UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE ED ELETTRICA E MATEMATICA APPLICATA

Report - Software Architectures for Enterprise Systems

Peer Review System: PeerFlow

Requirements Document

Group 1

Professor

Nicola Capuano - ncapuano@unisa.it

Team members

Name and Surname	**Student ID**	**E-mail**
Ciaravola Giosuè	0622702177	g.ciaravola3@studenti.unisa.it
Della Corte Mario	0622702354	m.dellacorte19@studenti.unisa.it
Polverino Alessandro	0622702352	a.polverino15@studenti.unisa.it

ACADEMIC YEAR 2024/2025

Contents

1 Introduction

1.1 Purpose

This document defines the software requirements for the development of an automated Peer Review system, named PeerFlow. The main objective of this system is to facilitate a collaborative student evaluation process based on predefined criteria. PeerFlow will support the creation, management, and evaluation of academic assignments, providing an efficient platform for teachers and students. This document is intended for all project stakeholders, including clients, developers, testers, and project managers, to establish a clear and shared understanding of PeerFlow's functionalities and expectations.

1.2 Scope

The Peer Review system (PeerFlow) aims to support the following main functions:

- Assignment creation by teachers, including descriptions, assessment rubrics, deadlines, and involved students.
- Submission of assignments by students (text plus attachments).
- Peer assignment for review purposes – may be automatic (random) or manual (defined by the teacher).
- Review process where students provide motivated evaluations according to defined rubrics.
- Aggregation and visualization of evaluation results for students and teachers.

1.3 Definitions, Acronyms, and Abbreviations

- **API:** Application Programming Interface. A set of rules and protocols for building and interacting with software applications.
- **Assignment:** Work assigned by a teacher, to be submitted and/or reviewed.
- **BL:** Business Logic. The part of the system that encodes the real-world business rules that determine how data can be created, stored, and changed. Referenced in requirement IDs (e.g., FR-SYS-002.BL).
- **CI/CD:** Continuous Integration / Continuous Delivery. Practices for automating the stages of software development, from integration to delivery.
- **CPU:** Central Processing Unit. The primary component of a computer that executes instructions.
- **Docker:** A platform for developing, shipping, and running applications in containers.
- **FR:** Functional Requirement. A description of a service or function that the software system must provide. Used as a prefix in requirement IDs (e.g., FR-SYS-001).
- **HTTP:** HyperText Transfer Protocol. The foundation of data communication for the World Wide Web.
- **Kubernetes:** An open-source container orchestration system for automating software deployment, scaling, and management.
- **LCP:** Largest Contentful Paint. A user-centric metric for measuring perceived load speed because it marks the point in the page load timeline when the page's main content has likely loaded.
- **MOOC:** Massive Open Online Course. An online course aimed at unlimited participation and open access via the web.
- **MoSCoW Prioritization:** A prioritization technique used in management, business analysis, project management, and software development to reach a common understanding with stakeholders on the importance they place on the delivery of each requirement; it stands for **M**ust have, **S**hould have, **C**ould have, and **W**on't have.
- **NFR:** Non-Functional Requirement. A requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors (e.g., performance, security).
- **Peer Review:** A collaborative evaluation process where students assess the work of their peers based on predefined criteria.
- **PeerFlow:** The name of the automated Peer Review system.
- **PII:** Personally Identifiable Information. Any data that could potentially identify a specific individual.
- **QA:** Quality Attribute. A measurable or testable property of a system that is used to indicate how well the system satisfies the needs of its stakeholders. Often used in the context of Non-Functional Requirements.

- **Rubric:** A set of predefined criteria used for evaluation, often including descriptions of levels of quality.
- **SOA:** Service-Oriented Architecture. A software design style where services are provided to the other components by application components, through a communication protocol over a network.
- **SRS:** Software Requirements Specification. A document that describes what a software system should do and how it should perform.
- **Student:** User enrolled in one or more courses, responsible for submitting assignments and reviewing peers' work.
- **Teacher:** User with the role of instructor, responsible for creating courses, assignments, and overseeing the review process.
- **TTFB:** Time To First Byte. A metric used as an indication of the responsiveness of a web server or other network resource. It measures the duration from the user or client making an HTTP request to the first byte of the page being received by the client's browser.
- **UI:** User Interface. The means by which the user and a computer system interact, in particular the use of input devices and software. Referenced in requirement IDs (e.g., FR-SYS-002.UI).
- **VM:** Virtual Machine. An emulation of a computer system. Virtual machines are based on computer architectures and provide functionality of a physical computer.

1.4 References

- Project Assignment document.
- IEEE 830-1998 Standard, "Recommended Practice for Software Requirements Specifications".

1.5 Overall System Description

PeerFlow will be a web platform accessible via a user interface. It will be designed to support a large and highly variable number of users, typical of MOOC environments like Coursera or Udacity. The system's architecture will be based on a Service Oriented/Microservices Architecture approach, to ensure horizontal scalability, high availability, and low latency. The main users will be teachers and students, with distinct roles and functionalities. PeerFlow will allow teachers to create assignments and evaluation rubrics, monitor the submission and review process, and access detailed evaluation reports. Students will be able to submit their work (text and file attachments), receive assigned submissions from peers to review, complete structured reviews with motivated feedback, and view the evaluations received after the process is completed.

2 Overall Description

2.1 Product Perspective

PeerFlow will be a stand-alone application with a dedicated user interface. It is designed to operate within an academic context, with the goal of improving the evaluation and learning process in online education environments, particularly for MOOCs. Its microservices architecture will ensure that it can be integrated or extended in the future, if necessary, to connect with other educational systems or learning management platforms.

2.2 Production Functions

The system (PeerFlow) will provide the following main functions:

- **Assignment Management:**
  - Creation of assignments by teachers, including description, assessment rubrics, deadlines, and students involved.
  - Viewing of assigned submissions for students.
  - Submission of assignments by students (text and file attachments).
  - Monitoring the submission process by teachers.
- **Peer Review Management:**
  - Automatic (random) or manual (defined by the teacher) peer assignment for review purposes.
  - Receiving assigned submissions from peers to review for students.
  - Completing structured reviews with motivated feedback by students, based on defined rubrics.
- **Results and Reporting Management:**
  - Aggregation of evaluations.
  - Viewing evaluations received by students after the process is completed.
  - Accessing detailed evaluation reports for teachers.
  - Monitoring the review process by teachers.
- **Authentication and Profiling:**
  - Managing login, roles (teacher, student), and user profiles.

2.3 User Characteristics

PeerFlow will primarily be used by two categories of users:

- **Teachers:**
  - Responsible for creating assignments and evaluation rubrics
  - Must be able to define assessment rubrics and deadlines.
  - Need to access detailed evaluation reports and monitor the submission and review process.
- **Students:**
  - Responsible for submitting their work (text and file attachments) and reviewing peers' work.
  - Need an intuitive interface for uploading files and providing structured feedback.
  - Must be able to view the evaluations received after the process is completed.

2.4 Constraints

- **Architectural:** The system must adopt a Service Oriented/Microservices Architecture.
- **User Interface:** The User Interface (UI) must be included.
- **Development:**
  - Use of a Version Control System is mandatory.
  - Defining, implementing, and using a CI/CD pipeline is mandatory.
  - Use of Automatic Testing is mandatory.
  - Use of Containerization (Docker) is mandatory.
  - Use of Container Orchestration (Kubernetes) is mandatory.
  - Use of Infrastructure-as-Code Tools is optional.
- **Deployment:** Deployment can occur on Microsoft Azure or on an Ubuntu VM (On Premise).
- **Security:** The system must ensure the security of user data and submissions.

2.5 Assumptions and Dependencies

- It is assumed that users (teachers and students) have access to a stable internet connection.
- It is assumed that teachers will provide clear and complete assessment rubrics for assignments.
- The success of the system depends on the active participation of students in the review process.

3 Specific Requirements

This section details all the specific requirements of the PeerFlow system. These requirements describe the functionalities and non-functional attributes that the system must possess. They are categorized to provide clarity and ensure completeness, serving as the foundation for the design, development, and testing phases of the project. Each requirement should be clear, unambiguous, verifiable, and traceable.

3.1 Functional Requirements

This subsection outlines the specific functionalities that PeerFlow must provide to its users (teachers and students). Each functional requirement describes an action that the system must perform, how it should react to specific inputs, or what it should allow users to do. These requirements will often be further detailed with use cases, user stories, or detailed input/output descriptions in subsequent sections or annexes. Functional requirements are prioritized using "MoSCoW Prioritization":

- **M - Must have**: Non-negotiable product needs that are mandatory for the team.
- **S - Should have**: Important initiatives that are not vital, but add significant value.
- **C - Could Have**: Nice to have initiatives that will have a small impact if left out.
- **W - Will not have**: Initiatives that are not a priority for this specific time frame.

For a better understanding of the system's operations, each subsection within the subsequent "Functional Requirements" section will be accompanied by a concise activity diagram to visually clarify the main activity flow of the described requirements.

3.1.1 General System

ID	Priority	Requirement
FR-SYS-001	M	The system shall support two distinct user roles: Teacher and Student.
FR-SYS-002	M	The system shall require users to authenticate before accessing role-specific functionalities.
FR-SYS-002.UI		The system's UI shall present a typical login page (email and password) as the initial point of interaction for unauthenticated users attempting to access the system.
FR-SYS-002.BL		The system's BL shall validate submitted user credentials (handling the invalid credential case), identify the authenticated user's pre-assigned role and enforce access restrictions by granting access only to the system functionalities, data, and user interface views appropriate for the identified user role.
FR-SYS-003	M	The system shall allow new users (Students or Teachers) to sign up.
FR-SYS-003.UI		The system's UI shall provide a clearly identifiable link or button (e.g., labeled "Sign Up" or "Register") on the login page, which navigates the user to a dedicated registration page. The Sign Up Page should select if the user is a Student or a Teacher and insert name, surname, email (unique) and password, with a Sign Up button to confirm.
FR-SYS-003.BL		The system's BL, should validate (check if a user with the same email already exists) and register the new user.

