Task 0: Logging

Tasks	Time Consumed
Task 1	45 minutes
Task 2	3 hours 30 minutes
Task 3	2 hours

Task 2: Use Case Model

E-Concordia Drive

1. Actor-Goal List

Actor	Goal
Admin	Approves the Lesson Manages the Lesson Delete the Lesson View Students Information View Trainers Information
Trainers	Create Lecture Videos Quiz Creation Lesson Modification View Lesson status
Students	Watch Video Lecture View Course Progress Check video lecture status Attend Quizzes View Course Grades Pay Course Fees

2. Use Case Models

Id: UC-1

Use Case: Lesson Approval

Description
Trainer creates the lesson and drafts it. Once he drafts the video, a notification is being received by admin. Then admin checks the content of the lesson and check for content and its quality. If admin finds it perfect, then he approves the video and publish it, otherwise he sends comment to trainer regarding changes. Finally, Students can watch the video uploaded and can learn and practice from it.

Level: User Level

Primary Actor
Admin, Trainer

Supporting Actors
Student

Stakeholders and Interests
Developers = They develop various functionalities.
Testers = They test all these functionalities.
Quebec Traffic Department = They can verify the quality of lectures uploaded once admin approves it.

Pre-Conditions
Lesson should be created with required slide/quiz description.
Post Conditions

Success end condition
A good quality lesson will be published, and students can watch them anytime and anywhere. Also, if the lesson has some quizzes, then they can solve it too.

Failure end condition:
If the lesson's quality is not good or up-to mark, then video will not be approved, and a note will be send to the trainer regarding changes to be made.

Main Success Scenario

1. Lesson created by Trainer.
2. All related information to be mentioned alongside the slides.
3. Admin checks the draft video's quality and the content.
4. If admin feels the content is perfect, he will publish the video.
5. Students purchase this newly published lesson.
6. Students can now learn and practice this Lesson.

Special Requirements

1. User Interface requirements
2. Access Control
3. Availability
4. Software Interoperability

Id: UC-2

Use Case: Lesson Management

Description
Admin keeps track of the lesson/slides/video drafted and published. He continuously checks the drafted video list and check for its quality. Once he/she feels the lesson is perfect he/she will publish it. They will also keep track of published video.

Level: Sub-function

Primary Actor
Admin

Supporting Actors

Student, Trainer

Stakeholders and Interest
Developers = They develop Lesson Management functionalities.
Testers = They test Lesson Management functionalities.

Pre-Conditions
Lesson should be Created.
Lesson might be published.

Post Conditions

Success end condition
Admin managing all the lessons perfectly. They must be able to publish, reject, edit or delete the published video.

Failure end condition:
Admin not able to access the lessons. They are not able to publish, reject, edit, or delete the published video.

Main Success Scenario

1. Admin self identifies themselves.
2. Admin Checks the Notification.
3. Admin checks for drafted video, if they found it, they publish it or rejects it.
4. Admin checks list of published videos, if they found any issue with the published video, they edit it or delete it depending on the situation.

Special Requirements

1. User Interface requirements
2. Access Control
3. Availability
4. Software Interoperability
5. Consistency

Id: UC-3

Use Case: Delete published video

Description
Admin checks for the published video's quality, if they found any content not related to traffic or any rule which is no more applicable, then they can delete the published video. So that students do not study any outdated lesson.

Level: Sub-function

Primary Actor
Admin

Supporting Actors

Student

Stakeholders and Interests
Developers = They develop Lesson deletion functionalities, only done by admin.
Testers = They test lesson deletion functionality and role definition for lesson deletion.

Pre-Conditions
Lesson should be created and then published.

Post Conditions

Success end condition
Admin successfully able to delete the published lesson.

Failure end condition:
Admin not able to delete the published lesson.

Main Success Scenario

1. Admin self identifies itself
2. Admin checks the content of the published lesson.
3. They found something outdated or out-of-content lesson.
4. They delete the published lesson.

Special Requirements

1. User Interface requirements
2. Access Control
3. Availability
4. Consistency
5. Reliability

Id: UC-4

Use Case: Manage Student Information

Description
Admin can view various information of students. They can keep track record of all student information like what courses they have enrolled, personal details, their performance, and many more.

