Teams Management System – Team Creation sub-system

Requirements Document

Document id: TMS-Req-01

This document describes requirements of the Team Creation sub-system (TCS) of the Teams Management System (TMS).

A. Problem

Students are usually asked to work on assignments and projects in teams as part of courses. Setting up of these teams can be difficult as students usually have a very short time to organize themselves, and do not always know which other students are available. Collaboration between team members may also prove difficult as a project is going on because of lack of contact information. For instance, some students want to keep their phone and email information confidential from their teammates, but still need to be in contact.

The TMS will help Instructors and Students set up and manage teams for assignments and projects as part of courses. The TCS is a part of the TMS that will be used for setting up teams.

B. Background information

Backgroung information about this project are provided in the overview document **T**MS-G-01 and the domain overview document TMS-D-01.

C. Environment and system models

The TMS system is to run as a client server system. The server part shall run on an application server. Information about courses, students and instructors shall be accessed from a data store. Clients shall access the system using web browsers.

D. Functional requirements

F1. The TMS-TCS shall allow instructors to start up teams creation for courses that they had registered.

F1.1. Teams creation shall be started up by specifying the following properties:

- course code
- minimum number of students per teams
- maximum number of students per teams
- deadline for teams creation

F2. The TMS-TCS shall allow instructors to modify properties set up for teams (see F1.1) anytime before a set deadline for teams creation.

F2.2. The new properties shall apply to all existing and future teams.

F3. The TMS-TCS shall allow instructors to visualize a list of all teams created for their courses.

F3.1. Information provided for each team shall include:

- a unique team identifier
- a team name
- the team date of creation
- information about the team members with a mention of the team liaison
- a status of a team as *complete, incomplete*

F3.2. Information provided for each team member shall include:

- the student's number
- the student's name
- the student's study program
- the course section in which the student is enrolled
- the student's email address

F3.2. A team status shall be set up as *incomplete* if it doesn't have the minimum number of students per teams (see F1.1, F5.)

F3.3. A team status shall be set up as *complete* if it does have the maximum number of students per teams (see F1.1, F5.)

F4. The TMS-TCS shall allow instructors to allow a team to have less members than the minimum set (see F1.1).

F5. The TMS-TCS shall allow instructors to allow a team to have more members than the maximum set (see F1.1).

F6. The TMS-TCS shall allow instructors to add members to a team.

F7. The TMS-TCS shall allow instructors to remove members from a team.

F8. The TMS-TCS shall allow students to create teams once an instructor started up teams creation for a course in which they are registered (see F1.)

F8.1. A team shall be created by specifying:

- a name
- a possible empty set of members

F8.2. A student creating a team shall be automatically member of that team.

F8.3. A student creating a team shall be automatically the team liaison for that team.

F9. The TMS-TCS shall allow students to visualize a list of all teams that are not *complete*.

F10. The TMS-TCS shall allow students to ask to be added as members of teams that are not *complete*.

F11. The TMS-TCS shall allow students to quit teams in which they are members anytime before the deadline for teams creation.

F12. The TMS-TCS shall allow all members of a team to visualize the following information about their team:

- team identifier
- team name
- team date of creation
- team members names with a mention of the team liaison
- a status of a team as *complete* or *incomplete*

F13. The TMS-TCS shall allow all members of a team to visualize a list of names of students who asked to join their team (see F10.).

F14. The TMS-TCS shall allow a team liaison to add students from the list of students who asked to join that team as members of the team (see F10.)

F14.1. The TMS-TCS shall prevent adding a student to a team with a status *complete.*

F15. The TMS-TCS shall allow a team liaison to specify another team member as the team liaison.

F16. The TMS-TCS shall send an email notification to a student when he/she is added to team or removed from a team.

F17. The TMS-TCS shall allow new Users (Instructors, Students) to register

F17.1. A User shall register by providing the following information:

- User identification (employee or student number)
- Login Password
- Name (First and Given)
- Email address

F17.2. The TMS shall authenticate all access based on the User identification and Password

E. Non-functional requirements and constraints

NF1. Only instructors teaching a course shall be able to start up teams creation for that course.

NF2. Only instructors teaching a course shall be able to set up and modify teams parameters for that course.

NF3. Only students registered to a course shall be able to create teams for that course.

