SCSJ2203: Software Engineering

SOFTWARE REQUIREMENT SPECIFICATION - SRS

E-RUBRIC SYSTEM

Version 2.0

17/6/2020

School of Computing, Faculty of Engineering

Prepared by: NIGHT

Table of Contents

1. Introduction

1.1 Purpose

This SRS describes the requirements for the E-Rubric System in details. The specified requirements are described by using use case diagram, together with the support of sequence diagram and activity diagram. The description of requirements allows the target audience to understand the structure and process of the whole proposed system. The intended audience for this SRS is the stakeholders including lecturers, students, course coordinator and CICT staff.

1.2 Scope

The software product is E-Rubric System, an improved computerized system used by lecturer and students especially in their assignment. Generally, the system allows the course coordinator to upload or generate rubrics using a built-in generator. The lecturer also allowed to upload an assignment, or any course material to the system and allow the students to download. The deadlines are set by the lecturer upon upload the assignments. Lecturer also has to grade the student based on the rubric and give feedback to the students. Lecturers can assign the students into groups, allowing them to peer review one another. There are also a built-in plagiarism checker and grammar checker that assist the lecturer in their grading. Students can upload their assignments and view grading and reviewing feedback from their lecturer. There are also peer review systems for students to review their group mates accordingly. The system also alerts students before the submission deadline expires. The course coordinator will upload the course material along with the lecturers. The lecturer also can generate a student's performance report.

The objective and goal of this system are to synchronize every lecturer so that their marking standard will be the same as they are strictly following the same rubrics. The benefits of the system are that it allows the students to view their grade in details. Besides, it also automates the process of calculating peer review marks of the students.

1.3 Definitions, Acronyms and Abbreviation

- SD – System Documentation
- CICT – Department of Digital Services
- UTM – University Technology Malaysia
- ACID – Academic Computing ID
- DBMS – Database Management System
- GUI – Graphical User Interface
- HTML – Hypertext markup language
- CSS – Cascading style sheets
- SQL – Structured Query Language
- PHP – Personal Home Page

1.4 References

[1] Exemplars (nd). Introducing Rubrics to Students. Retrieved 18 March 2020 from https://exemplars.com/resources/assessment/rubrics/how-to

[2] Rcampus. (nd). Retrieved 19 March 2020 from https://www.rcampus.com/indexrubric.cfm

[3] Rubric and Evaluation form. (nd). Retrieved 19 March 2020 from http://www.fp.utm.my/buku3/rubric.html

[4] SDLC-Software Development Life Cycle (n.d.). Retrieved 20 March 2020 from https://www.projectmanager.com/software/use-cases/sdlc

[5] Teammates. (nd). Retrieved 19 March 2020 from http://teammatesv4.appspot.com/

1.5 Overview

The current version of system requirement specification involved description of introduction, specific requirements. The purpose, scope, definitions, acronyms and abbreviations as well as reference are described in the introduction, while the interfaces, system features which consists of use cases, requirements, constraints, and software system attributes are explained in specific requirement.

2. Overall Description

2.1 Product Perspective

2.1.1 System Interfaces

The system shall be able to communicate with external UTM ACID system for login credential

The system shall be able to communicate with database store data.

2.1.2 User Interfaces

a) Logical characteristics: The GUI of the system is based on HTML, CSS, and JavaScript, where it is accessible from any device with any web browser.

b) Aspects:

- If the user does not log in, there are limited pages can be access by the user, including the login page.
- After login, the website should show different content for different type of user respectively (course coordinator, lecturer, and student)
- On the event of website server down or error, it should show error message understandable by the users.

2.1.3 Hardware Interfaces

The E-Rubric System is a web-based system, hence it required:

Web server based on Linux environment

Database server

Internet bandwidth speed: 1 GB/s

Website requirement: support mobile and PC with web browser

2.1.4 Software Interfaces

1. Web hosting server is needed to host the website and to handle request of client.

Name: Microsoft Windows server 2019

Mnemonic: Windows server 2019

Version number: 10

Source: https://www.microsoft.com/en-us/cloud-platform/windows-server

2. Web hosting panel is required to manage the website. It includes DBMS, file hosting, and more.

Name: cPanel

Mnemonic: cPanel

Version number: version 86

Source: https://cpanel.net/

3. UTM ACID integration is required for user login.

Name: Academic Computing ID

Mnemonic: ACID

Version number: none

Source: https://digital.utm.my/akaun-acid/

4. Turnitin API integration support that allow lecturer to check plagiarism.

Name: Turnitin

Mnemonic: Turnitin

Version number: none

Source: https://help.turnitin.com/ithenticate/ithenticate-developer/api/api-guide.htm

2.1.5 Communication Interfaces

Support HTTPS protocol to enable secure connection between client and server. Besides, it also supports FTP protocol for file transfer.

2.1.6 Memory constraints

Since the UTM community is large, high memory and storage are needed to satisfy the user's usage.

Web server RAM: 12GB

2.1.7 Operations

This system shall identify different type of user during the login process. Different features are available to different type of users, and some features are limited to some users.

Most of the operations of this system are interactive operations. For instance, interactive operations are run on real time such as create, read, update, and delete operation to the database.

The system shall backup the information to a separate database for recovery.

2.1.8 Site adaptation requirements

There is no site adaption for this system. It is just a website that can be run through any web browser.

2.2 Product functions

Credential

This function provides security layer to protect the data. Different user can login with their own credential and only can edit the data assigned to them.

Add rubric / View rubric

This function allows certain user to add rubric to the database, and other users to view the rubric. Each rubric will append to an assignment.

Add assignment

This function allows lecturers to add new assignment details and links to the database and allows other users to view the assignment.

Upload assignment

This function allows students to upload their assignment to the database and wait to be graded by the lecturer. Students only can upload assignment before the due date.

Assign group

This function allows the lecturer to assign the students to different group for each assignment.

Grade assignment

This function allows lecturer to grade the students' assignments and give feedback to them. Lecturer only can grade the assignment after the due date.

View grading / view performance report

This function allows student and lecturer to view their assignment grade. This function only can be accessed after the assignment has be graded.

2.3 User characteristics

a) Student –Includes all UTM students that have assignment in their syllabus. They should familiar with basic functions to operate a website. Main responsible is to download learning material and to upload assignment, as well as view their marks.
b) Lecturer – Takes important role of using this system. Main responsible is to upload new assignment as well as grade the students' assignments. Should be familiar with this system as UTM currently also have similar system
c) Course coordinator – Responsible to upload rubric to the system. Should be well trained to operate such system.

2.4 Constraints

The system shall provide access only to the user that authenticated.
The system shall prevent type of user from accessing and updating certain data.
The system shall store encrypted user data to prevent data abuse.

2.5 Assumptions and dependencies

Assumptions made are all user have stable internet connection, as well as browser that support normal website functionalities.

3. Specific Requirements

3.1 Overall Description

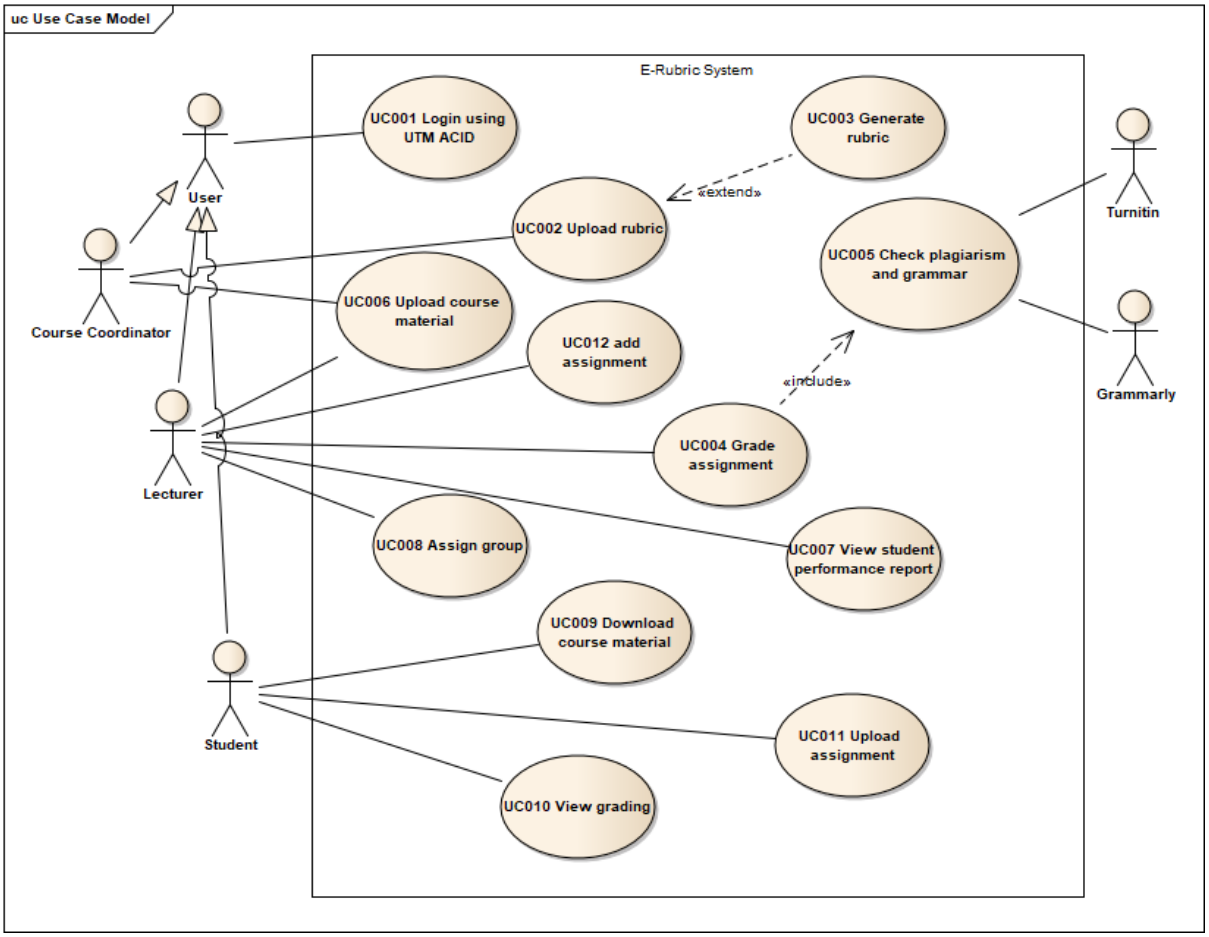

Figure 2.1: Use Case Diagram for E-Rubric System

The three classes, Lecturer, CourseCoordinator and Student are inherited from superclass User. Student class is aggregate with Assignment class, where one student can have 0 or many assignments. Course class and Assignment class has composition relation, where one Course owns the 0 or many Assignments. Assignment class has aggregation relation with rubric, where the rubric can belong to many Assignment. Assignment class also composite with AssignmentReport, AssignmentGroup, and PeerReview classes. Those classes are part of Assignment and cannot exists without Assignment class. One assignment only can have 1 report and 1 PeerReview class attached to it. One AssignmentGroup can have many Assignment class. The AssignmentGroup also has many Students (as the group members), and Student can have many AssignmentGroup (from different assignment).

