Charter Scope and Requirements Document

Project: Peer Review Application by Learnification Technologies

University Of British Columbia Okanagan

COSC 499 - Summer 2023

Date: May 29th, 2023

Group Members:

Charlotte Zhang (charlottezhang6@gmail.com)

Shila Rahman (shilarahman16@gmail.com)

Prabhmeet Singh Deol (arsh.appleid@gmail.com)

Lance Xu (lance924852785@gmail.com)

Sehajvir Singh Pannu (pannusehajvir@gmail.com)

TABLE OF CONTENTS

1. Project Purpose

The purpose of this peer review application is to foster a collaborative learning environment, empowering students to learn from each other and simplifying the management and evaluation of assignments for instructors. This enables students to provide constructive and helpful feedback to each other.

2. Project Objective

The objective of this project is to develop a user-friendly platform that facilitates the entire assignment and evaluation process for students and instructors. The goal is to create a seamless system where students can easily submit their assignments and receive timely feedback. Additionally, students will have the opportunity to evaluate a set number of assignments assigned by the instructor, providing valuable input on their peers' work. The platform will also enable students to evaluate the contributions of their fellow group members for group assignments. Moreover, students themselves will be evaluated by their peers in a fair and anonymous manner. This peer evaluation process aims to promote collaboration, accountability, and the development of valuable skills. For instructors, the platform will provide a convenient interface to create assignments, manage evaluations, and view feedback. Instructors will have a comprehensive overview of students' progress, enabling them to assess individual and group performance effectively. Overall, the project's objective is to create an inclusive and interactive learning environment where students can submit, evaluate, and receive feedback on their assignments, while instructors have the necessary tools to manage assignments, evaluations, and track student progress. The platform will empower students to engage actively in the learning process and foster a collaborative and enriching educational experience.

3. Success Metrics

a. Minimum number of clicks to accomplish a task.
b. Students can perform the processes of peer evaluation on the platform.
c. Instructors can gather all the evaluations provided by the students for individual assignments and group assignments.

d. Both students and instructors had a good experience after using the platform features.
e. Stakeholders recognize that the final project product meets their expectations.

4. High-level Requirements

a. The project aims to develop a comprehensive and user-friendly platform that facilitates efficient management of student assignments, evaluations, and course information.
b. The system should provide students with easy access to course details, clear assignment tracking, streamlined evaluation processes, and seamless communication.
c. Additionally, instructors should have robust capabilities for managing assignments, evaluations, and student information, allowing for flexibility in grading, team/group management, and account administration.
d. The project seeks to enhance the overall learning experience by promoting effective assignment management, timely feedback, and collaborative evaluation among students and instructors.

5. Assumptions And Constraints

Assumptions:

a. The developers will be able to code in React.js and Node.js
b. The capstone team is able to complete all the deliverables in the given timeframe.
c. Stakeholder and team availability is not an issue.
d. Developers will not run into any computer problems.
e. The Project Manager can arrange the schedule reasonably and monitor the product quality effectively.
f. Minimal change of project requirements to finish on time.

Constraints

a. Time constraint
b. Developers need to learn new concept and language
c. Clients and teams availability

6. High-Level Project Description and Boundaries

a. The platform focuses on students anonymously evaluating other students' assignments and giving feedback. It will not include features unrelated to evaluations, such as course content delivery or student enrollment management.

b. The system will support a predefined set of questions determined by the instructor for evaluations, including multiple-choice, short answer, and rating scale questions. It will not support additional question types that are not within the defined scope.

c. The system will allow instructors to manage evaluations and view student performance. It will not provide administrative features user management, course scheduling, or grade management.

d. The system will be developed for use on web browsers and will not be optimized for specific mobile platforms or devices.

e. The project will focus on the development of the evaluation system and its core functionalities. It will not include extensive customization options or integration with external systems beyond and defined scope.

f. The project will consider integrations with external systems or APIs as necessary for its functioning. However, the project's boundaries will be set to include only the essential integrations, and any additional integrations may be subject to further assessment and approval from the Sponsor.

