1.1 Introduction

#### Purpose

The purpose of this document is the Use-Case Model comprehensively. It describes the structure of UCM and lists down the actors involved and their goals in the model.

#### Scope

This Use-Case Model applies to E-Concordia Drive website to be developed by the Chronos Tech Solutions. This website will help the students learn about driving to obtain their driver's permit, while staying at their home.

### 1.2 Actor-Goal List

Actor	Goal
Student	<ul style="list-style-type: none"><li>• Login</li><li>• View Lessons</li><li>• Take and Submit Quizzes</li></ul>
Trainer	<ul style="list-style-type: none"><li>• Login</li><li>• Submit/Update the Lesson Content</li><li>• Submit/Update the quizzes</li></ul>

Admin	<ul style="list-style-type: none"> <li>• Login</li> <li>• Approve the content posted by trainer</li> <li>• Send comments to trainers with required changes</li> <li>• Add/Remove a user which includes students and trainers.</li> <li>• Manage content.</li> </ul>
-------	---

Table 1.1: Actor Goal List.

## 1.3 Use Case Model

### 1.3.1 Use Case ID: UC001

**Use Case:** Student Registration

**Description:**

Students should be able to create an account online by filling out the personal details. After registration students will be provided 3 unique identifiers used to login

**Level:** User Goal

**Primary Actor:**

Students

**Supporting Actor:**

Mobile Devices

**Stakeholders and Interests:**

Students

**Pre-Conditions:**

Students should have access to any mobile device and internet connection

**Post-Conditions:**

Will be determined later

**Main Success Scenario:**

1. User visits the website and clicks on the sign-up button
2. User fills in the personal details: First Name, Last Name, Date of Birth and Address
3. User selects the license type from the drop-down list

4. User selects the payment type as credit card/debit card and clicks submit.
5. After the successful payment student is provided with unique File Number, Traffic Number and Student Number the combination of which will be used to login into the account.
6. All the 3 unique numbers will be stored in EDC database along with student information for account authentication

**Extensions:**

If the payment by card fails, users should be given option to either make payment by cheque, cash or Interac

**Special Requirements:**

The payments using Credit/Debit needs to be secured and the system should be secure enough to prevent data leakage.

### **1.3.2 Use Case ID: UC002**

**Use Case:** Trainer Registration

**Description:**

Trainers should be able to create an account online by filling out the personal details.

**Level:** User Goal

**Primary Actor:**

Trainer

**Supporting Actor:**

Mobile Devices

**Stakeholders and Interests:**

Trainers

**Pre-Conditions:**

Students should have access to any mobile device and internet connection

**Post-Conditions:**

Will be determined later

**Main Success Scenario:**

1. User visits the website and clicks on the sign-up button
2. User fills in the personal details: First Name, Last Name, Date of Birth, Teacher ID, E-mail ID and Address.
3. User chooses the username and password of their choosing.
4. User clicks on submit and the profile goes to admin for approval.

5. After the admin approves/rejects an email notification is sent to the trainer.
6. The username and password(hotkey format) will be stored in EDC database for account authentication

**Extensions:**

1. If the username already exist then trainer has to choose a new username.
2. Password should be of format: 1 uppercase, 1 lowercase, 1 special character and 1 numeric character.

**Special Requirements:**

None

**1.3.3 Use Case ID: UC003**

**Use Case:** Read content on student dashboard

**Description:**

After the student is logged in this use case will occur. It deals with what is displayed on the dashboard page

**Level:** User Goal

**Primary Actor:**

Students

**Supporting Actor:**

Mobile Devices and EDC Database

**Stakeholders and Interests:**

Stakeholders will be Students Stakeholder interest will be to let students see their details on dashboard along with lesson icons

**Pre-Conditions:**

Students should have registered.

**Post-Conditions:**

None

**Main Success Scenario:**

1. User visits the website and Login using the 3 unique numbers provided at time of registration
2. Website displays license type, student number, traffic number , student name, lesson icons from the trainer backend, student progress, lesson names, enrollment and expiry date etc.
3. Students can click on the lesson icons to start it.
4. Students can click on start again button once to lesson is complete

**Extensions:**

None

**Special Requirements:**

None

**1.3.4 Use Case ID: UC004**

**Use Case:** View videos by students

**Description:**

This use case deals with students viewing the videos posted by the trainers.

**Level:** User Goal

**Primary Actor:**

Students

**Supporting Actor:**

Mobile Devices and EDC Database

**Stakeholders and Interests:**

Students will be stakeholders and if the videos are not available they might not be able to learn something.

