

Introduction to software engineering

Phase01-Tournaments software-Section51

Muath Alghamdi -202031040

Fahad Almutairi - 202171210

Naif Abdulaziz - 202038160

Ziyad Alyahya - 202039560

Table tasks contribution

Muath Alghamdi	Records scores - Add new sport/games-View tournament
Fahad Almutairi	Remove new sport/games-View Profile-Setup students profiles
Naif Abdulaziz	View table/bracket- Log in- Modify tournament- Generate table of matches
Ziyad Alyahya	Registration- Record tournaments- Stop Registration

rec Use case ID:			
Use Case Name:	Records scores		
Created by:		Last Updated by:	
Date Created:		Last Revision Date:	
Actors:	Admin		
Description:	Adding the scores of a match or scores of matches to the table		
Trigger:	Clicked on: Record scores		
Preconditions:	<ul style="list-style-type: none">-The user is logged in- The user is admin- Is in tournament period.-There is at least one finished match in the tournament (to be able to record its scores)		
Postconditions:	-The table/bracket is updated with the given scores		

	<ul style="list-style-type: none"> -The scores are available in tournament information. -The ability to modify or delete the scores.
Normal Flow:	<ol style="list-style-type: none"> 1-Admin clicks on record scores 2-System validates the user access ability 3-System: valid access ability 4- System shows the table of matches 5-The admin selects a specific match 6-The system shows the previous scores of this match or an empty slot if there were not any scores. 7-The admin will add/modify/delete the selected match's scores. 8-The admin enters a score 9-The admin clicks save 10-The system will validate the given score 11- System: valid score 12-The system updates the table with the given score.
Alternative Flows:	<p>In normal flow number 5. Instead of selecting a match:</p> <ol style="list-style-type: none"> 5- The admin searches for a specific match 6- The system show the match's records 7- Go back to Normal flow 6 <p>In normal flow number 8. Instead of entering a score, the admin delete a score:</p> <p>The system asks the user to confirm.</p> <ul style="list-style-type: none"> -The admin selects confirm -The System will delete the previous record -The System will update the table after deleting the record of the selected match.
Exceptions:	<p>In normal flow 3:</p> <ul style="list-style-type: none"> - The system could not identify the user as admin - The system inform this user that record score is only for admins <p>In normal flow 11:</p> <ul style="list-style-type: none"> - The input score were not valid - The system inform the admin that the score is not valid - The system asks the admin to enter the score again
Assumptions:	
Notes and Issues:	

Use case ID:			
Use Case Name:	Add new sport/games		
Created by:		Last Updated by:	
Date Created:		Last Revision Date:	
Actors:	admin		
Description:	Adding a new sport/games to the list of sport\games		
Trigger:	Clicked: Add new sports/games		
Preconditions:	<ul style="list-style-type: none"> -The user is logged in -The user is admin -list is not full 		
Postconditions:	<ul style="list-style-type: none"> -The sport/game is added to the list -The admin now can create a tournament out of this sport/game 		
Normal Flow:	<ol style="list-style-type: none"> 1- The admin clicks on add new sports/games 2- The system validate the user's ability 3- System: valid ability 4- The system will prompt the user to enter the desired name of the sport 5- The admin enters a name 6- The system will validate the given name 7- System: valid name 8- The System will check that there is no duplicate support 9- The system adds the sport to the list 10- The sport is now available in the tournaments creation with its photo if there were any. 		
Alternative Flows:	<p>In normal flow 4, instead of entering a name:</p> <ol style="list-style-type: none"> 5- The system will show a list of sports with their photos and a brief description 6- The admin selects the desired sport 7- go back to normal flow 10 		
Exceptions:	<p>In normal flow 2:</p> <ul style="list-style-type: none"> - The system found that the user is not an admin - The system denies access <p>In normal flow 7:</p> <ul style="list-style-type: none"> -The system found that the input name is invalid - The system informs the admin that the name is not valid <p>-The system asks the admin to enter the name again</p>		

	<p>In normal flow 8:</p> <ul style="list-style-type: none"> - The system found an already existing sport - The system will inform the user
Assumptions:	
Notes and Issues:	

Use case ID:			
Use Case Name:	Remove new sport/games		
Created by:		Last Updated by:	
Date Created:		Last Revision Date:	
Actors:	Admin		
Description:	Removing a sport/games from the list of sport\games		
Trigger:	Clicking: remove sport/game		
Preconditions:	The user is logged in The user is admin The desired sport/game exists to be able to delete it		
Postconditions:	The sport/game is deleted from the list The tournaments of this sport will be deleted		
Normal Flow:	<ol style="list-style-type: none"> 1- The admin clicks on remove sports/games 2- The system validate the user's ability 3- System: valid ability 4- The system will prompt the user to enter the desired name of the sport 5- The system will validate the given name 6- System: valid name 7- The system removes the sport from the list 8- The related tournament of this sport will be deleted 		
Alternative Flows:	<p>In normal flow 4, instead of entering a name:</p> <ol style="list-style-type: none"> 5- The system will show a list of all sports in the system 6- The admin selects the desired sport 7- go back to normal flow 7 		
Exceptions:	<p>For normal flow 3, the system found that the user is not an admin</p> <ul style="list-style-type: none"> - Decline access <p>For normal flow 6, the system found that the input name is invalid</p>		

	The system inform the user that the name is invalid The system asks the user to re enter the name
Assumptions:	
Notes and Issues:	

;Use case ID:	1.1		
Use Case Name:	View tournament		
Created by:	Section51	Last Updated by:	
Date Created:		Last Revision Date:	
Actors:	Student , admin		
Description:	View the all information of the tournament		
Trigger:	Click (View tournament)		
Preconditions:	<ul style="list-style-type: none"> • User is logged in. • There is a tournament in the system. • 		
Postconditions:	<ul style="list-style-type: none"> • View all the tournament details. 		
Normal Flow:	<ol style="list-style-type: none"> 1. The student or admin clicks view tournament 2. The student or the admin Enter the tournament name 3. The system will show: <ol style="list-style-type: none"> a. name of the tournament b. type of tournament c. participation type (teams or individuals). d. the sport/game of the tournament. 		
Alternative Flows:	<ol style="list-style-type: none"> 1. The student or the admin enters a code for a specific tournament. Go to Step 3 in Normal flow. 		
Exceptions:	<ul style="list-style-type: none"> • If the tournament name does not exists the system will inform the user • The system will ask to user to re enter the name • If the tournament sport/game does not exists the system will show an error message. 		
Assumptions:			

Notes and Issues:		
Use case ID:	1.1	
Use Case Name:	View profile	
Created by:	Section51	Last Updated by:
Date Created:		Last Revision Date:
Actors:	Student , admin	
Description:	View the all information of the profile	
Trigger:	Click (View profile)	
Preconditions:	<ul style="list-style-type: none"> • There is a profile in the system. • User login. 	
Postconditions:	<ul style="list-style-type: none"> • View the profile details. • View information of the profile user. 	
Normal Flow:	<ol style="list-style-type: none"> 4. The student or admin clicks view profile 5. The student or the admin Enter the user name 6. The system validate the given name 7. The system will show: <ol style="list-style-type: none"> a. the user name b. the ID of user In case of a student c. the tournament he is in. d. date of tournament he is in. e. the rank of the student. 	
Alternative Flows:	<p>The student or the admin Enter the ID to view profile.</p> <ol style="list-style-type: none"> 2. Go to Step 2 in Normal flow 	
Exceptions:	<ul style="list-style-type: none"> • If the participant name does not exists the system will show an error message. • If the participant ID does not exists the system will show an error message. 	
Assumptions:		
Notes and Issues:		

Use case ID:	1.1		
Use Case Name:	Setup students profiles		
Created by:	Section51	Last Updated by:	
Date Created:		Last Revision Date:	
Actors:	admin		
Description:	the ability to modify students profiles details		
Trigger:	Click (modify profile)		
Preconditions:	<ul style="list-style-type: none"> • The admin login to the system • The student profile is exist 		
Postconditions:	<ul style="list-style-type: none"> • The new student profile information is updated 		
Normal Flow:	8. The admin clicks modify profile 9. The admin clicks search 10. The admin Enter the student name 11. The system will show options to modify the profile 12. The admin will select some options. 13. The admin will click submit 14. The system will check for the following: <ul style="list-style-type: none"> a. Changes do not conflict b. name is valid 15. The system will update the new information to students' profile.		
Alternative Flows:	1 -In normal flow 3, instead of entering a name the admin will enter ID 2-go back to normal flow 4		
Exceptions:	<ul style="list-style-type: none"> • If the student name does not exists the system will inform the admin • The admin can re enter the name • If the student ID does not exists the system will inform the admin • The admin can re enter the ID • If the changes made by the admin conflict the previous information of student, show an error message after pressing submit 		
Assumptions:			
Notes and Issues:			

Use case ID:	1.3		
Use Case Name:	Log in		
Created by:	Section51	Last Updated by:	

Date Created:	30/03/2023	Last Revision Date:	
Actors:	Admin, Student		
Description:	The ability to login to the system		
Trigger:	Click (log in)		
Preconditions:	<ul style="list-style-type: none"> • The user is an authorized admin or student. • The system is operating. • The user has a username and a password 		
Postconditions:	<ul style="list-style-type: none"> • The admin is on the main page (logged in). • The system shows all the functionalities that the admin can access. 		
Normal Flow:	16. The user enters the software. 17. The user enters the password and the username. 18. The user clicks log in. 19. The system checks the following: <ul style="list-style-type: none"> a. the validity of the username and password. b. if the user is authorized as an admin or student. c. if the system is under maintenance. 20. If the checking is correctly passed the user is logged in		
Alternative Flows:	if the user forgets his name or password. <ol style="list-style-type: none"> 3. He can click forget name or pass to reassign them. 4. Go to Step 2 in Normal flow. 		
Exceptions:	In the normal flow step 4. <ol style="list-style-type: none"> 1. The system does not validate username or the password. <ol style="list-style-type: none"> a. Show an error message and ask the user to enter them again. b. If it is done 3 times lock the log in for 10 mints 2. If the user is not authorized of system is under maintenance display an error message 		
Assumptions:			
Notes and Issues:			

Use case ID:	1.3		
Use Case Name:	Modify tournament		
Created by:	Section51	Last Updated by:	
Date Created:	30/03/2023	Last Revision Date:	
Actors:	Admin		
Description:	The ability to Modify a tournament		
Trigger:	Click (modify on the tournament)		
Preconditions:	<ul style="list-style-type: none"> • The admin is logged in. • There is a tournament to Modify. 		
Postconditions:	<ul style="list-style-type: none"> • The changes are updated so the tournament is modified. 		