7. High Level Risks

a. Limited number of hours to finish the project
b. Limited number of weekly client meeting
c. No in-person client meeting as the client is not in Kelowna

d. Developer(s) can't pick up React.js, Node.js, express.js to code functionality for the project.
e. Developer(s) might drop the class
f. Developer(s) might get sick and not be able to code in a timely fashion
g. Weak team communication

8. Summary Milestone Schedule

Milestone No.	Deliverables	Due Date
1	Draft - Charter, Scope and Requirement Document	May 29, 2023
2	Final Submission - Charter, Scope, Requirements and Planning Milestone Document	June 4, 2023
3	Draft - Design and Testing Document	June 4, 2023
4	Final Submission - Design and Testing Document	June 7, 2023
5	Design Presentation	June 9, 2023
6	Mid-project Status Update and Assessment and MVP Design Review Presentation	July 5, 2023
7	Testing Event	August 4, 2023
8	Practice Presentations and Feedback	August 11, 2023
9	Final Report and Code Submission and Presentations	August, 2023 (date TBA)

9. Stakeholders List

a. Parsa Rajabi (Project Sponsor)
b. Shila Rahman (Client Liasion + Frontend Developer)
c. Lance Xu (Full stack Developer)
d. Prabhmeet Singh Deol (Full Stack Developer, Integration Lead)
e. Sehajvir Singh Pannu (Full Stack Developer, Team Lead)
f. Charlotte Zhang (Project Manager + Frontend Developer)

g. University of British Columbia

10. Functional and non-functional requirements

Functional Requirements:

a. Functional Requirements for Students:

1. Account Management:

a. The system must provide a user registration feature that allows students to create an account and accurately capture and store the following information: first name, middle name, last name, student ID, email, and password.

a. The system must provide a secure login functionality for students to access the platform using their registered email and password, ensuring authentication and verification of credentials.

b. The system must provide students with the ability to reset their passwords in the event of forgetting or needing to change them, offering a secure and accessible password recovery process that ensures the privacy and security of user accounts.

c. The system must provide students with the ability to log out of the system, implementing appropriate security measures to ensure a secure disconnection and protect against unauthorized access.

d. The system should allow students to manage their account details on the platform, including their name, email, and password, enabling them to update and modify their personal information securely, ensuring control over their account and maintaining account security.

2. Assignment Management:

a. The system should provide students with the ability to view a comprehensive list of assignments on the platform, displaying relevant details such as assignment title, instructions, rubric, points, start and end dates, and accepted

submission links, ensuring easy access to assignment information for monitoring and tracking progress.

b. The system should allow students to view a list of 'x' number of other students' assignments, determined by the instructor when creating the assignment, for the purpose of anonymous peer evaluation.
c. The system should allow students to submit assignments only as links (e.g., Google Drive, Dropbox) on the platform, as specified by the instructor when creating assignments for students.
d. The system should provide students with the ability to edit their assignment submission on the platform before the deadline, enabling them to make necessary revisions or updates to their work.
e. The system must perform link validation to ensure the submitted links for assignments are valid, specifically checking if the URL contains the appropriate format specified by the instructor (e.g., for a Google Docs submission, the URL should contain 'https://docs.google.com/'). This validation process ensures the reliability and integrity of the submitted work.

3. Evaluation:

a. The system should provide all students with the ability to evaluate their teammates through a comprehensive survey that includes matrix questions, short and long answer questions, multiple answer questions, and multiple-choice questions. This allows for the collection of both quantitative and qualitative feedback.
b. The system should require each student within a team to provide feedback to all other members of their team, ensuring comprehensive and inclusive evaluation among team members.
c. If a student evaluating another student does not give a full mark on a question, the system should show a text input box where the student must provide a small comment specifying areas for improvement for the particular question. This ensures the provision of constructive feedback and opportunities for growth. The system should not allow a student to proceed until they have provided a comment feedback.

d. The system should allow students to view the scores and comments associated with their assignment evaluations, without revealing the identity of the reviewers. This ensures transparency and provides valuable feedback for self-assessment and improvement.

b. Functional Requirements for Instructors:

4. Instructor Account:

a. The system must allow instructors to log in securely to the system using their registered email and password, ensuring secure access to their account and associated functionalities.
b. The system should provide instructors with the ability to reset their passwords in case of forgetting or needing to change them, ensuring a secure and accessible password recovery process
c. The system should provide instructors with the ability to log out of the system, ensuring a secure disconnection and safeguarding their account from unauthorized access.
d. The system should provide instructors with access to a feature that allows them to manage their account details on the platform, ensuring control over their personal information and account settings.
e. The system should provide instructors with the ability to enroll students to a course by importing their student IDs using a CSV file. The CSV file format for student enrollment should follow a specific format to be determined.