**Pre-Conditions:**

Students should have registered.

**Post-Conditions:**

None

**Main Success Scenario:**

1. User visits the website and Login using the 3 unique numbers provided at time of registration
2. Website dashboard page displays the video icons taken from the trainer back-end.
3. Students clicks on the lesson icons and the website navigates them to the lesson page.
4. lesson name will be picked from EDC database and lesson description will have any additional information necessary related to the lesson.
5. Student clicks on the START NOW button to start the lesson.

**Extensions:**

Video does not start.

1. An error/broken feed message is displayed if it is because of time out.
2. Student should check their internet connection.



3. If there is no issue with internet contact someone at school to look at the system.

User is logged out of the system.

1. Student is logged out after being idle for 30 minutes.
2. Repeat from step 1 of main scenario.
3. Video will start from where student left it at.

**Special Requirements:**

1. System should be able to handle 350000 users at same time.
2. The video screen should be interactive which means the orientation should change if the device orientation is changes.

### 1.3.5 Use Case ID: UC005

**Use Case:** Students take quizzes

**Description:**

This use case deals with students attempting the quiz posted by the trainers.

**Level:** User Goal

**Primary Actor:**

Students

**Supporting Actor:**

Mobile Devices and EDC Database

**Stakeholders and Interests:**

Owner and Trainer. They want quiz so that students can practice what they have learned

**Pre-Conditions:**

Students should have registered.

**Post-Conditions:**

Success end condition

All the questions are answered.

Failure end condition

None.

**Main Success Scenario:**

1. User visits the website and Login using the 3 unique numbers provided at time of registration
2. Website dashboard page displays the video icons taken from the trainer back-end.
3. Students clicks on the lesson icons and the website navigates them to the lesson page.

4. Student should complete all the slides before moving to the quiz section.
5. Students should select the correct answers to the questions.
6. Click FINISH after successfully answering all the questions to complete the lesson

**Extensions:**

User is logged out of the system.

1. Student is logged out after being idle for 30 minutes.
2. Repeat from step 1 of main scenario.
3. Video will start from where student left it at.

**Special Requirements:**

1. System should be able to handle 350000 users at same time.
2. The video screen should be interactive which means the orientation should change if the device orientation is changes.
3. Interactive screen also means that user can select the answers in a touchscreen device as well.
4. Quiz answers should be of type true/false, select correct option, drag and match, rearrange in correct order.

### 1.3.6 Use Case ID: UC006

**Use Case:** Create a lesson

**Description:**

This use case deals with trainers creating a lesson for students.

**Level:** User Goal

**Primary Actor:**

Trainer and Admin

**Supporting Actor:**

Mobile Devices and EDC Database

**Stakeholders and Interests:**

Trainer and Driving school owner. School owner has an interest that the content uploaded by the trainer should be of a perfect quality and not misleading.

**Pre-Conditions:**

Trainer should be approved by the admin.

**Post-Conditions:**Success end condition

Content is approved by admin is successfully published.

#### Failure end condition

Content is not approved by admin and sent back for changes.

#### **Main Success Scenario:**

1. Trainer logs in using the unique username and password which will be authenticated using the EDC database.
2. Trainer is navigated to the dashboard page and a menu bar is displayed with link to different pages.
3. Trainer clicks on ALL LESSONS link and quick navigation pane is opened which has options like, All Lessons, Drafts, Pending and Notification.
4. From the quick navigation click on ALL LESSONS link and user is take to lessons management page.
5. Click on manage lessons and a popover window opens.
6. Select the lesson name and language from list drop down taken from EDC database servers. Write lesson description to be shown in student UI. Upload lesson icon by choosing a file.
7. Click submit and the lesson will be ready for content insertion

#### **Extensions:**

User is logged out of the system.

1. Trainer is logged out of system with unsaved work.
2. Start over from step 1.

#### **Special Requirements:**

1.  $\oplus$  sign should be used to expand the lesson.
2. system should allow to save different versions of a lesson.
3. lesson description should be displayed in student UI.

### **1.3.7 Use Case ID: UC007**

**Use Case:** Insert Content to lessons

#### **Description:**

This use case deals with trainers inserting slides and quizzes to the lessons.

**Level:** User Goal

#### **Primary Actor:**

Trainer and Admin

#### **Supporting Actor:**

Mobile Devices and EDC Database

**Stakeholders and Interests:**

Trainer and Driving school owner. School owner has an interest that the content uploaded by the trainer should be of a perfect quality and not misleading.