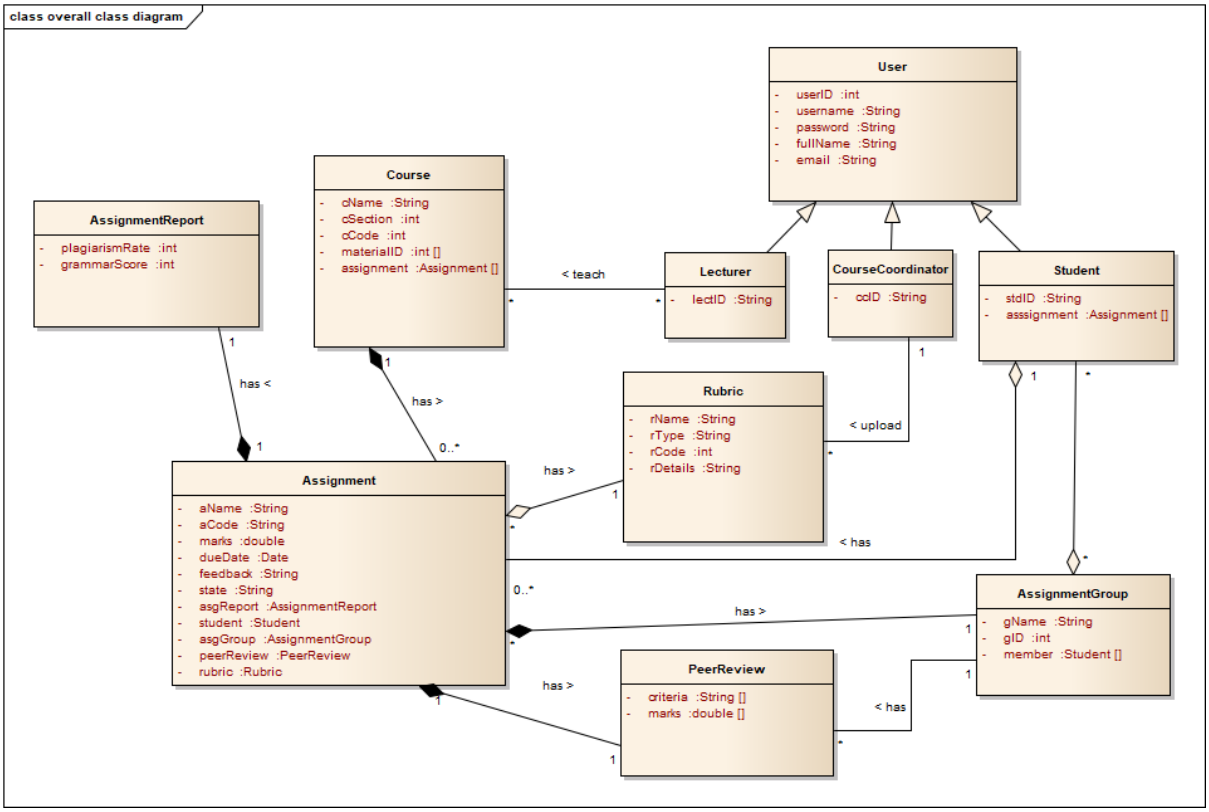

Figure 2.2: Domain Model for E-Rubric System

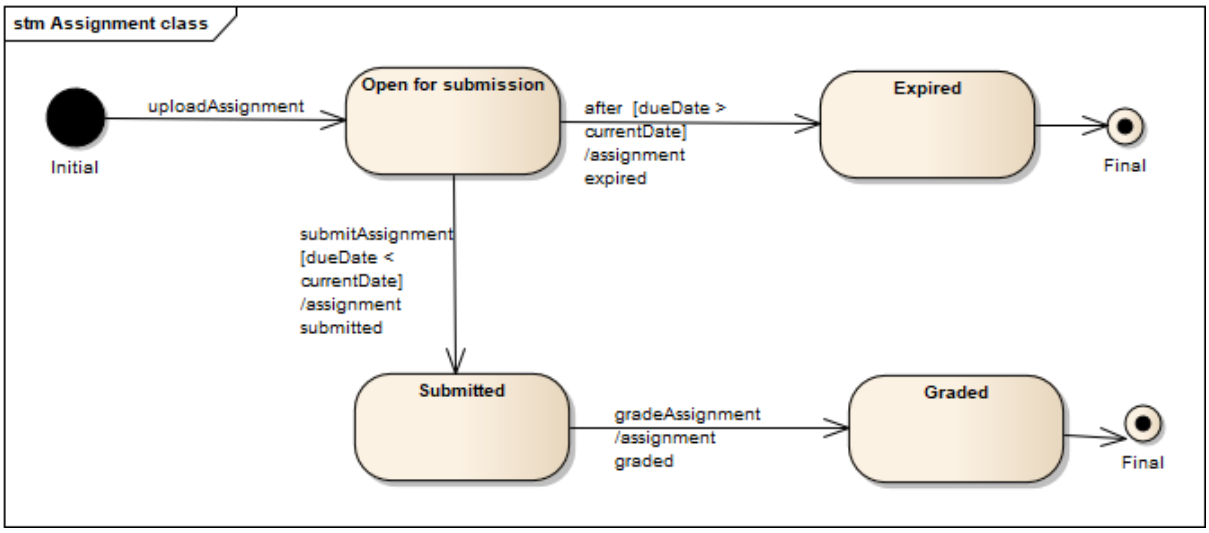

Figure 2.3: State Machine Diagram for Assignment class

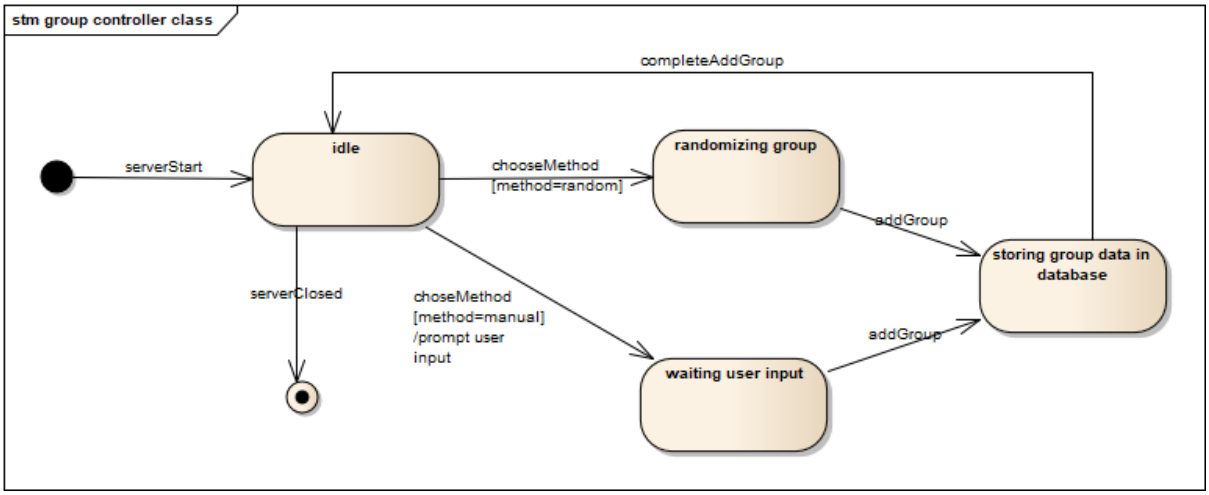

Figure 2.4: State Machine Diagram for Group controller class

3.2.1 UC001: Use Case Login using UTM ACID

<table>
<tr><th colspan="5">Use case: Login using UTM ACID</th></tr>
<tr><td rowspan="3">History Log:</td><td colspan="2"><ID></td><td colspan="2"><Changes></td></tr>
<tr><td colspan="2">1.0.0</td><td colspan="2">Create initial use case</td></tr>
<tr><td colspan="2">2.0.0</td><td colspan="2">Added exception flow</td></tr>
<tr><td>Version</td><td colspan="4">2.0.0</td></tr>
<tr><td>Created by:</td><td>Tan Chong Lim</td><td>Last update by:</td><td colspan="2">Tan Chong Lim</td></tr>
<tr><td>Date created</td><td>11/05/2020</td><td>Last Revision Date:</td><td colspan="2">7/6/2020</td></tr>
<tr><td colspan="5">ID: UC001</td></tr>
<tr><td colspan="5">Actors: Student, Lecturer, Course coordinator</td></tr>
<tr><td colspan="5">Preconditions:
1. User must exist in the database.</td></tr>
<tr><td colspan="5">Flow of events:
1. The system displays the login page.
2. Users enter his/her credential (Username and password).
3. Users click the login button.
4. If credential is wrong, E1 is performed, else, continue flow.
5. The system redirects the user to his/her homepage.</td></tr>
<tr><td colspan="5">Exception Flow E1:
1. System display failed login message
2. Return to normal flow 1</td></tr>
<tr><td colspan="5">Postconditions:
1. User has logged in.</td></tr>
<tr><td rowspan="3">Related Requirements(s):</td><td>ID</td><td colspan="2">Requirement</td><td>Priority</td></tr>
<tr><td>FR UC001-01</td><td colspan="2">During the login process, the system shall be able to verify the user credential with UTM ACID database.</td><td>Basic</td></tr>
<tr><td>CR UC001-012</td><td colspan="2">The system shall encrypt the user login information to prevent any data leak.</td><td>Performance</td></tr>
</table>

Table 2.1: Use Case Description for Login using UTM ACID

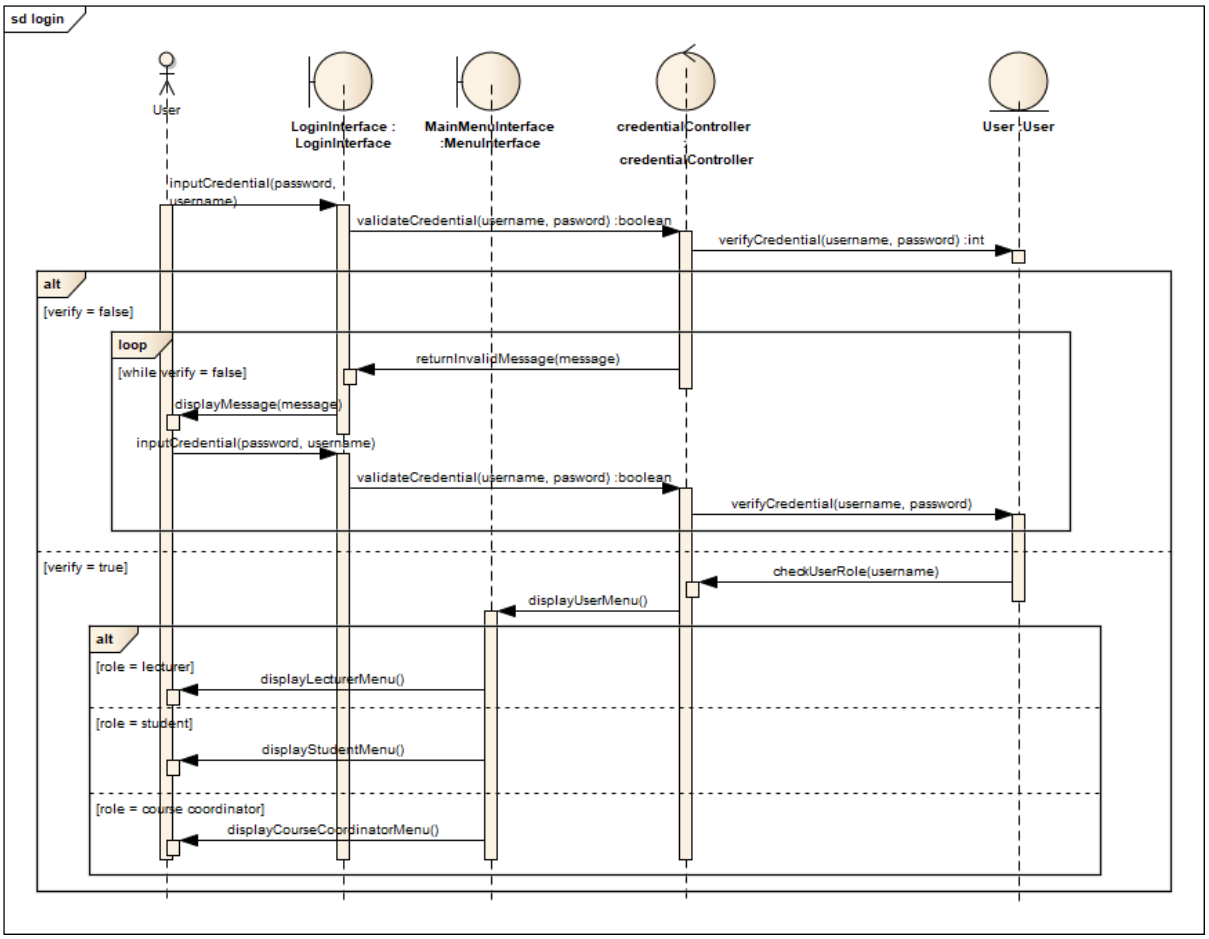

Figure 2.4: Sequence Diagram for Login using UTM ACID

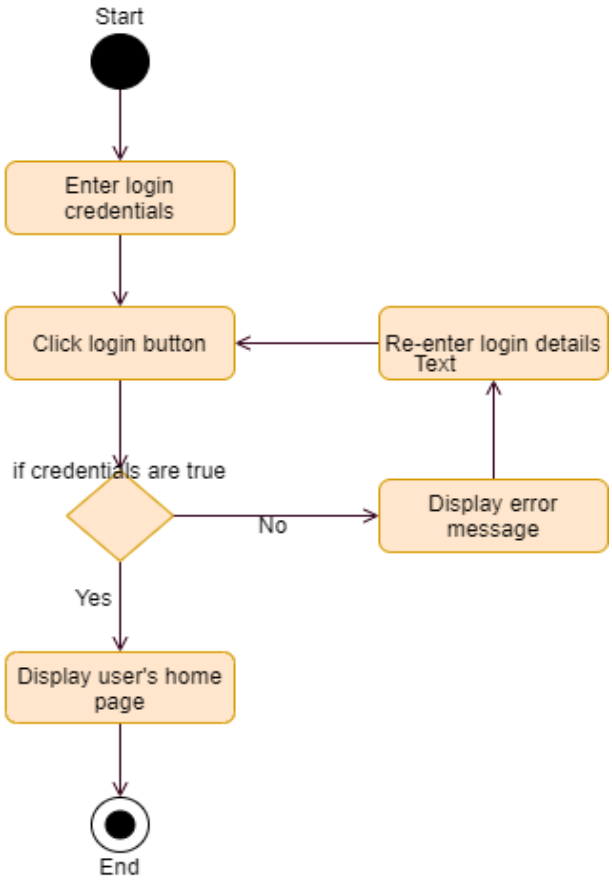

Figure 2.5: Activity Diagram for Login using UTM ACID

3.2.2 UC002: Use Case Upload Rubric

<table>
<tr><th colspan="4">Use case: Upload Rubric</th></tr>
<tr><td rowspan="3">History Log:</td><td><ID></td><td colspan="2"><Changes></td></tr>
<tr><td>1.0.0</td><td colspan="2">Create initial use case</td></tr>
<tr><td>1.1.0</td><td colspan="2">Improve the flow and exception</td></tr>
<tr><td>Version</td><td colspan="3">1.1.0</td></tr>
<tr><td>Created by:</td><td>Tan Jun Han</td><td>Last update by:</td><td>Tan Jun Han</td></tr>
<tr><td>Date created</td><td>11/05/2020</td><td>Last Revision Date:</td><td>12/6/2020</td></tr>
<tr><td colspan="4">ID: UC002</td></tr>
<tr><td colspan="4">Actors: Course coordinator</td></tr>
<tr><td colspan="4">Preconditions:
1. User must be logged in
2. Assignment must be created before
3. User selected an assignment.</td></tr>
<tr><td colspan="4">Flow of events:
1. If the user selects generate rubric, UC003 is performed. Else, continue the flow.
2. The course coordinator selects the upload icon.
3. The course coordinator select file from local device.
4. If file format is not supported or file size exceeded, E1 is performed.
5. Confirm upload.
6. System display upload status message.
7. Use case ends.</td></tr>
<tr><td colspan="4">Postconditions:
New rubric for the assignment is uploaded to the database together with the assignment.</td></tr>
<tr><td colspan="4">Exception E1:
1. prompt error message showing the types of error.
2. return to normal flow 2</td></tr>
<tr><td colspan="4">Related Requirement: UC003</td></tr>
<tr><td rowspan="5">Related Requirements(s):</td><td>ID</td><td colspan="2">Requirement</td><td>Priority</td></tr>
<tr><td>FR UC002-01</td><td colspan="2">The system shall provide course coordinator with the ability to upload rubric to the database.</td><td>Basic</td></tr>
<tr><td>FR UC002-02</td><td colspan="2">The system shall be able to attach the uploaded file together with the selected assignment.</td><td>Basic</td></tr>
<tr><td>FR UC002-03</td><td colspan="2">If a course coordinator chooses to generate rubric, the system shall provide template for the course coordinator to input the rubric details.</td><td>Performance</td></tr>
<tr><td>QR UC002-04</td><td colspan="2">The system shall check and verify every file before upload to the server</td><td>Performance</td></tr>
</table>

