COMP 6030 : Software Engineering

Coursework – Individual Part

Group : 15

Student Number : 19212742

Subsystem : Cla-Clo Teacher

Introduction:

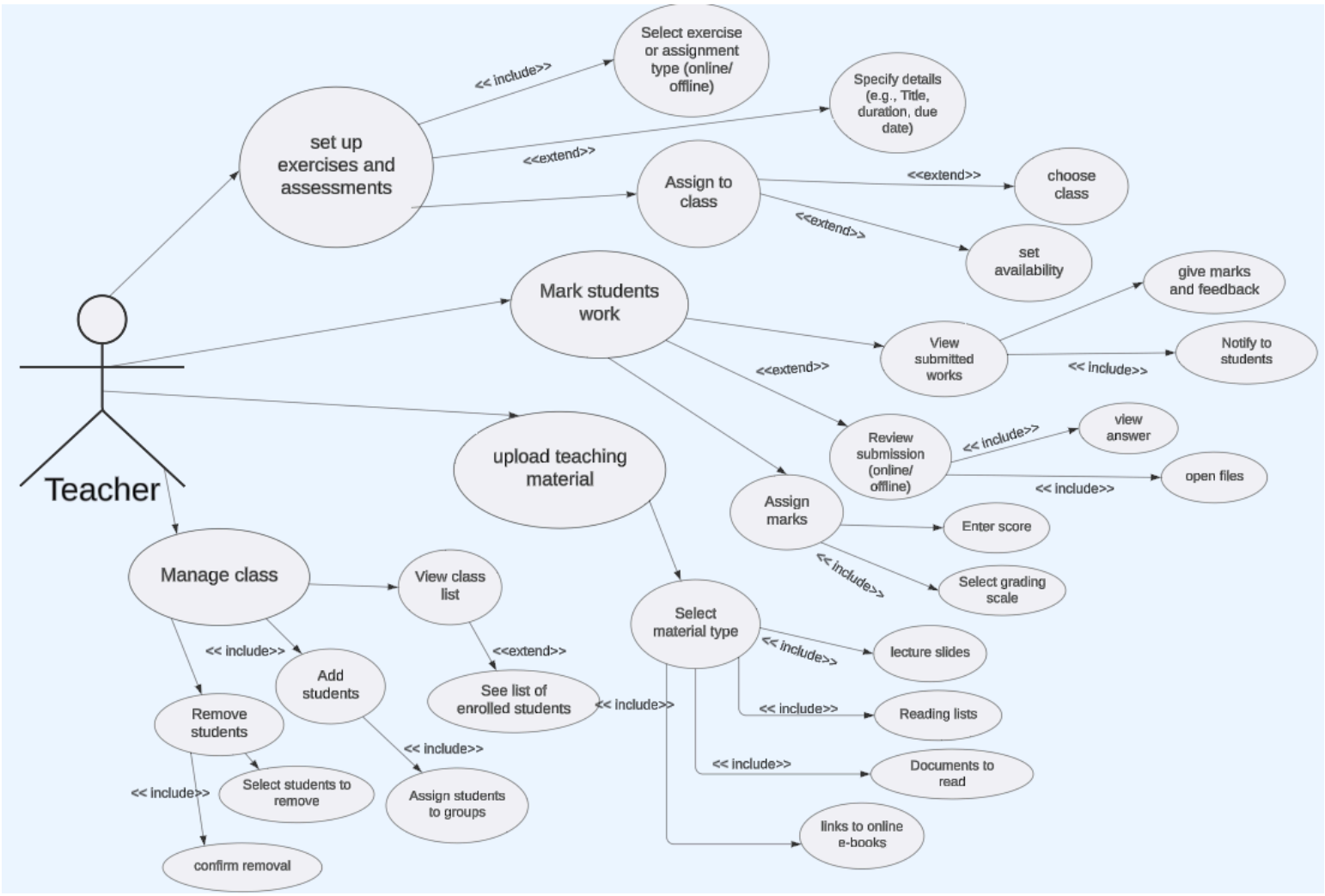
In this coursework, I am responsible for creating the ClaClo-Teacher subsystem, which helps teachers manage their tasks. This subsystem allows teachers to give students exercises and coursework, receive and mark their submissions, and provide feedback. As part of the project, I will work on defining the requirements for the teacher subsystem, designing how it works, and creating detailed models to show its structure and behavior. I will also create a test plan to make sure the subsystem works properly and connects well with the rest of the ClaClo system, ensuring smooth interaction with students, administrators, and university managers.