5. Evaluation Team:

a. The system should allow instructors to create evaluation teams/groups by importing them using a CSV file. The CSV file should contain group names and the corresponding group members. The specific format for the CSV file will be determined.
b. The system must provide instructors with the ability to add or remove students from teams/groups, both during and after the group creation process. This feature ensures the necessary flexibility to manage team/group membership as required.

6. **Assignment Management :**
    a. The system must provide instructors with the ability to create a new assignment for a course. Instructors should be able to enter the assignment name, assignment marks, assignment rubric, assignment instructions, assignment start date, assignment end date, and select the type of link allowed for submissions.
    b. The system must provide instructors with the ability to set and change the final marks of assignments, ensuring flexibility in grading and accommodating adjustments, including the provision of 'fudge points'.

7. **Evaluation Survey Management:**
    a. Instructors must have access to create evaluation forms and select the questions to include, allowing them to design customized assessment criteria and collect specific feedback based on their evaluation requirements.
    b. Evaluations can ONLY be edited/updated BEFORE a student uses that criteria -- once the evaluation has been "used" by a single student, the instructor should not be able to modify it.
    c. Instructors must have the ability to toggle or set options that allow students to view the feedback they have received, providing control over the visibility and accessibility of feedback for student learning and improvement.
    d. Instructors have the authority to determine whether students can access their feedback in individual response format or as a summarized version, enabling control over the level of detail provided to students for their assessment feedback.
    e. The website must provide a visual analysis or representation of the feedback and marks given by students, clearly indicating the source of feedback and the recipients, facilitating a comprehensive overview of the evaluation process.
    f. The evaluation data must have the capability to be easily exported to Excel format, allowing for convenient analysis, storage, and further manipulation of the data as needed.

c. Functional Requirements for Super Admin

1. **Account Management:**
    a. The system should have a super admin account that is created using a postman call during the initial system boot. The super admin should have the privilege to grant access to other instructors within the system.

d. System Functional Requirements:

8. **Evaluation**
    a. Calculations must be performed for the quantifiable data derived from the feedback, allowing for the allocation of specific weightings to each question as set by the instructor, ensuring accurate assessment of scores based on predetermined weightings.
    b. The system should implement a notification mechanism, such as email or in-app notifications, to alert students when their feedback becomes visible to them. This feature ensures timely communication and enables students to access their assessment results promptly.

9. **Visibilty**
    a. The system should provide a clear and organized list of courses for students, displaying relevant details such as course name, course number, and course image. This feature ensures easy navigation and access to course information.
    b. The system should provide a clear and organized list of evaluations for students, displaying relevant details such as group assignment title and due date. This feature ensures easy access to evaluation information for monitoring and tracking progress.
    c. The system must provide a clear and organized list of grades for students that display relevant details for each assignment, such as assignment title, overall grades, view feedback (if the student being evaluated did not receive a full mark for some evaluation question).
    d. When instructors click on "View Feedback," the system must provide a preview of the assignment and a summary of the individual evaluation with

the corresponding student names in an instructor view, facilitating the assessment and review process for instructors.

e. When clicked on "Overall Grade," the system must provide a clear and organized rubric of assignments for students and instructors, displaying various evaluation criteria such as overall grade and different sections for example Section A (Team Performance), Section B (Respect and Behaviour) etc.

Non-Functional Requirements:

a. Security

a. The system must grant access to user accounts when users enter the correct username and password, ensuring secure authentication and authorized access to the platform.
b. The application must not create an account for a user until they create a strong password (At least 12 characters long but 14 or more is better. A combination of uppercase letters, lowercase letters, numbers, and symbols)

b. Compatibility

a. The system must be designed to run on computers and tablets, considering the impact of display screen size on the user interface (UI) to ensure optimal user experience and usability, while not requiring specific adaptation for mobile devices such as cell phones.

c. Localization

a. The time of assignments deadline must be based on the Instructors set time zone.

d. Usability

a. Users must be able to predict the meanings of icons, such as inferring that tapping a button with a picture of a magnifying glass will open a search bar, ensuring intuitive and user-friendly interface design.
b. The user interface elements, layouts, and interactions must be consistent across different devices and screen dimensions, ensuring a unified and seamless user experience regardless of the device used.

e. Accessibility

a. The web application must pass the audit test, such as Google Lighthouse, ensuring adherence to best practices, performance optimization, accessibility, and other relevant criteria for an optimal user experience.