Table 2.2: Use Case Description for Upload rubric

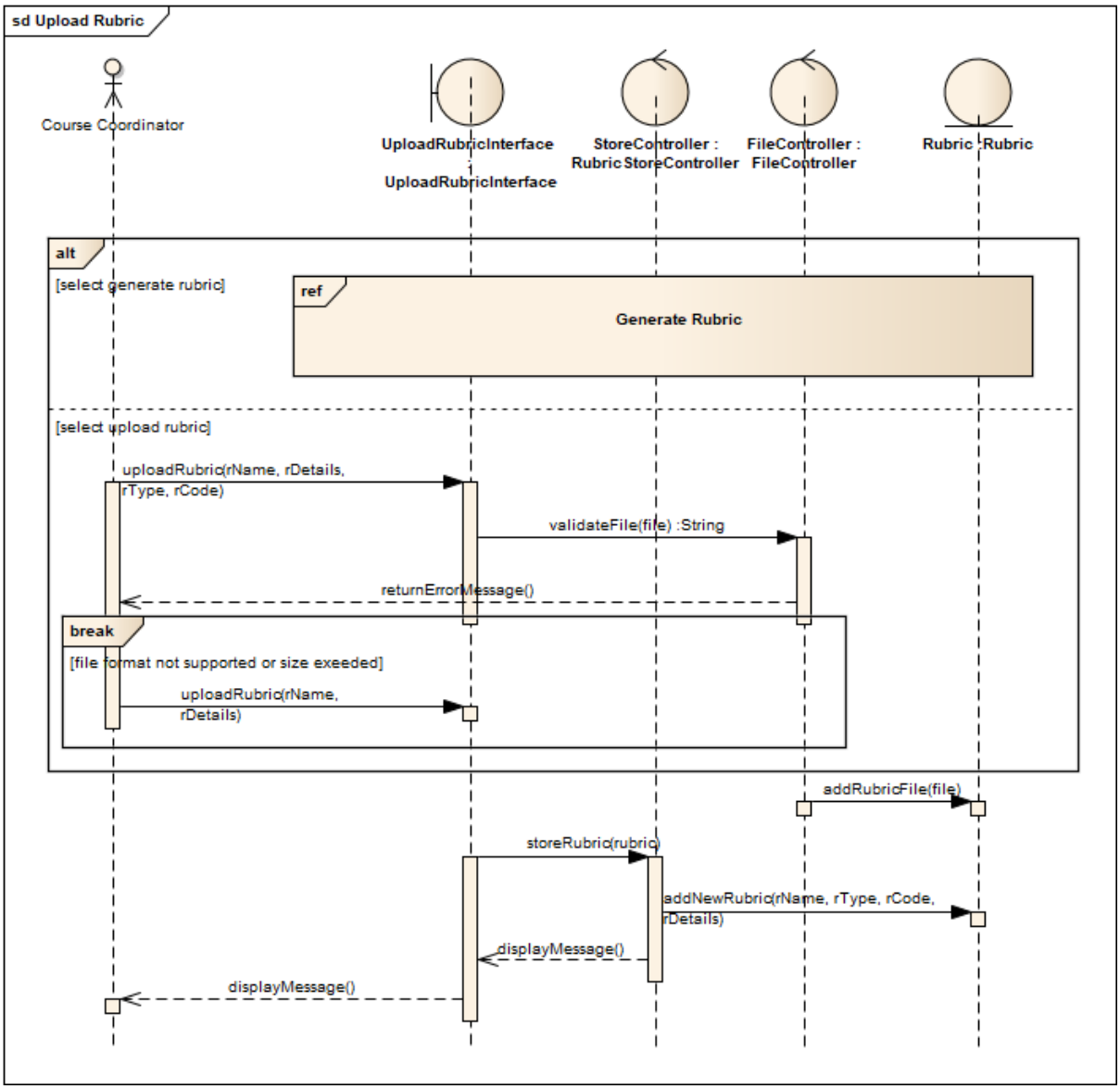

Figure 2.6: Sequence Diagram for Upload rubric

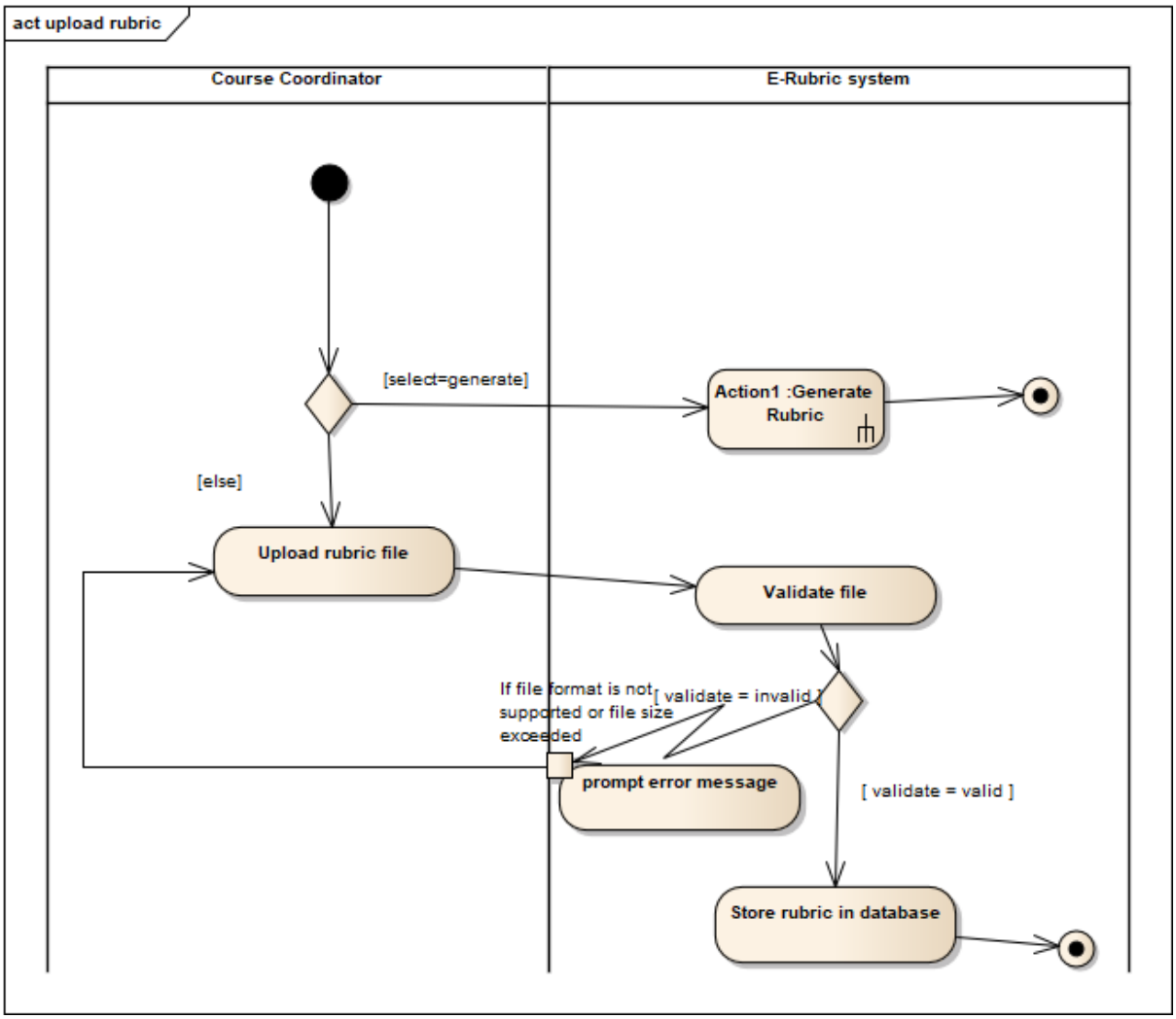

Figure 2.7: Activity Diagram for Upload rubric

3.2.3 UC003: Use Case Generate rubric

<table>
<tr><th colspan="5">Use case: Generate Rubric</th></tr>
<tr><td rowspan="3">History Log:</td><td colspan="2"><ID></td><td colspan="2"><Changes></td></tr>
<tr><td colspan="2">1.0.0</td><td colspan="2">Create initial use case</td></tr>
<tr><td colspan="2">1.1.0</td><td colspan="2">Added exception flow</td></tr>
<tr><td>Version</td><td colspan="4">1.1.0</td></tr>
<tr><td>Created by:</td><td colspan="2">Tan Jun Han</td><td>Last update by:</td><td>Tan Jun Han</td></tr>
<tr><td>Date created</td><td colspan="2">11/05/2020</td><td>Last Revision Date:</td><td>12/6/2020</td></tr>
<tr><td colspan="5">ID: UC003</td></tr>
<tr><td colspan="5">Actors: Course coordinator</td></tr>
<tr><td colspan="5">Preconditions:
1. The user must be logged in.
2. Assignment must be created before.
3. User selected an assignment.</td></tr>
<tr><td colspan="5">Flow of events:
1. The course coordinator selects Generate Rubric button.
2. The course coordinator chooses the rubric type from the drop-down menu.
3. The rubric template prompts for input.
4. The course coordinator input rubric details.
4.1 If course coordinator closes the page, E1 is performed.
5. The course coordinator click generate rubric
6. The use case ends.</td></tr>
<tr><td colspan="5">Postconditions:
New marking rubric is generated for the specified assignment.</td></tr>
<tr><td colspan="5">Exception E1:
1. The system saves the unfinish rubric as draft and will show in the drop-down menu.
2. Return to normal flow 1.</td></tr>
<tr><td colspan="5">Related Requirement: UC002</td></tr>
<tr><td rowspan="3">Related Requirements(s):</td><td>ID</td><td colspan="2">Requirement</td><td>Priority</td></tr>
<tr><td>FR UC003-01</td><td colspan="2">The system shall be able to generate marking rubric together with the selected assignment.</td><td>Basic</td></tr>
<tr><td>FR UC003-02</td><td colspan="2">The system shall be able to save the unfinish rubric before the user exits without any user input.</td><td>Excitement</td></tr>
</table>

Table 2.3: Use Case Description for Generate rubric

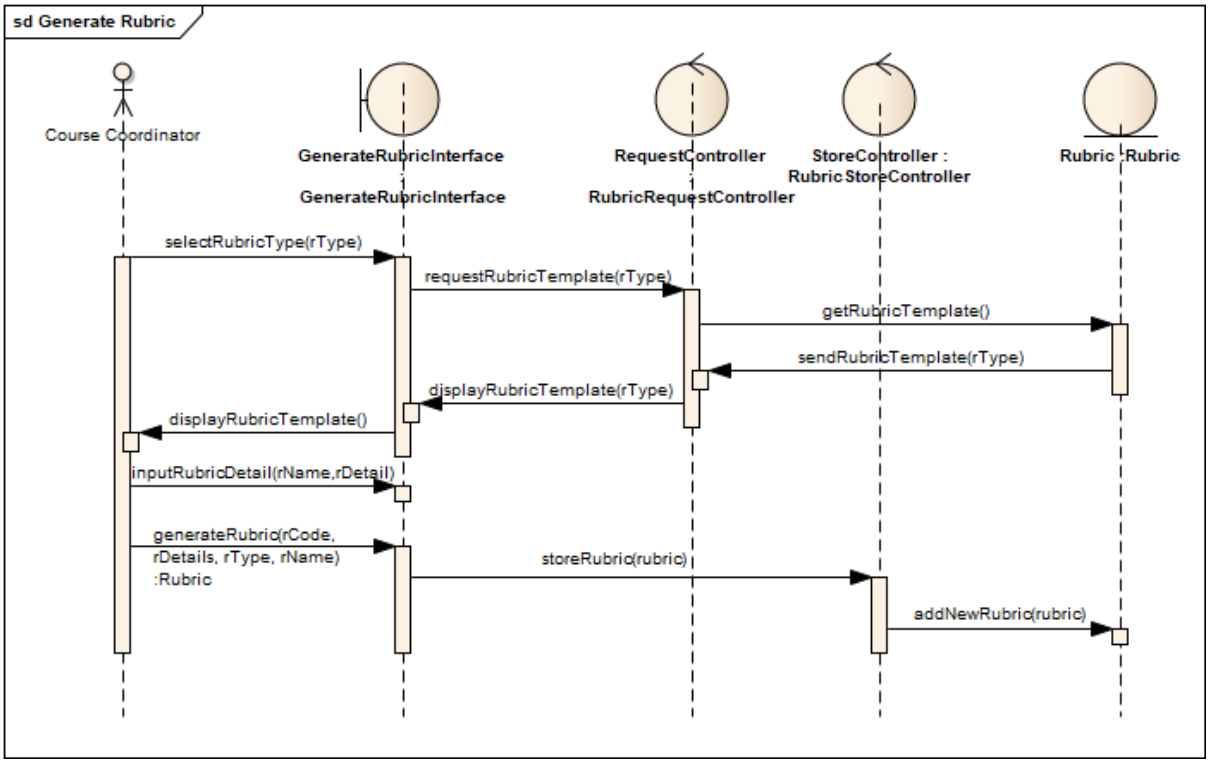

Figure 2.8: Sequence Diagram for Generate rubric

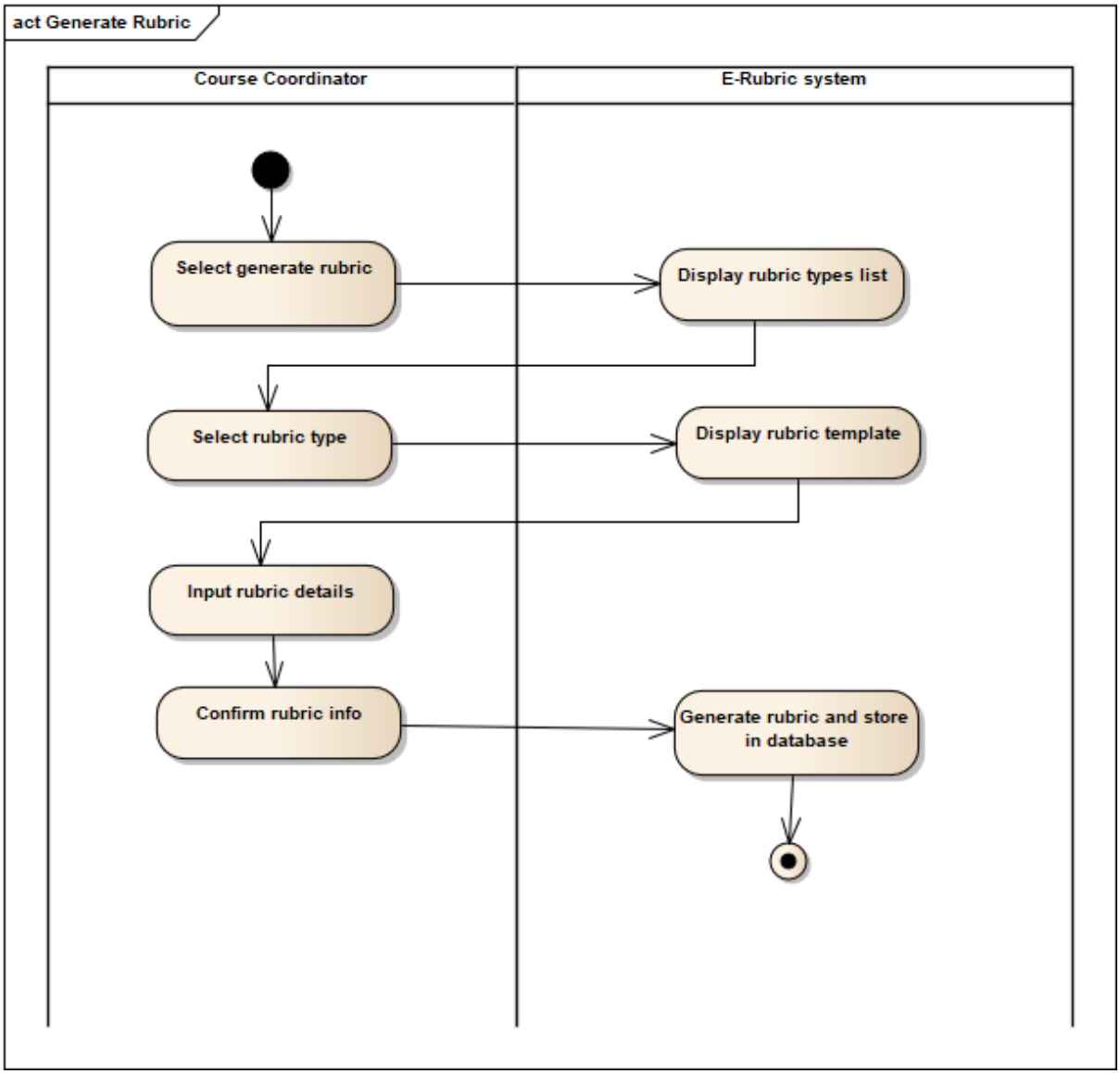

Figure 2.9: Activity Diagram for Generate rubric

3.2.4 UC004: Use Case Grade Assignment

Use case: Grade Assignment			
History Log:	<ID>	<Changes>	
	1.0.0	Create initial use case	
	1.0.1	Improved some grammar	
Version	1.0.1		
Created by:	Tan Chong Lim	**Last update by:**	Tan Chong Lim
Date created	11/05/2020	**Last Revision Date:**	7/6/2020
ID: UC004			
Actors: Lecturer			
Preconditions: 2. User must be logged in as lecturer. 3. There must existing assignment in database.			
Flow of events: 1. The lecturer select grade assignment. 2. The system display list of student assignment. 3. The lecturer select check plagiarism and grammar, UC005 is performed. 4. The system display list of student assignment report. 5. The lecturer select any student's assignment file from the list. 6. The system display the student's assignment information together with marking template 7. The lecturer review and give marks to the student. 8. If the lecturer select feedback, A1 is performed. 9. The use case ends.			
Postconditions: Assignment grade is assigned to the student.			
Alternative flow A1: 1. A feedback form is prompt for input 2. The lecturer input any feedback. 3. The use case ends.			
Postconditions: Feedback is given to the students.			
Related Requirement: UC005			

