

**STUDENT MANAGEMENT PROJECT**

**Software Requirement Specification**

**Class: SE1619-NET**

**Member:**

* **Đặng Tất Thành**
* **Nguyễn Hữu Tiến Anh (Leader)**
* **Nguyễn Minh Đức**
* **Thái Minh Quang**
* **Đan Công Duy**

– Hanoi, May 2022 –

**Record of changeS**

| **Date** | **A\* M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- |
| 10/5 | A | ANHNHT | Thêm phần product vision I.1 |
| 11/5 | A | ANHNHT | Thêm phần Business Rules I.7 |
| 11/5 | A | THANHDT | Thêm phần actor list I.4.4.2 |
| 12/5 | A | DUCNM | Thêm phần major features I.3 |
| 12/5 | A | QUANGTM | Thêm phần product context I.2 |
| 12/5 | A | QUANGTM | Thêm phần use case I.4.4.1 |
| 13/5 | A | THANHDT | Thêm phần Assumptions & Dependencies I.5 |
| 13/5 | M | DUCNM | Thêm phần Limitation and Exclusions I.6 |
| 15/5 | M | THANHDT | Sửa phần Major features I.3 |
| 16/5 | M | ANHNHT | Sửa phần Assumptions & Dependencies I.5 |
| 16/5 | M | DUYDC | Sửa phần Mind map I.3 |
| 30/5 | A | ANHNHT | Thêm phần User detail feature II.1 |
| 30/5 | A | THANHDT | Thêm phần Class detail feature II.2 |
| 30/5 | A | DUCNM | Thêm phần Subject detail feature II.3 |
| 30/5 | A | DUYDC | Thêm phần Team detail feature II.4 |
| 30/5 | A | QUANGTM | Thêm phần Iteration detail feature II.5 |

\*A - Added M - Modified D - Deleted

**Table of Contents**

[I. Product Overview](#_heading=h.gjdgxs)

1. [Product Vision](#_heading=h.30j0zll)
2. [Product Context](#_heading=h.1fob9te)
3. [Major Features](#_heading=h.3znysh7)
4. [User Requirements](#_heading=h.2et92p0)