Technical Requirements:

1. The front end system will be written in ReactJs, Next.js libraries and TailwindCss. These would be the most important frameworks for the front end.
2. Back end will be written using Node.js and Express.js following MVC design choice.
3. The database driver for this application will be MySQL.
4. The project must make use of Github as a version control system and Github Projects as a Kanban Board.
5. Github Actions will be used to run and build the CI/CD pipeline.
6. Every piece of code that is written should be associated with a ticket number using Github issues and the ticket number should be noted in the branch name of Github.
7. All github branches must go through pull request (PR) review and be reviewed by other team members and pass automatic tests before they're merged into the development branch for further development.
8. Development branch must be smoke-tests using automatic testing before PR'ed into production branch
9. Understanding of how docker containers work with docker networks.
10. API testing through postman.
11. Production environment setup through ssh.

User Requirements:

1. User Requirements for Students:

a. Account Management

i. The student shall be able to create a new account by providing their first name, middle name, last name, student ID, email, and password, ensuring a seamless and user-friendly registration process.

ii. The student shall be able to securely log in to the platform using their registered email and password, expecting a straightforward login process that protects the privacy and security of their account.

iii. The student shall have the option to reset their password in case of forgetting or needing to change it, expecting a user-friendly and easily accessible password reset process.

iv. The student must have the ability to easily access the logout functionality and receive confirmation of a successful disconnection, ensuring a secure disconnection from the system and safeguarding their account from unauthorized access.

v. The student shall have access to an easily accessible and intuitive account management feature that provides clear instructions for updating their name, email, and password details, ensuring a seamless and user-friendly experience when managing their account.

vi. Students must have access to a clear and organized list of courses within the system, displaying important details such as course name, course number, and course image. This user-friendly interface allows for easy navigation and quick access to relevant course information, enhancing the overall user experience for students.

b. Assignment Management

i. The student shall be able to easily access a comprehensive list of assignments on the platform, displaying assignment title, instructions, rubric, points, start and end dates, and accepted submission links. This

will allow them to conveniently monitor and track their progress, ensuring effective assignment management and submission.

ii. The student shall have the ability to access and review a list of 'x' number of other students' assignments, as determined by the instructor, to perform anonymous peer evaluation. This feature will facilitate the fair assessment and feedback exchange among students.

iii. The student must be able to submit assignments exclusively as links (e.g., Google Drive, Dropbox) on the platform, following the submission method defined by the instructor when creating assignments. This ensures a consistent and streamlined process for submitting assignments

iv. The student must have the option to edit their assignment submission on the platform before the deadline, allowing them to make required revisions or updates to their work. This feature ensures flexibility and the opportunity for students to improve their submissions within the given timeframe.

v. The student expects the system to perform link validation for the submitted assignment links, verifying that the URLs are valid and meet the specified format set by the instructor. This validation process enhances the accuracy and trustworthiness of the submitted work.

c. Evaluation

i. Students must have access to a clear and organized list of evaluations within the system, displaying important details such as group assignment title and due date. This user-friendly interface allows for easy access to evaluation information, enabling students to monitor and track their progress in completing evaluations for group assignments

ii. The student must be able to access and complete a comprehensive survey to evaluate their teammates, which includes matrix questions, short and long answer questions, multiple answer questions, and multiple-choice questions. This feature ensures that both quantitative and qualitative feedback can be provided, enabling a comprehensive evaluation process.

iii. The student must be mandated by the system to provide feedback to all other members of their team, ensuring a comprehensive and inclusive evaluation process within the team. This requirement fosters a collaborative and supportive environment where each team member receives feedback from their peers.

iv. When evaluating another student, if a full mark is not given on a question, the student must provide a small comment specifying areas for improvement for that particular question. This requirement ensures that constructive feedback is provided and opportunities for growth are identified. The system should enforce the completion of the comment feedback before allowing the student to proceed, promoting thorough and valuable evaluations

v. The student must have the ability to access and view the scores and comments provided by reviewers for their assignments. The feedback should be presented anonymously, without revealing the identity of the reviewers. This feature promotes transparency, enables self-assessment, and facilitates valuable feedback for personal improvement.

vi. Students must receive notifications, either through email or within the application, when their feedback becomes available. This notification mechanism ensures timely communication, allowing students to promptly access and review their assessment results. By providing timely feedback notifications, the system enhances the overall user experience for students.