	ID	**Requirement**	**Priority**
Related Requirements(s):	**FR UC004-01**	After the assignment overdue, the system shall provide lecturer with the ability to give marks to students' assignments.	Basic
	FR UC004-02	The system shall provide lecturer with the ability to give feedback on student's assignments	Basic
	QR UC004-03	The system shall prevent user from	Basic

		submitting if user input is not a number	

Table 2.4: Use Case Description for Grade Assignment

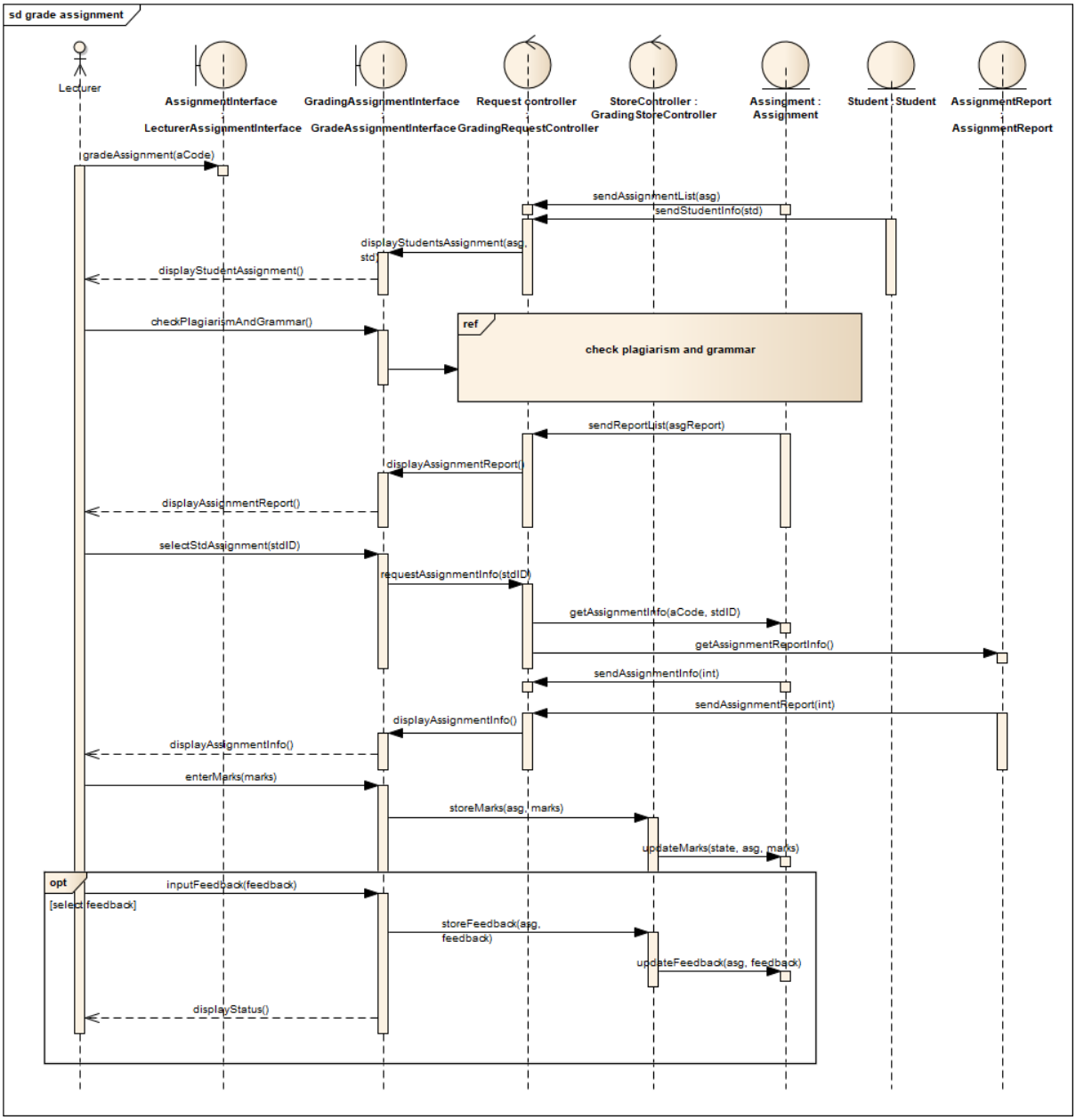

Figure 2.10: Sequence Diagram for <Grade Assignment >

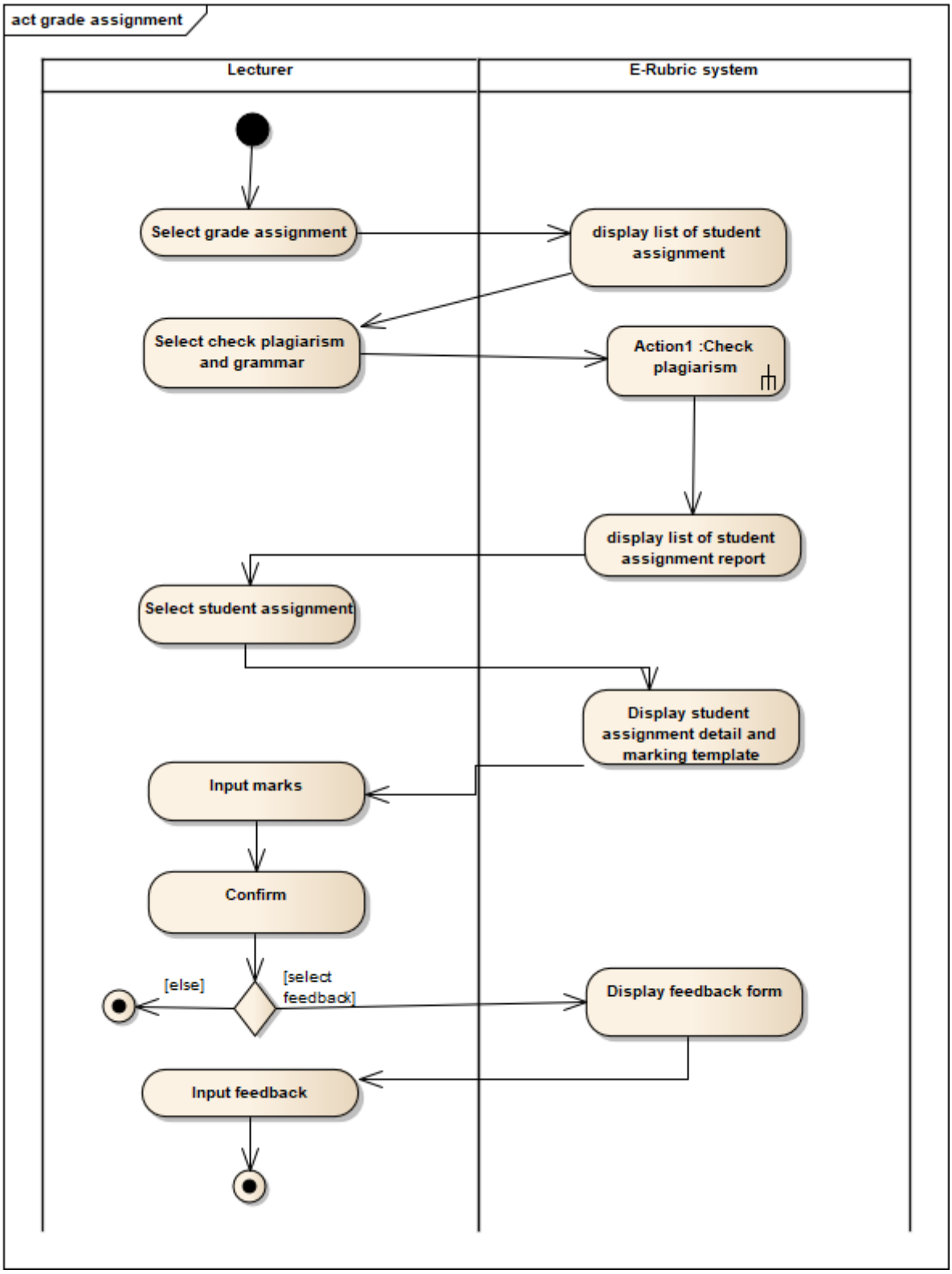

Figure 2.11: Activity Diagram for < Grade Assignment >

3.2.5 UC005: Use Case Check plagiarism and grammar

<table>
<tr><th colspan="4">Use case: Check plagiarism and grammar</th></tr>
<tr><td rowspan="3">History Log:</td><td><ID></td><td colspan="2"><Changes></td></tr>
<tr><td>1.0.0</td><td colspan="2">Create initial use case</td></tr>
<tr><td>1.1.0</td><td colspan="2">Improved the flow</td></tr>
<tr><td>Version</td><td colspan="3">1.1.0</td></tr>
<tr><td>Created by:</td><td>Tan Chong Lim</td><td>Last update by:</td><td>Tan Chong Lim</td></tr>
<tr><td>Date created</td><td>11/05/2020</td><td>Last Revision Date:</td><td>7/6/2020</td></tr>
<tr><td colspan="4">ID: UC005</td></tr>
<tr><td colspan="4">Actors: Lecturer, Turnitin, Grammarly</td></tr>
<tr><td colspan="4">Preconditions:
4. User must be logged in as lecturer.
5. There must existing assignment in database.</td></tr>
<tr><td colspan="4">Flow of events:
1. The lecturer select check plagiarism and grammar.
2. Turnitin API receive students' assignment and generate report for it.
2.1 If error occur at Turnitin server, E1 is performed.
3. Grammarly receive students' assignment and generate report for it.
2.1 If error occur at Grammarly server, E1 is performed.
4. The lecturer review the generated report on the page.</td></tr>
<tr><td colspan="4">Postconditions:
Plagiarism & grammar report file is generated.</td></tr>
<tr><td colspan="4">Exception E1 :
1. display error message
2. return back to grade assignment page</td></tr>
<tr><td colspan="4">Related Requirement: UC004</td></tr>
<tr><td rowspan="2">Related Requirements(s):</td><td>ID</td><td>Requirement</td><td>Priority</td></tr>
<tr><td>FR UC005-01</td><td>When the lecturer chooses check plagiarism and grammar, the system shall be able to connect to external API of Turnitin and Grammarly to get the result.</td><td>Excitement</td></tr>
</table>