NF4. Only a team member designated as team liaison and instructors shall be able to add new members to that team.

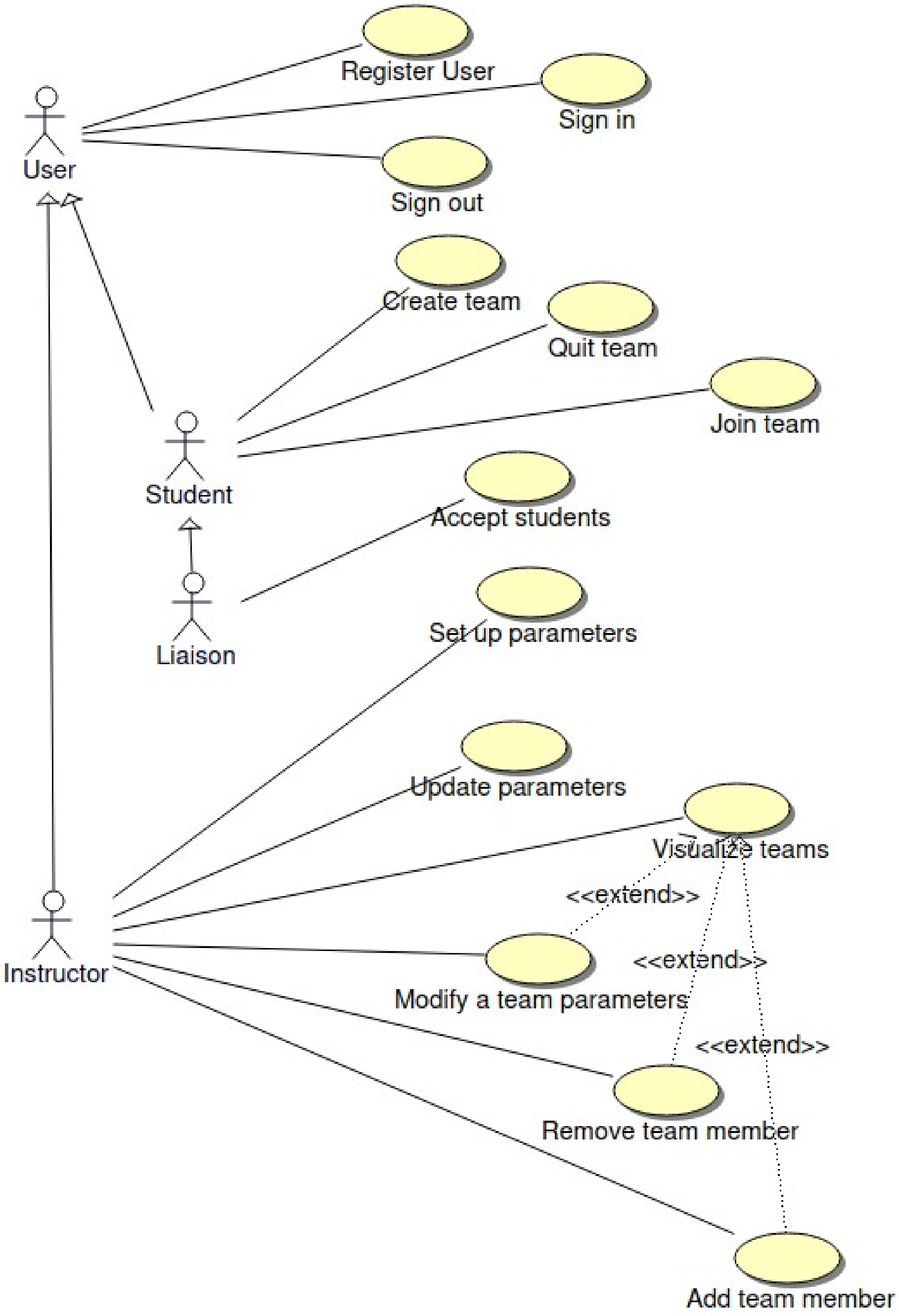
NF5. A student shall not be member of more than one team at a time for a course.

NF6. Any interaction initiated at the server shall be completed within 6 seconds.

NF7. Access to the TMS system functions shall be possible using any W3C compliant browser.

NF8. A working system satisfying requirements for the TMS-TCS component shall be delivered within 2 months.