**Pre-Conditions:**

Trainer should be logged in to a system and a lesson must have been created.

**Post-Conditions:**Success end condition

Content is approved by admin is successfully published.

Failure end condition

Content is not approved by admin and sent back for changes.

**Main Success Scenario:**

1. Trainer click on +Slide button given on the created lesson.A popover window opens.
2. Trainer add slide number, slide title, type of media(video/image/audio), slide description and slide duration.
3. Trainer chooses the media file and clicks submit.
4. Trainer click on +Quiz button given on the created lesson.A popover window opens.
5. Trainer selects lesson name, quiz type and enters quiz answer.
6. Provide the correct answer and click on save button

**Extensions:**

Slide title or question already exists

1. Systems pops an error saying duplicate content
2. Start over from step 1 or Step 4.

**Special Requirements:**

1. One click on +Slide and +Quiz will create a single slide and single question. to create multiple slides and quiz questions click on those buttons again.

**1.3.8 Use Case ID: UC008**

**Use Case:** Pending Lessons

**Description:**

This use case deals with pending lessons.

**Level:** User Goal

**Primary Actor:**

Trainer and Admin

**Supporting Actor:**

Mobile Devices and Admin

**Stakeholders and Interests:**

Trainers and Admins. They should know the status of lessons using which they can plan the work.

**Pre-Conditions:**

Admin should provide feedback for the draft content.

**Post-Conditions:**Success end condition

Pending lessons are submitted by the trainers after the changes are made.

Failure end condition

None

**Main Success Scenario:**

1. Trainer sends draft for admin approval.
2. Trainer receives comments from admin and the status of content is changed back to pending.
3. Trainer click on on the pending button from the navigation bar.
4. All the pending lessons are displayed.
5. Trainer makes changes and save it as draft and send for admin approval.

**Extensions:**

There are no pending lessons

1. Message should be displayed that there is no pending lesson.
2. Trainer should work on another task provided by admin.

**Special Requirements:** Special Requirement will be determined once the iteration starts.

**1.3.9 Use Case ID: UC009**

**Use Case:** Draft Lessons

**Description:**

This use case deals with content that is saved as draft.

**Level:** User Goal

**Primary Actor:**

Trainer and Admin

**Supporting Actor:**

Mobile Devices

**Stakeholders and Interests:**

Trainers and Admins. They should know the status of lessons using which they can plan the work.

**Pre-Conditions:**

Trainer should be logged in the website.

**Post-Conditions:**

Success end condition

Drafts are saved and notification is sent to the admin.

Failure end condition

No notification is sent to admin

**Main Success Scenario:**

1. Trainer creates a lesson. By default lesson is added as draft
2. Click on Drafts link in the navigation menu
3. All the drafts are displayed on the draft page.
4. Lesson information is displayed in the right pane.
5. Trainer can make changes and save it as draft and send for admin approval.

**Extensions:**

There are no drafts.

1. Message should be displayed that there is no draft lesson.
2. Trainer should work on another task provided by admin.

**Special Requirements:** Special Requirement will be determined once the iteration starts.

### **1.3.10 Use Case ID: UC010**

**Use Case:** Edit and delete lessons

**Description:**

This use case deals with lesson management which includes editing and deletion of lesson.

**Level:** User Goal

**Primary Actor:**

Trainer and Admin

**Supporting Actor:**

Mobile Devices

**Stakeholders and Interests:**

Trainers and Admins. Trainer should be able to edit lessons by creating versions of published and should not be able to delete published content. Admin should have all rights

**Pre-Conditions:**

First draft of the lesson should have been created.

**Post-Conditions:**

Success end condition

Lessons are edited successfully.

Failure end condition

Lesson editing is not allowed. Published lesson is deleted by trainer

**Main Success Scenario:**

1. Trainer creates a lesson. By default lesson is added as draft
2. Clicks on all lessons. Under the manage lesson header all lessons are displayed.
3. Click on the edit button.
4. A popover pane is opened and user clicks on create version and make changes to it and saves.
5. Add slides and quizzes in the new version. Saved as draft by default
6. Send to admin for approval. Change status to pending.
7. Click on delete to remove the draft.

**Extensions:**

Trainer is provided access to delete published content

1. This will be a bug in the system.
2. Developers/maintenance team is notified and issue is fixed at the earliest.
3. Admin should get a notification if this case ever happens.