Table 2.5: Use Case Description for Check plagiarism and grammar

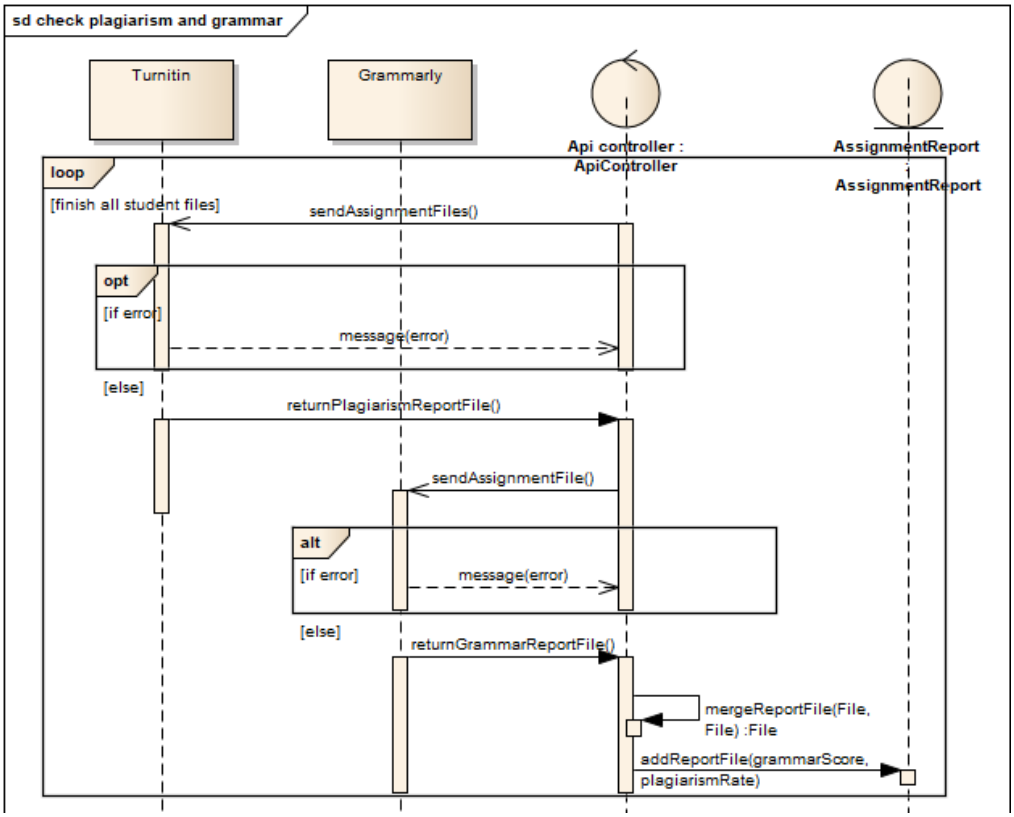

Figure 2.12: Sequence Diagram for Check plagiarism and grammar

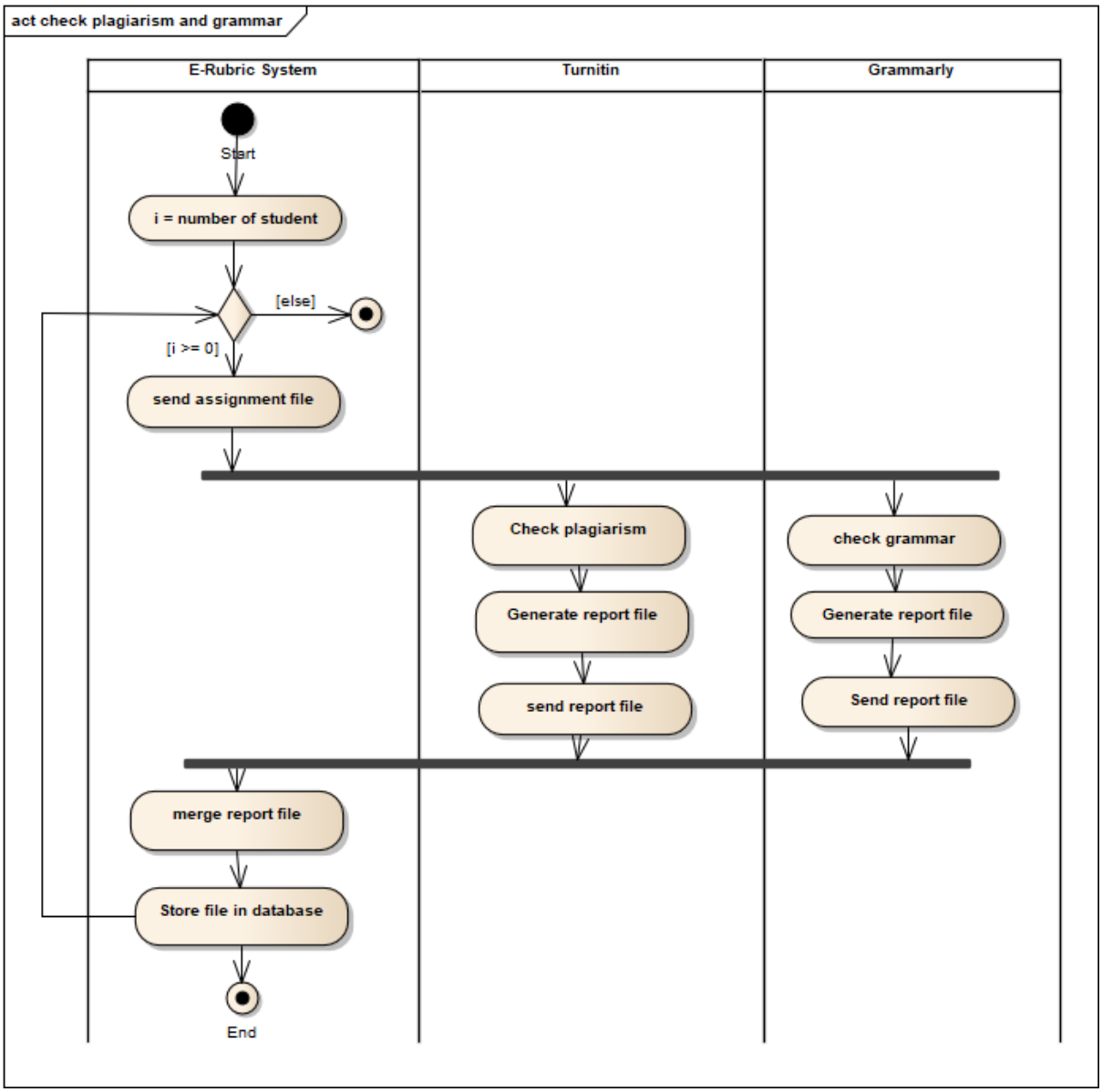

Figure 2.13: Activity Diagram for Check plagiarism and grammar

3.2.6 UC006: Use Case Upload course material

<table>
<tr><th colspan="5">Use case: Upload course material</th></tr>
<tr><td rowspan="2">History Log</td><td colspan="2"><ID></td><td colspan="2"><Changes></td></tr>
<tr><td colspan="2">1.0.0</td><td colspan="2">Create Initial use case</td></tr>
<tr><td>Version</td><td colspan="4">1.0.0</td></tr>
<tr><td>Created by:</td><td colspan="2">Nurul Ismat Tanni</td><td colspan="2">Last Update by: Nurul Ismat Tanni</td></tr>
<tr><td>Date created:</td><td colspan="2">13/5/2020</td><td colspan="2">Last Revision Date: 11/6/2020</td></tr>
<tr><td colspan="5">ID: UC006</td></tr>
<tr><td colspan="5">Actors: Course coordinator, Lecturer</td></tr>
<tr><td colspan="5">Preconditions: User must be logged in</td></tr>
<tr><td colspan="5">Flow of events:
1. The user select upload course material.
2. The system display upload interface.
3. Choose file to upload from local device.
4. If file format is not supported or file size exceeded, E1 is performed.
5. The file is stored in database.
6. The use case ends.</td></tr>
<tr><td colspan="5">Exception E1:
1. prompt error message
2. return to normal flow 3</td></tr>
<tr><td colspan="5">Postconditions:
Course material is uploaded to the database.
The students from the course receive notification about new uploads.</td></tr>
<tr><td colspan="5">Postconditions: Course material is uploaded to database</td></tr>
<tr><td rowspan="3">Related Requirements</td><td>ID</td><td colspan="2">Requirements</td><td>Priority</td></tr>
<tr><td>FR UC006-01</td><td colspan="2">The system shall provide lecturer and course coordinator with the ability to upload course material.</td><td>Basic</td></tr>
<tr><td>FR UC006-02</td><td colspan="2">The system shall check and verify every file before upload to the server</td><td>Performance</td></tr>
</table>

Table 2.6: Use Case Description for Upload course material

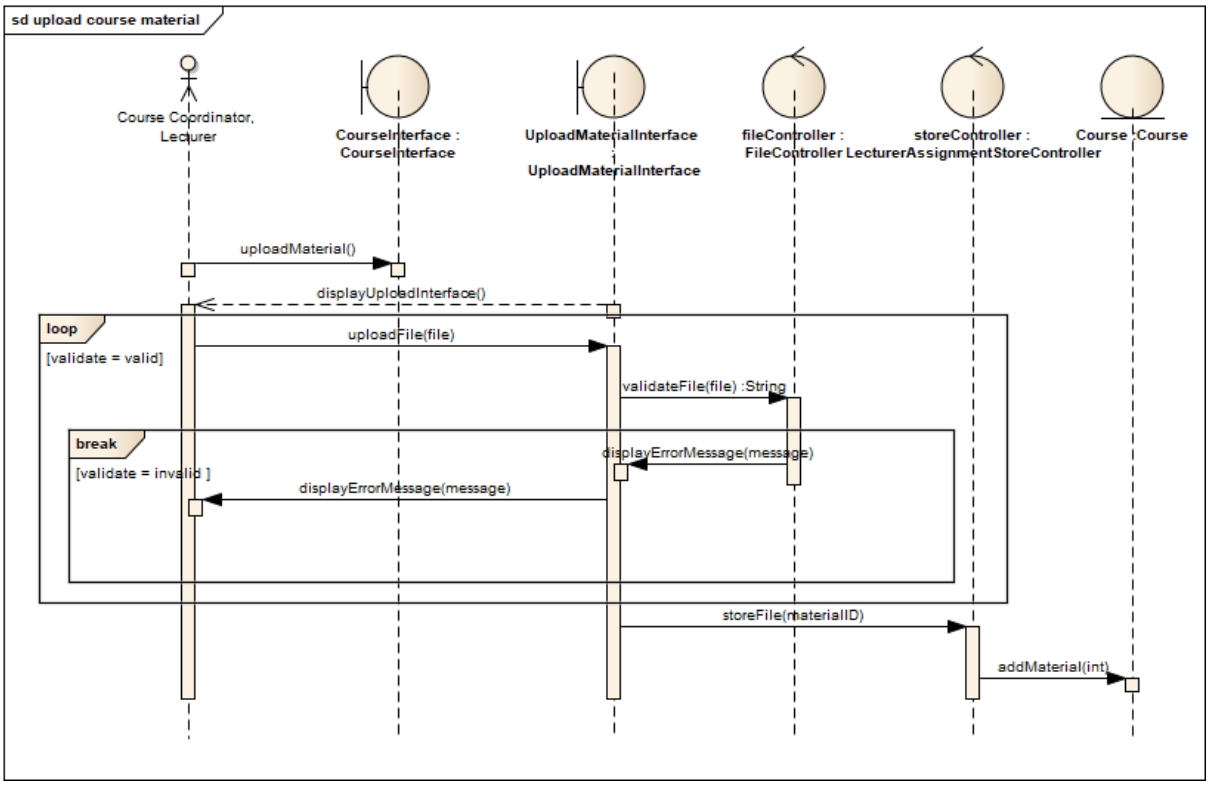

Figure 2.14: Sequence Diagram for Upload course material

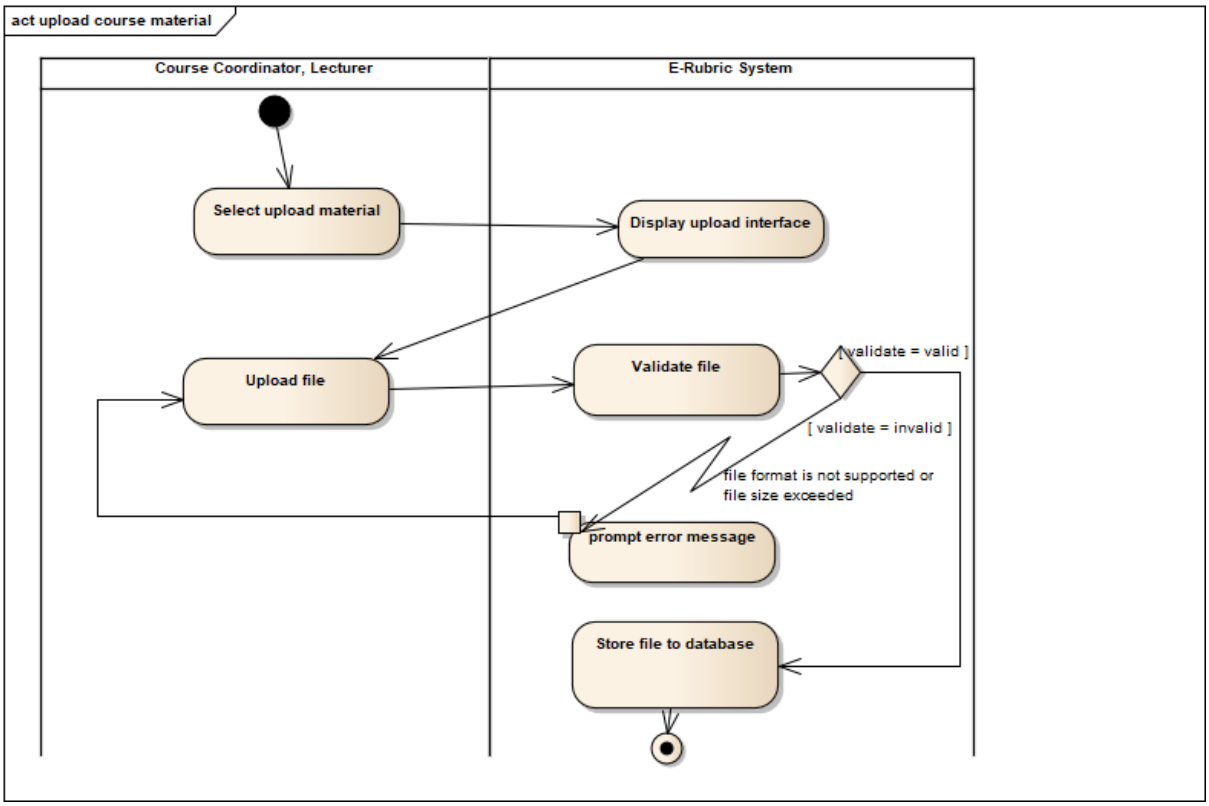

Figure 2.15: Activity Diagram for Upload course material

3.2.7 UC007: Use Case View students' performance report

<table>
<tr><th colspan="5">Use case: View students' performance report</th></tr>
<tr><td rowspan="3">History Log:</td><td colspan="2"><ID></td><td colspan="2"><Changes></td></tr>
<tr><td colspan="2">1.0.0</td><td colspan="2">Create initial use case</td></tr>
<tr><td colspan="2">1.1.0</td><td colspan="2">Improved the flow</td></tr>
<tr><td>Version</td><td colspan="4">1.1.0</td></tr>
<tr><td>Created by:</td><td colspan="2">Tan Chong Lim</td><td>Last update by:</td><td>Tan Chong Lim</td></tr>
<tr><td>Date created</td><td colspan="2">11/05/2020</td><td>Last Revision Date:</td><td>7/6/2020</td></tr>
<tr><td colspan="5">ID: UC007</td></tr>
<tr><td colspan="5">Actors: Lecturer</td></tr>
<tr><td colspan="5">Preconditions:
1. User must be logged in as lecturer.
2. There must existing assignment in database.
3. The assignment must be graded.</td></tr>
<tr><td colspan="5">Flow of events:
1. The lecturer select view students' performance report.
2. The lecturer select type of report
2.1 If type is table, display report in table
2.2 if type is chart, display report in chart
3. Use case ends.</td></tr>
<tr><td rowspan="2">Related Requirements(s):</td><td>ID</td><td colspan="2">Requirement</td><td>Priority</td></tr>
<tr><td>FR UC007-01</td><td colspan="2">After the assignment graded, the system shall provide lecturer with the ability to generate student performance report.</td><td>Basic</td></tr>
</table>