	<ul style="list-style-type: none"> The system shows all the updates that happened to the tournament.
Normal Flow:	<p>21. The admin enters the tournament tap. 22. The system will show all the available tournaments. 23. The admin will choose a tournament then clicks of modify. 24. the system shows the modification tap to the admin. 25. The admin modifies the tournament the clicks save. 26. The system checks the modifications. 27. The system saves the changes.</p>
Alternative Flows:	<p>The admin enters a code for a specific tournament.</p> <p>5. The system checks if it is a valid code. 6. Go to Step 4 in Normal flow.</p>
Exceptions:	<p>In the normal flow step 2.</p> <p>3. The system does not find any tournaments. 4. Display a message that there are no tournaments</p> <p>In step 6 if the modifications are not valid due to:</p> <ol style="list-style-type: none"> Conflicts with other tournaments. The are some typing mistakes. The tournament cannot be modified because it has passed its time. If the admin left an empty section <p>1. Display an error message to the admin and it will not save the changes.</p>
Assumptions:	
Notes and Issues:	

Use case ID:	1.1		
Use Case Name:	Generate table of matches		
Created by:	Section51	Last Updated by:	
Date Created:	30/03/2023	Last Revision Date:	
Actors:	Admin		
Description:	The ability to generate a table for all the matches in the tournament		
Trigger:	Click (Generate Table)		
Preconditions:	<ul style="list-style-type: none"> The admin is logged in. There is a tournament to generate a table to. The registering for the tournament is closed 		
Postconditions:	<ul style="list-style-type: none"> The table for the tournament is available to view. All information including time and place and other are available. The table can be published to all participants. 		
Normal Flow:	28. The admin clicks on Generate Table.		

	<p>29. The system will show all the available tournaments to show table to.</p> <p>30. The admin will choose a tournament.</p> <p>31. The system checks if the tournament is closed.</p> <p>32. The system will show the table with all the information to the matches.</p> <p>33. The admin can publish the table to all the participants.</p>
Alternative Flows:	<p>The admin enters a code for a specific tournament.</p> <p>7. The system checks if it is a valid code.</p> <p>8. Go to Step 4 in Normal flow.</p>
Exceptions:	<p>In the normal flow step 4.</p> <ol style="list-style-type: none"> If the tournament is still open the system will show an error message. The admin will be back on the main page. <p>In the Alternative flow.</p> <ol style="list-style-type: none"> If the code is not valid the system will ask the admin to enter again. Or go back to the main page.
Assumptions:	
Notes and Issues:	

Use case ID:	1.2		
Use Case Name:	View table/bracket		
Created by:	Section51	Last Updated by:	
Date Created:	30/03/2023	Last Revision Date:	
Actors:	Admin		
Description:	The ability to display the table for a tournament.		
Trigger:	Click (View table)		
Preconditions:	<ul style="list-style-type: none"> The admin is logged in. There is a tournament to view. 		
Postconditions:	<ul style="list-style-type: none"> The table/ bracket is displayed. The extra information about the details of the tournament can be displayed as well. 		
Normal Flow:	<p>34. The admin goes in the tournament tap.</p> <p>35. The system will show all the tournaments that are available.</p> <p>36. The admin chooses a tournament.</p> <p>37. The system will give the admin some choices.</p> <p>38. The admin chooses view bracket.</p> <p>39. The system displays the table for the selected tournament.</p>		
Alternative Flows:	<p>The admin searches for a code for the tournament.</p> <p>9. The system validates the code.</p>		

	10. Go to Step 4 in Normal flow.
Exceptions:	<p>In the normal flow step 2.</p> <p>5. The system dose not find any tournaments. 6. Display a message that there are no tournaments.</p> <p>In the Alternative flow</p> <p>1. The code is not valid. 2. The system asks the admin to enter the code again 3. Or go back to the main page</p>
Assumptions:	
Notes and Issues:	

Use Case Name:	Registration		
Created by:	Section55	Last Updated by:	
Date Created:	19/02/2023	Last Revision Date:	
Actors:	Student		
Description:	The ability to register in a tournament		
Trigger:	Click (register)		
Preconditions:	<ul style="list-style-type: none"> • The user is logged in. • Tournament registration period is open. • allowed to register in the tournament 		
Postconditions:	<ul style="list-style-type: none"> • Student is registered in the tournament. • The student is shown in the tournament list. • Number of available seats is updated. • A confirmation email is sent to the student 		
Normal Flow:	<ol style="list-style-type: none"> 1. Student clicks available tournament. 2. The system will show all the available tournaments. 3. The student will filter the results whether he wants Elimination tournaments or Round-robin tournaments and if it is a team tournament or an individual tournament. 4. The student chooses a tournament. 5. The student will fill in the needed information. 6. The students will click register. 7. The system will check for the following. <ol style="list-style-type: none"> a. There are enough seats. b. Student is not participating with another team in the tournament. 8. The system registers the student in the tournament. 		

Alternative Flows:	<p>1. Student register by the code:</p> <ul style="list-style-type: none"> a. The student enters the tournament code. b. Go to Normal flow step 6.
Exceptions:	<p>In the alternative flow 1b:</p> <ul style="list-style-type: none"> 1- If the tournament code isn't valid. 2- The system will show an error message and will not register him. <p>In the normal flow step 7a.</p> <ul style="list-style-type: none"> 3- The system found there are not enough seats. 4- The system will show the restriction message and will not register him. <p>In the normal flow step 7b.</p> <ul style="list-style-type: none"> 5- The system found there is a student participating with another team in the tournament. 6- The system will show the restriction message and will not register him.

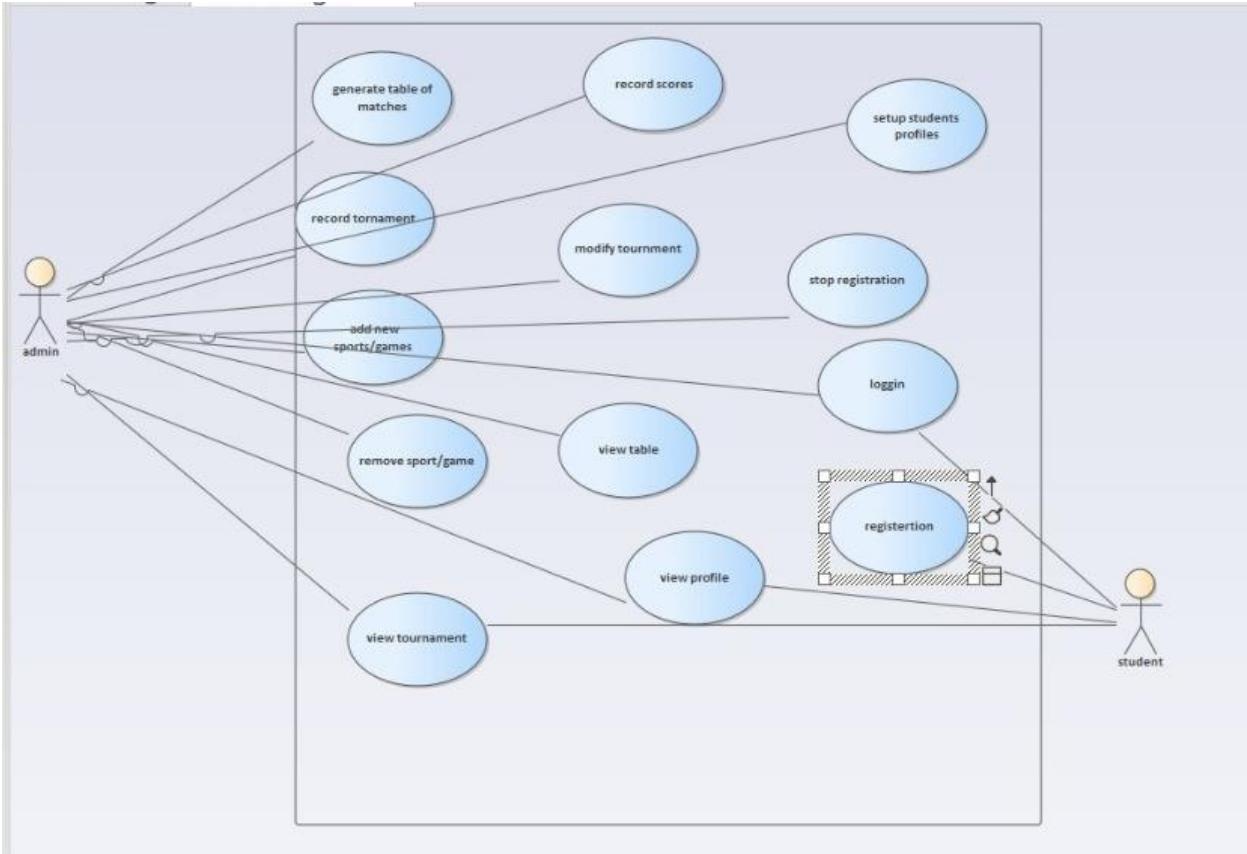
Use case ID:	1.1		
Use Case Name:	Stop Registration		
Created by:	Section55	Last Updated by:	
Date Created:	19/02/2023	Last Revision Date:	
Actors:	admin		
Description:	The admin has the authority to stop the registration for the tournament.		
Trigger:	Click (stop registration).		
Preconditions:	<ul style="list-style-type: none"> • The admin is logged in. • tournament registration period has begun. • There are enough participants. 		
Postconditions:	<ul style="list-style-type: none"> • No students can register for the competitions 		
Normal Flow:	<ol style="list-style-type: none"> 1. admin clicks Search for tournaments. 2. The system will show all the tournaments. 3. The admin will filter tournaments. 4. The admin chooses the tournaments. 5. The admin will click stop registration. 6. The system will check if the tournament registration period has begun. 7. The system will stop the registration for the tournaments. 		
Alternative Flows:	<ol style="list-style-type: none"> 1. Default by the system <ol style="list-style-type: none"> a. The registration period has ended. b. Then the system will stop the registration automatically. 2. The admin enters a code for a specific tournament. <ol style="list-style-type: none"> a. The admin enters the tournament code. b. Go to Normal flow step 6. 		
Exceptions:	<p>A. In the normal flow step 6.</p> <ol style="list-style-type: none"> 1. The system found that the tournament registration period has not begun. 2. The system will show the restriction message. <p>B. In the Alternative Flows step 2.</p> <ol style="list-style-type: none"> 1. The system found that the tournament code isn't valid. 2. The system will show an error message. 		