Table 3.1: General System Functional Requirements.

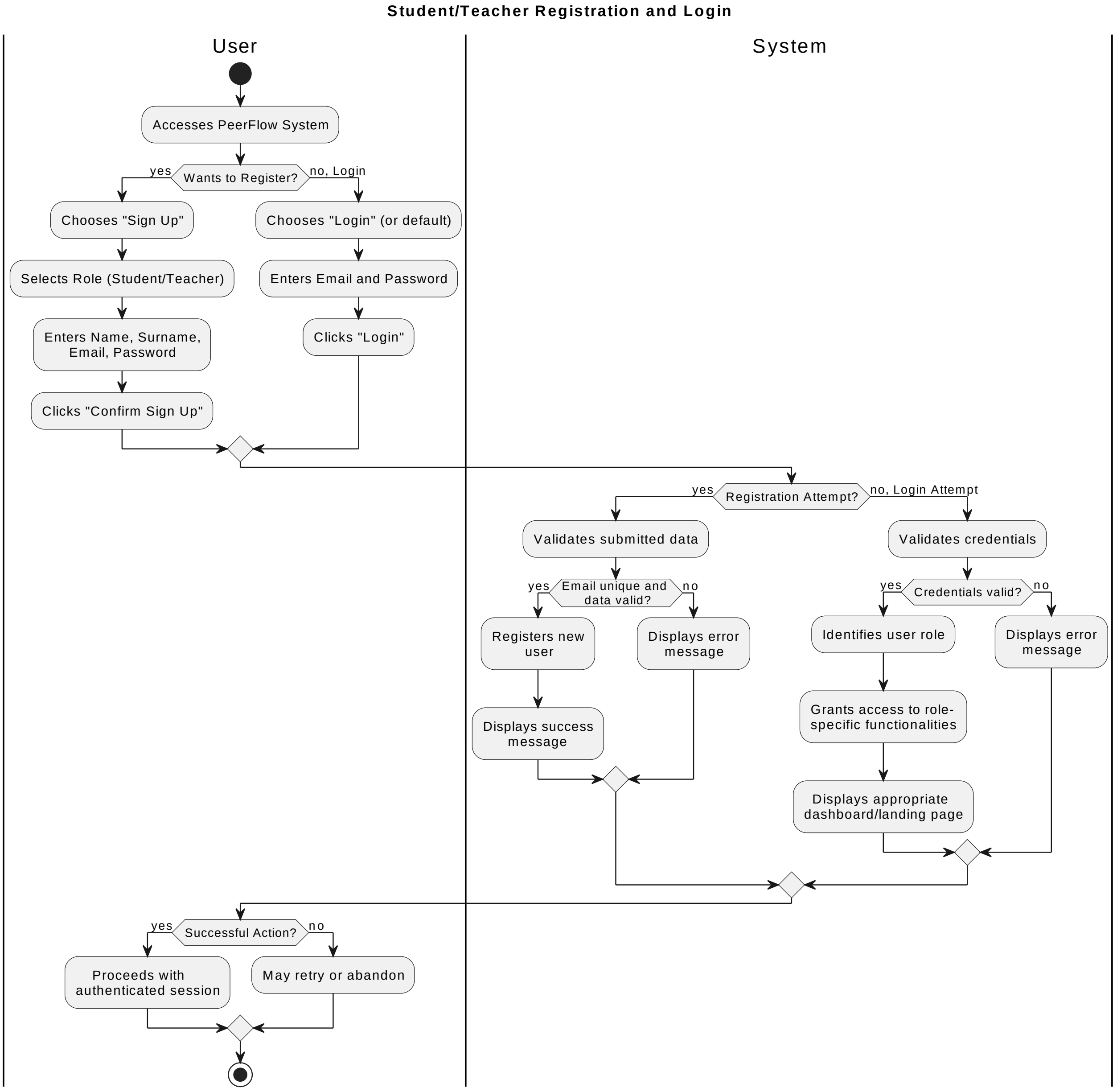

Figure 3.1: General System Student/Teacher Registration and Login Activity Diagram.

3.1.2 Assignment Management - Teacher

ID	Priority	Requirement
FR-ASG-001	M	The system shall enable authenticated Teachers to view a list of all assignments within the system.
FR-ASG-001.UI		The system's UI shall provide a dedicated section or page, accessible to authenticated Teachers, specifically for displaying assignments (for each show name, status [assigned, preparing review, under review, completed, deleted], and last modified date). The UI shall provide a mechanism to navigate from an assignment in the list to a more detailed view or management page for that specific assignment (name, description, deadline, current status, list of students involved, created data).
FR-ASG-001.BL		The system's BL shall provide the list of created assignments for the specific authenticated Teacher. In case of navigate into the detailed page of an assignment, the BL shall retrieve the necessary data fields (name, description, deadline, current status, list of students involved, created data).
FR-ASG-002	M	The system shall allow authenticated Teachers to create new assignments.
FR-ASG-002.UI		The system's UI shall provide a mechanism for authenticated Teachers to initiate the creation of a new assignment (e.g., a "Create New Assignment" button or link accessible from assignments list page). Upon initiating assignment creation, the UI shall present a form or dedicated page where the Teacher can input the details for the new assignment (name, description, deadline, list of students involved). The UI shall provide a button (e.g., "Create Assignment," "Save Assignment") to submit the new assignment details.
FR-ASG-002.BL		Upon submission of the new assignment creation form by an authenticated Teacher, the system's business logic shall validate the fields (check the deadline is in the future), generate a new assignment record in the data store and storing all provided details (Title, Description, Deadline, list of involved Student IDs, and add the Assignment to the appropriate list of the students involved).

Table 3.2: Teacher Assignment Management Functional Requirements (1-2).

ID	Priority	Requirement
FR-ASG-003	S	The system shall enable authenticated Teachers to modify the details of assignments, provided the modification attempt occurs before the assignment's current submission deadline has passed. Modifiable details include the assignment title, description, submission deadline, and the list of involved students.
FR-ASG-003.UI		From an assignment's detail view page, if the assignment's current submission deadline has not yet passed, the system's UI shall present a clearly identifiable "Edit Assignment" button or link to authenticated Teachers who created the assignment. This option shall be absent or disabled if the deadline has passed. The editing interface shall be pre-populated with the current details of the assignment being edited.
FR-ASG-003.BL		The business logic shall verify that the assignment's original submission deadline (the deadline stored before this modification attempt) has not yet passed. If it has passed, the modification shall be rejected or not shown. The business logic shall validate all submitted modified fields.

Table 3.3: Teacher Assignment Management Functional Requirements (3).

ID	Priority	Requirement
FR-ASG-004	S	The system shall enable authenticated Teachers to soft delete assignments (change its state to “deleted”), provided that the assignment’s peer review process has not been finalized and results have yet been aggregated and made available to students.
FR-ASG-004.UI		The system’s UI shall provide a ”Delete Assignment” button or link for assignments listed in the Teacher’s assignment overview or/and on an assignment’s detail page. This ”Delete Assignment” option shall only be visible and enabled for assignments created by the authenticated Teacher and for which the peer review process has not been finalized and results are not yet available. Upon initiating the deletion action, the UI shall present a confirmation dialog to the Teacher. The assignment shall remain in the list of teacher’s and student’s assignments, but with the “deleted” status (all the button disappear).
FR-ASG-004.BL		The system’s business logic shall first verify that the authenticated Teacher, then check the current status of the assignment to determine if it is eligible for deletion (results of the peer review phase have been made available to students). If the assignment is eligible for the deletion, the BL shall change the assignment status to “deleted”. All the other operation for that assignment are disabled.
FR-ASG-005	C	The system shall notify involved students (through email) when a new assignment is created, (soft) deleted or an existing one is significantly updated (description or deadline).

Table 3.4: Teacher Assignment Management Functional Requirements (4-5).