Table 2.7: Use Case Description for View students' performance report

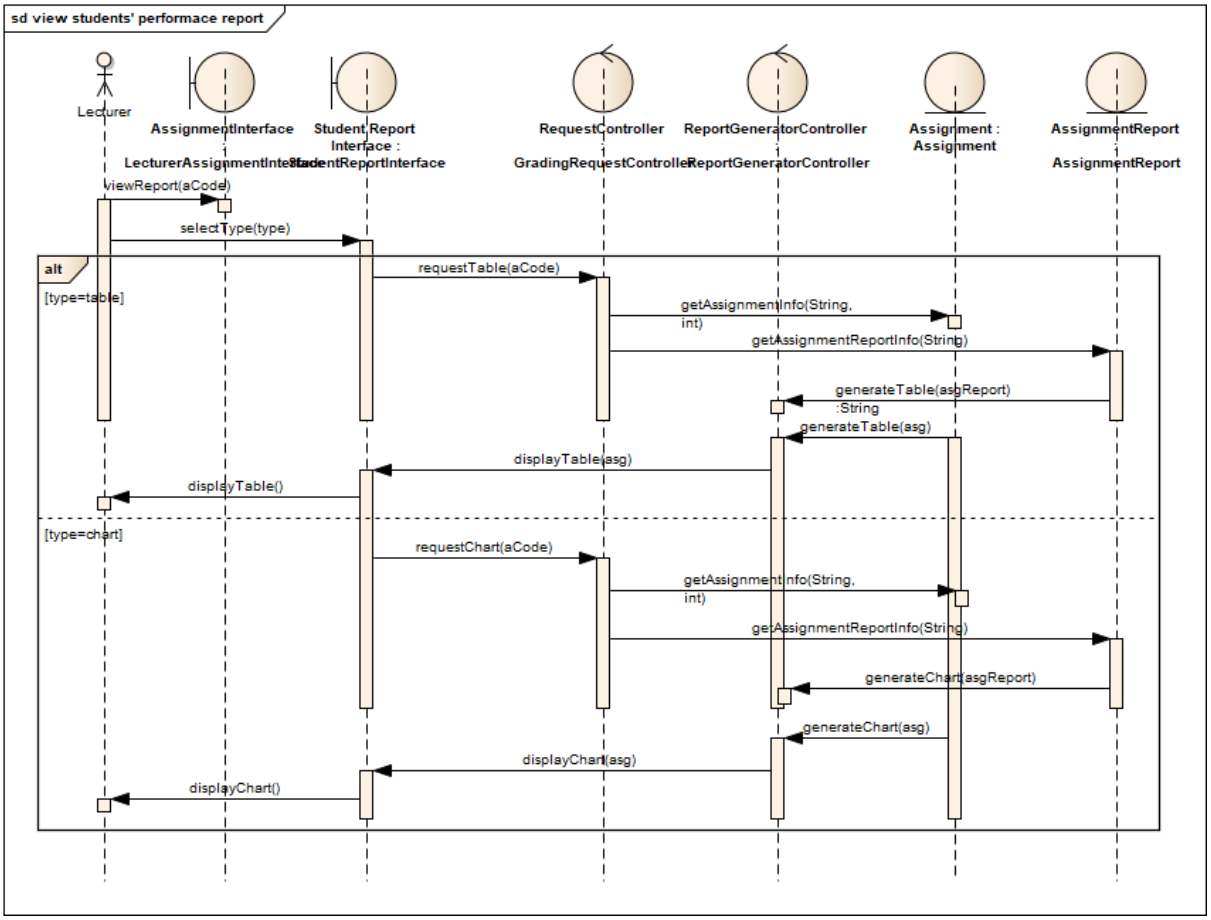

Figure 2.16: Sequence Diagram for View students' performance report

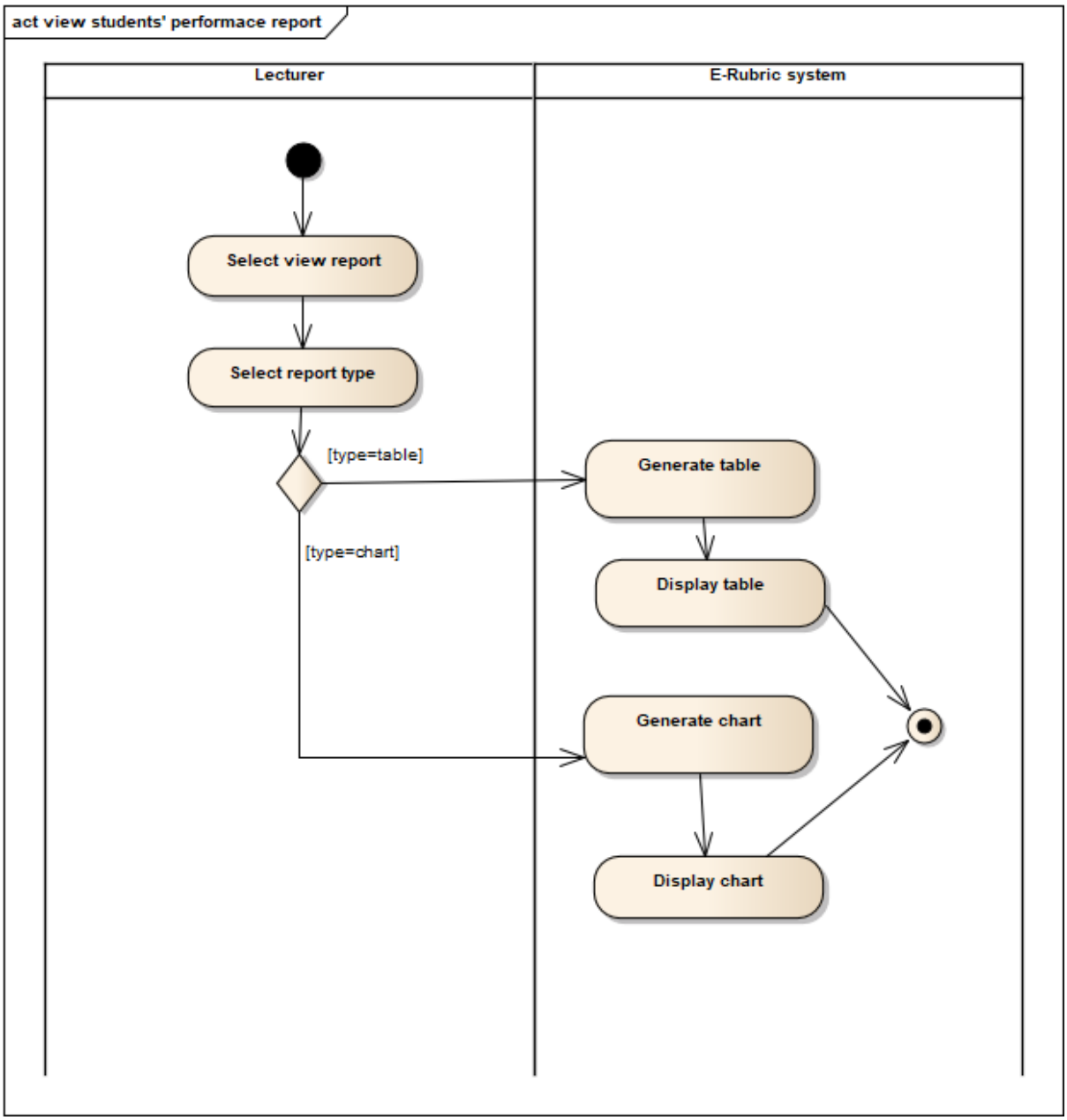

Figure 2.17: Activity Diagram for View students' performance report

3.2.8 UC008: Use Case Assign group

<table>
<tr><th colspan="5">Use case: Assign group</th></tr>
<tr><td rowspan="2">History Log</td><td><ID></td><td colspan="3"><Changes></td></tr>
<tr><td>1.0.0</td><td colspan="3">Create Initial Use case</td></tr>
<tr><td>Version</td><td colspan="4">1.0.0</td></tr>
<tr><td>Created by</td><td>Nurul Ismat Tanni</td><td>Last Update By</td><td colspan="2">Nurul Ismat Tanni</td></tr>
<tr><td>Date created</td><td>13/5/2020</td><td>Last Revision Date</td><td colspan="2">11/6/2020</td></tr>
<tr><td colspan="5">ID: UC008</td></tr>
<tr><td colspan="5">Actors: Lecturer</td></tr>
<tr><td colspan="5">Preconditions:
User must be logged in
There must be assignment in the database.</td></tr>
<tr><td colspan="5">Flow of events:
1. The lecturer select assign group
2. The lecturer choose method of assign.
2.1 If choose random, system randomize the selection of group members.
2.2 If choose manual, A1 is performed.
3. Use case ends.</td></tr>
<tr><td colspan="5">Postconditions: Students are assigned to group.</td></tr>
<tr><td colspan="5">Alternative flow A1:
1. The system display student list.
2. The lecturer assign all the students to their group respectively.
3. The use case ends.</td></tr>
<tr><td colspan="5">Post conditions: Students are assigned to group.</td></tr>
<tr><td rowspan="3">Related Requirements</td><td>ID</td><td colspan="2">Requirement</td><td>Priority</td></tr>
<tr><td>FR UC008-01</td><td colspan="2">The system shall provide lecturer with the ability to assign student to different groups.</td><td>Basic</td></tr>
<tr><td>FR UC008-02</td><td colspan="2">When requested by the lecturer, the system shall assign the students to different group in random.</td><td>Performance</td></tr>
</table>

Table 2.8: Use Case Description for Assign group

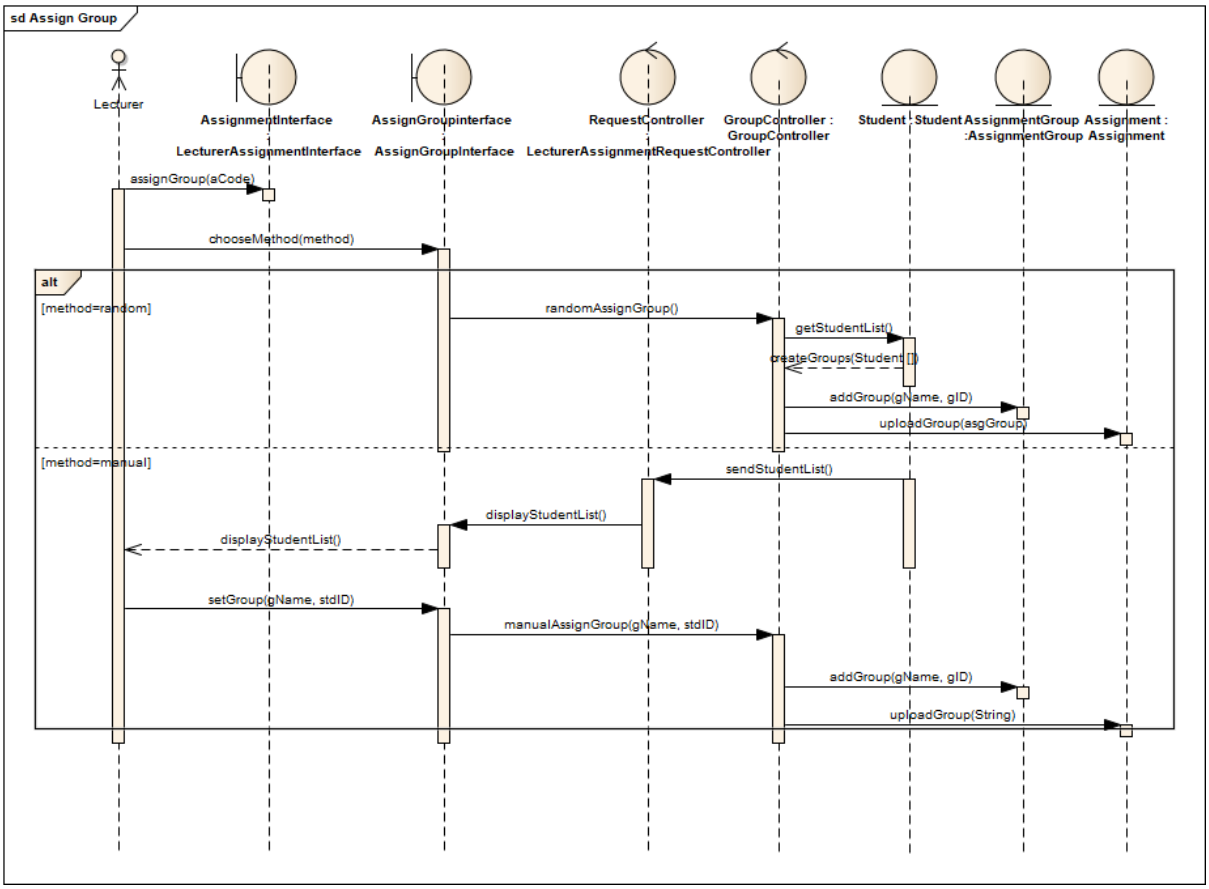

Figure 2.18: Sequence Diagram for Assign group

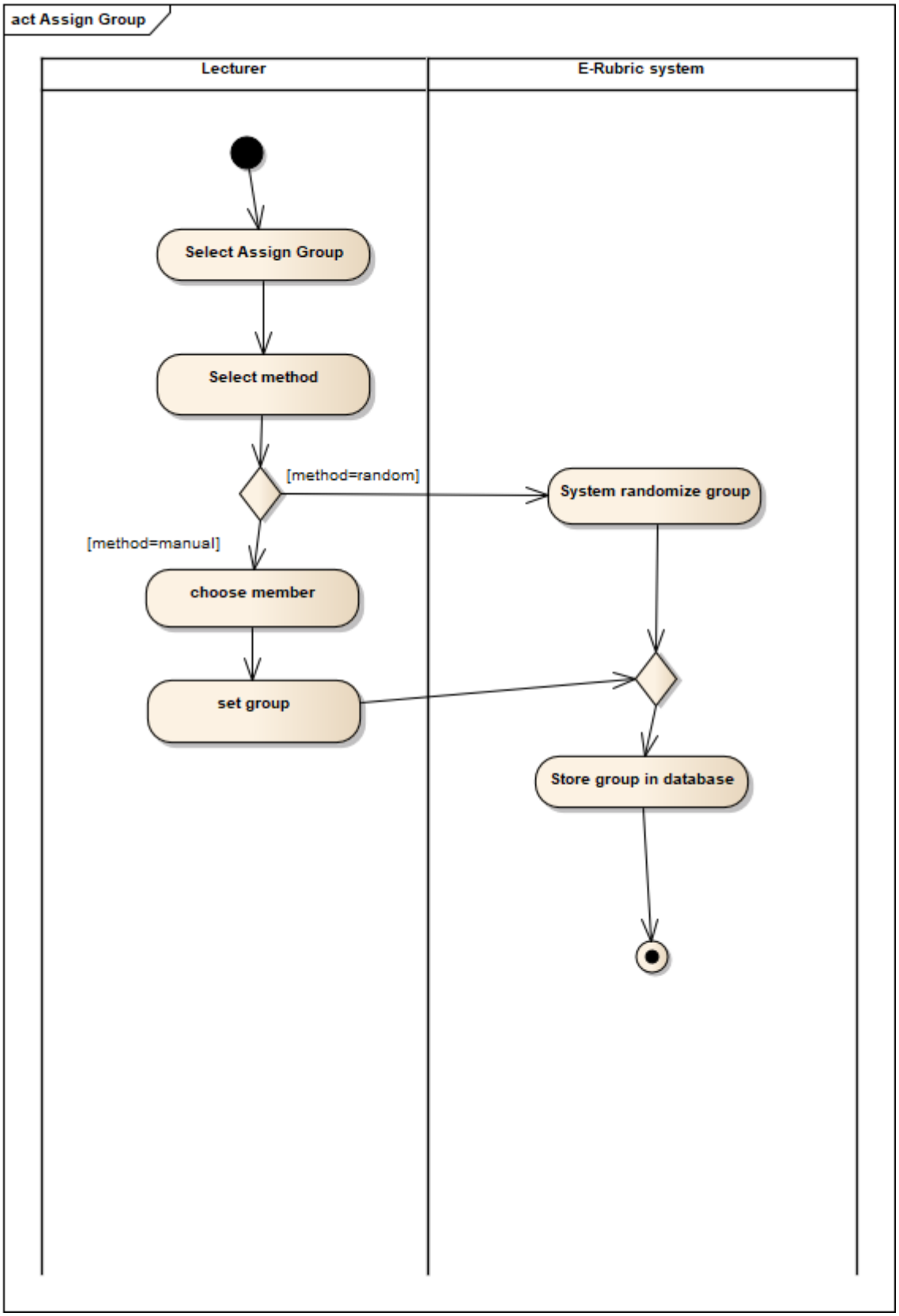

Figure 2.19: Activity Diagram for Assign group

3.2.9 UC009: Use Case Download course material

<table>
<tr><th colspan="4">Use case: Download Course Material</th></tr>
<tr><td rowspan="2">History Log:</td><td><ID></td><td colspan="2"><Changes></td></tr>
<tr><td>1.0</td><td colspan="2">Create initial use case</td></tr>
<tr><td>Version</td><td colspan="3">1.1.0</td></tr>
<tr><td>Created by:</td><td>Nurul Aneesha</td><td>Last update by:</td><td>Nurul Aneesha</td></tr>
<tr><td>Date created</td><td>11/05/2020</td><td>Last Revision Date:</td><td>10/06/2020</td></tr>
<tr><td colspan="4">ID: UC009</td></tr>
<tr><td colspan="4">Actors:Student</td></tr>
<tr><td colspan="4">Preconditions:
1. Student is logged on the system
2. Lecturer already uploaded the course Material</td></tr>
<tr><td colspan="4">Flow of Events:
1.Student select a course
2.The student select course material from the list
3. The student select download
4. The student wait for download to be done.
5. Use case end</td></tr>
<tr><td colspan="4">Postconditions:
1. course material file is downloaded by the student</td></tr>
<tr><td rowspan="3">Related Requirements(s):</td><td>ID</td><td>Requirement</td><td>Priority</td></tr>
<tr><td>FR UC009-01</td><td>The system should provide the ability for student to download material</td><td>Basic</td></tr>
<tr><td>FR UC009-02</td><td>The system should be able to print message for successful download</td><td>Performance</td></tr>
</table>