**Special Requirements:**

1. Trainer cannot delete the published lesson.
2. Trainer cannot make changes to slides and contents of published lesson only admin holds that right.

### **1.3.11 Use Case ID: UC011**

**Use Case:** Notifications for any changes

**Description:**

This use case deals with notification received by trainer and admin amongst themselves and how to deal with them

**Level:** User Goal

**Primary Actor:**

Trainer and Admin

**Supporting Actor:**

Mobile Devices

**Stakeholders and Interests:**

Owner, Trainer and Admin. Their interest aligns as without notifications there will be delays in corrections and uploading of content

**Pre-Conditions:**

Users should be logged in the system and some content should be posted.

**Post-Conditions:**Success end condition

Notifications are received by trainers and admins.

Failure end condition

Users do not get their notifications.

**Main Success Scenario:**

1. Trainer creates a lesson. By default lesson is added as draft.
2. Submit the content for approval. Admin receives a notification and status of lesson is moved to pending
3. Click on the notification link in the navigation pane.
4. User is navigated to the notification page. Notification page should have 6 columns namely, Description, Status, Name, Lesson, Date/Time, Action.
5. Click on view under action to see the required action to be taken.
6. Trainer is taken to the slide or the quiz question where changes are required.
7. Trainer make changes, add comments and clicks done.
8. Status of the comment is changes and admin gets the notification.

**Extensions:**

None **Special Requirements:**

1. Status is of 3 types: Lesson updated, Comment Updated and Quality Comment
2. A button should be provided to turn notifications on/off.
3. Server should be able to upload multiple(50 or more) and large media(upto 1GB per file) files

**1.3.12 Use Case ID: UC012**

**Use Case:** Admin Dashboard

**Description:**

This use case deals with what all activities happen on admin dashboard



**Level:** User Goal

**Primary Actor:**

Admin

**Supporting Actor:**

Mobile Devices

**Stakeholders and Interests:**

Admin. Admin when logged in should be able to see the latest updates and Comment summary

**Pre-Conditions:**

Users should be logged in the system and some content should be posted.

**Post-Conditions:**

Success end condition

Notifications are received by admins.

Failure end condition

Users do not get their notifications/updates.

**Main Success Scenario:**

1. Trainer creates a lesson and submit the content for approval.
2. Admin receives a notification
3. Click on the Dashboard from navigation pane
4. Admin is able to see the updates made to feedback and new lessons posted. Also they can see the Comment Summary.
5. Click on view under action to see the required action to be taken.
6. Trainer is taken to the slide or the quiz question where changes are made.
7. Admin approves the changes and contents and publish them for students

**Extensions:**

Content is posted by the trainer for approval but the comments are not mentioned for changes made.

1. Send back the content and ask for comments written for changes made.
2. Or check whether the required changes are made and approve the content.

**Special Requirements:**

1. Status is of 3 types: Lesson updated, Comment Updated and Quality Comment
2. A button should be provided to turn notifications on/off.
3. Server should be able to upload multiple(50 or more) and large media(upto 1GB per file) files

### 1.3.13 Use Case ID: UC013

**Use Case:** Manage Trainers

**Description:**

This use case deals with managing trainers by admin

**Level:** User Goal

**Primary Actor:**

Admin

**Supporting Actor:**

Mobile Devices and Trainers

**Stakeholders and Interests:**

Owner and Admin. Owner wants admin to have the rights to approve or reject trainers

**Pre-Conditions:**

Users should be logged in the system and some trainer profile should be registered.

**Post-Conditions:**

Success end condition

Trainers are successfully added or removed.

Failure end condition

Wrong trainers are removed or Trainers who filled incorrect information are approved.

**Main Success Scenario:**

1. Trainer creates a profile.
2. Admin gets a notification for new file to be added.
3. Admin validates the information and approves/reject.
4. Trainer receives an email if they are approved/rejected.
5. If a trainer leaves the school/is deceased admin remove them from the system.

**Extensions:**

None **Special Requirements:**

1. A popup message to ask for confirmation are you sure that the trainer is to be removed.
2. A button to revert the changes to be provided which will be active for 10 days. This will be used in case wrong trainer is removed.

### 1.3.14 Use Case ID: UC014

**Use Case:** Manage Students

**Description:**

This use case deals with managing students by admin

**Level:** User Goal

**Primary Actor:**

Admin

**Supporting Actor:**

Mobile Devices and Students

**Stakeholders and Interests:**

Owner and Admin. Owner wants admin to have the rights to approve or reject students

**Pre-Conditions:**

Users should be logged in the system and some student profile should be registered.