Level: Sub-function

Primary Actor
Admin

Supporting Actors
Student

Stakeholders and Interests
Developers = They develop view user information functionalities, only done by admin.
Testers = They test view user information functionality and role definition for viewing confidential information.

Pre-Conditions
Students must be registered into the system.

Post Conditions

Success end condition
Admin successfully able to view all the information of students registered into the system.

Failure end condition:
Admin not able to get any information of registered students.

Main Success Scenario

1. Admin self identifies itself.
2. Admin selects the view student information button.
3. Admin get all the information of the desired student.

Special Requirements

1. User Interface requirements
2. Access Control
3. Availability
4. Reliability

Id: UC-5

Use Case: Manage Trainers Information

Description
Admin can view various information of trainers. They can keep track record of all trainer information like what lessons they have designed, their performance, and personal information.

Level: Sub-function

Primary Actor
Admin

Supporting Actors
Trainer

Stakeholders and Interests
Developers = They develop view trainer information functionalities, only done by admin.
Testers = They test view trainer information functionality and role definition for viewing confidential information.

Pre-Conditions
Trainer must be registered into the system.

Post Conditions

Success end condition
Admin successfully able to view all the information of trainers registered into the system.

Failure end condition:
Admin not able to get any information of registered trainers.

Main Success Scenario

1. Admin self identifies itself.
2. Admin selects the view trainer information button.
3. Admin get all the information of the desired trainer.

Special Requirements

1. User Interface requirements
2. Access Control
3. Availability
4. Reliability

Id: UC-6

Use Case: Create Lecture Videos

Description
Trainer creates a lesson and then adds various slides to it. These slides contain traffic related material. The created lesson is added in a draft list. They can also add quizzes to a lesson. Once the quality of this lesson is checked and it is perfect then admin publish the lesson.

Level: User-level

Primary Actor
Trainer

Supporting Actors
Student, Admin

Stakeholders and Interests
Developers = They develop lesson creation, slide addition, quiz addition functionalities.
Testers = They test lesson creation functionality.
Quebec Traffic Department = They give content for the slide to the trainer in some topics.

Pre-Conditions
Trainer must have enough content to create the lesson.

Post Conditions

Success end condition
Trainers distinguish all the contents and design various slides with all the content it has. Once the slides are ready, he adds the slide in a lesson. Once all the slides, quizzes are added, he sends lesson for approval.

Failure end condition:
Admin not able to add slide in the lesson, which in turn does not create a lesson.

Main Success Scenario

1. Trainer self identifies itself
2. Trainer collects all the information/data to be added in a slide.
3. Trainer creates the slide.
4. Once the slide is created, he adds it in a lesson.
5. All this slides, quizzes collectively forms a lesson.

Special Requirements

1. User Interface requirements
2. Access Control
3. Reliability
4. Convenient
5. Availability
6. Integrity

Id: UC-7

Use Case: Quiz Creation

Description
Trainer creates various types of quizzes for students, which checks the student involvement in the lesson and the topic. Quiz types created by trainers are simple quiz. Multiple Choice Question, Reorder Quiz, Drag and Drop and True or False type questions.

Level: user-level

Primary Actor
Trainer

Supporting Actors
Student, admin

Stakeholders and Interests
Developers = They provide functionality to add various types of quizzes to the lesson.
Testers = They test functionality of quiz creation.
Quebec Driving Issuing Department = They can provide enough content for quiz creation to trainer.

Pre-Conditions
Trainer must have quiz content.

Post Conditions

Success end condition
Trainer identifies various sections where he/she can add the quiz to the lesson. Once they have the content ready for quiz, quiz is being added to the lesson.

Failure end condition:
Admin not able to add the quiz in any lesson.

Main Success Scenario

1. Trainer self identifies itself.
2. Trainer creates the content of the quiz.
3. Trainers add the quiz to appropriate lesson.

Special Requirements

1. User Interface requirements
2. Access Control
3. Availability
4. Reliability
5. Integrity

Id: UC-8

Use Case: Lesson Modification

Description
Trainer creates the lesson and drafts it. Once the video is cited in the draft list, admin checks the quality of the lesson, if admin send any comments to the lesson, then trainer must follow the instruction mentioned in the comment. Here the instructions could be editing any slide or quiz. Thus, the trainer modifies the lesson according to the comment given by admin.