Table 2.9: Use Case Description for Download course material

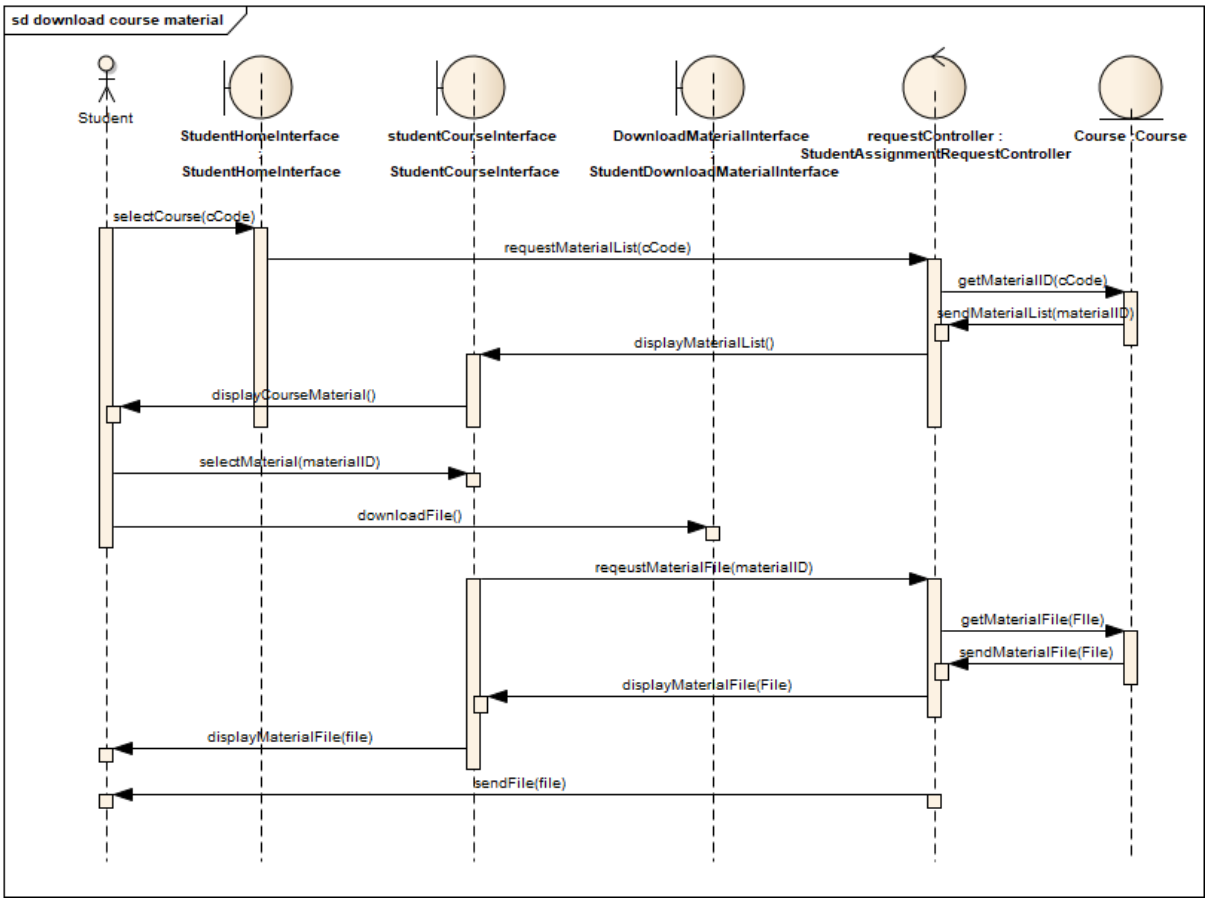

Figure 2.20: Sequence Diagram for Download course material

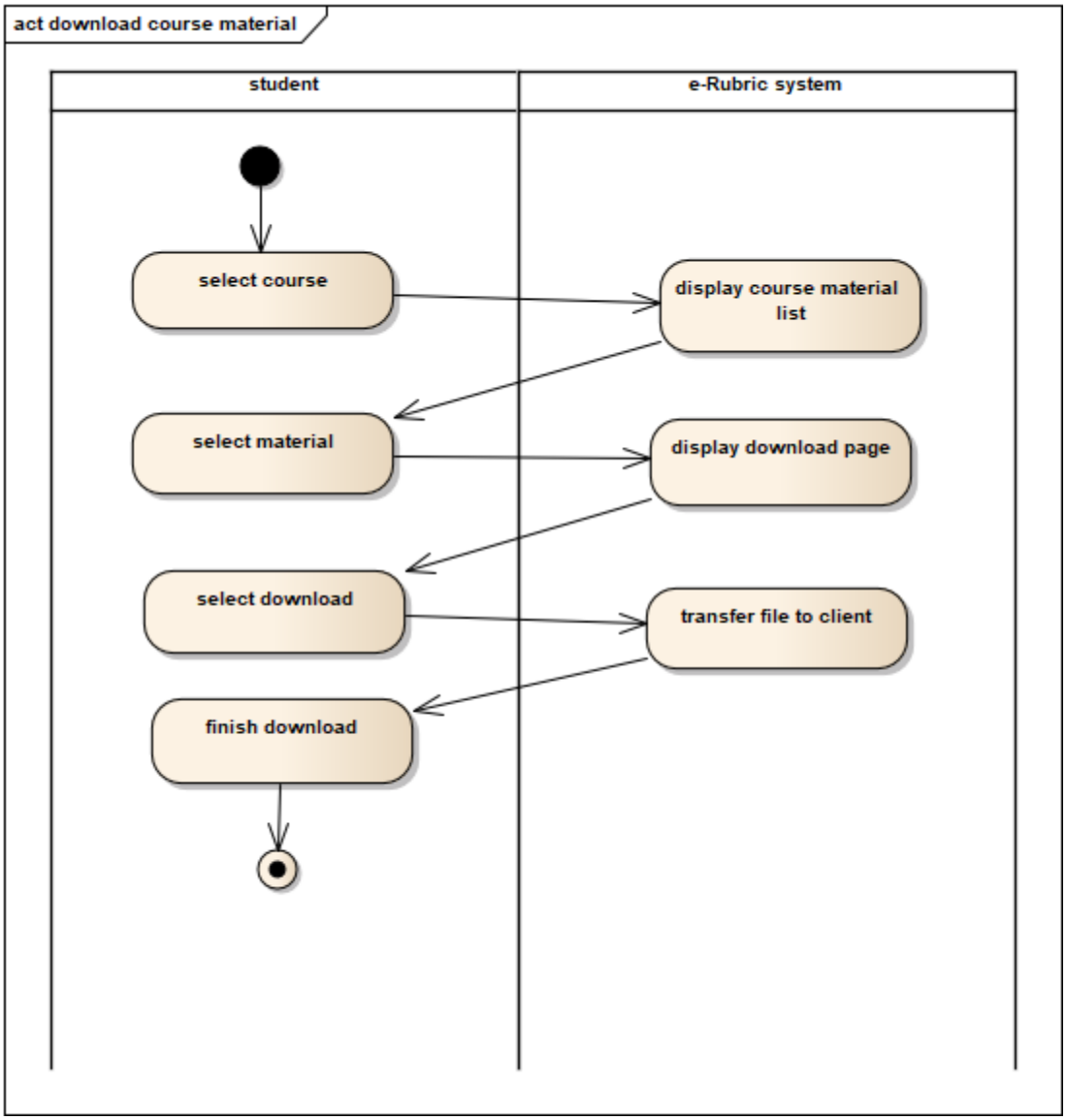

Figure 2.21: Activity Diagram for Download course material

3.2.10 UC010: Use Case View grading

<table>
<tr><td colspan="4">Use case:View Grading</td></tr>
<tr><td rowspan="3">History Log:</td><td colspan="2"><ID></td><td><Changes></td></tr>
<tr><td colspan="2">1.0</td><td>Create initial use case</td></tr>
<tr><td colspan="2">1.1.0</td><td>Improve the flow</td></tr>
<tr><td>Version</td><td colspan="3">1.1.0</td></tr>
<tr><td>Created by:</td><td>Nurul Aneesha</td><td>Last update by:</td><td>Nurul Aneesha</td></tr>
<tr><td>Date created</td><td>11/05/2020</td><td>Last Revision Date:</td><td>10/06/2020</td></tr>
<tr><td colspan="4">ID: UC010</td></tr>
<tr><td colspan="4">Actors:Student</td></tr>
<tr><td colspan="4">Preconditions:
1.Student is logged on the system
2.Uploaded assignment in the system
3.Assignment graded by lecturer</td></tr>
<tr><td colspan="4">Flow of Events:
1.The system display the list of registered course
2.Select on course
3.The system display list of assignment that has been graded by the lecturer
4. Student click on assignment that they want to view
5.The system display the grading and feedback from lecturer</td></tr>
<tr><td colspan="4">Alternative flow:
1. No grade information available</td></tr>
<tr><td rowspan="2">Related Requirements(s):</td><td>ID</td><td>Requirement</td><td>Priority</td></tr>
<tr><td>FR UC010-01</td><td>The system should be able to display grades and feedback for each assignment and course to the student.</td><td>Basic</td></tr>
</table>

Table 2.10: Use Case Description for View grading

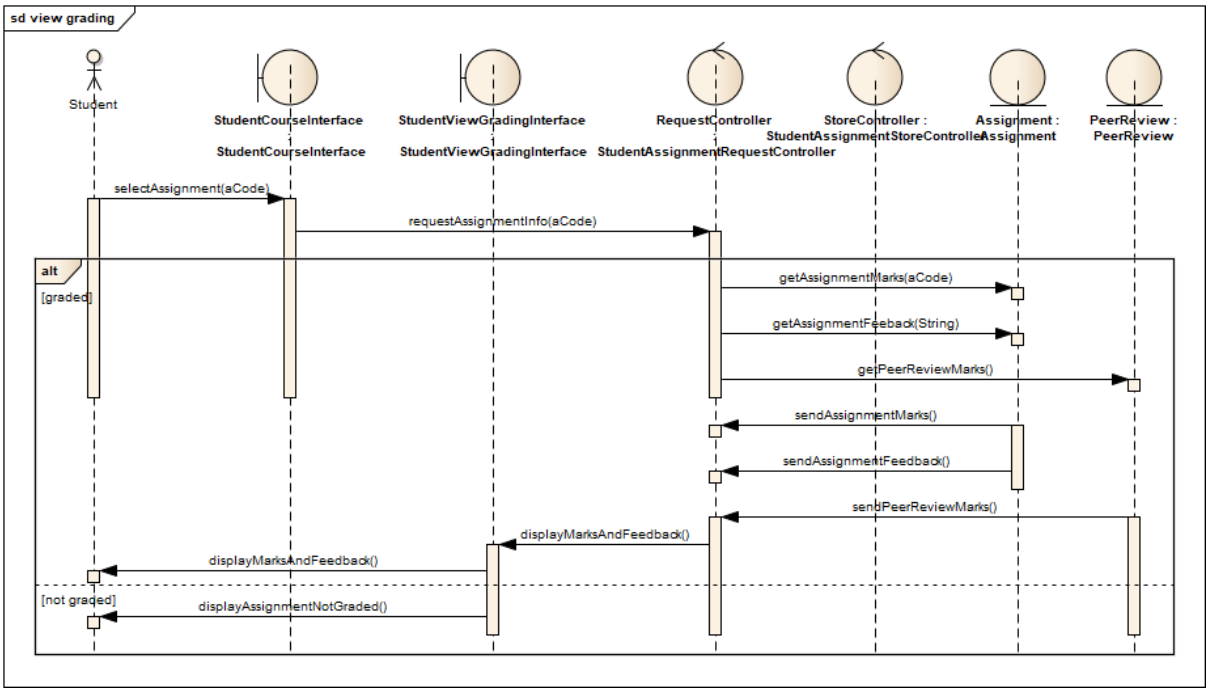

Figure 2.22: Sequence Diagram for View grading

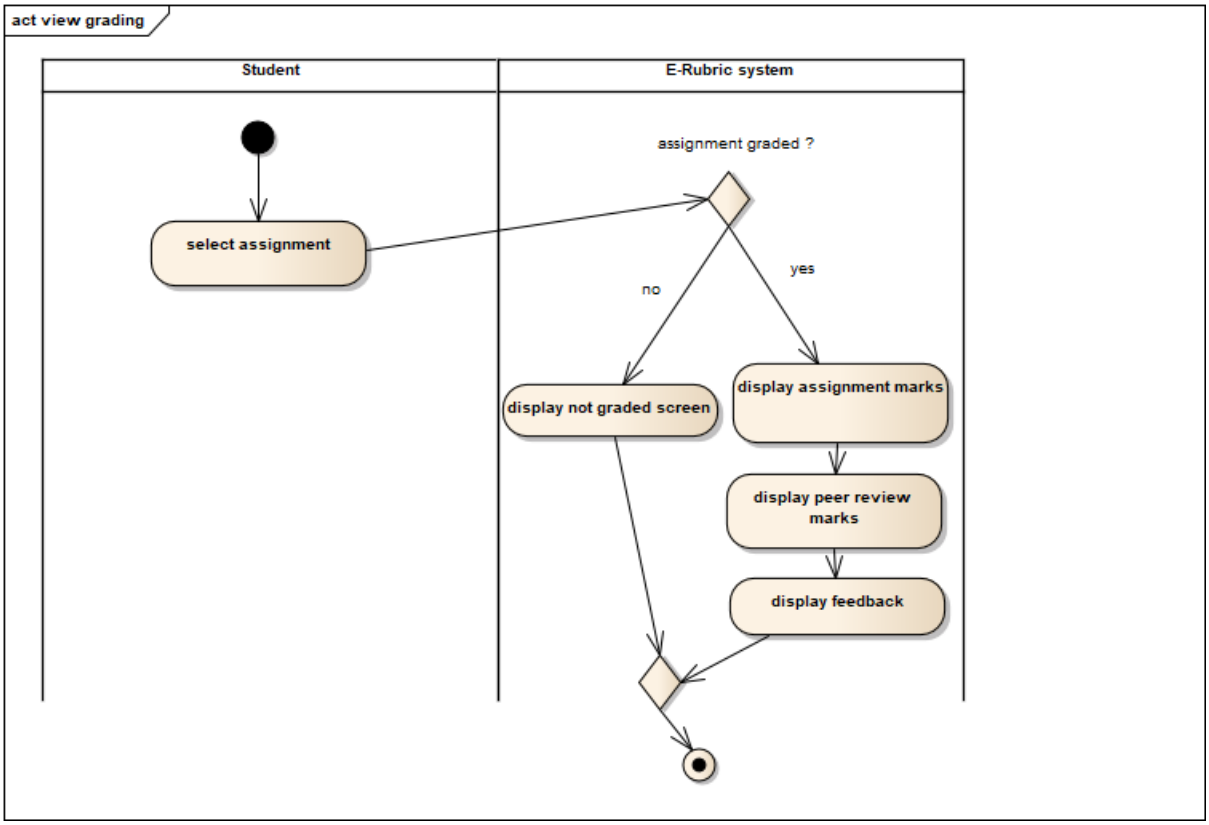

Figure 2.23: Activity Diagram for View grading

3.2.11 UC011: Use Case Upload Assignment

<table>
<tr><th colspan="4">Use case: Upload Assignment</th></tr>
<tr><td rowspan="2">History Log:</td><td colspan="2"><ID></td><td><Changes></td></tr>
<tr><td colspan="2">1.0</td><td>Create initial use case</td></tr>
<tr><td>Version</td><td colspan="3">1.0</td></tr>
<tr><td>Created by:</td><td>Nurul Aneesha</td><td>Last update by:</td><td>Nurul Aneesha</td></tr>
<tr><td>Date created</td><td>11/05/2020</td><td>Last Revision Date:</td><td>10/06/2020</td></tr>
<tr><td colspan="4">ID: UC011</td></tr>
<tr><td colspan="4">Actors:Student</td></tr>
<tr><td colspan="4">Preconditions:
1. Student has logged in to his/her account.</td></tr>
<tr><td colspan="4">Flow of Events:
1. The student select assignment
2. The system redirects the Student to the upload page.
3. The Student upload the assignment file and select “Submit”.
3.1 if file format is not supported or file size exceeded limit, E1 is performed.
4. The system displays a submission completed message.
5. If the Student click on the “Peer Review” button, A1 is performed. Else, continue flow.
6. The use case ends.</td></tr>
<tr><td colspan="4">Alternative flow:
2. System display peer review page
3. The student enter peer review marks for his teammates.</td></tr>
<tr><td colspan="4">Postconditions:
1. Student had submitted the assignment.
2. Student has peer review his teammates</td></tr>
<tr><td rowspan="2">Related Requirements(s):</td><td>ID</td><td>Requirement</td><td>Priority</td></tr>
<tr><td>FR UC011-01</td><td>The system should be able to provide the ability to upload document to student</td><td>Basic</td></tr>
</table>