F. Use Cases



Use Case: Register User

Title: Register User

Summary: A user (Instructor or Student) wants to register to the TMS by providing required registration information. If successful, the user receives an acknowledgment. If information is incomplete, an error message is displayed.
Primary Actor: User (Instructor or Student)

Priority: 6
Goal: A User wants to register to the TMS in order to identify himself to the system and use it's functions
Precondition: The TMS is operational
Postcondition: The User is registered

Main Scenario

1. The user selects the option to register
2. The TMS prompts the user for required information
3. The user submits the necessary details.
4. The TMS confirms registration with an acknowledgment message

Alternatives

3. a. Incomplete information provided
3. a. 1. The TMS shows an error message requesting complete information.

Use Case: Sign in

Title: Sign in

Summary: A user wants to log in by providing credentials. The system verifies authorization before showing available operation options according to the user's type. If unauthorized, an error message is displayed.
Primary Actor: User (Instructor or Student)

Priority: 12
Goal: The user wants to log in to access the system's features.
Precondition: The TMS is operational
Postcondition: The user is successfully logged in, and the system displays available operation options according to the type of user.

Main Scenario

1. The user enters their login credentials
2. The TMS verifies the user's authorization
3. The TMS presents the user with operation choices

Alternatives

2. a. The user is not authorized to use system
2. a. 1. The TMS shows an error message for unauthorized access

Use Case: Sign out

Title: Sign out

Summary: A user wants to log out. The TMS confirms the action with an acknowledgment message. The user is no longer logged in.
Primary Actor: User

Priority: 13
Precondition: The user is logged in, and the TMS is operational
Postcondition: The user is logged out

Main Scenario

1. The user initiates the log-out process
2. The TMS confirms with a log-out acknowledgment message

Use Case: Set up parameters

Title: Set up parameters

Summary: An instructor wants to set up team parameters by providing input, which is validated by the system. If valid, the TMS confirms teams creation readiness. Otherwise, it displays an error message.
Primary Actor: Instructor

Priority: 4
Precondition: The Instructor is logged in, the Instructor's operation choice menu is being displayed
Postcondition: Teams can be started up

Main Scenario

1. The instructor selects the option to set up team parameters
2. The TMS prompts the instructor to enter the parameters
3. The instructor provides the required parameters
4. The TMS validates the input
5. The TMS confirms successful setup with an acknowledgment message

Alternatives

4. a. Teams Parameters are not valid
4. a. 1. The TMS shows an error message for incorrect team parameters

Use Case: Update parameters

Title: Update parameters

Summary: An instructor wants to update team parameters, which are validated by the TMS. If valid, the system confirms teams creation readiness based on the new parameters. Otherwise, an error message is shown for invalid parameters.
Primary Actor: Instructor

Priority: 7
Precondition: The Instructor is logged in, and the Instructor's operation choice menu is displayed
Postcondition: Teams can be started up

Main Scenario

1. The instructor selects the option to update team parameters
2. The TMS prompts the instructor for the new parameters
3. The instructor submits the updated parameters
4. The TMS validates the new input
5. The TMS confirms the update with an acknowledgment message

Alternatives

4. a. Teams Parameters are not valid
4. a. 1. The TMS displays an error message for incorrect team parameters

Use Case: Visualize teams

Title: Visualize teams

Summary: An instructor wants to view the list of student teams by selecting the corresponding option, and the TMS displays all teams.
Primary Actor: Instructor

Priority: 11
Precondition: The Instructor is logged in, the Instructor's operation choice menu is displayed
Postcondition: The TMS displays the list of student teams

Main Scenario

1. The instructor selects the option to view student teams
2. The TMS shows a list of all teams

Use Case: Modify a team parameters

Title: Modify a team parameters

Summary: An instructor wants to modifies a specific team parameters by selecting the team, editing its details, and having the TMS validate and update them. If the parameters are invalid, an error message is displayed.
Primary Actor: Instructor

Priority: 8
Precondition: The instructor is logged in, and the list of teams is displayed
Postcondition: The team's parameters are successfully updated

Main Scenario

1. The instructor selects a team
2. The TMS shows the selected team's information
3. The instructor opts to modify the team’s parameters
4. The TMS displays the current team parameters
5. The instructor edits the parameters
6. The TMS validates the new parameters
7. The TMS updates the parameters