**Post-Conditions:**

Success end condition

Students are successfully added or removed.

Failure end condition

Wrong students are removed or students who filled incorrect information are approved.

**Main Success Scenario:**

1. Students creates a profile.
2. Admin gets a notification for new file to be added.
3. Admin validates the information and approves/reject.
4. Students receives an email if they are approved/rejected.
5. If a student leaves the school/is deceased admin remove them from the system.

**Extensions:**

None **Special Requirements:**

1. A popup message to ask for confirmation are you sure that the student is to be removed.
2. A button to revert the changes to be provided which will be active for 10 days. This will be used in case wrong student is removed.

### **1.3.15 Use Case ID: UC015**

**Use Case:** Manual Registrations

**Description:**

This use case deals with admins registering students to the system on phone.

**Level:** User Goal

**Primary Actor:**

Admin

**Supporting Actor:**

Students

**Stakeholders and Interests:**

Owner, Trainer and Students. Owner wants admin to add students on call as some might not know how to register

**Pre-Conditions:**

Students should be able to call on the company number provided in advertisements

**Post-Conditions:**Success end condition

Student is registered successfully.

Failure end condition

Student is not registered as they might not know how to operate a mobile device/computer.

**Main Success Scenario:**

1. Admin receives call from people
2. Admin open the student login page and fill personal details of students
3. Admin fill details for what type of license they want.
4. Admin click submit.
5. Student is accepted right at the moment as admin has already approved.
6. Student receives an email with 3 unique identification numbers used for login.

**Extensions:**

Students do not know how to operate a computer hence profile cannot be registered for online classes.