Use case ID:	1.1		
Use Case Name:	Record tournaments		
Created by:	Section55	Last Updated by:	
Date Created:	19/02/2023	Last Revision Date:	
Actors:	admin		
Description:	The admin will record the information of the tournaments.		
Trigger:	Click (record tournaments).		
Preconditions:	<ul style="list-style-type: none"> • The admin is logged in. • There are no tournaments of the same sport/game at the same time as the new tournament. 		
Postconditions:	<ul style="list-style-type: none"> • The recorded tournament will appear in the tournaments list. 		
Normal Flow:	<ol style="list-style-type: none"> 1. admin clicks tournaments. 2. The system will show all of the tournaments. 3. The admin will click record tournament. 4. The admin will fill in the tournament name and choose the remaining information. 5. The system will check for the following. <ol style="list-style-type: none"> a. All tournament information has been filled in. b. There is no time conflict between the new tournament and any registered tournament of the same type. 6. The system will put the new tournaments on the tournament list. 		
Alternative Flows:	1-In normal flow 4, instead of entering a name, the admin will enter a code for the tournament 2-go back to normal flow 5		
Exceptions:	<ol style="list-style-type: none"> 1. In the normal flow step 5. a: <ol style="list-style-type: none"> a. The system found that the admin didn't fill in all the tournament information. b. The system will not create the tournament and will show a message. 2. In the normal flow step 5. b: <ol style="list-style-type: none"> a. The system found there is a conflict between the new tournament and any registered tournament of the same type. b. The system will not create the tournament and will show a message. 3. In the alternative flow 1: <ul style="list-style-type: none"> - The system found that the code is already possed by other tournment 		

Non-Functional requirements:

- 1- Efficient with regards to memory: it should delete or save the information of players and scores of the matches in the appropriate data-structures so that it can reduce the memory space and does not make wasteful use of resources.
- 2- Efficient with regards to time: it should be fast and does not take long time to generate the matches and view their results and their other related operations for admins. It should also be fast for students in such a way that they can immediately register without waiting for the program to load.
- 3- Security: it should implement some type of authentication to prevent any outside access on the admin's && students' profiles. It should also secure the personal information of the profiles in such a way that it does not get leaked.
- 4- Reliability: the system must perform without failure in use cases during a tournament period.
- 5- Availability: The program should be available before, during and after the tournaments for users.
- 6- Usability: the program should be easy to use for all users. It should provide an easy way for admins to view or add sport/ games. It should also have an appropriate UI to display the tournaments info.
- 7- Maintainability: the program should be written in a way that is easy to maintain. Requirements might change each semester ,for example, a new type of tournaments arrangement might be introduced. Thus, Designing and implementing these requirements should cost as minimum as possible if needed.
- 8- Portability: the program can be run on tablets or computers without a change in its behavior.

Use Case Diagram:

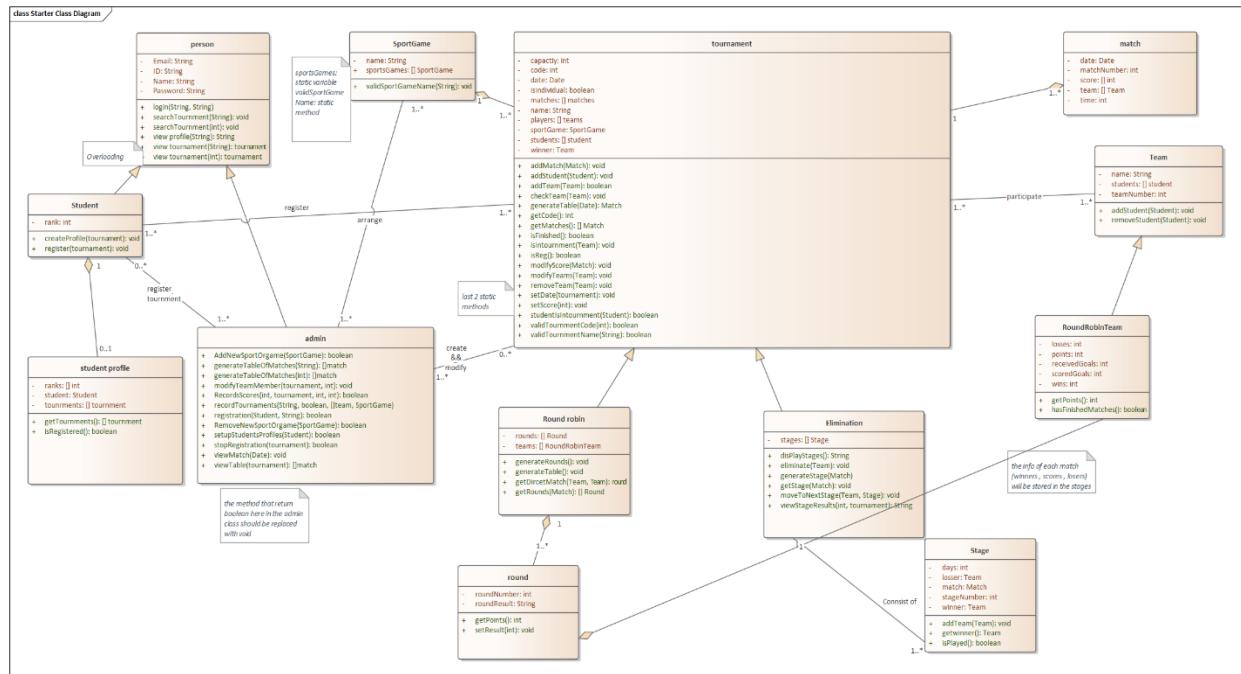


Phase 2:

Table tasks contribution (most work was common)

Muath Alghamdi	Class diagram+ act diagrams
Fahad Almutairi	Class diagram + sequence diagrams + Review the work
Naif Abdulaziz	Class diagram + sequence diagrams
Ziyad Alyahya	Class diagram + act diagrams