4.1. Use cases

4.2. Actor list

1. Assumptions & Dependencies
2. Limitation and Exclusions
3. Business Rules

II. Use Case Specifications

1. User detail feature

2. Class detail feature

3. Subject detail feature

4. Team detail feature

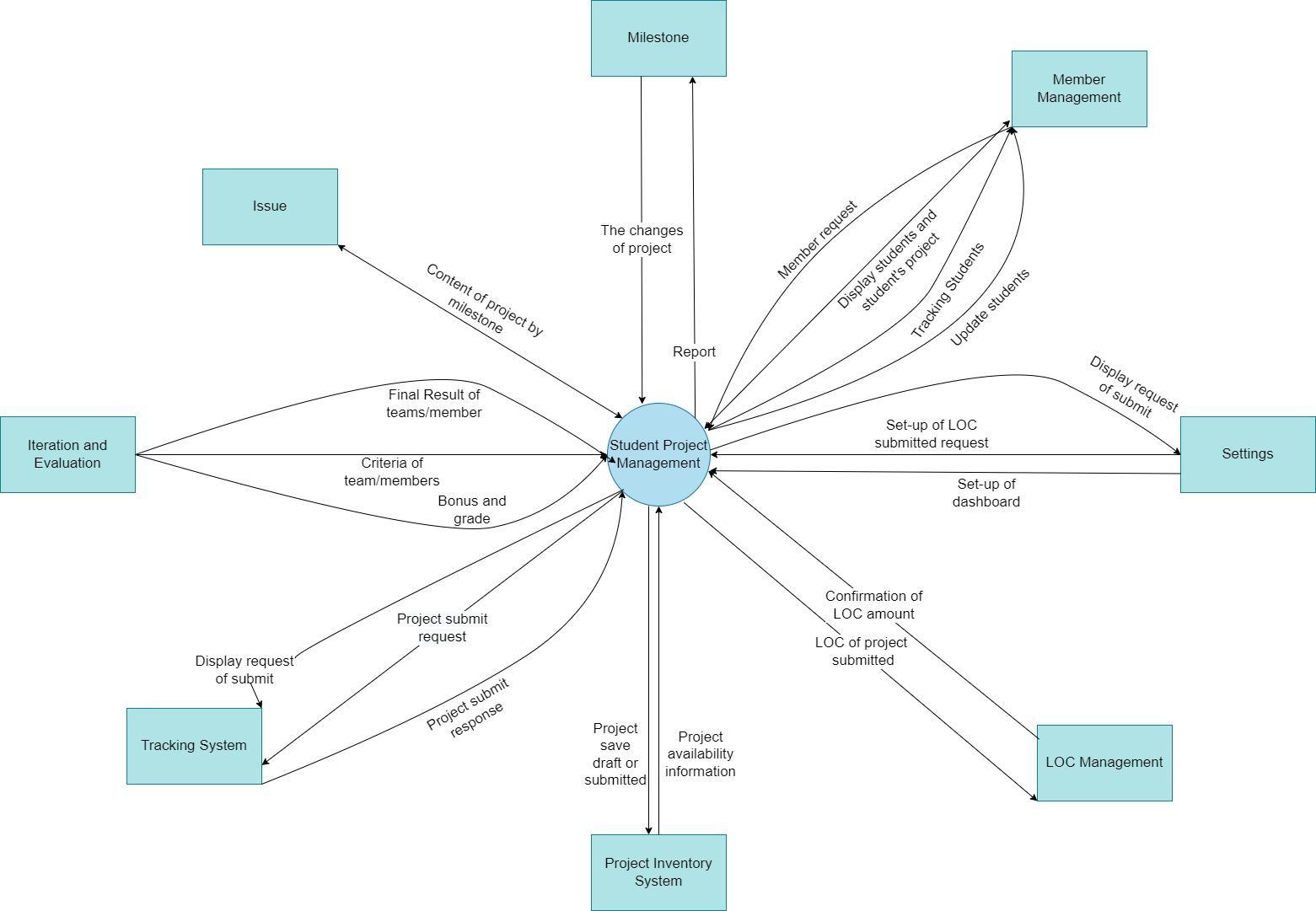
5. Iteration detail feature

# I. Product Overview

## 1. Product Vision

Student Project Management - a system for managing projects of  subjects. In this system, students will be able to track their project progress along with the evaluation and comments of the lecturer. As for teachers, they will be able to manage student projects more easily, as well as easily give timely evaluation comments to students who complete the project. And the important part is indispensable to manage the score and course status (pass or not pass).

## 2. Product context



## 3. Major feature

FE-01: home and dashboard.