Level: user-level

Primary Actor
Trainer, admin

Supporting Actors
Student.

Stakeholders and Interests
Developers = They provide functionality to modify lesson.
Testers = They test functionality of modifying the lesson.

Pre-Conditions
Trainer must have created lesson.

Post Conditions

Success end condition
Trainer must be able to modify the lesson once any error or any comment is being attached to the video.

Failure end condition:
Admin not able to modify the lesson.

Main Success Scenario

1. Trainer self identifies itself.
2. Trainer creates the lesson.
3. Admin attaches any comment if any / trainer wants to edit the draft lesson.
4. Trainer modifies the content of the lesson.

Special Requirements

1. User Interface requirements
2. Access Control
3. Availability
4. Consistency

Id: UC-9

Use Case: Lesson Status

Description
Once the trainer completes the lesson, the lesson is drafted, and the admin is notified. Now, admin can approve the lesson and can published, if the quality of the content is alright otherwise, they will reject the lesson. The status that that the video is published, approved but not published or rejected or any comment is being posted on a lesson can be seen in the "view status" of a lesson.

Level: user-level

Primary Actor
Trainer, admin

Supporting Actors
Student.

Stakeholders and Interests
Developers = They provide functionality to modify lesson.
Testers = They test functionality of modifying the lesson.

Pre-Conditions
Trainer must have created lesson.

Post Conditions

Success end condition
If any comment is being posted on any lesson, trainer can see the status successfully and can work on it.

Failure end condition:
Admin not able to view the comments posted on any lesson.

Main Success Scenario

1. Trainer self identifies itself.
2. Trainer creates the lesson.
3. Admin attaches any comment if any.
4. If no comments and quality is perfect, admin publish the lesson.
5. Trainers view all the status of the lesson whether published, rejected or any comment posted.

Special Requirements

1. User Interface requirements

2. Availability
3. Consistency
4. Reliability

Id: UC-10

Use Case: Lesson Information

Description
The lesson created by trainer, once approved after quality check is being published and made available to all the students registered. Now if students want to take that course, then they to purchase the course and then can learn and practice the lesson they bought.

Level: user-level

Primary Actor
Student

Supporting Actors
Trainer, Admin.

Stakeholders and Interests
Developers = They provide functionality to view lesson.
Testers = They test functionality of playing video lecture.

Pre-Conditions
Trainer must have created lesson.
Admin must have published the lesson.
Students must have purchased the lesson.

Post Conditions

Success end condition
If the lesson published is bought by the student, then he/she can view the video lecture and can read the content of the lesson.

Failure end condition:
Student not able to purchase the course. If bought, then then they might not able to play the video lecture.

Main Success Scenario

1. Student self identifies itself.
2. Trainer creates the lesson.
3. Admins publish the lesson.
4. Students purchase the lesson.
5. Student able to view the lesson.

Special Requirements

1. User Interface requirements
2. Access Control
3. Availability

4. Software Interoperability
5. Device Interaction
6. Performance

Id: UC-11

Use Case: Cite Progress

Description
Students when bought a course and completes some lesson of it, can track their progress when they again re-login into the system. It shows how much lessons does the student has completed and how many marks does he/she scored in the quiz.

Level: user-level

Primary Actor
Student

Supporting Actors
Trainer, Admin

Stakeholders and Interests
Developers = They provide functionality to view student progress.
Testers = They test functionality developed by developers.

Pre-Conditions
Trainer must have created lesson.
Admin must have published it.
Student must have purchased it.

Post Conditions

Success end condition
Student when starts any course, can track his/her status as in how many video lectures they have seen, what amount of content they have covered. Also, the quizzes they covered.

Failure end condition:
Student not able to see his/her own progress of any course purchased by him/her.

Main Success Scenario

1. Student self identifies itself.
2. Trainer creates the lesson.
3. Admins publish the lesson.
4. Students purchase the course.
5. Student tracks their course coverage.