Table 2.11: Use Case Description for Upload assignment

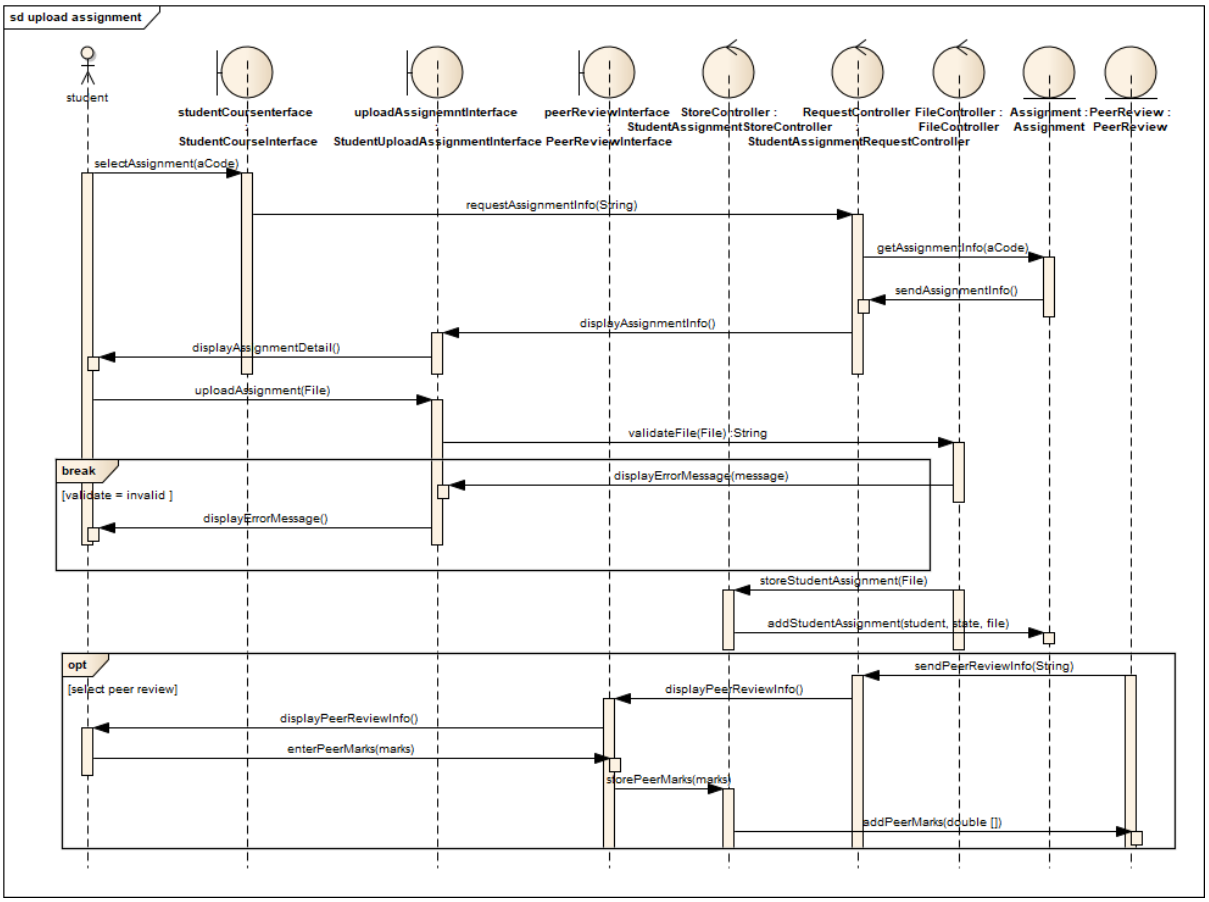

Figure 2.24: Sequence Diagram for Upload assignment

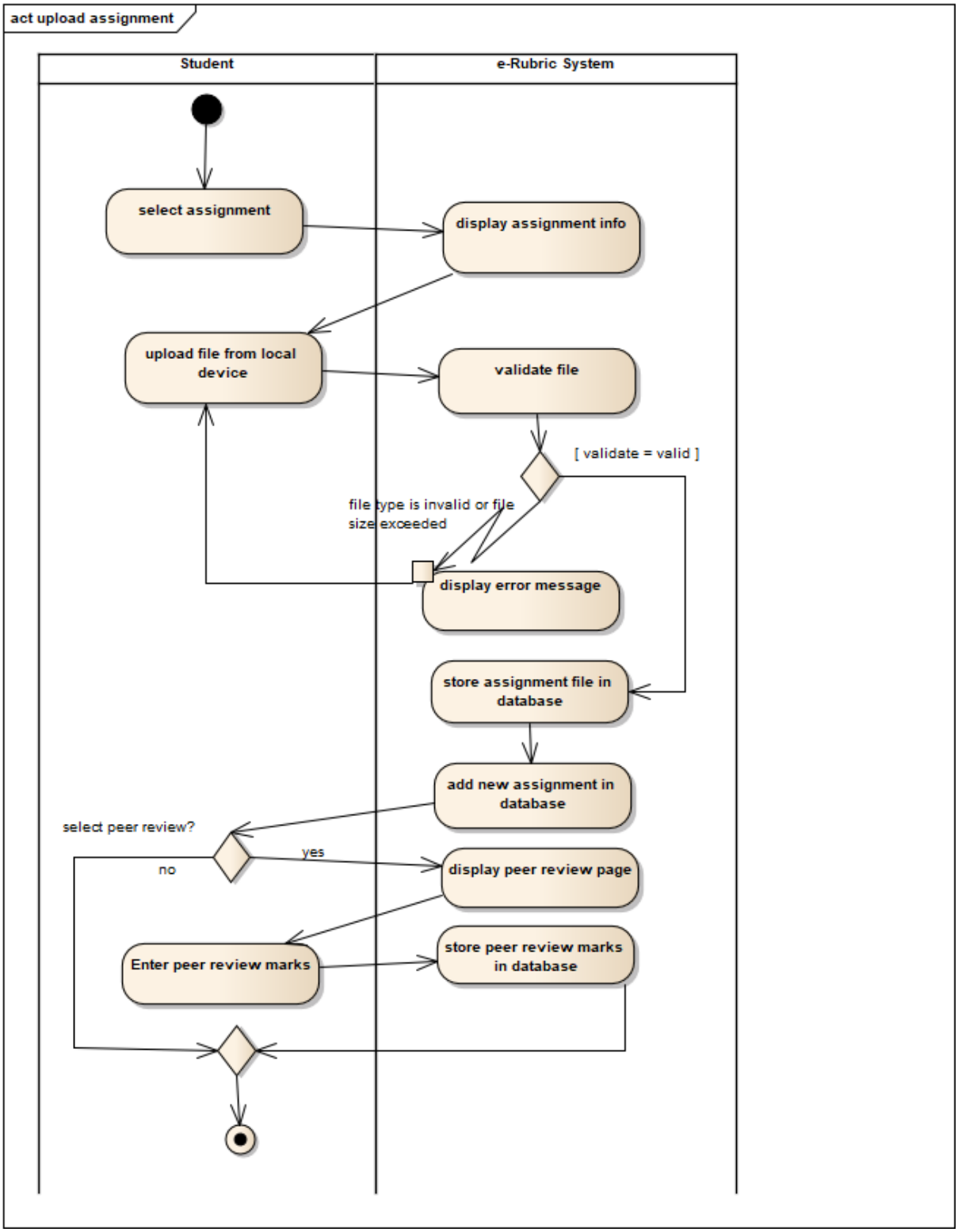

Figure 2.25: Activity Diagram for Upload assignment

3.2.12 UC012: Use Case Add Assignment

<table>
<tr><th colspan="5">Use case: Add Assignment</th></tr>
<tr><td rowspan="2">History Log:</td><td colspan="2"><ID></td><td colspan="2"><Changes></td></tr>
<tr><td colspan="2">1.0.0</td><td colspan="2">Create initial use case</td></tr>
<tr><td>Version</td><td colspan="4">1.0.0</td></tr>
<tr><td>Created by:</td><td colspan="2">Tan Chong Lim</td><td>Last update by:</td><td>Tan Chong Lim</td></tr>
<tr><td>Date created</td><td colspan="2">11/05/2020</td><td>Last Revision Date:</td><td>7/6/2020</td></tr>
<tr><td colspan="5">ID: UC012</td></tr>
<tr><td colspan="5">Actors: Lecturer</td></tr>
<tr><td colspan="5">Preconditions:
1. User must be logged in as lecturer.</td></tr>
<tr><td colspan="5">Flow of events:
1. The lecturer select assignment from the list
2. The lecturer select add assignment.
3. The lecturer input the assignment details.
4. The lecturer choose file to upload form local device.
5. If file format is not supported or file size exceeded, E1 is performed.
6. Select save.</td></tr>
<tr><td colspan="5">Exception E1:
1. prompt error message
2. return to normal flow 4</td></tr>
<tr><td colspan="5">Postconditions:
New assignment is added to database.</td></tr>
<tr><td rowspan="2">Related Requirements(s):</td><td>ID</td><td colspan="2">Requirement</td><td>Priority</td></tr>
<tr><td>FR UC012-01</td><td colspan="2">The lecturer shall be able to add new assignment</td><td>Basic</td></tr>
</table>

Table 2.12: Use Case Description for Add Assignment

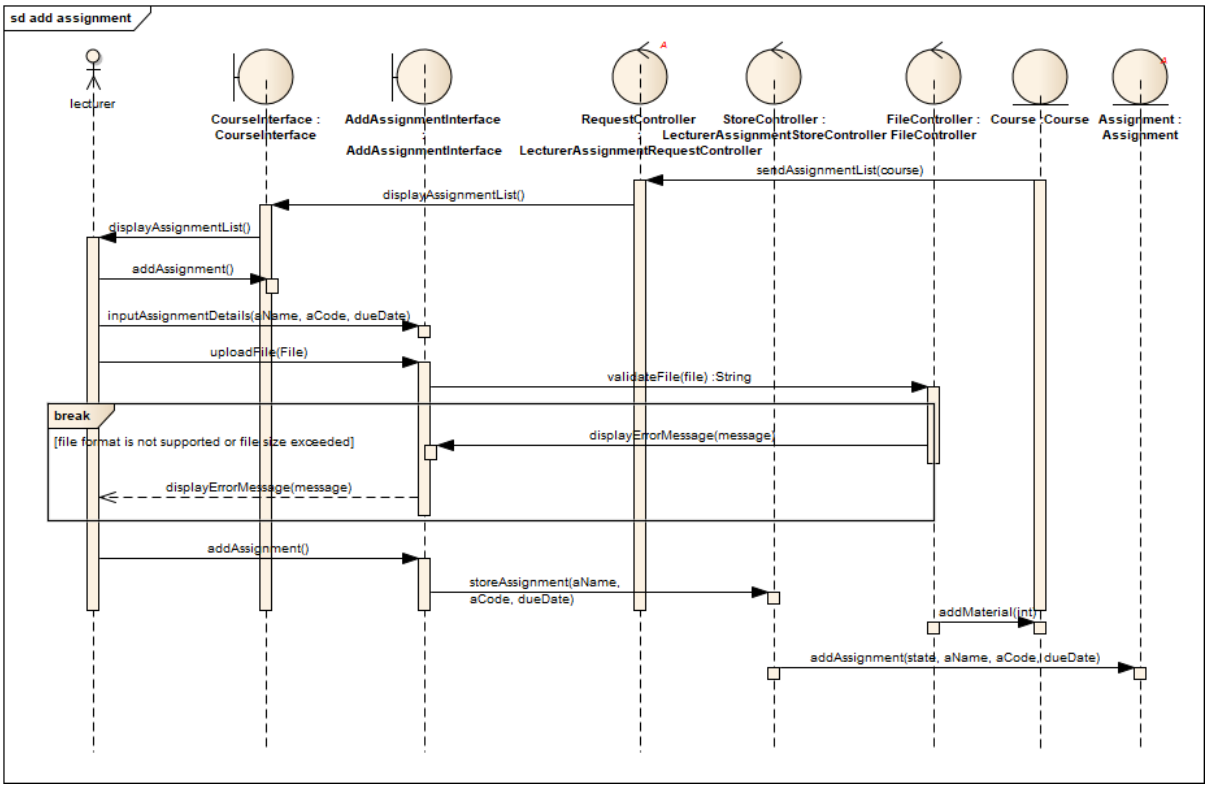

Figure 2.26: Sequence Diagram for Add assignment

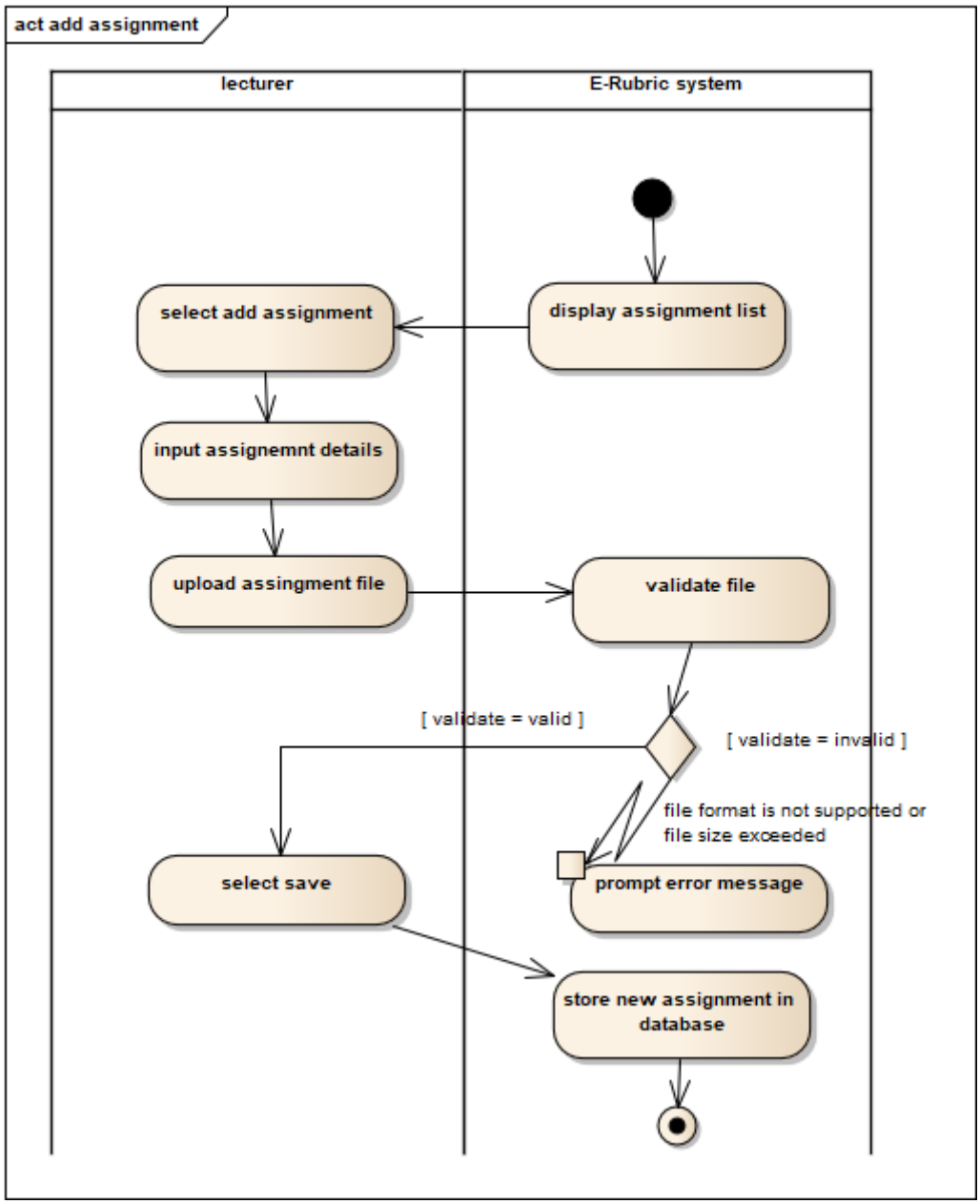

Figure 2.27: Activity Diagram for Add assignment

3.3 Performance Requirements

- The system should response to user request in less than 5 seconds
- The system shall be able to update the database in less than 5 seconds for every request.
- When success upload assignment, the system shall send notification email to the students within 5 seconds.
- The system shall support at least 5000 concurrent users.

3.4 Design Constraints

- The system shall not display any content that violate the copyright law.
- The rubric generator shall be based on UTM Assessment for rubric handbook.
- The system shall be based on Angular JS as the frontend, while Spring Framework as the backend, with MySQL as the database.

3.5 Software System Attributes

- The system shall online and running 24hours every day.
- The system shall backup the database on 11.59pm every day.
- The admin shall use a strong password and encryption key for all backend services.
- The system shall check and validate every user input to prevent SQL injection attack.
- The system shall be able to install in any operating system that preinstall Java, and MySQL server.

2.7 Other Requirements

- The system grading system shall be precise in calculation and accurate to 2 decimal places.
- The system shall be able to run on any device equipped with any web browser.