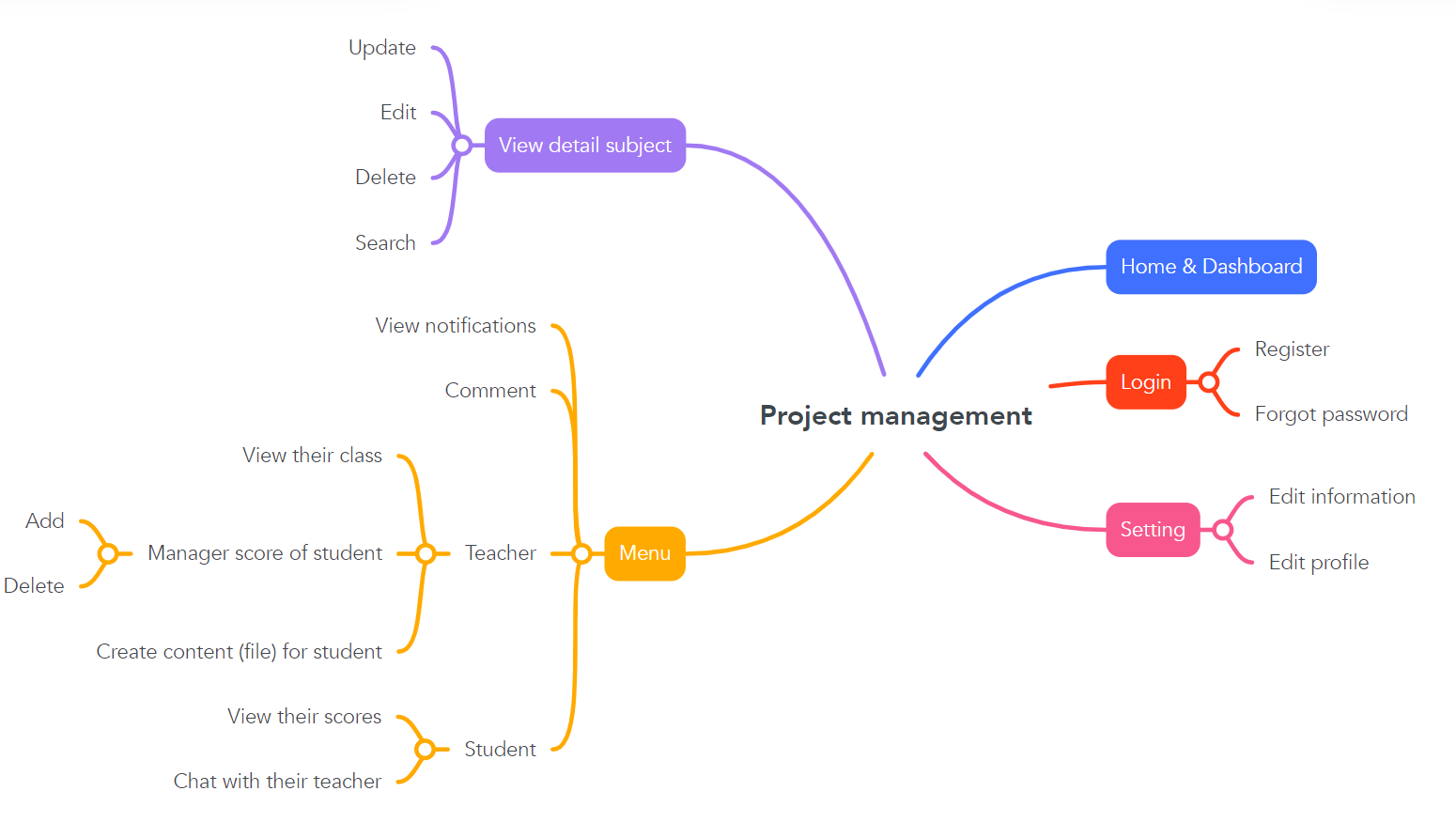
FE-02: login, register.

FE-03: setting list, setting detail.

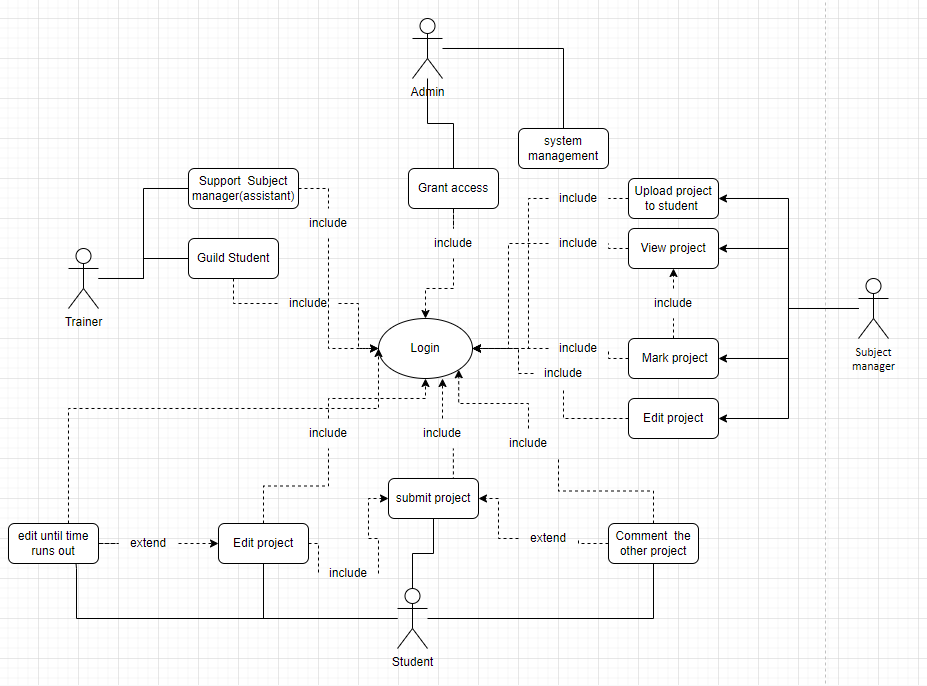
FE-04: user profile, comment, manage LOC.