Special Requirements

1. Useability
2. Access Control
3. Availability
4. Consistency
5. Software Interoperability

Id: UC-12

Use Case: Track Course Status

Description
The status of the course purchased by the student whether they have started the course, or it is pending (not started yet) or have finished it. This all can be track by the student.

Level: user-level

Primary Actor
Student

Supporting Actors
Trainer, Admin

Stakeholders and Interests
Developers = They provide functionality to track the course.
Testers = They test functionality to track the course.

Pre-Conditions
Trainer must have created lesson.
Admin must have published it.
Student must have purchased it.

Post Conditions

Success end condition
Student successfully able to track their course like whether they have started learning it or have finished it or have kept it pending.

Failure end condition:
Student not able to track their course status correctly. They are getting wrong information.

Main Success Scenario

1. Student self identifies itself.
2. Trainer creates the lesson.
3. Admins publish the lesson.
4. Students purchase the course.
5. Student tracks their course status(ongoing , completed, pending).

Special Requirements

1. Useability
2. Access Control
3. Availability
4. Consistency
5. Software Interoperability
6. Reliability

Id: UC-13

Use Case: Attend Quizzes

Description
Students when purchase any course, they might get some quiz being added in some lesson to make the lesson quite interesting and lively. Students attend these quizzes and solve it.

Level: user-level

Primary Actor
Student

Supporting Actors
Trainer, Admin

Stakeholders and Interests
Developers = They provide functionality to answer the quiz.
Testers = They test functionality developed by developers.

Pre-Conditions
Trainer must have created lesson.
Admin must have published it.
Student must have purchased it.

Post Conditions

Success end condition
Students able to solve any type of quiz from any course they have purchased.

Failure end condition:
Student not able to solve the quiz from the course that are owned by student.

Main Success Scenario

1. Student self identifies itself.
2. Trainer creates the lesson.
3. Admins publish the lesson.
4. Students purchase the course.
5. Student attend any type of quiz encountered.

Special Requirements

1. Useability
2. Access Control
3. Availability
4. Consistency
5. Software Interoperability
6. Device Interaction

Id: UC-14

Use Case: View Grades

Description
Students can view the grades that they obtained by solving various types of quizzes that are included in the lesson that they have purchased.

Level: user-level

Primary Actor
Student

Supporting Actors
Trainer, Admin

Stakeholders and Interests
Developers = They provide functionality to view grades.
Testers = They test functionality to view grades.

Pre-Conditions
Trainer must have created lesson.
Admin must have published it.
Student must have purchased it.
Student must have given quiz.

Post Conditions

Success end condition
Students able to view grades of various quizzes that they have solve from the course that they have purchased.

Failure end condition:
Student not able to see their grades even after solving the quiz.

Main Success Scenario

1. Student self identifies itself.
2. Trainer creates the lesson.
3. Admins publish the lesson.
4. Students purchase the course.
5. Students solves the quizzes.
6. Students able to see their grades.

Special Requirements

1. Useability
2. Access Control
3. Availability
4. Consistency

Id: UC-15

Use Case: Course Fee Payment

Description
Students able to pay the fees of the courses that they want to purchase.

Level: user-level

Primary Actor
Student

Supporting Actors
Trainer, Admin

Stakeholders and Interests
Developers = They provide functionality to pay the fees.
Testers = They test functionality to pay the fees securely.

Pre-Conditions
Trainer must have created lesson.
Admin must have published it.

Post Conditions

Success end condition
Student when liked a particular course published by admin, purchase it and then pay the amount of the course.

Failure end condition:
Student not able to finish the payment process.

Main Success Scenario

1. Student self identifies itself.
2. Trainer creates the lesson.
3. Admins publish the lesson.
4. Students pay for the course, he/she wants to purchase.