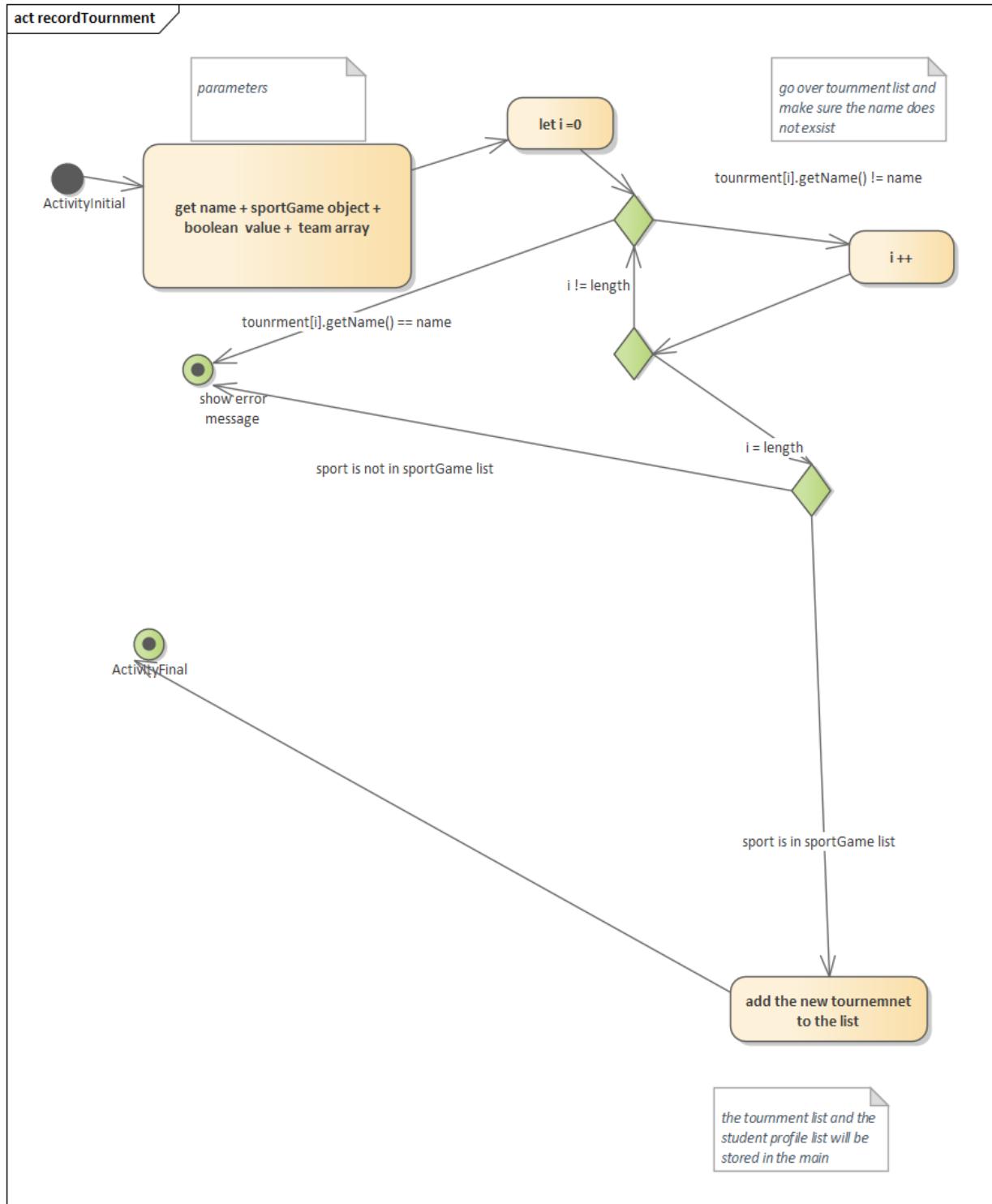
Class Diagram



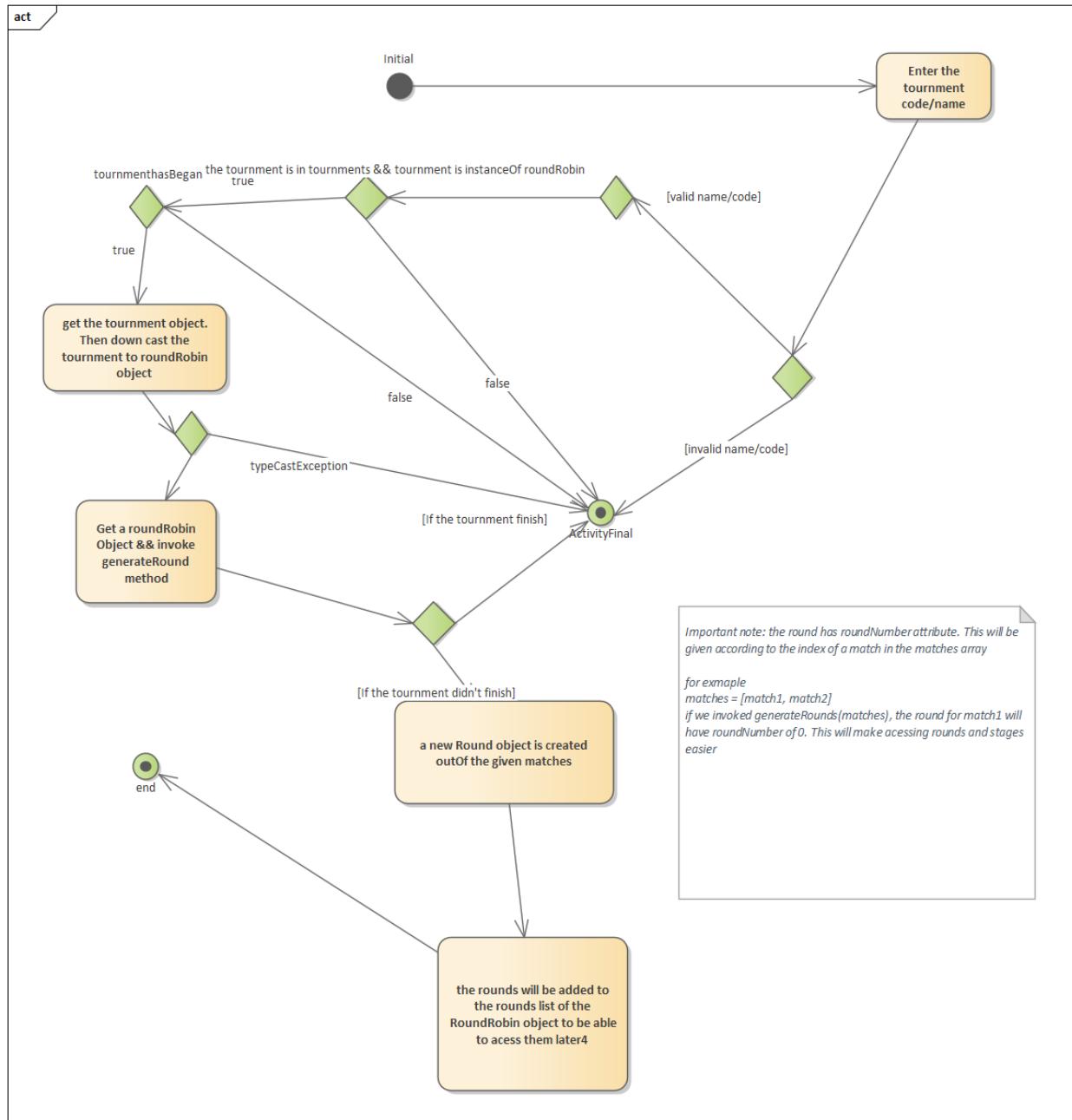
(if it is empty search for starter class)

Activities Diagrams

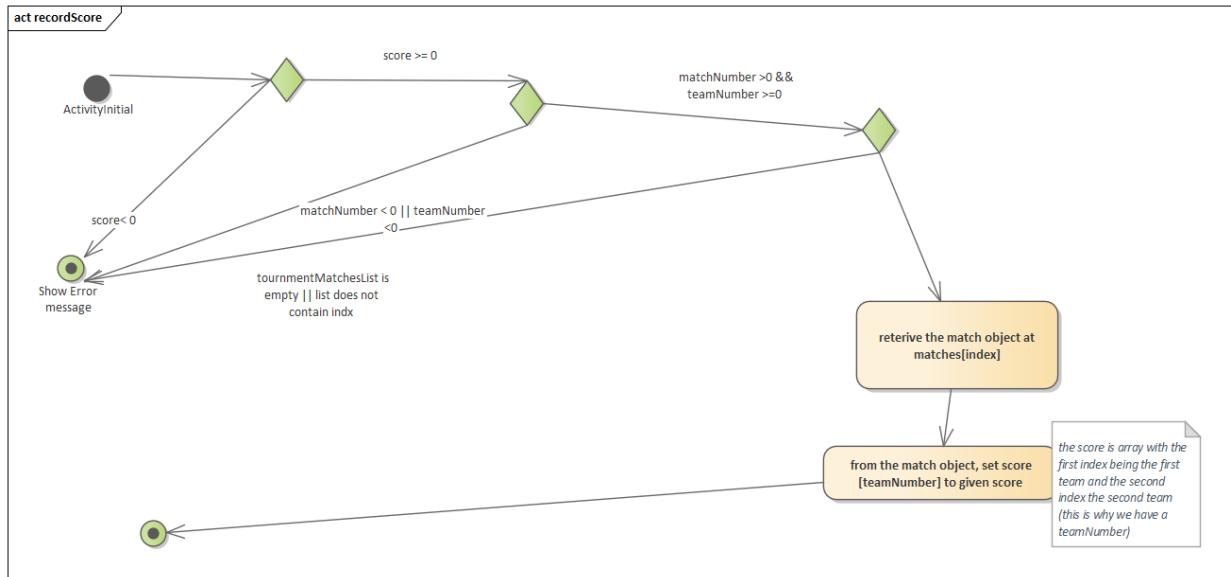
Record Tournament (note the team array will be student array if the boolean value == false that is if it is indv)