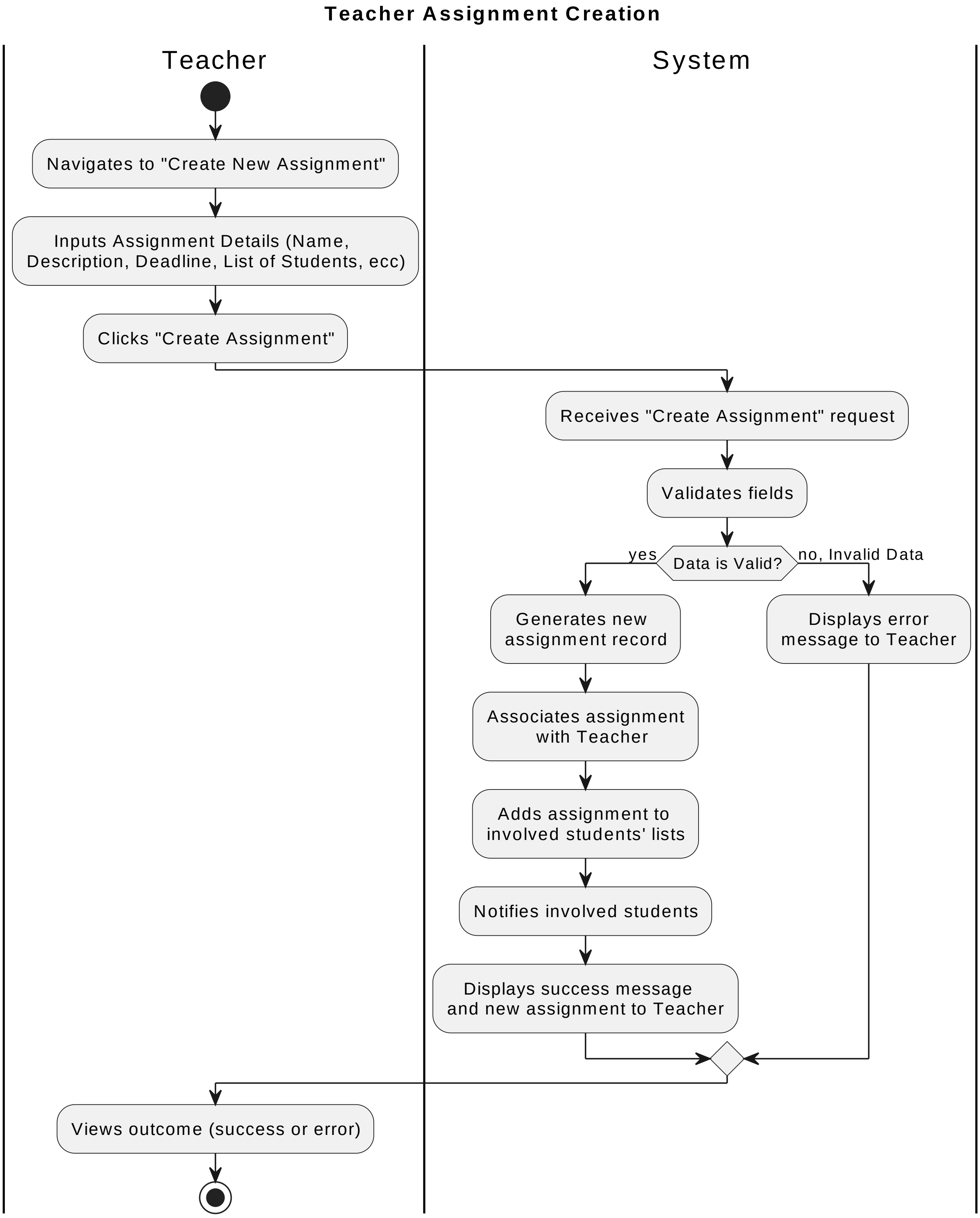
Teacher Assignment Creation
Teacher
System
Navigates to "Create New Assignment"
Inputs Assignment Details (Name, Description, Deadline, List of Students, ecc)
Clicks "Create Assignment"
Receives "Create Assignment" request
Validates fields
yes
Data is Valid?
no, Invalid Data
Generates new assignment record
Displays error message to Teacher
Associates assignment with Teacher
Adds assignment to involved students' lists
Notifies involved students
Displays success message and new assignment to Teacher
Views outcome (success or error)

Figure 3.2: Teacher Assignment Creation Activity Diagram.

3.1.3 Assignment Management - Student

ID	Priority	Requirement
FR-ASG-006	M	The system shall allow authenticated Students to view the list of assignments they are involved in.
FR-ASG-006.UI		The system's UI shall provide a dedicated section or page, accessible to authenticated Students, specifically for displaying assignments they are involved in (for each show name, status, and last modified date). The UI shall provide a mechanism to navigate from an assignment in the list to a more detailed view or management page for that specific assignment (name, description, deadline, current status, list of students involved, created data).
FR-ASG-006.BL		The system's BL shall provide the list of assignments for the specific authenticated Student. In case of navigate into the detailed page of an assignment, the BL shall retrieve the necessary data fields (name, description, deadline, current status, list of students involved, created data).

Table 3.5: Student Assignment List View Functional Requirements.

3.1.4 Submission - Student

ID	Priority	Requirement
FR-SUB-001	M	The system shall enable authenticated Students, who are involved in a specific assignment, to submit their work for that assignment before its submission deadline. Submissions shall consist of textual input and may optionally include file attachments of specified types.
FR-SUB-001.UI		From an assignment's detail view page, if the current date/time is before the assignment's submission deadline and the authenticated Student has not yet submitted their work the system's UI shall present a clearly identifiable "Submit Work" or "Upload Submission" button/link. This option shall be absent or disabled otherwise. Activating the "Submit Work" option shall navigate the Student to a dedicated submission page/modal for that assignment, where he can submit a text, and the possibilities of upload a file (PDF, TXT o JPG). After submit with a button, the UI shall return to the details page of the assignment with the submission shown.
FR-SUB-001.BL		The BL, shall verify that the current system date/-time is before the assignment's official submission deadline. If possible should validate the input and register text (mandatory), attachments (if present), timestamp of submission. If all verifications and validations pass the system shall store all the input, the date timestamp and mark this assignment as "submitted" for the student.
FR-SUB-002	M	The system shall allow students to view/download their own submitted work in the detailed assignment page.

Table 3.6: Student Assignment Submission Functional Requirements (1-2).

ID	Priority	Requirement
FR-SUB-003	S	The system shall enable authenticated Students, who are involved in a specific assignment, to edit all the part of their submitted work for that assignment before its submission deadline. Submissions shall consist of textual input and may optionally include file attachments of specified types.
FR-SUB-003.UI		From an assignment's detail view page, if the current date/time is before the assignment's submission deadline and the authenticated Student has submitted their work the system's UI shall present a clearly identifiable ”Edit Work” or ”Edit Submission” button/link. This option shall be absent or disabled otherwise. Activating the ”Edit Work” option shall navigate the Student to the same page/modal described in the FR-SUB-001.UI for that assignment, with the field filled with their upload, with the possibilities to edit everything. After submit with a button, the UI shall return to the details page of the assignment with the new submission shown.
FR-SUB-003.BL		The BL, shall verify that the current system date/-time is before the assignment's official submission deadline. If possible, shall retrieve the previous submission to fill the fields, and should validate, if edited, the input and register text (mandatory), attachments (if present), and update the timestamp of submission. If all verifications and validations pass the system shall store all the input. If all the fields are empty, the BL shall remove the “submitted” mark for the student.

Table 3.7: Student Assignment Submission Functional Requirements (3).

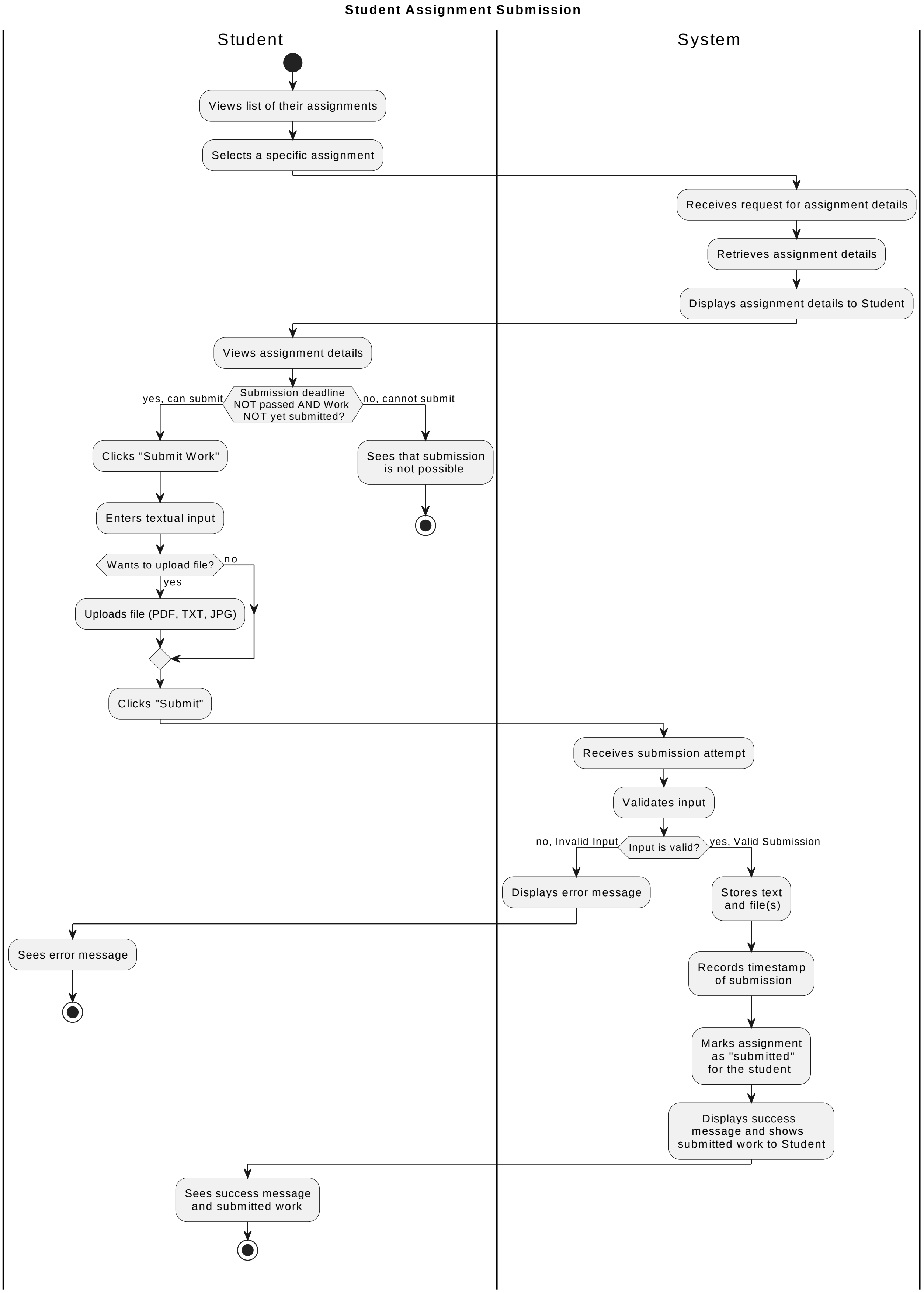
Student Assignment Submission
Student
System
Views list of their assignments
Selects a specific assignment
Receives request for assignment details
Retrieves assignment details
Displays assignment details to Student
Views assignment details
Submission deadline NOT passed AND Work NOT yet submitted?
yes, can submit
no, cannot submit
Clicks "Submit Work"
Sees that submission is not possible
Enters textual input
Wants to upload file?
no
yes
Uploads file (PDF, TXT, JPG)
Clicks "Submit"
Receives submission attempt
Validates input
Input is valid?
no, Invalid Input
yes, Valid Submission
Displays error message
Stores text and file(s)
Sees error message
Records timestamp of submission
Marks assignment as "submitted" for the student
Displays success message and shows submitted work to Student
Sees success message and submitted work

Figure 3.3: Student Assignment Submission Activity Diagram.