Task1: Software modelling and specification

(a) Use Case model

Here is the link for the use case model that I have made on Lucidchart:

https://lucid.app/lucidchart/15a0ff26-db28-4b70-b805-6b2bd66136bf/edit?viewport_loc=-3150%2C-750%2C2800%2C1546%2C.Q4MUjXso07N&invitationId=inv_773ac9f8-af9d-40e2-8643-2a3d3af4eea6

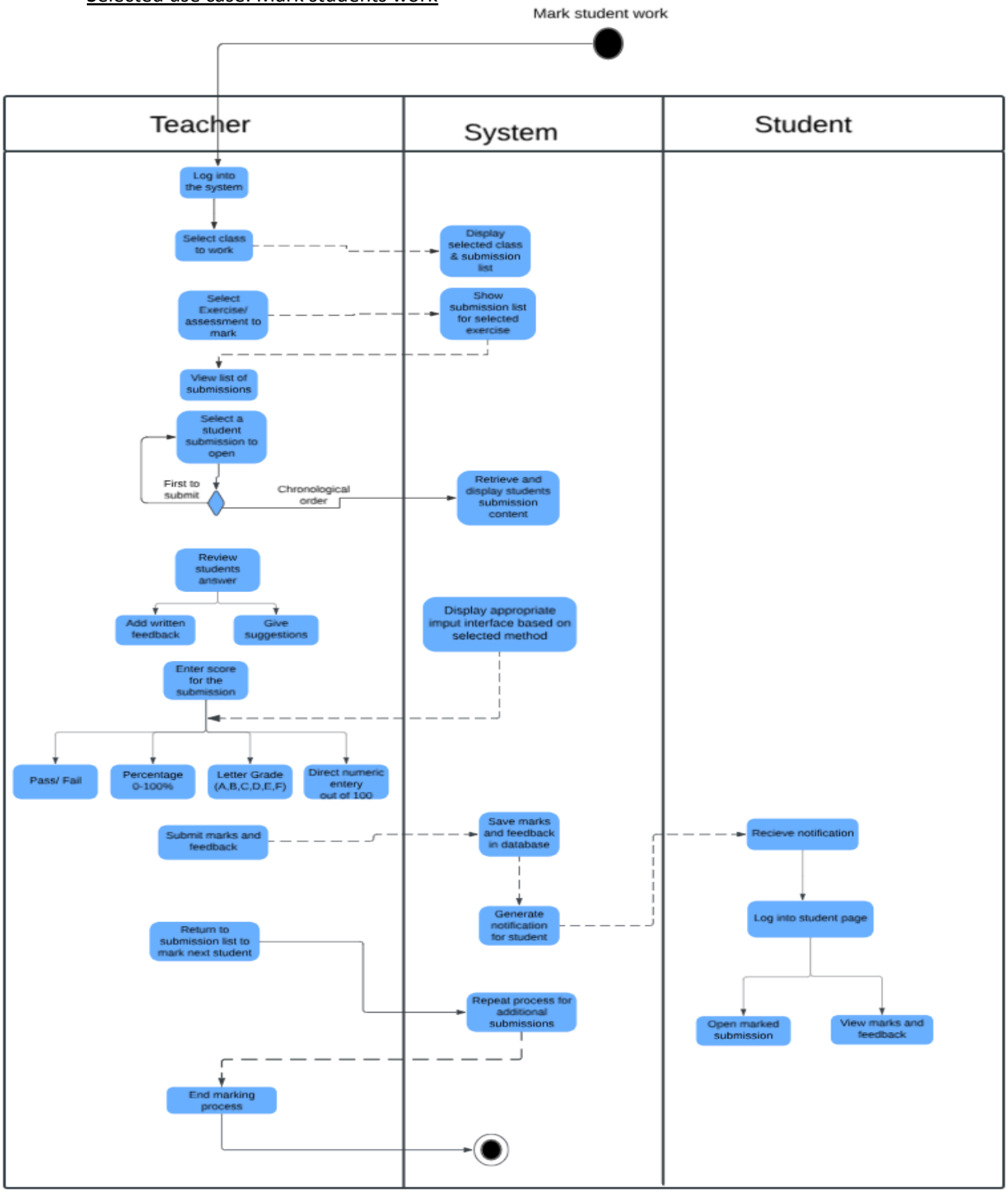


(b) Activity model

Here is the link for the Activity model that I have made on Lucidchart:

https://lucid.app/lucidchart/3f867bb2-4c58-46ac-89a6-f2bb035cd8fa/edit?viewport_loc=-2037%2C-1312%2C4234%2C1967%2C0_0&invitationId=inv_bab2d493-9cac-4750-9c28-fcebb523d409