2. User Requirements for Instructors

a. Instructor Account:

i. Instructors should have the ability to securely log in to the system using their registered email and password. This login feature ensures that instructors can access their account and utilize the functionalities provided by the system.

ii. Instructors must have the option to reset their passwords in case they forget or need to change them. The system should offer a secure and easily accessible password recovery process, enabling instructors to

regain access to their account while maintaining the security of their account credentials.

iii. Instructors must be able to easily log out of the system, ensuring a secure disconnection from their account and protecting it from unauthorized access. This feature promotes account security and privacy for instructors using the system.

iv. Instructors must have the ability to manage their account details on the platform, enabling them to control and update their personal information and account settings. This feature ensures that instructors have autonomy and control over their account within the system.

v. Instructors must have the option to enroll students to a course by importing their student IDs using a CSV file. The system would have a particular format that is acceptable for the CSV file, which will be used for importing and organizing student enrollment data. This feature facilitates efficient enrollment management for instructors.

b. Evaluation Team:

i. Instructors must have the ability to create evaluation teams/groups by importing them using a CSV file. The CSV file should follow a specific format and include group names and the members assigned to each group. This feature streamlines the process of creating evaluation teams and enhances organizational efficiency for instructors.

ii. Instructors must be able to add or remove students from teams/groups, allowing them the necessary flexibility to manage team/group membership during or after the initial group creation. This feature empowers instructors to make adjustments and updates to team/group compositions as needed.

c. Account Management:

i. Instructors must have access to a feature that allows them to create new assignments for a course. Instructors should be able to input the assignment name, assignment marks, assignment rubric, assignment instructions, assignment start date, assignment end date, and select the type of link allowed for submissions. This functionality empowers

instructors to efficiently design and manage assessments within the designated course.

ii. Instructors must have the ability to set and modify the final marks of assignments, allowing for flexibility in grading and accommodating any necessary adjustments, such as the provision of 'fudge points'. This feature empowers instructors to effectively manage the grading process, ensuring accurate assessment and grading outcomes.

11. Project Management Methodology and Workflow

1. Methodology: Kanban

The peer evaluation application project will be implemented using a Kanban methodology. Kanban focuses on visualizing and optimizing the flow of work, promoting a smooth and efficient project execution. The project will utilize a Kanban board to manage tasks, collaborate effectively, and continuously improve the workflow.

2. Workflow

1. Product Backlog Refinement:
    a. Collaborate with stakeholders to identify and prioritize features and user stories.
    b. Break down user stories into smaller tasks or requirements and estimate their effort.
    c. Continuously refine and update the product backlog.
2. Task Management
    a. Organized and Visualized user stories and tasks on a Kanban board with columns such as "To Do," "In Progress," "Testing," and "Completed."
    b. Pull tasks into the appropriate columns as work progresses.
3. Continuous Development:
    a. Create a new branch for each task or feature.
    b. Implement the functionality based on the task requirements and test cases.