3.1.5 Peer Review Assignment - Teacher

ID	Priority	Requirement
FR-PRV-001	S	The system shall enable authenticated Teachers to define and associate a new assessment rubric with a specific assignment, only after the assignment's submission deadline has passed and provided a rubric has not already been finalized for it. A rubric shall consist of one or more criteria, each with a title, a detailed description, and a defined scoring range (e.g., 0 to 5 points).
FR-PRV-001.UI		On an assignment's detail view page, if the assignment's submission deadline has passed and no rubric has yet been finalized for it, the system's UI shall present a clearly identifiable "Define Rubric" or "Set Evaluation Rubric" button/link to the authenticated Teacher. Activating this option shall navigate the Teacher to a rubric creation interface specifically for that assignment. The rubric creation interface shall allow the Teacher to define one or more evaluation criteria. For each criterion, input: title description, scoring range (define a minimum and maximum numerical score range (e.g., Min: 0, Max: 5)). Provide an option to remove criteria during the definition process. The UI shall provide a button (e.g., "Save Rubric," "Finalize Rubric") to submit the defined rubric for the assignment. A "Cancel" button shall also be provided. Upon successful creation and association of the rubric, the UI shall provide clear visual feedback (e.g., a success message) and display the defined rubric on the assignment's detail page.
FR-PRV-001.BL		The BL shall verify that the submission deadline for the target assignment has indeed passed. The BL shall verify that a rubric has not already been finalized and associated with this assignment. The BL shall validate the submitted rubric data: at least one criterion must be defined; for each criterion, a Title, Description, and a valid Scoring Range (e.g., min score ¡= max score, non-negative integers) must be provided. If all verifications and validations pass, the system shall create a new rubric record in the data store, including all its criteria, and associate this newly created rubric with the target assignment.

Table 3.8: Teacher Peer Review Creation Functional Requirements (1).

ID	Priority	Requirement
FR-PRV-002	S	The system shall enable authenticated Teachers to edit the defined assessment rubric with a specific assignment they have created.
FR-PRV-002.UI		On an assignment's detail view page, near the defined rubrics, the system's UI shall present a clearly identifiable “Edit Rubric” button/link to the authenticated Teacher. Activating this option shall navigate the Teacher to the rubric creation interface populated with the defined data during the creation. The UI shall provide a button (e.g., ”Save Rubric,” ”Finalize Rubric”) to submit the edited rubric for the assignment. A ”Cancel” button shall also be provided in order to cancel the update done.
FR-PRV-002.BL		The BL shall verify that a rubric exists. The BL shall retrieve the defined rubrics data to populate the fields. The BL shall validate the submitted rubric data: at least one criterion must be defined; for each criterion, a Title, Description, and a valid Scoring Range (e.g., min score ¡= max score, non-negative integers) must be provided. If all verifications and validations pass, the system shall update the rubric record in the data store, including all its criteria.
FR-PRV-003	S	The system shall enable authenticated Teachers to delete the defined assessment rubric with a specific assignment.
FR-PRV-003.UI		On an assignment's detail view page, near the defined rubrics, the system's UI shall present a clearly identifiable “Delete Rubric” button/link to the authenticated Teacher who created the assignment and rubric. Activating this option shall remove the created rubric.
FR-PRV-003.BL		The BL shall verify that a rubric exists. The BL shall delete the rubrics associated with the assignment.

Table 3.9: Teacher Peer Review Creation Functional Requirements (2-3).

ID	Priority	Requirement
FR-PRV-004	M	The system shall enable authenticated Teachers to configure and initiate the peer review process for a specific course assignment they manage. This configuration shall occur after student submissions for the assignment have closed and an evaluation rubric has been finalized (as per FR-PRV-001). Configuration options shall include setting the number of peer reviewers per submission, confirming the finalized rubric (create one if not present), and choosing between automatic or manual reviewer assignment. The system must prevent any student from being assigned their own submission to review. Once initiated, the peer review phase can only be halted by the Teacher deleting the entire assignment; no other mechanism to stop or pause the review phase for an individual assignment will be provided.
FR-PRV-004.UI		On an assignment's detail view page, if student submissions have closed (and peer review has not yet been initiated), the system's UI shall present a "Setup Peer Review" or "Configure Reviewers" button/link to the authenticated Teacher. Activating this option shall navigate the Teacher to a dedicated peer review configuration interface for that assignment. This interface shall display: a read-only view or summary of the finalized rubric for confirmation (if created, instead the “Create Rubric” button shall be shown); an input field for "Number of Reviewers per Submission" (numerical input); a selection mechanism for "Reviewer Assignment Mode" with options: "Automatic" and "Manual." If "Automatic" assignment mode is selected, no further student selection UI is immediately required on this initial setup screen. If "Manual" assignment mode is selected, the UI shall additionally present: a list of students who have submitted work for this assignment. For each submitting student (or their submission), a mechanism to select the specified number of unique peer reviewers from the pool of other students who have also submitted work for this assignment. The UI must prevent the selection of a student to review their own submission. The interface shall include a button (e.g., "Start Peer Review," "Initiate Reviews").

Table 3.11: Teacher Peer Review Creation Functional Requirements (4).

ID	Priority	Requirement
		(e.g., The UI shall provide clear visual feedback upon successful initiation of the peer review phase (e.g., a success message, assignment status update) or indicate specific input errors if initiation fails "Number of reviewers must be at least 1," "Not enough eligible reviewers for automatic assignment," "All submissions must have the required number of manual reviewers assigned," "Rubric not confirmed"). Once the peer review phase is initiated for an assignment, the "Setup Peer Review" option shall no longer be available for that assignment. UI elements related to modifying the review configuration (other than deleting the entire assignment) shall be absent or disabled.
FR-PRV-004.BL		Upon a Teacher's request to initiate the peer review phase for an assignment, the system's BL shall verify: the authenticated Teacher, the assignment's submission deadline has passed, and submissions are closed, the peer review phase has not already been initiated for this assignment. The BL shall validate the provided configuration parameters. The system shall then automatically assign the specified number of unique peer reviewers to each submission, ensuring no student is assigned their own submission. The system shall create the necessary peer review pairing records in the data store (linking reviewer, reviewee's submission, and the assignment). The system shall update the status of the assignment to "Peer Review in Progress." The system shall make the assigned submissions and the confirmed rubric available to the assigned student reviewers. If any verification or validation fails, the system shall prevent the initiation of the peer review phase and provide appropriate error information to be surfaced by the UI. Once an assignment's status is "Peer Review in Progress," the business logic shall prevent any further modifications to its peer review configuration (number of reviewers, assignment mode, pairings) or rubric. The only way to alter this state via Teacher action is through assignment deletion.
FR-PRV-005	S	The system shall notify students on email when they have been assigned peer reviews.

Table 3.13: Teacher Peer Review Creation Functional Requirements (4-5).