Selected use case: Mark students work

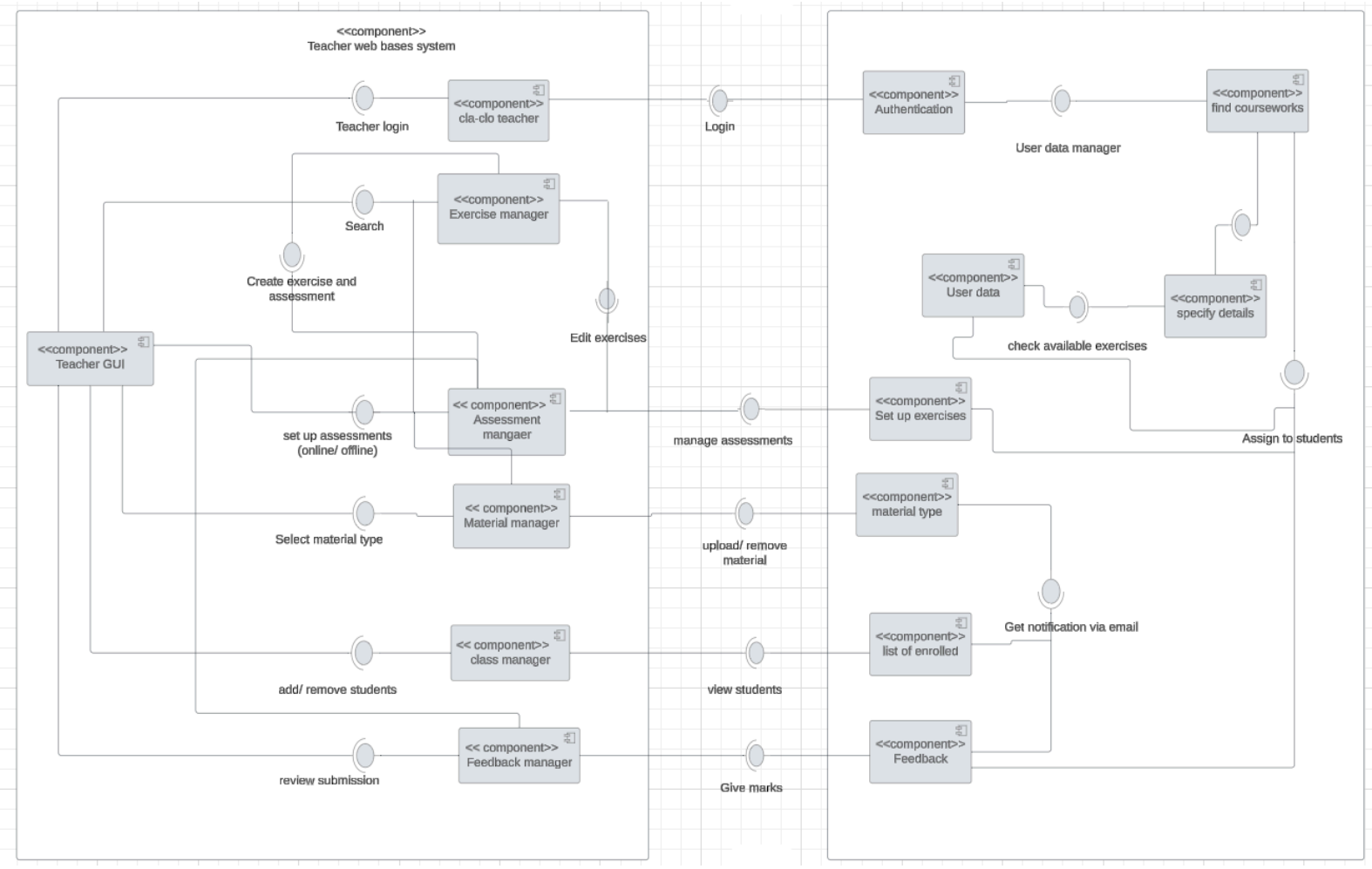


Task2: Software Architectural Design

(a) Architecture of the subsystem

Here is the link that I have made on Lucidchart:

https://lucid.app/lucidchart/fdf9706d-5045-4374-90d4-f1fcf8abc093/edit?viewport_loc=-1223%2C-1372%2C3830%2C1699%2C0_0&invitationId=inv_e9fa55f2-a5f4-4056-b8a1-84d2ae4a0a2c

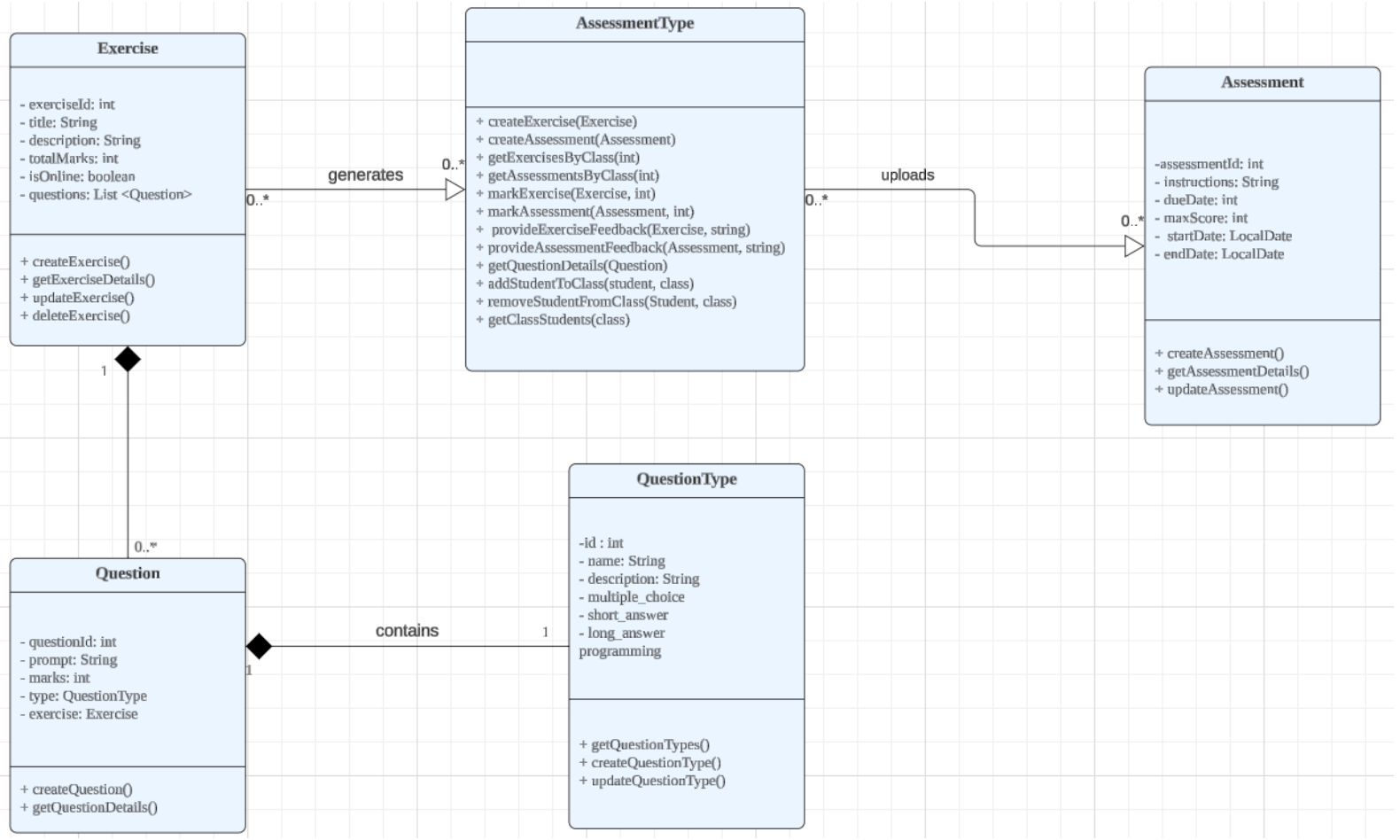


Task3: Software Detailed Design

(a) Structural model

Here is the link for the structural model that I have made on Lucidchart:

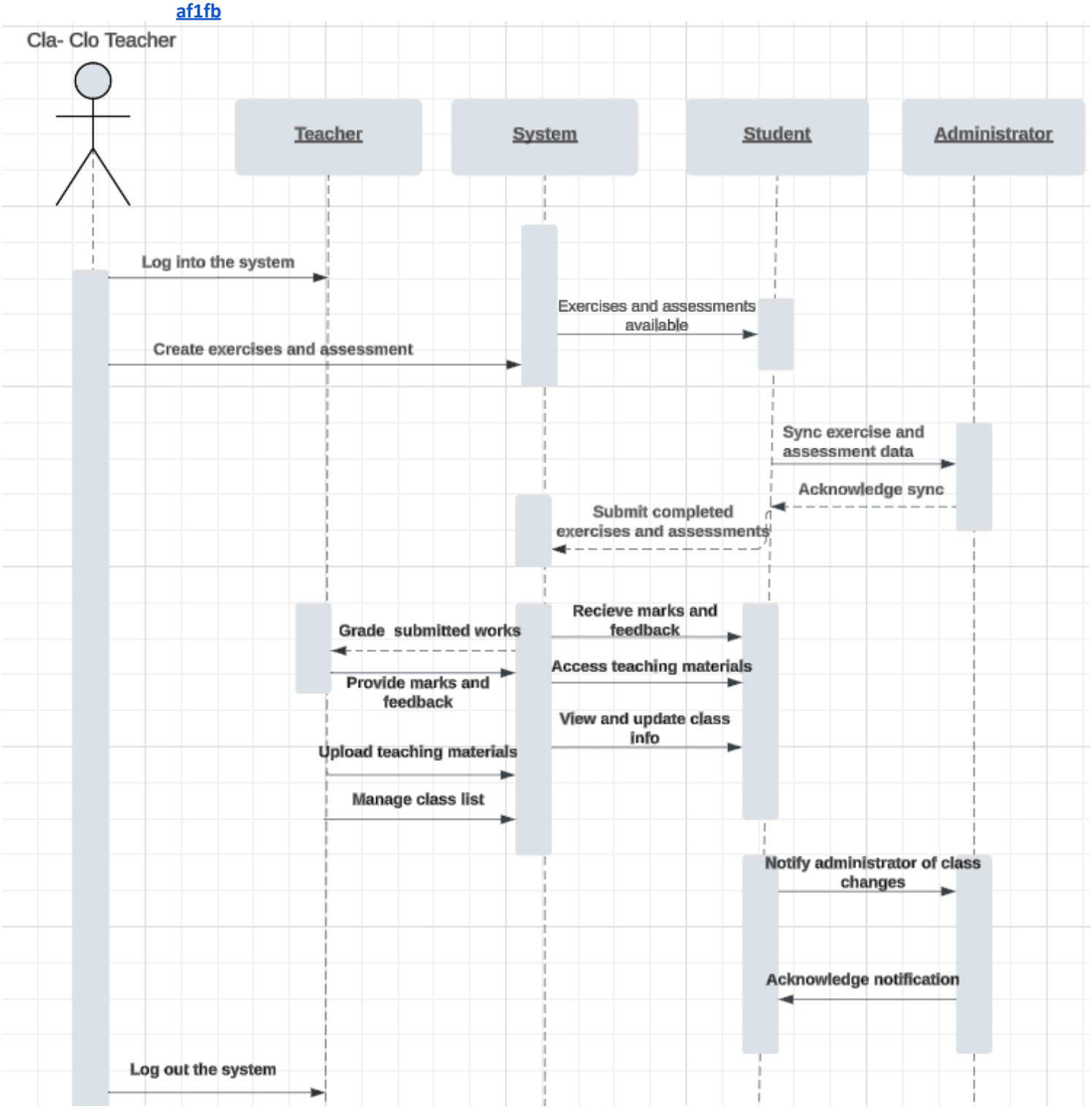
https://lucid.app/lucidchart/e520b7a0-bcfd-4a8d-a2c0-31cb0dbfe7b4/edit?viewport_loc=-2000%2C-678%2C2994%2C1391%2C0_0&invitationId=inv_a1eddabd-f557-4540-8f7d-f3c9cbb1df34