**Special Requirements:**

1. Student should be able to use website on a computer.
2. A special number shall be provided to students to call upon.

## 1.4 UML Diagrams

### 1.4.1 UML Use Case Diagram

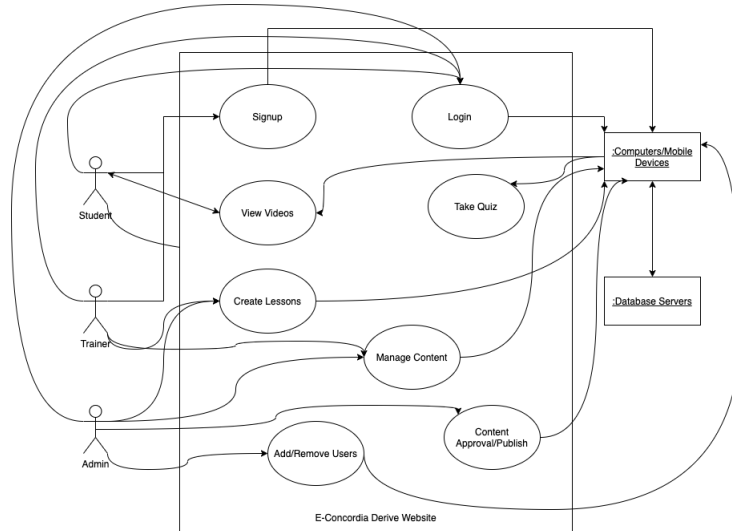


Figure 1.1: UML Use Case Diagram

### 1.4.2 UML Sequence Diagram

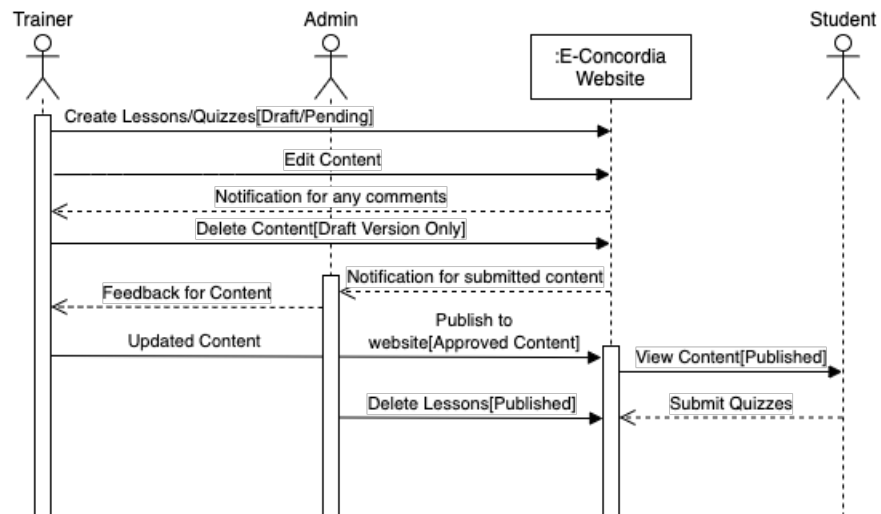


Figure 1.2: UML Sequential Diagram

### 1.4.3 UML Activity Diagram

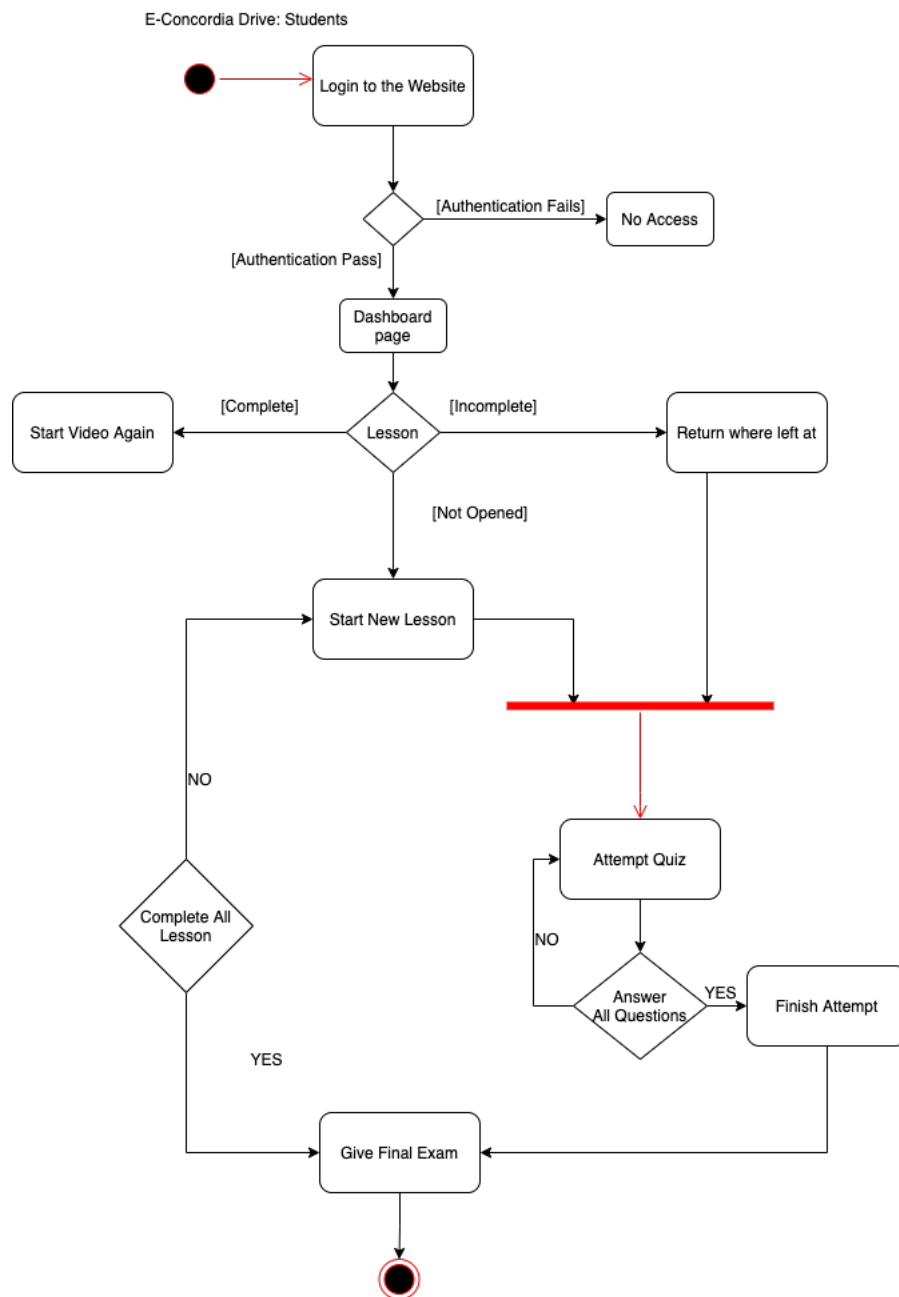


Figure 1.3: UML Activity Diagram

## 1.5 Appendix

<b>Section</b>	<b>Time Spent</b>
Introduction	0.5 hours
Actors	0.5 hours
Use Case Model	10 hours
UML Diagrams	1.5 hours
<b>Total</b>	<b>12.5 hours</b>

Table 1.2: Appendix - Chapter 1

## Chapter 2

# Supplementary Specification

### 2.1 Introduction

In this section we define the purpose, scope, all the definitions, acronyms and abbreviations and references.