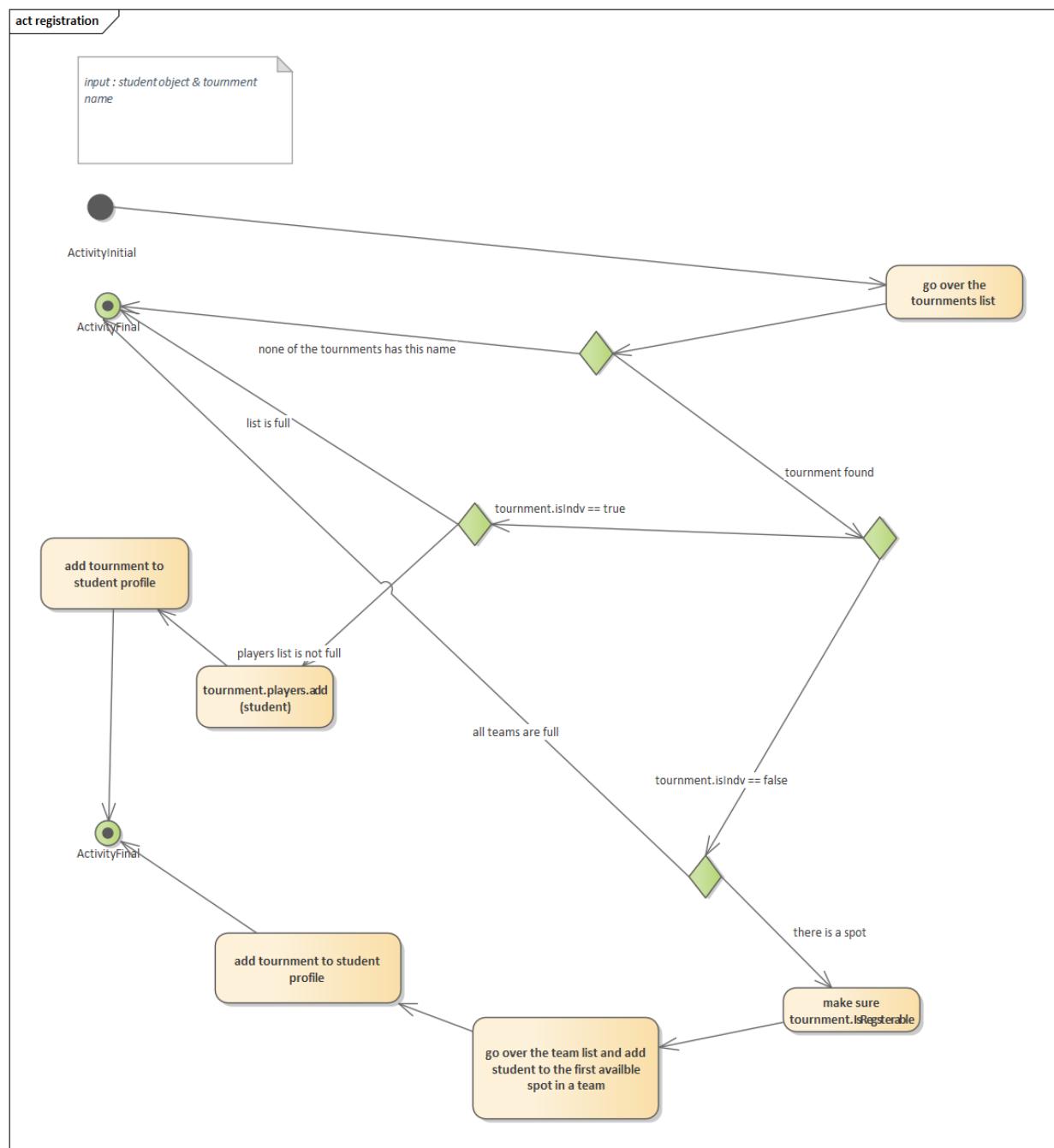
Generate rounds



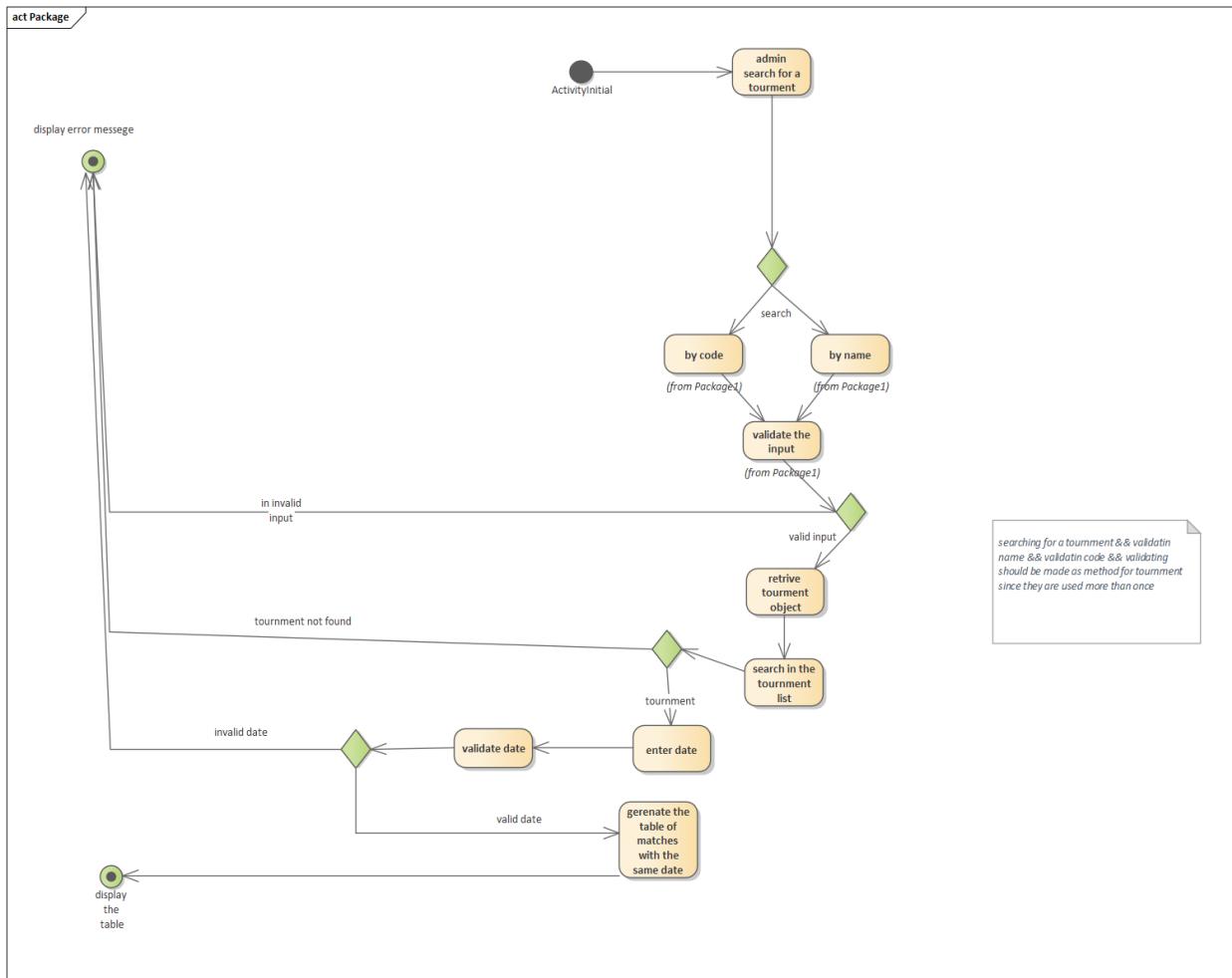
Record score (for tournaments with teams. The same logic can be applied for indv tournaments)