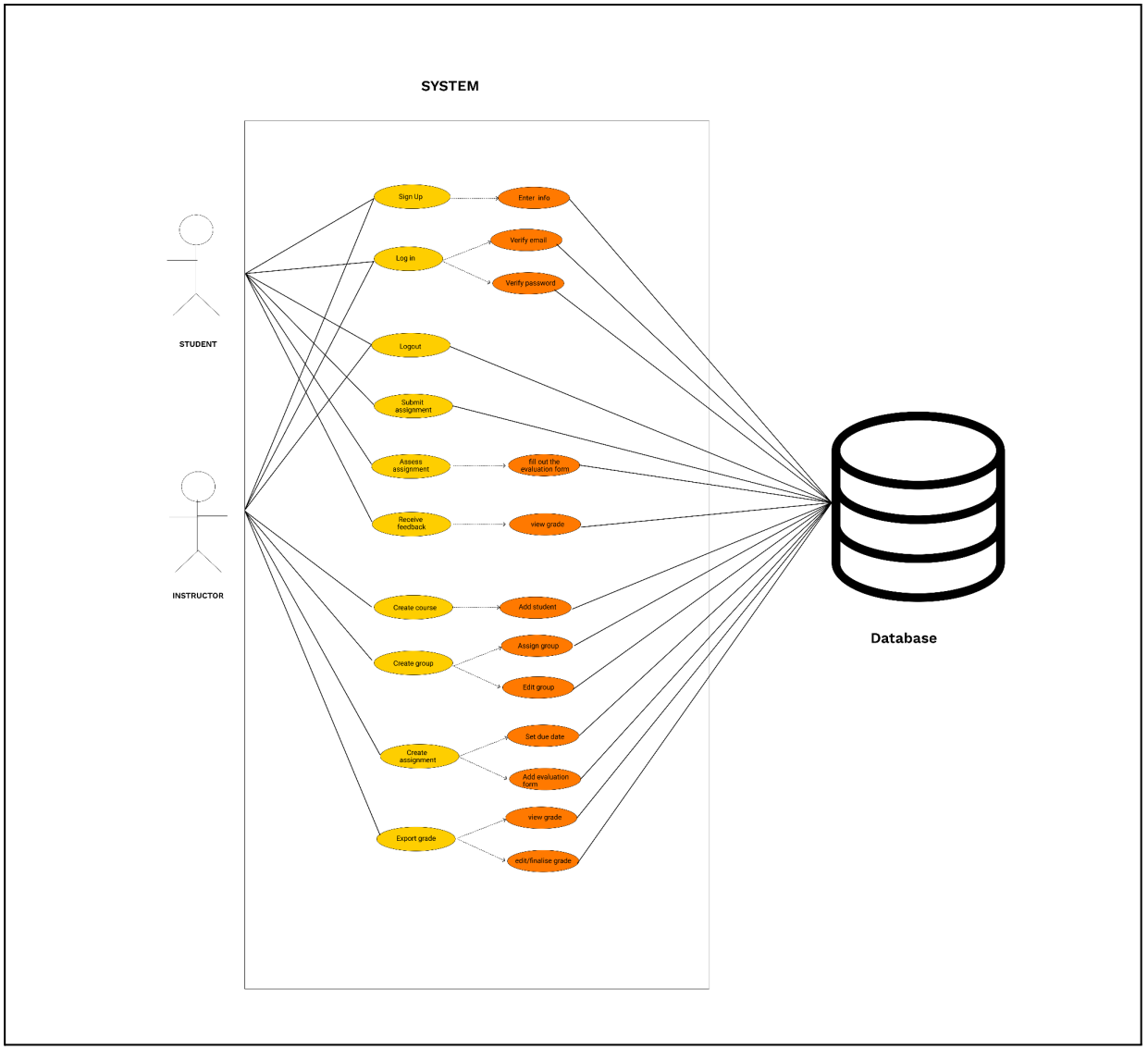
c. Conduct code reviews and seek feedback from team members for quality assurance.

4. Continuous Delivery:
    a. Perform thorough testing of the implemented functionality to verify its behavior and ensure it meets the requirements.
    b. Once a task is completed and thoroughly tested,address a pull request and merge the branch into the main development branch if code review reveals no problems.
    c. Deploy the updated code to the staging or production environment for evaluation once every 2 weeks.
5. Ongoing Collaboration:
    a. Hold regular meetings every Monday, Wednesday and Friday at 5:00 pm to review the Kanban board, discuss progress, and address any challenges.
    b. Foster effective communication and collaboration among team members.
    c. Continuously monitor and update the Kanban board to reflect the current status of tasks.
6. Continuous Improvement:
    a. Reflect on the project's progress and performance during retrospectives.
    b. Identify areas for improvement and formulate strategies for enhancing the workflow.

12. User Group and UML Use Case

Our platform has two main user groups:

1. **Students:** Students are able to login, submit assignment, assess assignment and receive feedback
2. **Instructors:** Instructors are able to login, create courses, add students, create groups, assign groups, edit grade, export grade and archive courses.