(b) Behaviour model

Here is the link for the use case model that I have made on Lucidchart:

https://lucid.app/lucidchart/87b9116b-9996-4fb2-811d-b9cb9c4cd958/edit?viewport_loc=-1854%2C-771%2C2683%2C1189%2C0_0&invitationId=inv_6b58af18-cef2-4ff5-9714-254ca45af1fb



Task4: Software testing

(a) Unit test plan

Unit test case for Teacher component from architecture design on subsystem

Test case	Methods and Parameters	Expected Output
LoginRegistration successful	userRegistration (123456,19212742@brookes.ac.uk)	Registration and login successful. Teacher redirected to their dashboard.
LoginRegistration failed	userRegistration (54945945,54945945@brookes.ac.uk)	Error message – mismatched passwords or invalid email format Teacher is not registered or logged in.
Create exercises and assessment successful	createAssessmnet(assessmnet_type=”quiz”, passing_ score = 60)	Online exercises set up successfully and made available for download and visible to students
Create exercises and assessment failed	createAssessmnet(assessmnet_type=”quiz”, passing_ score = 60, invalid_ dates= “2023-15-01”)	Error message- exercises and assessment creation fails due to file size exceeding limits or missing information
Upload teaching materials successful	uploadTeachingMaterial(file_name="lesson1.pdf", file_type="PDF", file_size="1MB", content_type="application/pdf")	“Your file ‘lesson1.pdf’ has been uploaded successfully” Redirect to a dashboard or confirmation screen.
Upload teaching materials failed	uploadTeachingMaterial(file_name="lesson1.pdf", file_type="PDF", file_size="30MB", content_type="application/pdf")	Error message- File type not supported. Please upload PDF or DOCX files and a file smaller than 20MB.
Mark students work successful	markStudentWork(student_id=12345, class_id="3rd year", subject_id="Mathematics", marks_obtained=85, status="Passed", grading_date="2024-12-05T12:00:00Z")	Student marked successfully All students have been graded successfully Student ‘12345’ has been marked as passed and notified via the system
Mark students work failed	markStudentWork(student_id=12345, class_id="3rd year", subject_id="Mathematics", marks_obtained="invalid_data", status="Failed", grading_ date="2024-12-05T12:00:00Z")	Error message-student Marking failed due to invalid marks data. Marks and feedback are not saved in the database

Manage classes successful	**manageClass(action="add", class_id="3rd year", student_id=12345, student_name="Varshitha Bairi")** **manageClass(action="remove", class_id="3rd year", student id=12345)**	**Class list retrieved successfully** **Student added successfully to 3rd year** **Student removed from the class successfully** **System reflects updated class membership**
Manage class failed	**manageClass(action="add", class_id="3rd Year", student_id=12345)**	**Failed to retrieve class. Invalid class ID or no students found** **Error: unable to add students. Invalid details or students already exist.** **Error: student not found in the class**

(b) System test plan

Use case- Mark students work- successful marking process

Pre condition:

- The teacher is logged into the system.
- The system has student submissions for the selected class/exercise.
- The system has saved the marks and feedback.
- The student is notified.

Teacher	System
1. Logs into the system	2. The system successfully authenticates the teacher and displays their dashboard.
3. Selects the class to work on	4. The system displays the selected class and submission list.
5. Selects an exercise/assessment to mark	6. The system shows the list of submissions for the chosen exercise
7. Opens a student submission.	8. The system retrieves and displays the content of the selected student's submission.
9. Reviews the submission and adds written feedback and/or suggestions	10. The system displays the input interface for feedback.
11. Enters a score (e.g., Pass/Fail, percentage, letter grade, or numeric score)	12. The system allows the input and validates it.
12. Submits the marks and feedback	13. The system saves the feedback and marks in the database. Generates a notification for the student.
14. Returns to the submission list to mark the next student	15. The system redirects to the submission list.