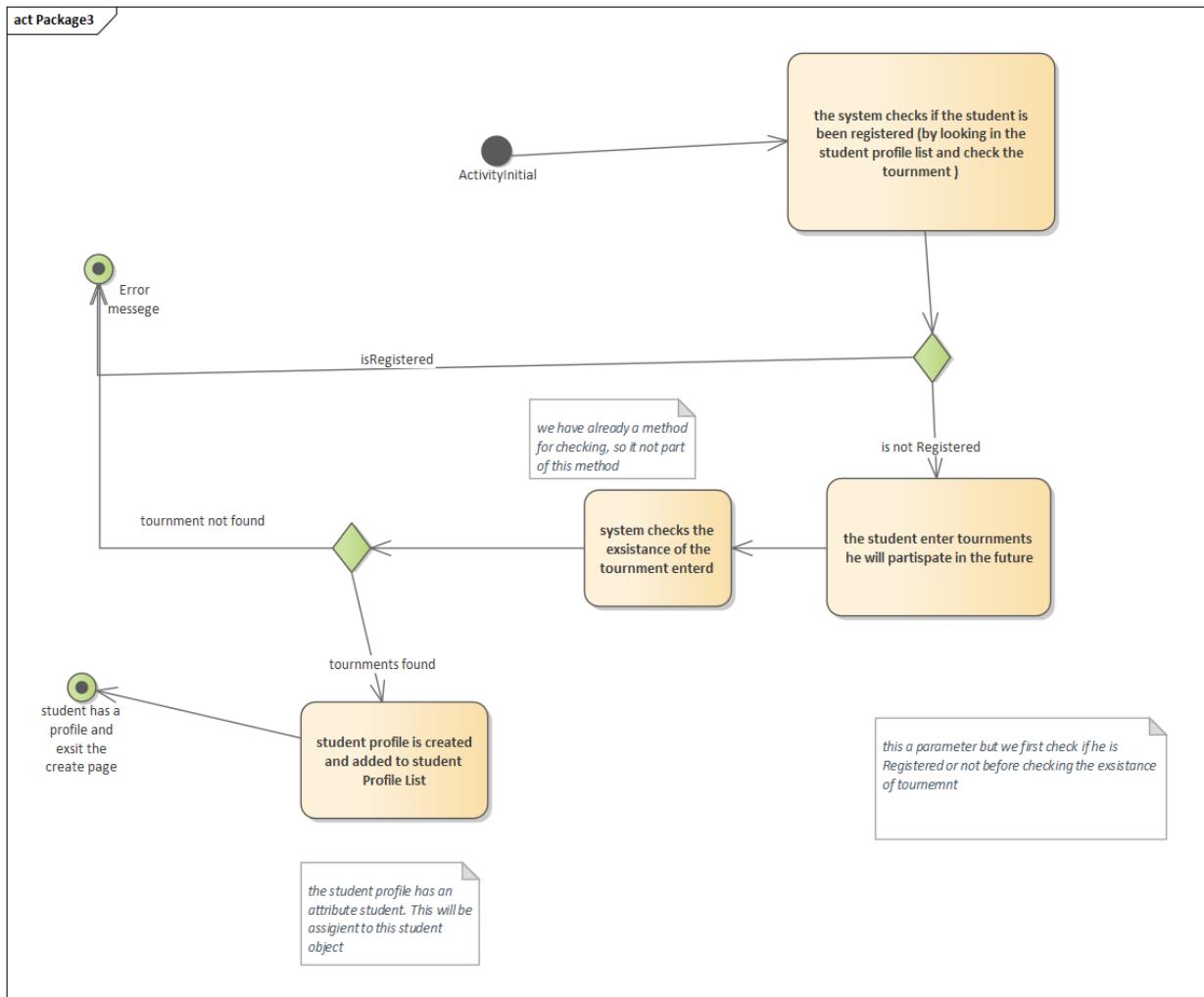
Registration



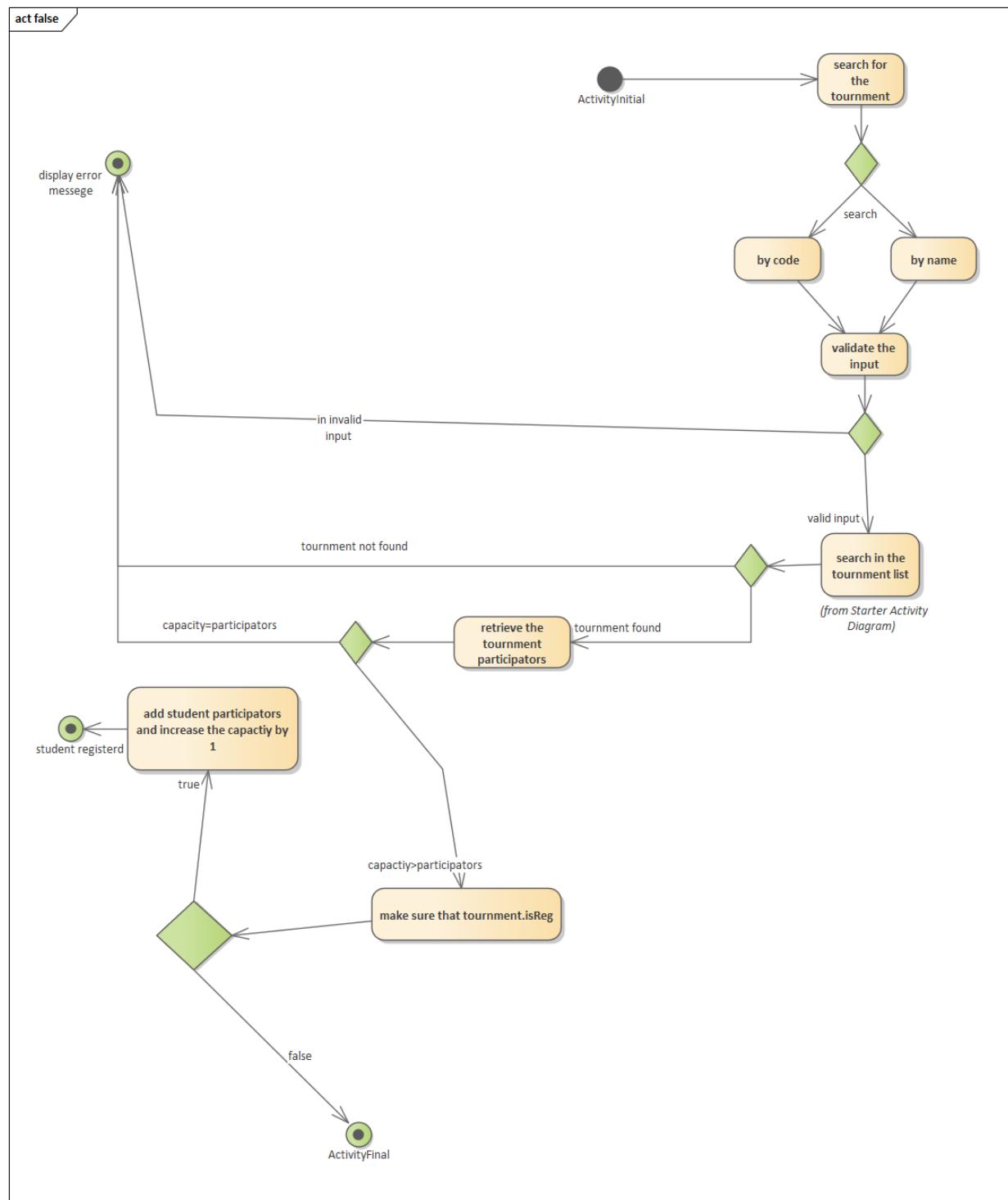
generateTableOfmatches



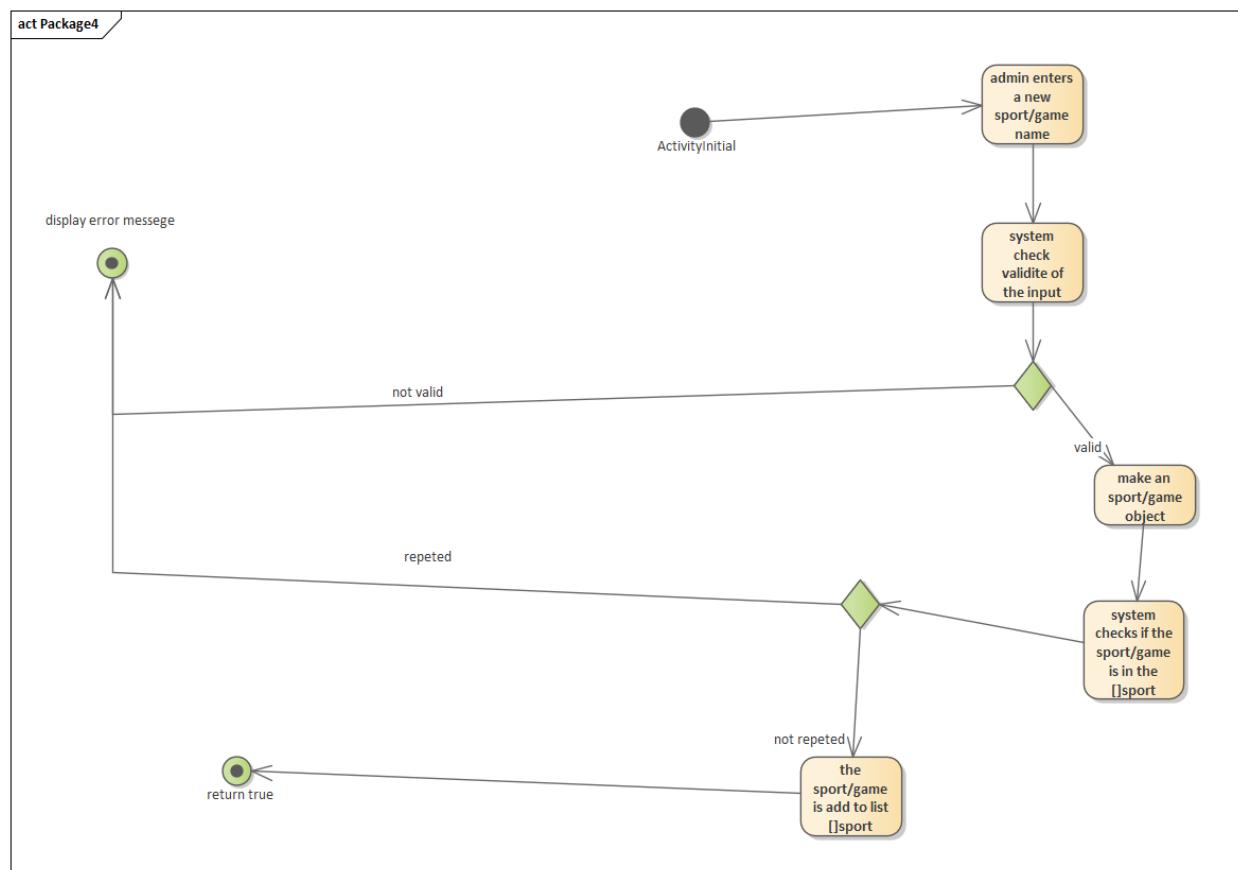
createPofile



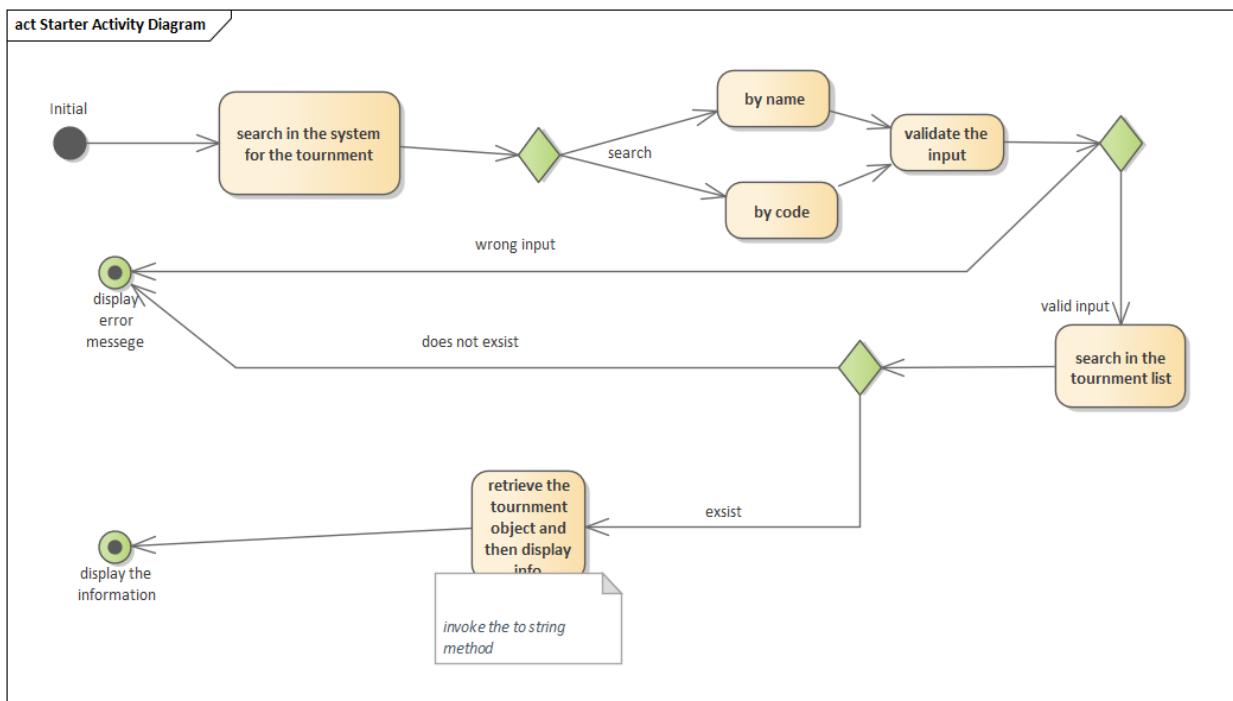