Special Requirements

1. Useability
2. Access Control
3. Availability
4. Software Interoperability
5. Security
6. Confidentiality
7. Integrity

Diagrammatic Notations:

- Use Case Diagram:

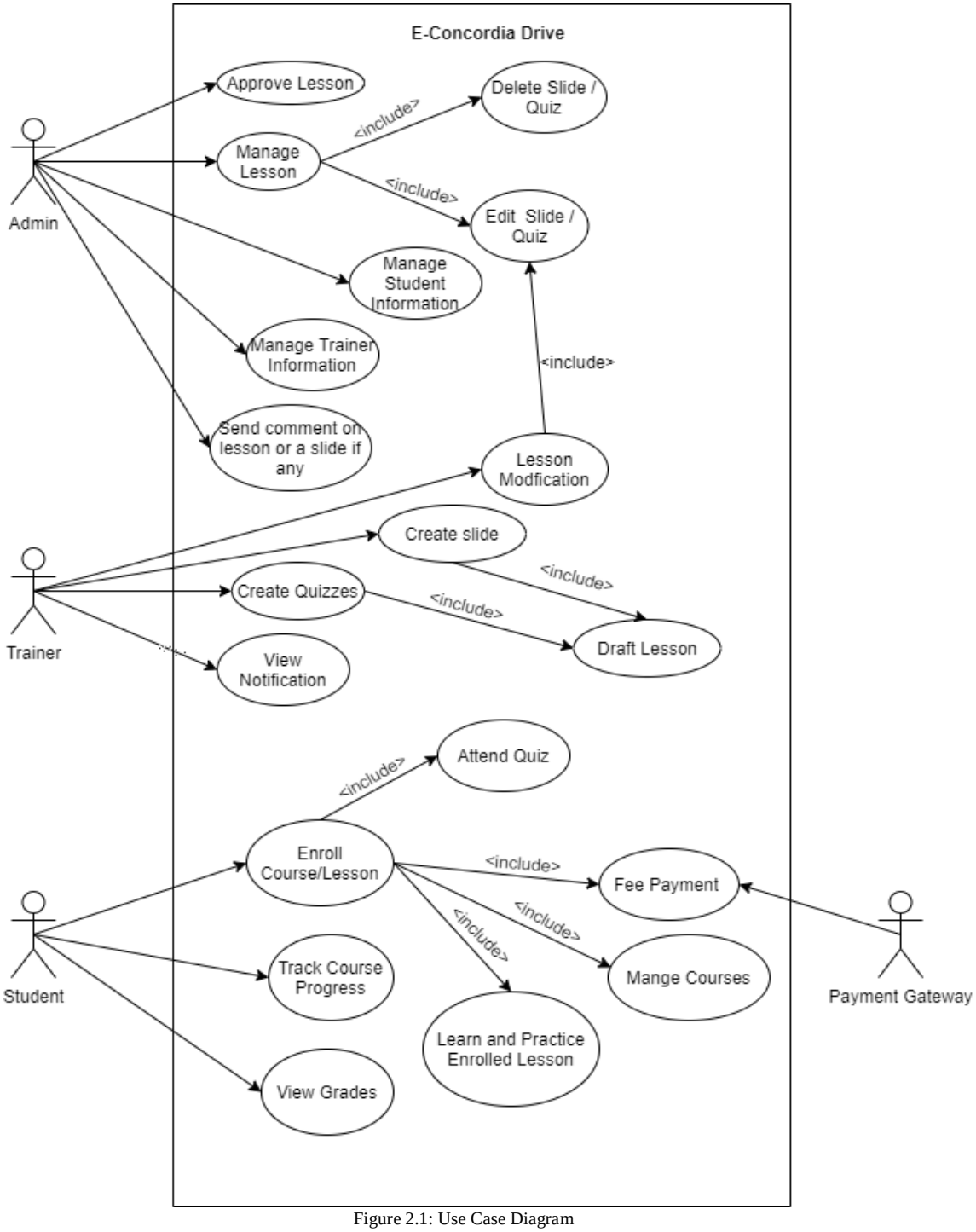

Figure 2.1: Use Case Diagram

- Sequence Diagram:

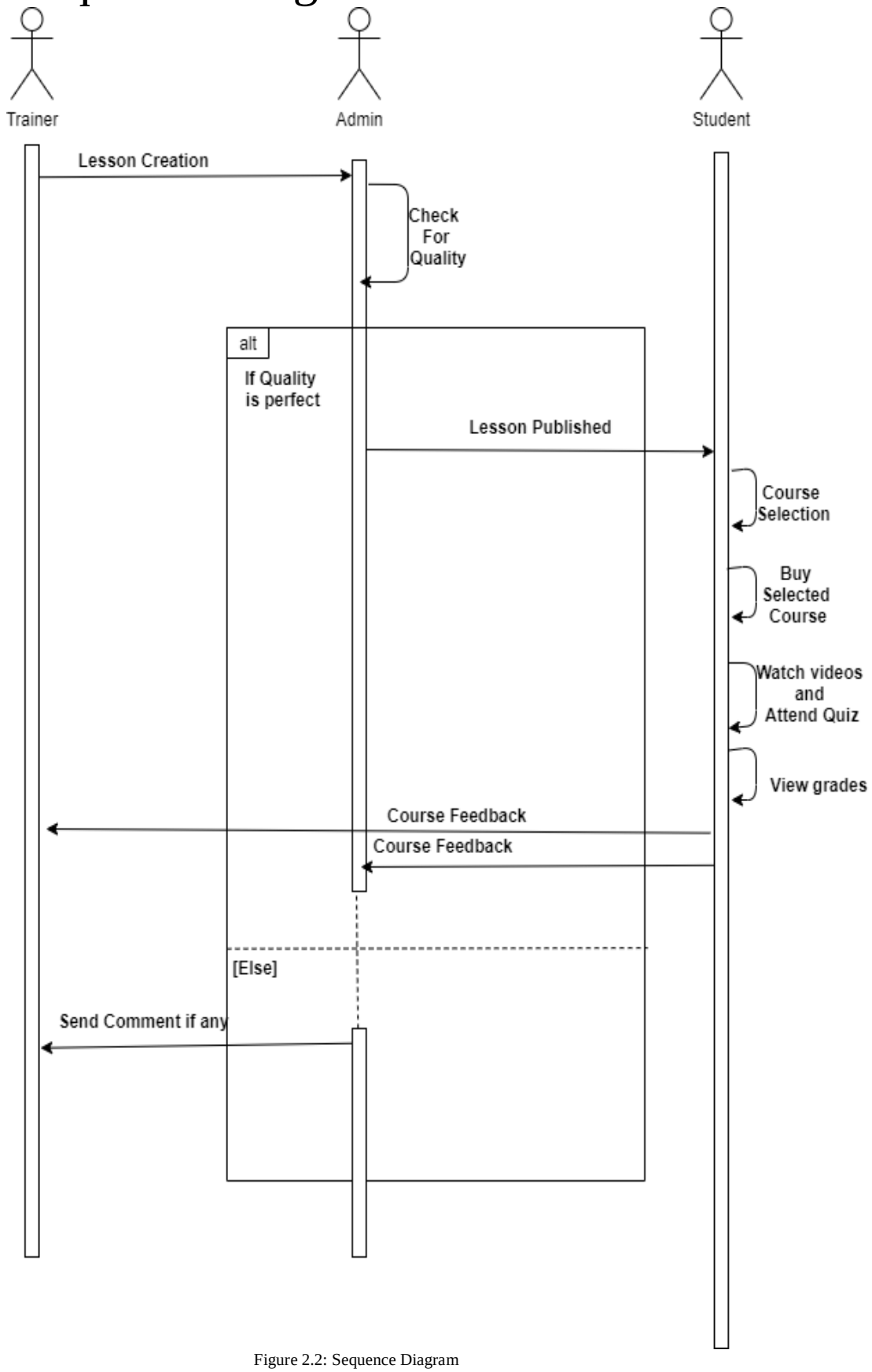

Figure 2.2: Sequence Diagram

- Activity Diagram:

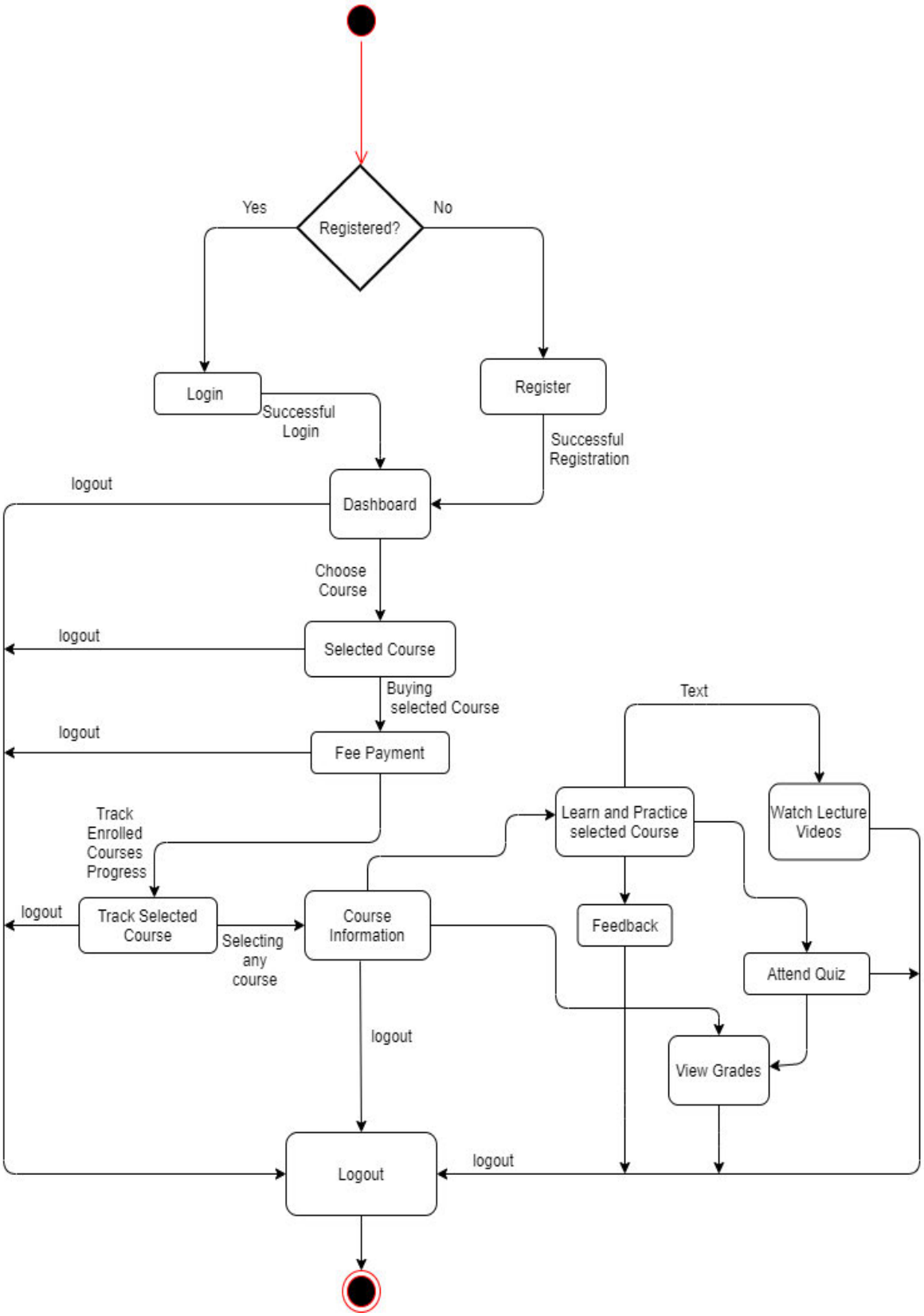

Figure 2.3: Activity Diagram

Task 3: Supplementary Specification and Glossary

E-Concordia Drive

Supplementary Specification

1. Introduction

Purpose

The document is intended to specify the requirements of E-Concordia Drive system. This supplementary specification captures requirements that were undermined in the use-case model. The supplementary specification and the use-case models collectively point out all requirements of the system.

1.1. Scope

E-Concordia system, an interactive online platform, caters different driving courses exclusively for the students at Concordia University.

Students can buy any driving course from the system and can learn and practice from it. Here students can watch driving videos, attend quizzes, and can track their progress.

The content is being designed by trainers, who are masters in the driving field and owned a driving school. They create lessons, quizzes and then post it which is check by admin for quality purpose. Once it is approved, it is made available to students.