Test data:

- Input
    - ➔ Student ID
    - ➔ Assignment/Exam ID
    - ➔ Marks
    - ➔ Feedback comments
    - ➔ Grading criteria
- Stored data (on cloud)
    - ➔ Student database
    - ➔ Course database
    - ➔ Assignment/Exam records
    - ➔ Marking schemes
    - ➔ User authentication database

➢ Stored data (on device)

➔ Temporary marking session data

➔ Local cache of student assignment details

➔ Offline marking capabilities

- output

➔ Successful mark submission confirmation

➔ Updated student grade record

➔ Feedback sent to student

➔ Audit log of marking process

- Test process

1. Setup Test Context

- Prepare test student data
- Load sample assignment
- Configure test grading criteria

2. Authentication

- Verify assessor credentials
- Check permission levels for marking

3. Load Student Assignment

- Retrieve assignment details
- Verify assignment eligibility for marking

4. Mark Assignment

- Input marks according to grading criteria
- Enter detailed feedback
- Validate mark ranges and consistency

5. Save Marks

- Submit marks to system
- Verify successful database update
- Generate confirmation receipt

6. Notification

- Confirm marks sent to student
- Update grade database
- Create marking audit trail

Use case- Mark students work-no submissions available

Precondition:

- The teacher is logged into the system.
- The system has no student submissions for the selected class/exercise.
- The teacher is notified of the lack of submissions

Teacher	System
1. Logs into the system	2. The system successfully authenticates the teacher and displays their dashboard.
3. Selects the class to work on	4. The system displays the selected class and submission list.
5. Selects an exercise/assessment to mark	6. The system informs the teacher that no submissions are available for marking.

Test data:

- Input
  - ➔ Course ID
  - ➔ Assignment/Exam ID
  - ➔ Assessor's user credentials
  - ➔ Expected submission deadline
- Stored data (on cloud)
  - ➔ Course enrollment database
  - ➔ Assignment submission records
  - ➔ Student registration database

➢ Stored data (on device)

➔ Local cache of course details

➔ Offline submission tracking

➔ Temporary marking interface state

- output

➔ "No submissions" notification

➔ List of students with missing submissions

➔ System-generated reminder report

➔ Marking interface status update

- Test process

7. . Setup Test Context

- Prepare course and assignment details

- Verify submission deadline
- Configure notification settings

8. Authentication

- Verify assessor credentials
- Check marking permissions

9. Check Submissions

- Query submission database
- Identify students with no submissions
- Generate missing submission report

10. Handling No Submissions

- Trigger notification system
- Create tracking record for late submissions
- Update course management dashboard

11. System Response

- Display "No submissions" message
- Provide options for manual intervention
- Log marking attempt for audit purposes

Use case- Mark students work- error in saving marks/ feedback

Precondition:

- The teacher is logged into the system.
- The system has student submissions for the selected class/exercise.
- The teacher is informed of the error.
- The marks and feedback are not saved until the issue is resolved.

Teacher	System

1. Logs into the system	2. The system successfully authenticates the teacher and displays their dashboard.
3. Selects the class to work on	4. The system displays the selected class and submission list.
5. Selects an exercise/assessment to mark	6. The system shows the list of submissions for the chosen exercise
7. Opens a student submission.	8. The system retrieves and displays the content of the selected student's submission.
9. Adds written feedback and enters a score.	10. The system allows the input and validates it.
11. Submits the marks and feedback	12. If there's an error (e.g., database failure), the system displays an error message and prompts the teacher to retry.

Test data:

- Input
  - ➔ Student ID
  - ➔ Assignment/Exam ID
  - ➔ Marks
  - ➔ Feedback comments
  - ➔ Assessor's user credentials
- Stored data (on cloud)
  - ➔ Student database
  - ➔ Marking database
  - ➔ Error logging system
  - ➔ Backup marking records

➢ Stored data (on device)

➔ Local marking session data

➔ Temporary mark storage

➔ Error recovery cache

- output

➔ Error messages

➔ System failure notification

➔ Mark saving status

➔ Error log details

- Test process

12. Setup Test Context

- Prepare course and assignment details
- Verify submission deadline
- Configure notification settings

13. Authentication

- Verify assessor credentials
- Check marking permissions

14. Check Submissions

- Query submission database
- Identify students with no submissions
- Generate missing submission report

15. Handling No Submissions

- Trigger notification system
- Create tracking record for late submissions
- Update course management dashboard

16. System Response

- Display "No submissions" message
- Provide options for manual intervention
- Log marking attempt for audit purposes