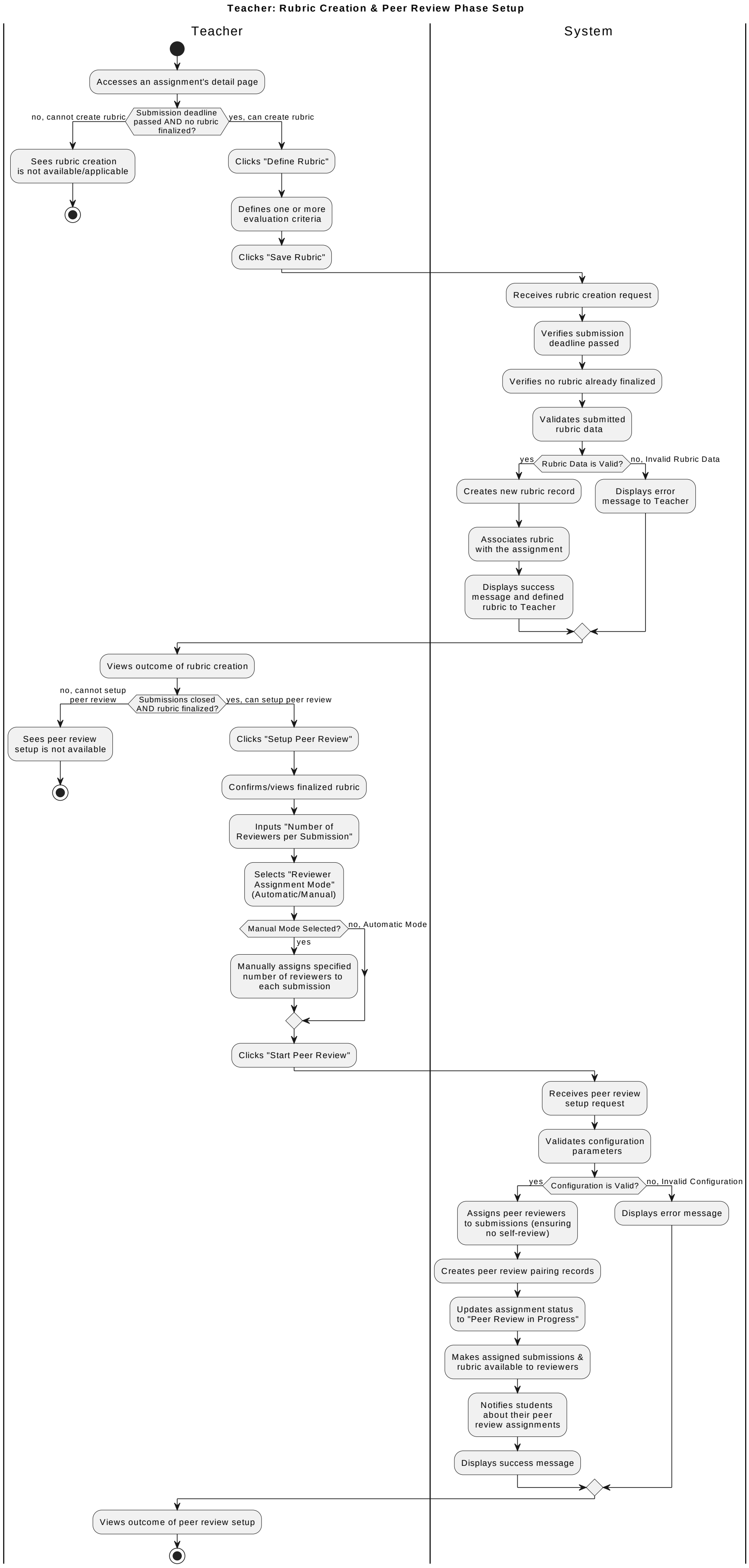
Teacher: Rubric Creation & Peer Review Phase Setup
Teacher
System
Accesses an assignment's detail page
Submission deadline passed AND no rubric finalized?
no, cannot create rubric
yes, can create rubric
Sees rubric creation is not available/applicable
Clicks "Define Rubric"
Defines one or more evaluation criteria
Clicks "Save Rubric"
Receives rubric creation request
Verifies submission deadline passed
Verifies no rubric already finalized
Validates submitted rubric data
Rubric Data is Valid?
yes
no, Invalid Rubric Data
Creates new rubric record
Displays error message to Teacher
Associates rubric with the assignment
Displays success message and defined rubric to Teacher
Views outcome of rubric creation
Submissions closed AND rubric finalized?
no, cannot setup peer review
yes, can setup peer review
Sees peer review setup is not available
Clicks "Setup Peer Review"
Confirms/views finalized rubric
Inputs "Number of Reviewers per Submission"
Selects "Reviewer Assignment Mode" (Automatic/Manual)
Manual Mode Selected?
no, Automatic Mode
yes
Manually assigns specified number of reviewers to each submission
Clicks "Start Peer Review"
Receives peer review setup request
Validates configuration parameters
Configuration is Valid?
yes
no, Invalid Configuration
Assigns peer reviewers to submissions (ensuring no self-review)
Displays error message
Creates peer review pairing records
Updates assignment status to "Peer Review in Progress"
Makes assigned submissions & rubric available to reviewers
Notifies students about their peer review assignments
Displays success message
Views outcome of peer review setup

Figure 3.4: Teacher Peer Review Creation Activity Diagram.

3.1.6 Reviewing - Student

ID	Priority	Requirement
FR-REV-001	M	The system shall enable authenticated Students to access and view the specific assignment submissions that have been allocated to them for peer review, along with the associated evaluation rubric, once an assignment has entered the peer review stage.
FR-REV-001.UI		When an assignment is in the ”Peer Review” stage, the system's UI shall present to an authenticated Student a dedicated section or page listing all submissions currently assigned to them for review for that specific assignment, including for each submission an identifier (the submitting student's name must not be visible), the title, the deadline and the status of the review (“In progress”, “Completed”). The UI shall provide a button to view/access the evaluation Rubric associated with the parent assignment from the list view. From the list of assigned submissions, the Student shall be able to select an individual submission to open a detailed review interface for that specific submission, including the submitted text content and a list of any attached files, allowing the Student to view or download them.
FR-REV-001.BL		When an authenticated Student requests to view submissions assigned to them for review for a specific assignment that is in the ”Peer Review” stage the business logic shall query the data store to identify all submission records specifically allocated to this Student for review within that assignment, and for each allocated submission retrieve the necessary data for display as specified in the FR-REV-001.UI. The business logic shall retrieve the evaluation Rubric associated with the parent assignment when required. When the Student requests to view a single assignment, the business logic shall retrieve the content of the assignment (text and file attachments).
FR-REV-ANON-001	M	For each submission assigned to a student for peer review of a specific assignment, any personally identifiable information (PII) of the submitter must be concealed from the reviewer.

Table 3.14: Student Reviewing Functional Requirements (1).

ID	Priority	Requirement
FR-REV-002	M	The system shall enable authenticated Students to conduct and submit a peer review for an assigned submission, requiring them to provide a score (within the rubric's defined range) and a textual justification for each criterion in the associated evaluation rubric. Submission of the review shall only be possible once all required criteria are addressed.
FR-REV-002.UI		Within the detailed review interface for a single submission the system's UI shall provide a button for the Student to initiate the review input process. Once the review process is initiated, the UI shall display each criterion from the assignment's associated rubric. For each criterion, the UI shall provide its description, an input mechanism to provide a score, which is validated against a score range defined by the rubric, and a text input area to provide a textual explanation for the given score. The UI shall provide a button to submit the completed peer review. This button shall be disabled or result in a validation error if any rubric criterion has not been assigned a score or lacks a justification. Upon successful submission of the review, the UI shall provide clear visual feedback and shall upgrade the status of the review task in the student's list.
FR-REV-002.BL		Upon a Student's attempt to submit a peer review for an assigned submission: the system's business logic shall verify that the Student is authenticated and is assigned to review the target submission, and retrieve the associated rubric for the assignment. For each criterion in the rubric the business logic shall validate that a score and a textual justification have been provided, and that the score falls within the valid range specified in the rubric. If all validations pass for all criteria, the business logic shall create a new peer review record in the data store, that includes the provided scores and justifications, the identity of the reviewer, the original author of the work being reviewed and the timestamp of the review. The business logic shall update the status of the review task (to "Completed") if the validation is successful.

Table 3.15: Student Reviewing Functional Requirements (2).

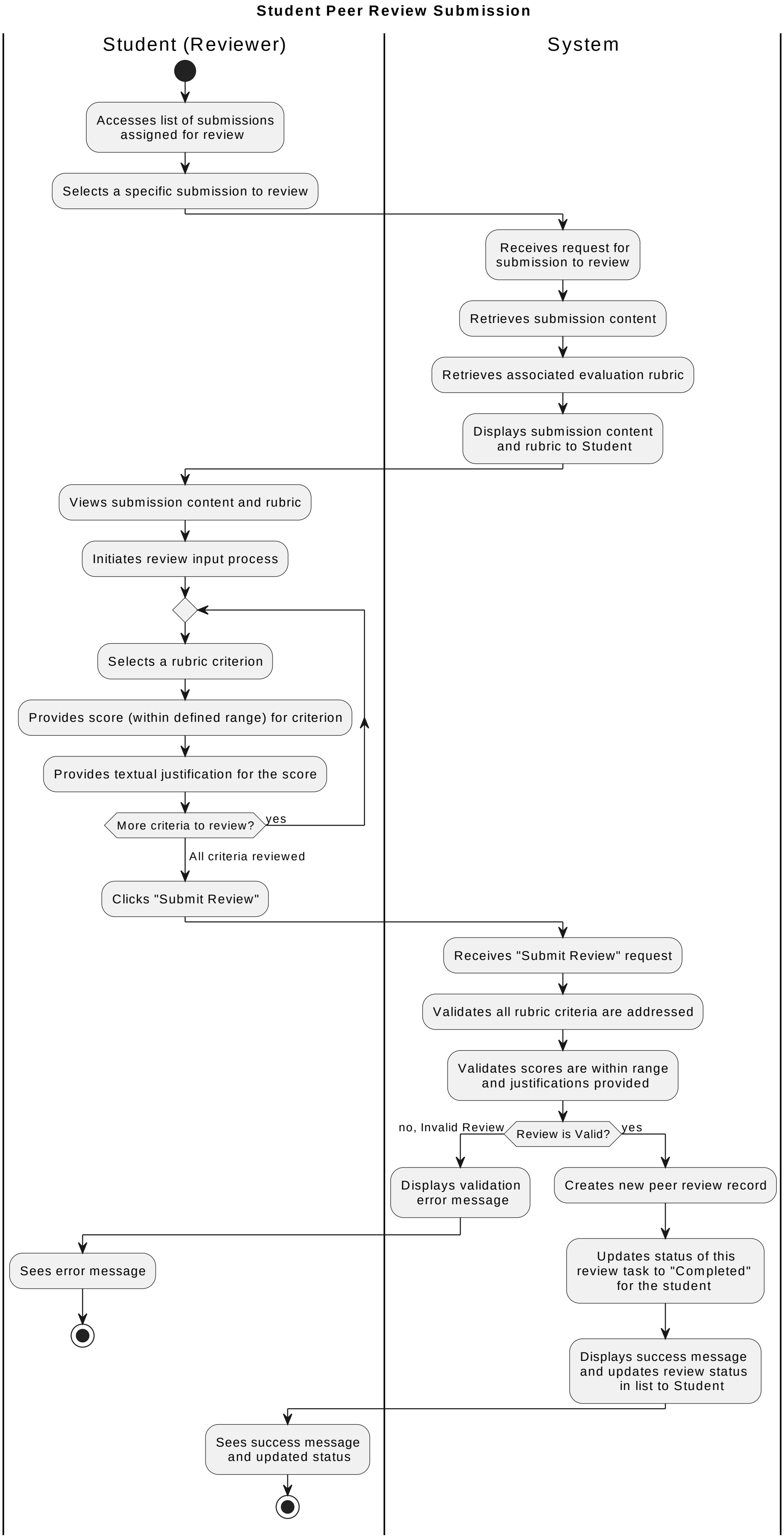
Student Peer Review Submission
Student (Reviewer)
System
Accesses list of submissions assigned for review
Selects a specific submission to review
Receives request for submission to review
Retrieves submission content
Retrieves associated evaluation rubric
Displays submission content and rubric to Student
Views submission content and rubric
Initiates review input process
Selects a rubric criterion
Provides score (within defined range) for criterion
Provides textual justification for the score
More criteria to review?
yes
All criteria reviewed
Clicks "Submit Review"
Receives "Submit Review" request
Validates all rubric criteria are addressed
Validates scores are within range and justifications provided
Review is Valid?
no, Invalid Review
yes
Displays validation error message
Creates new peer review record
Sees error message
Updates status of this review task to "Completed" for the student
Displays success message and updates review status in list to Student
Sees success message and updated status