1.2. References

- Collegiate sport system https://csis.pace.edu/~marchese/SE616_New/Samples/Example%20%20Supplementary%20Specification.htm
- Software Requirements Specification - https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document
- E-Concordia Drive Wireframe - https://drive.google.com/drive/folders/1ZwPfXS0qTLdKUHz-8RnGFNUoWHh_gsec

2. Functionality

This section describes various functional requirements that are common to more than one use case.

2.1. Payment Gateway Error

All payments should be track, whether it be a successful or unsuccessful. A receipt in a pdf format should be mail to the user, so that two reference can be noted. One in the database and other in the user's hand.

2.2. Remote Access

To access the system remotely, one need to connect their handset to an active internet connection. By which they can learn and practice driving lesson remotely.

2.3. Server Crash

To watch lecture videos anywhere anytime, video should be made available to all users if have they enrolled that course. Thus, if any issue in server or in backend of the system, users will not be able to access the video.

3. Usability

This section describes different requirements that relate to, or affect, the usability of the system.

3.1. Operating System Compliance

The system designed will be supported in windows having windows operating system as low as **Windows 7**.

For IOS devices, a minimum **ios 10** will be required to use the system.

Android devices should at least have 6.0 operating system to run the system.

3.2. Simple User Interface

E-Concordia drive interface should be easy-to-use and can be used without any kind of special knowledge.

3.3. "How-to-use" document

The system shall consist of a document name “How-to-use” which will have all the information of all functionalities, which make user understand the system very easily and rapidly.

3.4. Media player support

The system in which the user is accessing the system must have a media player support, as video will be there in all the courses. To play these videos smoothlesley, a media player support in the devise is more advisable.

4. Reliability

This section describes various reliability requirements.

4.1. Availability

E-Concordia drive must be available all the time, i.e., 24 hours a day, 7 days a week.

4.2. Accuracy

The different video qualities should be accurate as per user selected options. For e.g., User should be able to watch 360-pixel video if he/she has selected that option.

4.3. Mean Time to Repair

The mean time to repair should be very less. It should not exceed more than one day. By which the system will made be available for more time even after repair.

4.4. Bug or Defects Rate

The system should have minor bugs. Minor bugs from system point-of-view are some simple things that not working like a button not pressing, wrong video reference which can be solved in some hours while critical bugs include some functionality in the system not working, to solve such bugs it takes more than one working days.

5. Performance

The performance characteristics of the system are stated in this section.

5.1. Response Time

Response time of all transactions of E-Concordia drive should be completed within 1 minutes.

5.2. Concurrent Users

The system shall support 5000 users at a single and should be able to cater all functionalities that each user wants.

5.3. Resource Utilization

The system shall utilize low disk space and memory in the user's device. A cookie file will be stored in the user's device, other than this all the system's data will be stored in remote database. Thus, users will get what he/she wants in right manner in minimum time with minimum cost incurred with not no wastage of resources.

5.4. Throughput

The throughput of the system depends on the device's internet connection speed. The Internet speed of 25 Mbps is considered very good to watch videos online. High throughput is recommended.

6. Supportability

This section indicates any requirements that will enhance the supportability or maintainability of the system

6.1. Coding Standards

Coding standards shall be defined and followed by developers. The definition must include the rules and guidelines that determine programming style, procedures, and methods.

6.2. Naming Convention

Naming convention should be consistent for the project. All the constants used in project should be in capital letters, function name should start with d_function_name, and many such.

7. Design Constraints

This section should indicate any design constraints on the system being built.

7.1. Software Languages

System will be developed using HTML, JavaScipt, JQuery in the Frontend and PHP, JAVA, Python as backend languages. It will use MySQL database to store data.

7.2. Payment Gateway Integration

Paypal payment gateway shall be integrated to the system which will support online transactions, from which user can pay money for the course they enrol.

7.3. Internet Browsers

The system will be compatible with different browsers. In chrome, the version should be 4.0, safari should be 3.0 and Mozilla Firefox version 3.0.

8. Online User Documentation and Help System Requirements

There will be an attached pdf in the help section of the system named “How-to-use”, where users can seek help if they have any query regarding usage of functionality.

9. Glossary

HTML – Hypertext Markup Language

PHP – Hypertext Pre-processor