#### 2.1.1 Purpose

The purpose of this document is define the requirements of E-Concordia Drive website, which is an e-learning platform where students prepare for their driving exams. This supplementary specification document lists the requirements that are not listed in the use case model document. The UCM and SS documents capture the whole requirements and form a final SRS.

#### 2.1.2 Scope

This SS documents applies to E-Concordia Drive website which will be developed by Chronos Tech Solutions. They will develop a client server system which will create the interface between mobile devices and EDC servers.

E-Concordia drive is an E-learning platform for students to learn & practice driving lessons to obtain a license. It has got three main Users: Trainer, Student and Administrator (quality). Admin edits, comments, approves and publishes lessons uploaded by a trainer. Once approved the content is ready to be viewed by students.

This specification defines the non-functional requirements of the system; such as reliability, usability, performance, and supportability as well as functional requirements that are common across a number of use cases.

#### 2.1.3 Definitions, Acronyms, Abbreviations

Project Glossary is defined in section 9



### 2.1.4 References

1. REST API Tutorial <https://restfulapi.net/rest-architectural-constraints/>
2. T.S Lehari, *Vision Document E-Concordia Drive*, Summer 2021 SOEN 6481
3. Dr. R.Morales, *Trainer Wireframe*, SOEN 6481
4. Dr. R.Morales, *Student Wireframe*, SOEN 6481
5. Dr. R.Morales, *Admin Wireframe*, SOEN 6481
6. Use Case Model - Student Registration *UC001*, Summer 2021, E-Concordia Drive
7. Use Case Model - Traner Registration *UC002*, Summer 2021, E-Concordia Drive
8. Use Case Model - Read Content of Dashboard *UC003*, Summer 2021, E-Concordia Drive
9. Use Case Model - View Videos by students *UC004*, Summer 2021, E-Concordia Drive
10. Use Case Model - Students take quiz *UC005*, Summer 2021, E-Concordia Drive
11. Use Case Model - Create Lesson *UC006*, Summer 2021, E-Concordia Drive
12. Use Case Model - Insert Content to Lesson *UC007*, Summer 2021, E-Concordia Drive
13. Use Case Model - Pending Lessons *UC008*, Summer 2021, E-Concordia Drive
14. Use Case Model - Draft Lessons *UC009*, Summer 2021, E-Concordia Drive
15. Use Case Model - Edit/Delete Lessons *UC010*, Summer 2021, E-Concordia Drive
16. Use Case Model - Notification for any Changes *UC011*, Summer 2021, E-Concordia Drive
17. Use Case Model - Admin Dashboard *UC012*, Summer 2021, E-Concordia Drive
18. Use Case Model - Manage Trainer *UC013*, Summer 2021, E-Concordia Drive
19. Use Case Model - Manage Students *UC014*, Summer 2021, E-Concordia Drive
20. Use Case Model - Manual Registrations *UC015*, Summer 2021, E-Concordia Drive

## 2.2 Functionality

This section lists the functional requirements that are common to more than one use case.

### 2.2.1 Error Logs

1. A system error log should be maintained and shared with maintenance every night.
2. All the errors will be logged in ERR.LOGS table of the database
3. Error messages should have codes and description like ERROR123 - Taking more than usual buffer time.
4. Error log file will be sent by system 2 hours before maintenance time and errors will be resolved during maintenance time.

### **2.2.2 Remote Access**

All the functionalities should be available to the users if they have a supporting device and internet connection.

## **2.3 Usability**

This section includes all the requirements that affect the usability of the website/app(for future).

### **2.3.1 Training Time**

1. All the new trainers should be provided 1 week training to familiarize them with the website.
2. All the students will be given walk-through when they first visit the website.

### **2.3.2 Industry Standards**

Website will follow IBM CUA standards and IEC Reliability standards for touchscreen devices.

### **2.3.3 Chat Bots**

There will be a chat bot on each page and users can chat with the CSR if they encounter an issue.

### **2.3.4 Multiple Systems/Web-Browser Support**

If a mobile app is made it should be working on below systems:

1. iOS(iPhone, iPad)
2. MacOSX (11.0 and up)
3. Android
4. Windows 8/10

## **2.4 Reliability**

This section lists down all the reliability requirements for the system.