Register for student



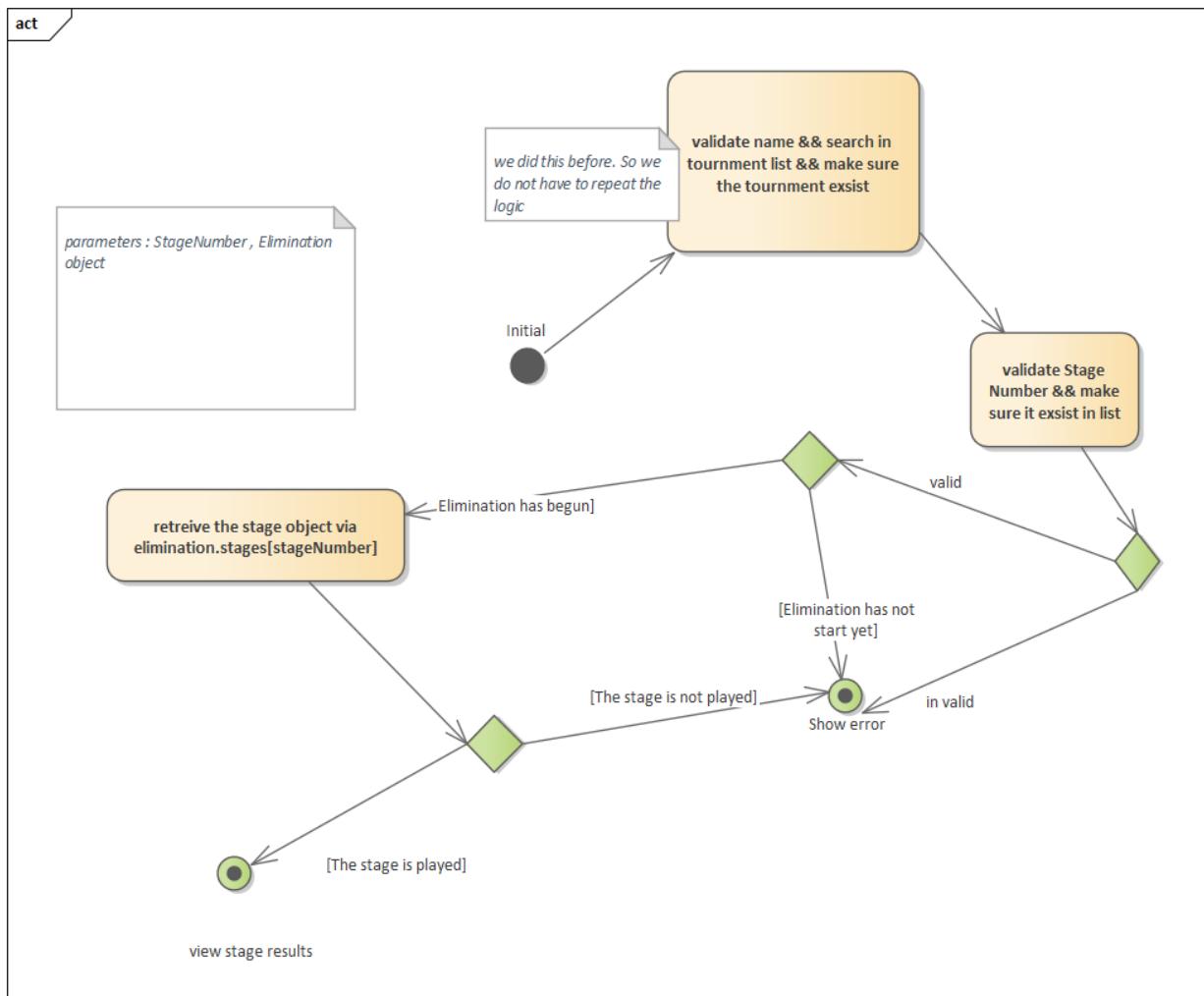
Add sportGame



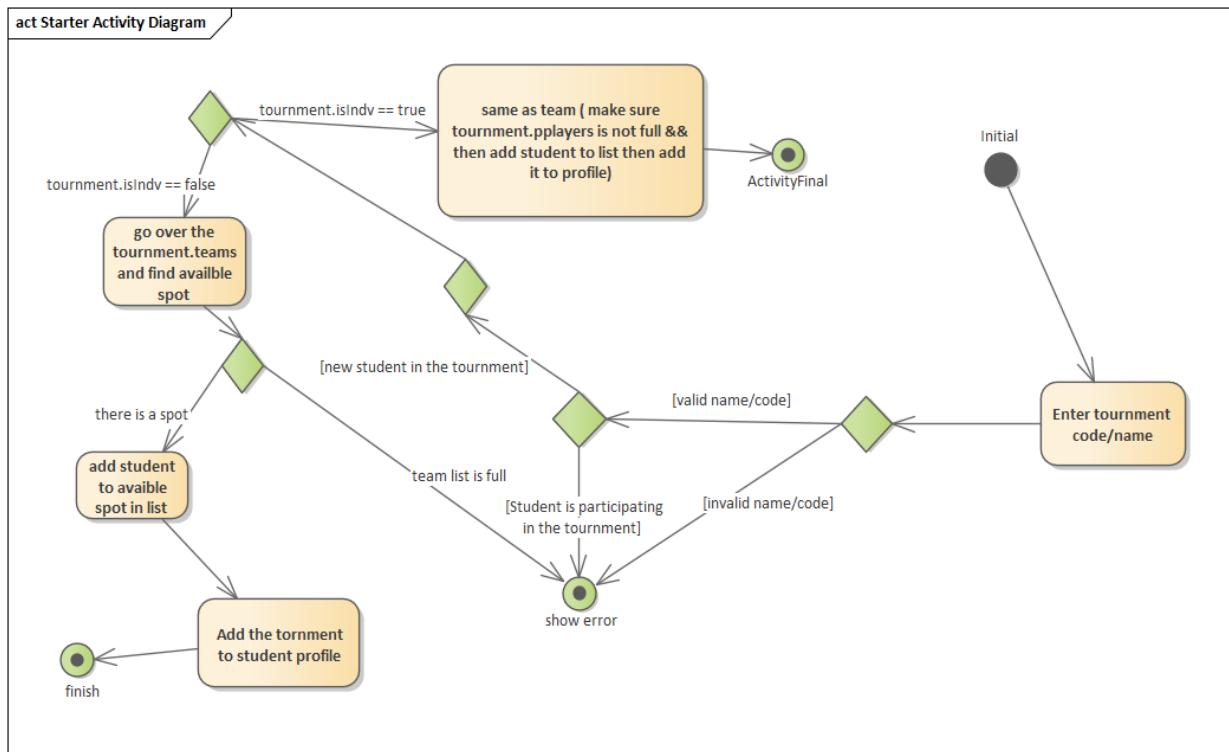
ViewTournament



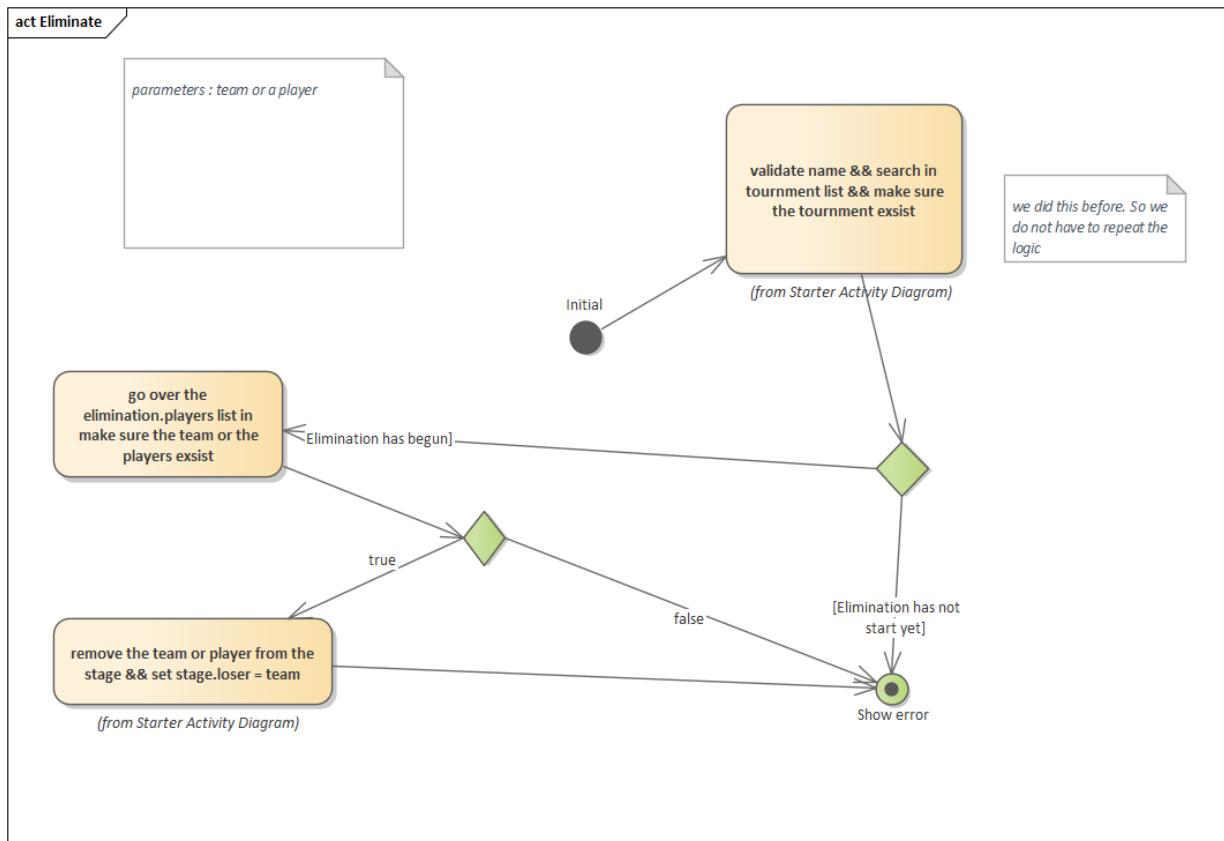
ViewStageResult