Figure 3.5: Student Peer Review Submission Activity Diagram.

3.1.7 Results & Monitoring - All Users

ID	**Priority**	**Requirement**
FR-RES-001	M	The system shall enable authenticated Teachers to monitor the submission status for assignments, particularly when the assignment is in a stage where submissions are expected, or when the deadline is expired and the peer review process is not started yet (e.g., ”assigned”, ”preparing review”). This includes viewing aggregate submission counts and details of individual student submissions.
FR-RES-001.UI		The system's UI shall provide a dedicated section or view, accessible to authenticated Teachers, when the assignment status is “assigned” or “preparing review” (prior to peer review distribution). The UI shall display the total number of students assigned to the assignment, the number of students who have submitted their work, and a list of submitted assignments, showing the student's name, the submission status and a timestamp of the submission.
FR-RES-001.BL		When an authenticated Teacher requests the detailed information of an assignment for which the peer review process is not started yet, the business logic shall retrieve the list of all students assigned to participate in this assignment, and determine their submission status by checking for a corresponding submission record; if it exists, retrieve the submission timestamp. The business logic shall calculate the aggregate statistics (total assigned, total submitted) as required by FR-RES-001.UI, and return them to the view.

Table 3.17: Results & Monitoring Functional Requirements (1).

ID	Priority	Requirement
FR-RES-002	M	The system shall enable authenticated Teachers to monitor the progress of peer reviews for assignments, specifically when the assignment is in the ”Under Review” stage. This includes viewing aggregate review completion statistics and detailed progress for individual reviewers and submissions.
FR-RES-002.UI		When an assignment is in the ”Peer Review” stage, the system's UI shall provide a dedicated section or view, showing the total number of expected peer reviews, number of peer reviews completed, total number of submissions requiring peer review, number of submissions for which all assigned peer reviews have been completed. The UI shall display a detailed list of all students assigned as reviewers for this assignment. For each reviewer, the list shall indicate the reviewers name or identifier, number of reviews assigned to them and number of reviews they have completed. The UI may provide an option to view a more detailed breakdown per submission, showing which reviewers are assigned to it and the status of each of those specific reviews.
FR-RES-002.BL		Upon a request from an authenticated Teacher to monitor the peer review progress for a specific assignment (and which is in the ”Under Review” stage), the business logic shall retrieve all peer review allocation records for the assignment (i.e., which student is assigned to review which submission) and calculate the aggregate statistics based on the status of the peer reviews. The business logic shall also calculate for each submission how many assigned reviews it has received and how many are completed. The business logic shall aggregate data for each reviewer, calculating the total reviews assigned to them and the number they have completed. Finally, the necessary data shall be returned to the view.

Table 3.19: Results & Monitoring Functional Requirements (2).

ID	Priority	Requirement
FR-RES-003	M	The system shall, after the peer review period for an assignment is completed, aggregate evaluation results and enable authenticated Teachers to view both overall assignment statistics (e.g., average scores per criterion, score distributions) and detailed reports for each individual student's submission, including the original submission content and all peer reviews received (scores, justifications, and reviewer identities).
FR-RES-003.UI		Once the peer review period for an assignment created by an authenticated Teacher is completed ("Completed" state), the system's UI shall provide a dedicated section or view for accessing peer review results. It shall show aggregated statistics for the entire assignment, including the average score for each criterion in the rubric, score distribution visualizations and the overall average or aggregated score. Alongside the overall results view, the UI shall display a list of all student submissions for the assignment. It shall include for each submission the name of the student who authored it, the number of peer reviews completed for this submission versus the number assigned, an aggregated or average score for the single submission, and a button to access a detailed report view. This view shall include the original submitted text content and associated files, and a section for each peer review received by the submission, including the name of the student who performed the review and score and explanation for each criterion in the rubric. The UI shall provide clear navigation to return from the detailed submission report view back to the overall assignment results view or the submission list.
FR-RES-003.BL		Upon a request from an authenticated Teacher to view peer review results for a specific assignment they created, The business logic shall retrieve all completed peer review records associated with all submissions for that assignment (including scores and explanations). For each criterion in the assignment's rubric, the business logic shall calculate the average score across all reviews for all submissions, and the distribution of scores, and finally return this data to the view. For the submission list and detailed report view, the business logic shall retrieve the list of all original submissions for the assignment, along with the identity of the student author for each, and associate the submission to its reviews.

ID	Priority	Requirement
		When a Teacher requests the detailed report for a specific submission, the business logic shall retrieve the content of the submission, and the data associated with each review, including the name of the student who performed that review, and finally return all this data to the view.
FR-RES-004	M	The system shall allow Students to view the aggregated evaluations and motivated feedback they received on their own submissions, after the review process is completed and results are released.
FR-RES-004.UI		Once peer review results for an assignment are officially released, the system's UI shall provide a dedicated section or view for an authenticated Student to access the feedback received on their submission for that assignment. This initial view for their submission shall display the student's own submitted work, average score for each criterion in the rubric, and an average or aggregated score for their submission. The UI shall also display a list of all the individual peer reviews, including the name of the reviewer, and average or aggregated score given in that specific review. Upon selecting a review from the list, a new view shall display for each criterion the score and explanation. The UI shall provide clear navigation to return from the detailed individual review view back to the summary feedback view or the list of reviews.
FR-RES-004.BL		Upon a request from an authenticated Student to view feedback received on their submission for a specific assignment (for which results have been released), the business logic shall retrieve all completed peer review records associated with the Student's specific submission for that assignment, and the Student's original submission content (text and file references). The business logic shall calculate an overall average or final aggregated score for the Student's submission based on a predefined aggregation method. For each peer review record retrieved (in 1.b), the business logic shall prepare the data for display, including: the identity (e.g., name) of the Student who performed the review, the scores and textual justifications provided for each criterion by that reviewer.

Table 3.22: Results & Monitoring Functional Requirements (3-4).