### **2.4.1 Availability, MTBF, MTTR, MDR**

1. **Availability:**
  - System will be available 24/7 for all the users.
  - System will have a downtime of 2 hours on every 2<sup>nd</sup> Saturday of the month for maintenance.
  - All the Error logs will be handled by maintenance team every day.

2. **Mean Time Between Failure:** MTBF shall exceed 500 hours.
3. **Mean Time to Repair:** MTTR should be 30 minutes
4. **Maximum bug or defect rate:** Maximum bug rate should be 5/KLOC. Categories of defects will be Critical, High Medium and Low(defined in glossary.

## 2.5 Performance

This section will list down all the performance requirements of the system.

### 2.5.1 Response Time

System should be able to complete 90% transactions in 3 minutes

### 2.5.2 Throughput

System should be able to complete 100 transactions per second.

### 2.5.3 Capacity

System should be able to handle 350000 customers at any given time.

### 2.5.4 Database/Server response time

The system shall provide access to the EDC server with no more than a 10 second latency.

### 2.5.5 Resource Utilization:

As it is a web application there will be very minimal resource utilization of a device. In future if an application is made it will require 200MB of disk space and enough storage space if downloadable resources are made available.

## 2.6 Supportability

This section covers the supportability requirements of the system.

1. System is supported on the readily web browsers in the market.
2. If an application is made in future it will be downloadable on devices supporting OS listed in subsection 3.4 .

## 2.7 Design Constraints

This section lists the Design constraints of the system.

### 2.7.1 Languages Supported:

Videos uploaded will support English, Arabic, French, Hindi, Chinese, Italian for now. There will be a room for language support expansion.

### 2.7.2 Development tools:

Development tools used are **phpStrom** and **MySQL**.

### 2.7.3 Management tools:

1. **Defect Management:** JIRA
2. **CI/CD:** Github
3. **Documentation:** Overleaf/Latex

### 2.7.4 Architectural Constraints:

1. Because this is a client server based application, the client and server application must be able to evolve separately without any dependency on each other.
2. All the webpages should be cacheable to improve the performance on client end and better scalability on the server side.

## 2.8 Online User Documentation and Help System Requirement

1. Screens should be interactive to make it easy to use on touch screen devices.
2. Contact us information should be provided.
3. A user should be guided through the system when they first login.
4. A help document should be in place for the help later on.

## 2.9 Glossary

### 2.9.1 Definitions

Name	Description
User	End User of the system which includes student, trainer and admin
Student	Person who registers to prepare for the driving test.
Trainer	Person who is a driving instructor and creates content for students.

Admin	Person who is also a QA and manages the driving school.
Developer	Person responsible for the development of website
Tester	Person responsible for testing the code
Data	<ul style="list-style-type: none"> <li>• <b>Definition 1</b> All the content on website.</li> <li>• <b>Definition 2</b> Also details of users registered.</li> </ul>
System	<ul style="list-style-type: none"> <li>• <b>Definition 1</b> E-Concordis Drive Website</li> <li>• <b>Definition 2</b> A computer/mobile device</li> </ul>
Server	The system which provides resources, data and services to other systems
Quality Check	To verify if the content uploaded is correct
Pending	Lesson is under review by the trainer and admin.
Draft	Initial stages where person working on content can edit it.
Published	Content is made available for students.

Table 2.1: Definitions.

### 2.9.2 Acronyms and Abbreviations

Acronym	Description
MVC	Model View Controller
MTTR	Mean Time to Repair
MTBF	Mean Time to Failure
KLOC	Thousand Lines of Code
IBM	International Business Machine
IEC	International Electrotechnical Commission
CUA	Common User Access
SRS	Software Requirement Specifications
SS	Supplementary Specifications
LTR and RTL	Left-to-Right and Right-to-Left

UCM	Use Case Model
UML	Unified Modelling Language

Table 2.2: Acronyms and Abbreviation.

## 2.10 Appendix

Section	Time Spent
Introduction	1 hours
Functionality	1 hours
Usability	1 hours
Reliability	0.5 hours
Performance	1 hours
Supportability	0.5 hours
Design Constraints	1 hours
ReliaOnline User Documentation and Help System Requirementbility	0.5 hours
Glossary	1 hours
<b>Total</b>	<b>7.5 hours</b>

Table 2.3: Appendix - Chapter 2