FE-05: edit user profile, create content for students.

FE-06: push file on system, view score of student.



**4. User Requirement**

**4.1. Use case**

#### 4.2. Actor list

| No | Actor | Description |
| --- | --- | --- |
| 1 | Admin | User management, monitoring and website security, making a new class for subject managers and their students. |
| 2 | Subject  manager | Push content like projects, assignments for students. Make a deadline for students. Grading for student's project, comment in student’s project. Manage the quantity of submitted students. |
| 3 | Trainer | View the whole students in one class, download submitted projects of students, comment for students. |
| 4 | Student | View their score (LOC) on the dashboard. They can only view their information.See the time left on content, assignment of the subject manager created. Can push projects by rar file to server, in running time they can update or delete their own project. |

#### 5. Assumptions & Dependencies

* Users as role students can upload files to submit projects for lecture to grade and comment.
* Teachers can create deadlines for students, after the required time students can’t do any work on this process.

#### 6. Limitations and Exclusions

* Users may require a real-time system counting down from the website.
* Users may require a larger file than file can accept.
* Users may need to rename a file or a note for their project.
* Users may out session in a long time not working.
* Student account just mush be a school's email account

**7. Business Rules**

| ***ID*** | ***Rule Definition*** | ***Type of Rule*** | ***Static or***  ***Dynamic*** | ***Source*** |
| --- | --- | --- | --- | --- |
| BR-1 | Students are only allowed to correct their work up to 3 times and must be within the time of submission. | Constraint | Dynamic | Student  Project Manager |
| BR-2 | Students can only see their group's projects, not other groups' projects. | Constraint | Static | Student  Project Manager |
| BR-3 | Students can only submit their project on time running in the system. | Constraint | Static | Student  Project Manager |
| BR-4 | Students can only see their schedule in private, other students can’t see the schedule of other people. | Constraint | Static | Student  Project Manager |
| BR-5 | Each student can only login with the school's email. | Constraint | Dynamic | Student  Project Manager |
| BR-6 | Only teachers can add, edit and delete the folder (file) of the class. | Constraint | Static | Student  Project Manager |
| BR-7 | Users not admin can only edit their information. | Constraint | Static | Student  Project Manager |
| BR-8 | Users can only edit their information after 3 days from the last edit. | Constraint | Static | Student  Project Manager |
| BR-9 | Login will automatically expire after 30 minutes. | Constraint | Dynamic | Student  Project Manager |
| BR-10 | System can not receive a folder too large (>25MB) to avoid overloading for the system. | Constraint | Dynamic | Student  Project Manager |
| BR-11 | System can only receive file rar, zip, doc, excel, pptx, txt. | Fact | Dynamic | Upload for project manager |
| BR-12 | System allows students and teachers to login whenever they want. | Fact | Static | Student  Project Manager |
| BR-13 | The system will be maintained once a month. | Fact | Static | Student  Project Manager |
| BR-14 | Student projects can be graded by the teacher or by the tool. | Fact | Dynamic | Student  Project Manager |
| BR-15 | The teacher can reject a student project if the student cheats. | Fact | Dynamic | Student  Project Manager |