Alternatives

6. a. Team parameters are not valid
6. a. 1. The TMS shows an error message

Use Case: Remove member from a team

Title: Remove team member

Summary: An instructor wants to remove a member from a team by selecting the team, choosing the member, and the TMS updates the team, notifies the removed member, and returns to the operation menu.
Primary Actor: Instructor

Priority: 10
Precondition: The instructor is logged in, and the list of teams is displayed
Postcondition: A member is removed from the selected team

Main Scenario

1. The instructor selects a team
2. The TMS shows the selected team's details
3. The instructor chooses a member to remove
4. The TMS removes the selected member from the team
5. The TMS notifies the removed member
6. The TMS returns to the instructor's operation menu

Use Case: Add member to a team

Title: Add team member

Summary: An instructor wants to add a student to a team by selecting the team and choosing the student from a list. The TMS updates the team, notifies the new member, and returns to the operation menu. If the student is already a member or the team is full, an error message is shown.
Primary Actor: Instructor

Priority: 5
Precondition: TMS is displaying the list of teams AND The Instructor is logged in
Postcondition: A Team has a member added

Main Scenario

1. The instructor selects a team
2. The TMS shows the selected team's details
3. The instructor browses and selects a student from the list of all students
4. The TMS adds the student to the team
5. The TMS notifies the added member
6. The TMS returns to the instructor's operation menu

Alternatives

3. a. Student is already a team member
3. a. 1. The TMS displays an error message
3. b. The team is complete
3. b. 1. The TMS displays an error message

Use Case: Create team

Title: Create team

Summary: A student wants to create a new team by selecting the option, providing a name and student list, which the TMS validates. If valid, the team is added; otherwise, appropriate error messages are displayed for duplicates or conflicts.
Primary Actor: Student

Priority: 1
Precondition: The student is logged in, and the TMS displays the student operation choices menu
Postcondition: A new team is successfully added

Main Scenario

1. The student selects the option to create a new team
2. The TMS requests the team name and a list of students
3. The student provides the required information
4. The TMS validates the provided details
5. The TMS adds the new team to the list of teams
6. The TMS returns to the student operation choices menu

Alternatives

4. a. The team name is already in use or the students list includes students that are already in another team
4. a. 1. The TMS displays an error message

Use Case: Quit team

Title: Quit team

Summary: A student wants to quits a team by selecting it and choosing to leave. The TMS updates their status, notifies them of the removal, and returns to the operation menu.
Primary Actor: Student

Priority: 9
Precondition: The student is logged in, the TMS displays the student operation choices a menu
Postcondition: The student is no longer a member of the team

Main Scenario

1. The student selects the team they are a member of
2. The TMS displays the information for the selected team
3. The student chooses the option to quit the team
4. The TMS removes the student from the team
5. The TMS notifies the student of their removal
6. The TMS returns to the student operation choices menu

Use Case: Join team

Title: Join team

Summary: A student wants to join a team by viewing incomplete teams, selecting the desired ones, and being added as a candidate member. The TMS then returns to the operation menu with the student's status set to pending.

Primary Actor: Student

Priority: 2

Precondition: The student is logged in, not a member of any team, and the TMS displays the student operation choices menu.

Postcondition: The student's status is set to pending

Main Scenario

1. The student opts to view incomplete teams
2. The TMS shows a list of incomplete teams
3. The student selects the teams they wish to join
4. The TMS adds the student to the list of candidate members for each selected team
5. The TMS returns to the student operation choices menu

Use Case: Accept students

Title: Accept students

Summary: A liaison wants to accept new students by reviewing requests, selecting candidates, and adding them to the team. If the maximum team size is exceeded, an error message is displayed.

Primary Actor: Liaison

Priority: 3

Precondition: The liaison is logged in

Postcondition: New students are successfully added to the team

Main Scenario

1. The liaison chooses to view the list of students who have requested to join the team
2. The TMS displays the list of requesting students
3. The liaison selects the students to accept
4. The TMS adds the selected students to the team
5. The TMS sends notifications to the newly added members

Alternatives

3. a. The maximum size of teams would be exceeded by adding the selected members

3. a. 1. The TMS displays an error message