13. Work Breakdown Structure

Task List	Estimated Hours					Average Estimate
	Prabhmeet	Sehajvir	Shila	Lance	Charlotte	
Milestone1: Requirements and Project setup	19	36	27	28.5	25	101.5
Documentation						
Charter and scope		5	2	2	4	13
Software description	4	1		2		7
Set of the Criteria		1	3			4
Functional and Non-functional requirements		2		2	4	8
Technical requirements	2	2		1	1	6
User requirements				1	1	2
Description of workflow and methodology				2	3	5
UML case diagrams			5			5
Work breakdown structure				6		6
Monitor project progress and track milestones		3	2		6	11
User Interface template design		2	6			8
User cases				3	3	6
Design document introduction		2		0.5		2.5
User Personas			3		3	6
System Architecture		3				3
UI mockups		3	6			9
Use case diagram						0
System Architecture						0
Technical specification						0
Test plan		3				
Database Setup						
Setup and Configure the database server	2					2

Implement data models and database server	2					2
Relationship setup for tables	2	9		9		20
Project Setup	4					4
Docker container for the database	3					3
Milestone2	46	45	46	45	45	218
Documentation						
User stories analysis					4	4
Fix Project Charter with respect to Prefessor Feedback		6				
Tasks into different columns on the project board	2		4		4	10
User Personas			3			
Implementaions						
Login page	6			3		9
Signup page		6			3	9
Retrieve password	3			6		9
View list of courses registered			6	3		9
View list of Courses they are enrolled in	3				3	6
View list of Assignments		3		6		9
View list of evaluation surevys	3	6				9
View list of Assignment grades	3			3		6
Assignment submission		3			6	9
Feedback summary page				3	6	9
User Interface						
Login Signup page			2			2
Evaluation Form			3			3
Account summary page			2			2
Icons and clear naviagtion			2			2
Dashboard UI for course list		3	2	3		8

Dashboard UI for assignment list		3	3	3		9
Dashboard UI for assignment feedback list		3	3	3		9
Testing						
Setup testing framework	6		6			12
Define test cases					3	3
Write test scripts	6	6	6	6		24
Run tests	2	2		2		6
Ensure website displays correct data	1				3	4
Vallidate database functionalities	1				3	4
Review						
MVP demo	4	4	4	4	4	20
Feedback summary	6				6	12
Milestone3	50	51	51	51	51	254
Documentation						
User stories analysis					4	4
Tasks into different columns on the project board					4	4
Implementaions						
View list of Assignments	6		3			9
Assignment management		6		3		9
Evaluation survey management	3		3			6
View list of Evaluation teams		3			6	9
Evaluation team management	6	3				9
Feedback management	3		6			9
View of the feedback summary		3		3	3	9
Exporting Feedback		6		3		9
Notification system				6	3	9

Evaluation Form Matrix Question		3		6		9
Evaluation Form Ranking Question			3		6	9
Evaluation Form Short and long answer question	3			6		9
Evaluation Form Check box questions	3		3		3	9
Evaluation Form Multiple choice questions		3		6		9
Quantifiable data calculation	2	4	4	2	4	16
User Interface						
Assignment management page			1			1
Evaluation survey management page			1			1
Evaluation team management page			1			1
Feedback management page			1			1
Feedback summary page			1			1
Compatibility with different screen size	2	2	1	2		7
Evaluation Form Matrix Question			1			1
Evaluation Form Ranking Question			1			1
Evaluation Form Short and long answer question			1			1
Evaluation Form Check box questions			1			1
Evaluation Form Multiple choice questions			1			1
Testing						
Define test cases	6				6	12
Write test scripts	6	6	6	6		24
Run tests	4	4	4	4		16

Ensure website displays correct data		4	4		4	12
Vallidate database functionalities	2				4	6
Review						
MVP demo	2	2	2	2	2	10
Feedback summary	2	2	2	2	2	10
Milestone4: Final Presentation	35	35	35	35	36	176
Documentation						
User stories analysis					5	5
Tasks into different columns on the project board					5	5
Implementaions						
Bug fixing	9	9	9	9		36
Finalizing the implementation	8	8	8	8		32
Testing						
Setup testing framework					3	3
Define test cases					3	3
Write test scripts	6	6	6	6		24
Run tests	4	4	4	4		16
Ensure website displays correct data					4	4
Vallidate database functionalities					4	4
Review						
Wrap-up and project evaluation	4	4	4	4	6	22
Project presentation and delivery	4	4	4	4	6	22
Total number of hours	150	167	159	159.5	157	749.5
Weekly Average of Hours Assigned	17	19	18	18	17	

14. Approvals

Client Signature	Date
Team Representative	Date