# II. Use Case Specifications

1. **User detail feature**

| ID and Name: | UC-01 User detail | | | |
| --- | --- | --- | --- | --- |
| Created By: | Nguyễn Hữu Tiến Anh | Date Created: |  | 30/5/2022 |
| Primary Actor: | Admin | | | |
| Description: | Admin can access the system to view the list of users. Here, the admin can update and delete users. | | | |
| Trigger: | An admin indicates that he wants to view user details. | | | |
| Preconditions: | PRE-1. Admin is logged into the system. | | | |
| Postconditions: | POST-1. Immediately after clicking delete user, a confirmation message will appear before deleting in the system.  POST-2. The user list will be made immediately after deletion or update. | | | |
| Normal Flow: | 1. Admin login to the system and check if this account is admin or not.  2. SPM displays a menu of available active functions and the other functions.  3. Admin select box view User management’.  4. System shows all of the users with the detail button to help admin access to each user.  5. System will show everythings about the class: list of students, roll number of students...  6. Admin will act with the system to make changes.  7. System notification after admin press save button.  8. Admin confirms acceptance of the changes on this user.  10. System sends Admin a message confirming changes are accepted or not.  11. Admin will change the status or role of users by clicking a button. | | | |
| Exceptions: | 1. When a user is deactivate, this account cannot login to the system. 2. when the number of users is full of 20 people. Admin needs to go to the next page. | | | |
| Priority: | High | | | |
| Frequency of Use: | Approximately 1000 users, average of one usage per day. Peak usage load for this use case is between 9:00 A.M. and 16:00 P.M. local time. | | | |
| Business Rules: | BR-7, BR-8, BR-12 | | | |
| Other Information: | 1. Admin shall be able to cancel the users editing process at any time prior to confirming it. 2. The default role is student if the admin doesn't enter the role of user. | | | |
| Assumptions: | Assume that there are multiple admins accessing and editing the same users. | | | |

**2. Class detail feature**

| ID and Name: | UC-02 Class detail | | | |
| --- | --- | --- | --- | --- |
| Created By: | ThanhDT | Date Created: | 30/05/2022 | |
| Primary Actor: | Patron | Secondary Actors: | Cafeteria Inventory System | |
| Description: | An admin accesses the system and goes to list all class functions to add more, change or delete class. Admin can view all the classes and then press to see detail of each class to see every member of the class. | | | |
| Trigger: | When admin want to make change or create new class | | | |
| Preconditions: | PRE-1. Admin is logged into the system by admin account | | | |
| Postconditions: | POST-1. Class right after being clicked on delete will appear a notification to confirm what they do.  POST-2. The list of classes must update right after it was deleted. | | | |
| Normal Flow: | 1.0 Admin work with classes  1. Admin login to the system and check if this account is admin or not.  2. SPM displays a menu of available active functions and the other functions.  3. Admin select box view ‘Class management’.  4. System shows all the lists of the class with the detail button to help admin access to each class.  5. System will show everythings about the class: list of students, number of students,...  6. Admin will act with the system to make changes.  7. System notification after admin press save button.  8. Admin confirms acceptance of the changes on this class.  10. System sends Admin a message confirming changes are accepted or not. | | | |
| Alternative Flows: | Admin can change lectures of this class | | | |
| Exceptions: | If the class is full of people (31 people, including lectures) the system will have a notification and do nothing.  If this student is joining two classes with the same subject, the system will have a notification and do nothing.  If his class already has this subject, the system will have a notification and do nothing. | | | |
| Priority: | High | | | |
| Frequency of Use: | Only the admin role can use this function. And this function can work 24/24 per day. | | | |
| Business Rules: |  | | | |
| Other Information: | 1. Admin can stop changing a class anytime and the system does nothing. | | | |
| Assumptions: | Assume that the system will automatically clear all the subjects of the class after a semester. | | | |

1. **Subject list feature**

| ID and Name: | UC-03 Subject list feature | | |
| --- | --- | --- | --- |
| Created By: | [Nguyen Minh Duc (K15 HL)](mailto:ducnmhe150901@fpt.edu.vn) | Date Created: | 30/05/2022 |
| Primary Actor: | Normal Users | Secondary Actors: | Admin |
| Description: | When the user accesses the system and wants to see the list of all the subjects the system has. The system will respond to the user with a list of subjects with any information of this subject they want to see. The system also has the search function to help the user easier to find the subject they want to join in. If the user has the high permission like admin, they can add new subjects, update or delete subjects. Admin have more functions than normal users. Admin can see all the subjects including the subjects that are not active. And admin can modify like add, update information or delete subjects. | | |
| Trigger: | A student wanna join to a subject | | |
| Preconditions: | PRE-1. That student logged in to the system.  PRE-2. That student has the role of student and the access to accept to learn that subject. | | |
| Postconditions: | POST-1. Are the subjects in the status of working.  POST-2. The system will check out the list class to learn this subject required.  POST-3. Respond to users of all the classes able to access the subject. | | |
| Normal Flow: | 1.0 Join a Subject.  1. User asks to view the menu for all subjects.  2. System displays a menu of all the subjects.  3. User selects one of all subjects from the menu.  4. System check out is the subject able to join.  5. System will check out all the classes that can be accessed in that subject and respond to the user. | | |
| Alternative Flows: | 1.1 User does not have permission to join subject  1. System respond to user a alert make them know the account does not in a status able to access  2. Return to the menu of subjects.  1.2 User did not login  2. Return to the login page. | | |
| Priority: | High | | |
| Frequency of Use: | Approximately 2000 users, average of one usage per day. Always ready to be access for everyone | | |
| Business Rules: |  | | |
| Other Information: | 1. Student can un-enrol anytime they want if the join in wrong subject  . Otherwise, the default date is the next day that the cafeteria is open. | | |
| Assumptions: | Assume about 1000 student enrol in a subject in the start of the course | | |