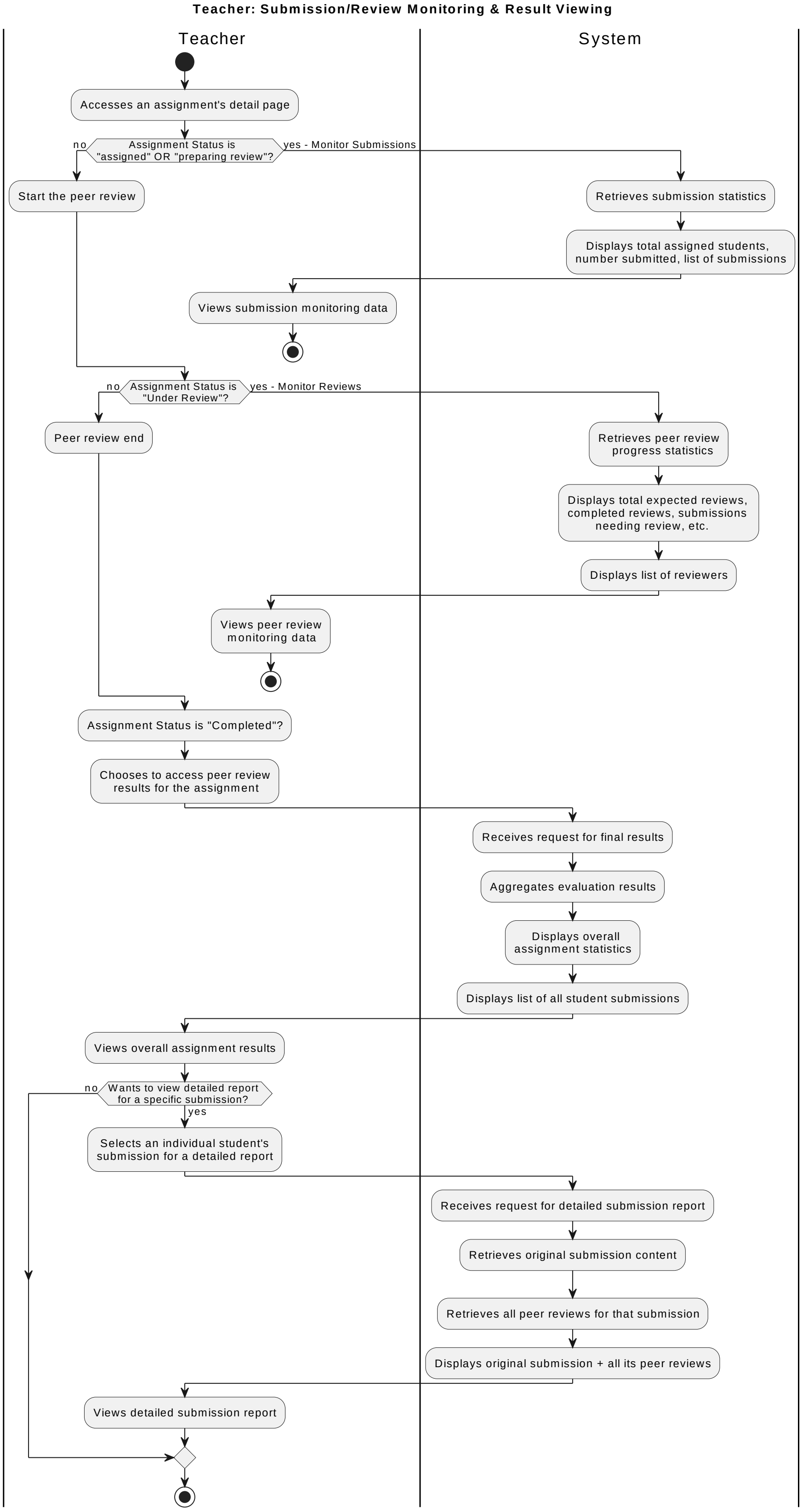
Teacher: Submission/Review Monitoring & Result Viewing
Teacher
System
Accesses an assignment's detail page
Assignment Status is "assigned" OR "preparing review"?
no
yes - Monitor Submissions
Start the peer review
Retrieves submission statistics
Displays total assigned students, number submitted, list of submissions
Views submission monitoring data
Assignment Status is "Under Review"?
no
yes - Monitor Reviews
Peer review end
Retrieves peer review progress statistics
Displays total expected reviews, completed reviews, submissions needing review, etc.
Displays list of reviewers
Views peer review monitoring data
Assignment Status is "Completed"?
Chooses to access peer review results for the assignment
Receives request for final results
Aggregates evaluation results
Displays overall assignment statistics
Displays list of all student submissions
Views overall assignment results
Wants to view detailed report for a specific submission?
no
yes
Selects an individual student's submission for a detailed report
Receives request for detailed submission report
Retrieves original submission content
Retrieves all peer reviews for that submission
Displays original submission + all its peer reviews
Views detailed submission report

Figure 3.6: Teacher Monitoring Review Activity Diagram.

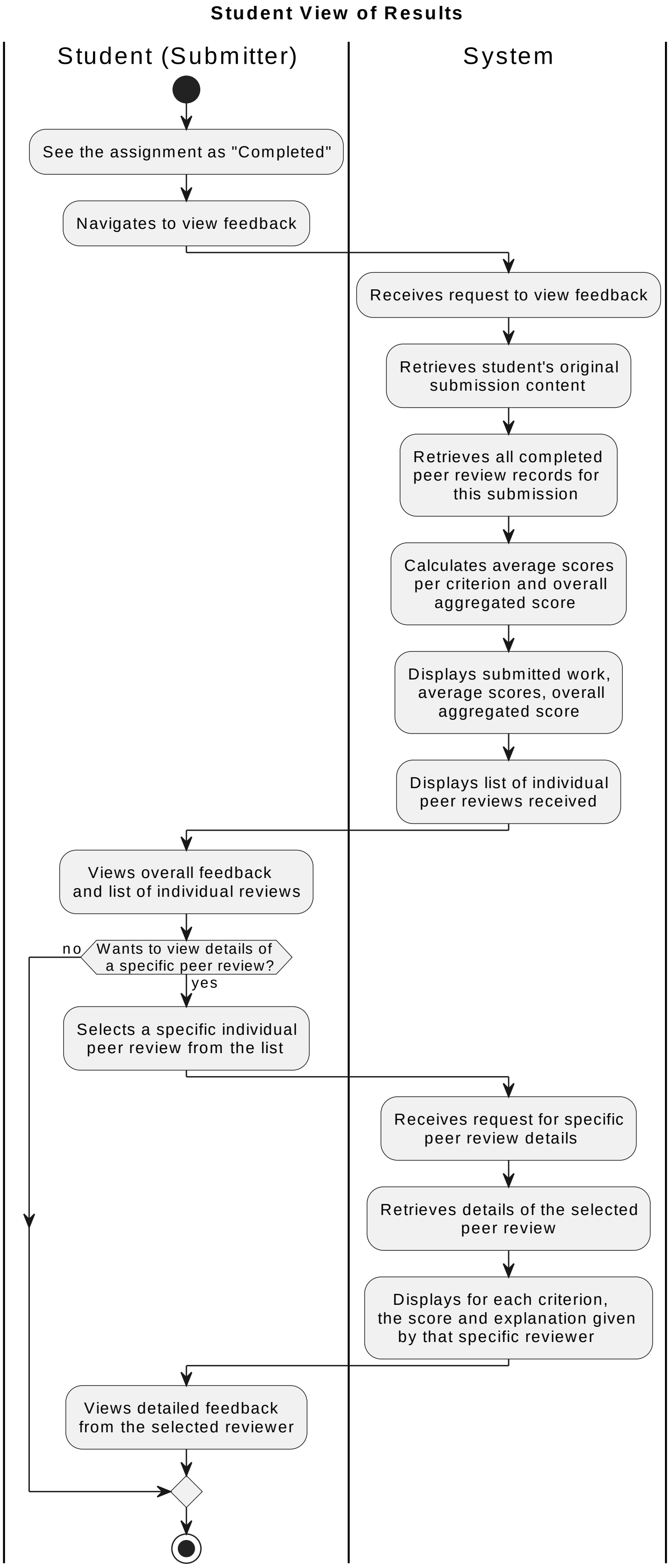
Student View of Results
Student (Submitter)
System
See the assignment as "Completed"
Navigates to view feedback
Receives request to view feedback
Retrieves student's original submission content
Retrieves all completed peer review records for this submission
Calculates average scores per criterion and overall aggregated score
Displays submitted work, average scores, overall aggregated score
Displays list of individual peer reviews received
Views overall feedback and list of individual reviews
Wants to view details of a specific peer review?
no
yes
Selects a specific individual peer review from the list
Receives request for specific peer review details
Retrieves details of the selected peer review
Displays for each criterion, the score and explanation given by that specific reviewer
Views detailed feedback from the selected reviewer

Figure 3.7: Students Receive Results Activity Diagram.