**4. Team detail feature:**

| ID and Name: | UC-04 Team detail | | |
| --- | --- | --- | --- |
| Created By: | Đan Công Duy | Date Created: | 30/5/2022 |
| Primary Actor: | Admin | Secondary Actors: | Manager Project Team |
| Description: | Admin accesses the system from the corporate intranet or from home, selects class, and creates a team for students. | | |
| Trigger: | Admin wants to create a new team or change team members. | | |
| Preconditions: | PRE-1. Admin is logged into the system. | | |
| Postconditions: | POST-1. Admin created a new team for students. | | |
| Normal Flow: | 1.0 Create a new team  1. Admin views the list of students in a class.  2. System displays a list of students.  3. Admin selects one or more students from the list to create a new team.  4. Admin confirms the select (continue) or modify team member (return step 3).  5. System displays the team which was newly created. | | |
| Alternative Flows: | 1.1 Create a new team similar to an existing team  1. Admin creates a new team similar to the existing team.  2. System notice error. | | |
| Exceptions: | 1.1. User not logged into system.   1. System will notify the user to login. 2. System will return to the login page.   1.2. If a student is in a team   1. System notice error. 2. Users need to create a different team. | | |
| Priority: | High | | |
| Frequency of Use: | Approximately 1000 users, average of one usage per day. Peak usage load for this use case is between 9:00 A.M. and 16:00 P.M. local time. | | |
| Business Rules: | BR-2, BR-5, BR-7, BR-12 | | |
| Other Information: | 1. Admin shall be able to cancel the creating team at any time. 2. Only the admin can edit the team. 3. Students can view their team after the admin creates the team. | | |
| Assumptions: | Assume that the system will automatically delete a team when a member of the team leaves class. | | |

**5. Iteration detail feature**

| ID and Name: | UC-05 Edit Iteration | | | |
| --- | --- | --- | --- | --- |
| Created By: | Thái Minh Quang | Date Created: | 30/5/2022 |  |
| Primary Actor: | Author | | | |
| Description: | A Author accesses the Project Management System from the corporate intranet or from home, list all the iteration that author can edit, add , remove, update iteration | | | |
| Trigger: | Author can edit the iteration as updating, deleting, adding,... | | | |
| Preconditions: | PRE-1. Author is logged into the system.  PRE-2. Author choose iteration list and then update, add, or delete the iteration | | | |
| Postconditions: | POST-1. After updating, adding, deleting,... data will be saved into the database. Then this will be showed to student | | | |
| Normal Flow: | 1.0 Author will give a fixed amount of time for students to do.  Add, delete and update | | | |
| Alternative Flows: | 1.1 Author can create multiple iterations for students in a given time period  Students can ask the teacher to edit the time | | | |
| Exceptions: | 1.0. If students submit their assignment without clicking on the submit button, that assignment can be rejected. | | | |
| Priority: | medium | | | |
| Frequency of Use: | Only authors can access the system. And they can access whenever they want | | | |
| Business Rules: | BR-1, BR-2, BR-3, BR-4, BR-11, BR-12, BR-14, BR-15 | | | |
| Other Information: | 1. Students can comment on the iteration that they do | | | |
| Assumptions: | Assume system will automatically follow the history of the iteration that student have done | | | |