3.1.8 Use Case Diagram

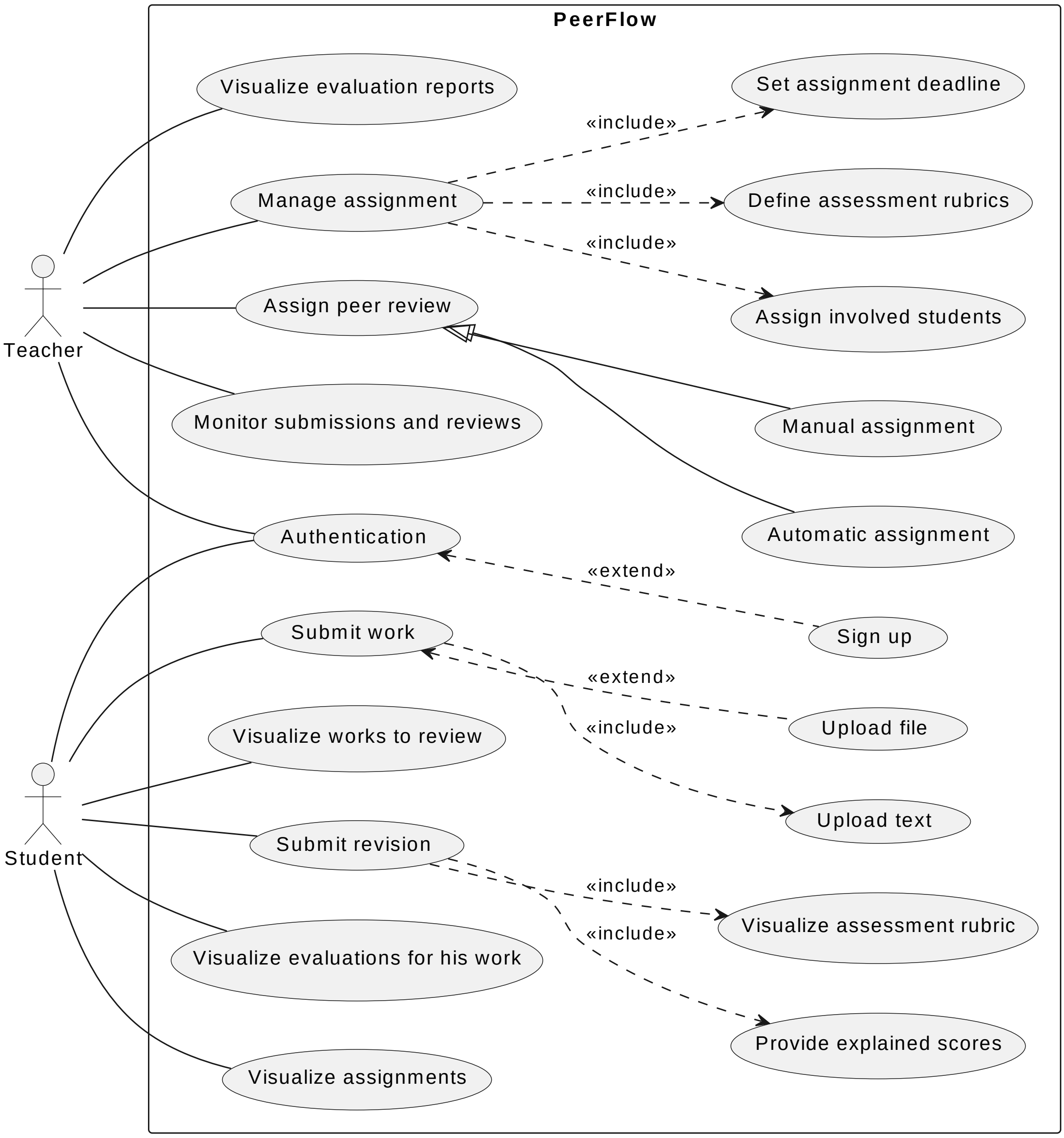

Figure 3.8: Complete Use Case Diagram

3.2 Non Functional Requirements

This subsection specifies the quality attributes that the PeerFlow system must exhibit. Unlike functional requirements that describe what the system does, non-functional requirements describe how well the system performs its functions. These attributes are critical for the overall success and user satisfaction of a MOOC-like platform. The quality attribute requirements will be specified using QA Scenarios:

1. **Source of Stimulus**: some entity (a human, a computer system, etc.) that generated the stimulus.
2. **Stimulus**: an event that requires a response by the system.
3. **Artifact**: the part of the system that is stimulated (may be the whole system).
4. **Environment**: the circumstances in which the scenario takes place (system in normal state, overloaded, shutting down, ...).
5. **Response**: the activity undertaken by the system (for runtime QA) or by developers (for dev. QA) as the result of the stimulus.
6. **Response Measure**: how to measure whether the response is satisfactory.

3.2.1 Modifiability

Scenario QA-MO-01: Modifying the Student Data Model

- **Source:** Developer.
- **Stimulus:** The project team decides to add a new mandatory field (e.g., ”Student ID Number”) to the student data model, requiring updates to the registration form and database query logic.
- **Artifact:** The Authentication and Profiling microservice (for user profile and registration) and potentially other services interacting with student data (e.g., Course Management, Assignment Management).
- **Environment:** The system is in the Development environment.
- **Response:** The student data model is successfully updated, the registration form is modified to include the new field, and all database queries for student information are adapted to handle the change without errors.
- **Response Measure:** The modification, including schema changes, form updates, and logic adjustments, is completed and unit-tested within 6 hours of effort, affecting a localized set of files within the relevant microservices.

Scenario QA-MO-2: Adding a New Submission File Type

- **Source:** Product Owner.
- **Stimulus:** The product owner decides to allow students to submit an additional file type (e.g., .zip for code archives) for assignments.

- **Artifact:** The Assignment Management microservice, specifically the file submission and storage components.
- **Environment:** The system is in the Development environment.
- **Response:** The system successfully accepts and stores the new file type, and the UI displays it correctly.
- **Response Measure:** The change to support the new file type is made and tested within 4 hours.

3.2.2 Integrability

Scenario QA-IN-1: Integrating a Basic Student Notification Service

- **Source:** Mission/system stakeholder.
- **Stimulus:** The project team decides to integrate a new, dedicated notification service to send basic email or in-app notifications to students regarding new assignment availability and peer review assignments.
- **Artifact:** The Assignment Management and Peer Review Management microservices (as triggers) and a new Notification microservice (as the provider of the notification functionality).
- **Environment:** The system is in the Development environment.
- **Response:** The Assignment Management and Peer Review Management services successfully trigger notifications via the new Notification service, and students receive timely and accurate alerts.
- **Response Measure:** The integration, including defining and implementing the Notification service's API, and modifying the triggering services, is completed and end-to-end tested within 8 hours, with notification delivery verified for correctness.

Scenario QA-IN-2: Enabling a Basic Health Check Endpoint for Service Discovery

- **Source:** Developers.
- **Stimulus:** A requirement arises to expose a basic health check endpoint for each microservice to facilitate automated service discovery and monitoring within the Kubernetes cluster.
- **Artifact:** All existing microservices, requiring the addition of a simple HTTP endpoint.
- **Environment:** The system is in the Development environment or Integration environment.
- **Response:** Each microservice provides a functional /health endpoint that returns a successful status (e.g., HTTP 200 OK), allowing Kubernetes to correctly determine service availability.
- **Response Measure:** The implementation of the health check endpoint for all microservices is completed and verified within 4 hours, requiring only the addition of a basic endpoint and no complex logic.

3.2.3 Deployability

Scenario QA-DE-1: Rolling Back a Malfunctioning Peer Review Management Service

- **Source:** System administrator.
- **Stimulus:** A newly deployed version of the Peer Review Management microservice exhibits unexpected behavior and high latency in the production environment.
- **Artifact:** The Peer Review Management microservice.
- **Environment:** The Production environment.
- **Response:** The system is rolled back to the previous stable version of the Peer Review Management service, and normal operation is restored.
- **Response Measure:** The rollback is initiated and completed within 10 minutes, with no data loss or negative impact on other services.

3.2.4 Availability

Scenario QA-AV-1: Service Instance Failure

- **Stimulus Source:** Unexpected failure (e.g., process crash, hardware issue on the VM) of a Review Service instance.
- **Stimulus:** The service instance stops responding to requests.
- **Artifact:** The Review Service.
- **Environment:** Normal operation, during student review processing.
- **Response:** The orchestrator detects the instance failure and automatically redirects traffic to healthy instances. A new instance is launched to replace the failed one.
- **Response Measure:**
  - Failover to a healthy instance occurs within 30 seconds of failure detection.
  - Users in session who were interacting with the failed instance might experience a single error, but subsequent attempts are successful.

3.2.5 Performance

Scenario QA-PE-1: Assignment Page Loading

- **Stimulus Source:** A student clicks to view the details of an assignment.
- **Stimulus:** HTTP GET request for the assignment description page, which includes rubrics and attachments.
- **Artifact:** The User Interface and the Assignment Service.
- **Environment:** Normal operation, with an average load of 50,000 concurrent users.
- **Response:** The system retrieves assignment information, rubrics, and attachment metadata, and renders the page in the student's browser.

- **Response Measure:**
  - Server-side response time (Time To First Byte - TTFB) for assignment data is less than 300 ms for the 95th percentile of users.
  - Total page load time (Largest Contentful Paint - LCP) is less than 2.5 seconds for the 75th percentile of users.

Scenario QA-PE-2: Saving a Review

- **Stimulus Source:** A student submitting a completed review for a peer's assignment.
- **Stimulus:** HTTP POST request containing review data (text, scores according to rubric).
- **Artifact:** The User Interface and the Review Service.
- **Environment:** Normal operation, peak review activity (e.g., day before review deadline).
- **Response:** The system validates and saves the review data in the database and confirms the operation to the student.
- **Response Measure:**
  - The processing time for the save request and confirmation to the user is less than 1 second for the 90th percentile of users.
  - In case of failed validation, feedback is provided within 500 ms.

Scenario QA-PE-3 - Scalability: Submission Peak

- **Stimulus Source:** A large number of students (e.g., 100,000) attempting to submit their assignments concurrently.
- **Stimulus:** Multiple file submission requests (text and attachments) within the last 30 minutes before the deadline of a popular assignment.
- **Artifact:** The Assignment Service and the underlying storage infrastructure.
- **Environment:** Normal operation, but during a projected peak load period.
- **Response:** The system accepts all submissions without errors, horizontally scaling Assignment Service instances and necessary storage resources to handle the load.
- **Response Measure:**
  - All submission requests are successfully processed within 1 minute of reception.
  - The error rate for submissions remains below 0.01%.

Scenario QA-PE-4 - Scalability: Increase in Registered Users

- **Stimulus Source:** The registration of 1 million new users to the MOOC platform over a week following a promotional campaign.
- **Stimulus:** Gradual but significant increase in the number of active users, profile creation requests, and general platform interactions (assignment viewing, etc.).
- **Artifact:** All services, with particular emphasis on the Authentication Service and Assignment Service.
- **Environment:** Normal operation, with a trend of increasing load.
- **Response:** The system handles the increased number of users and their activities while maintaining performance and availability levels, scaling resources transparently.
- **Response Measure:**
  - Average response time for login remains less than 2 seconds.
  - Average response time for viewing assignment information remains less than 2 seconds.
  - The platform supports a 50% increase in concurrent users (from 500,000 to 750,000) without degradation of key performance metrics, with a proportional increase in resources (not exceeding a 60% increase).

3.2.6 Testability

Scenario QA-TE-1: Unit Test Coverage

- **Stimulus Source:** Developers, QA
- **Stimulus:** A new feature is integrated or an existing code section is modified.
- **Artifact:** The affected code module(s).
- **Environment:** Development.
- **Response:** The execution of unit tests must achieve a specified percentage of line code coverage, calculated using a standard code coverage tool.
- **Response Measure:** The overall line code coverage for unit tests must be equal to or greater than 80%.

List of Figures

List of Tables