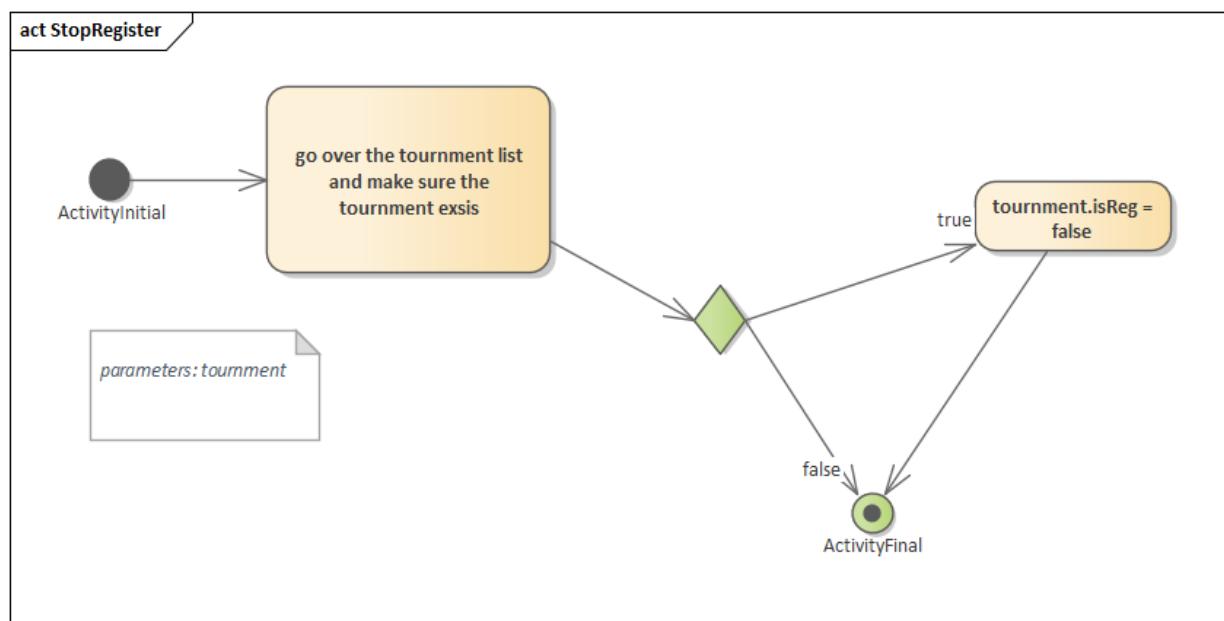
addStudent



Eliminate

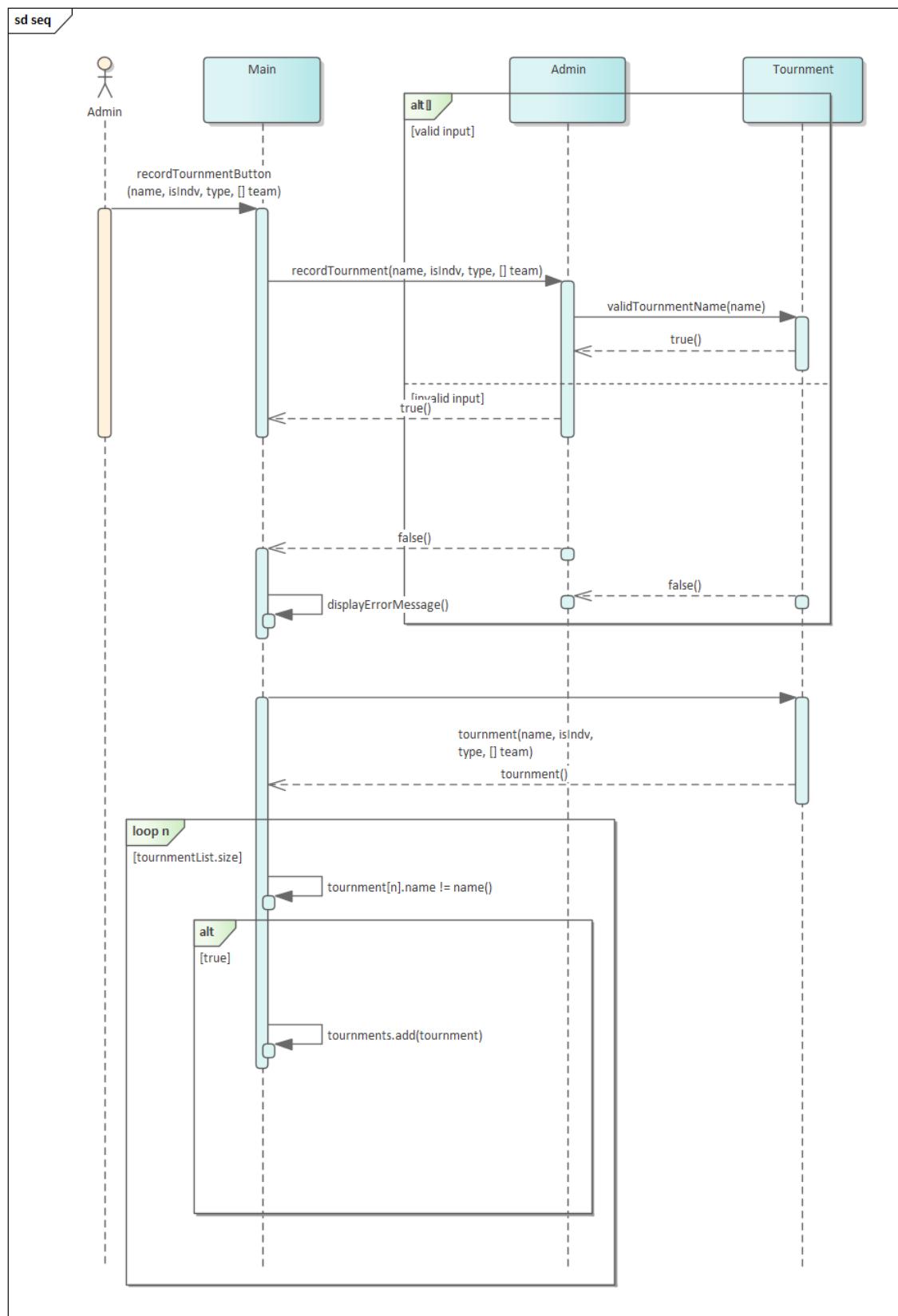


StopReg

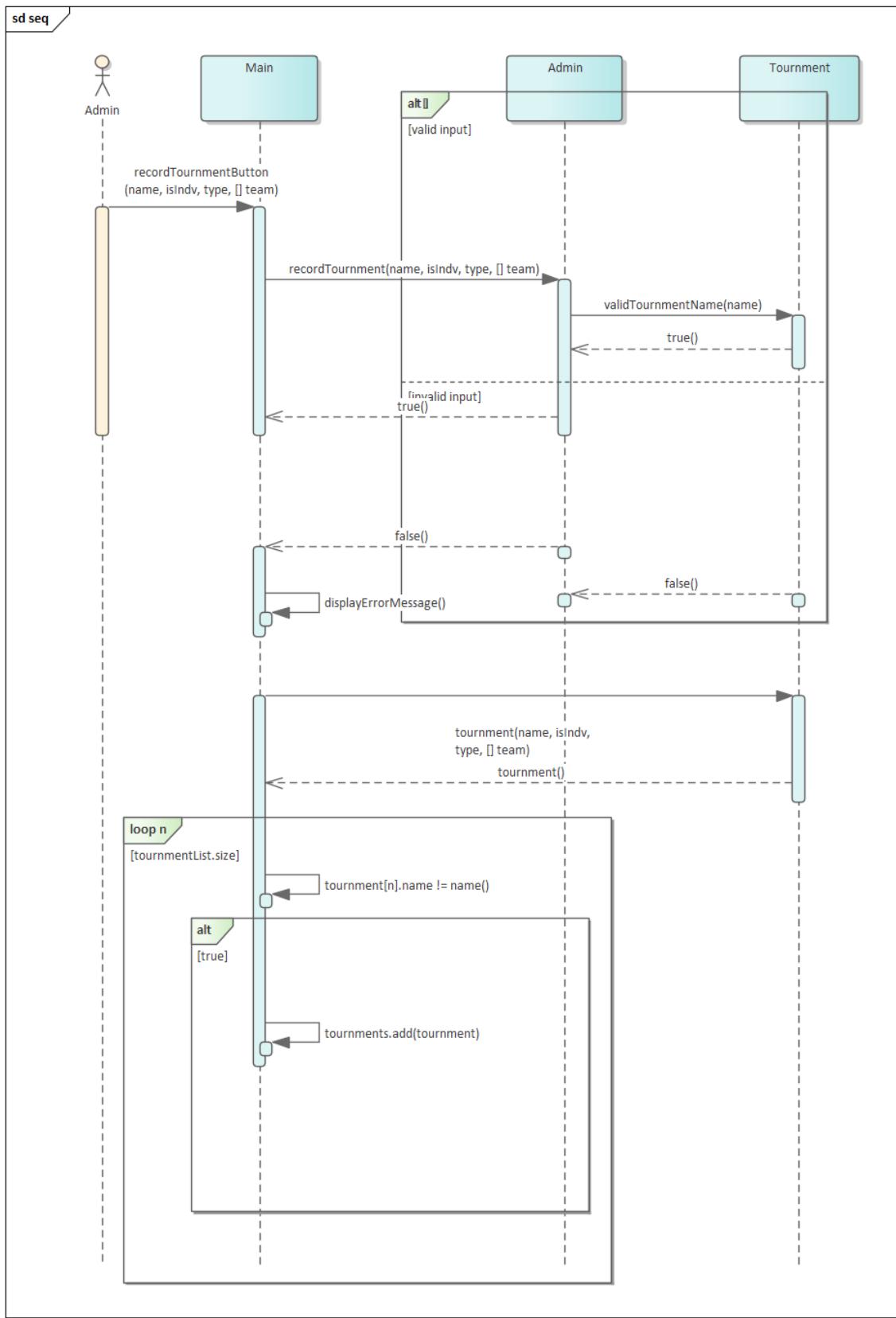


Sequence

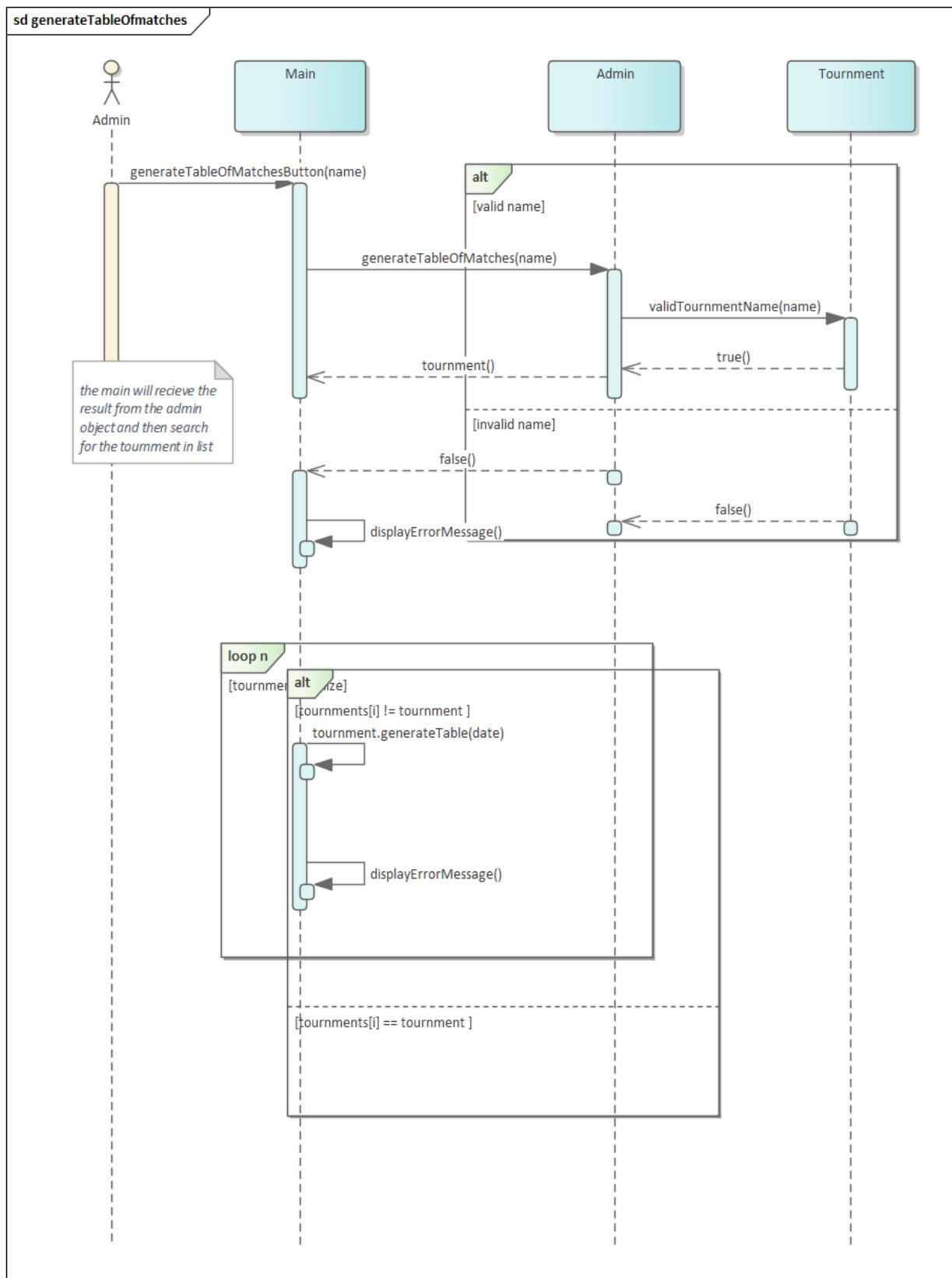
Recording a tournament



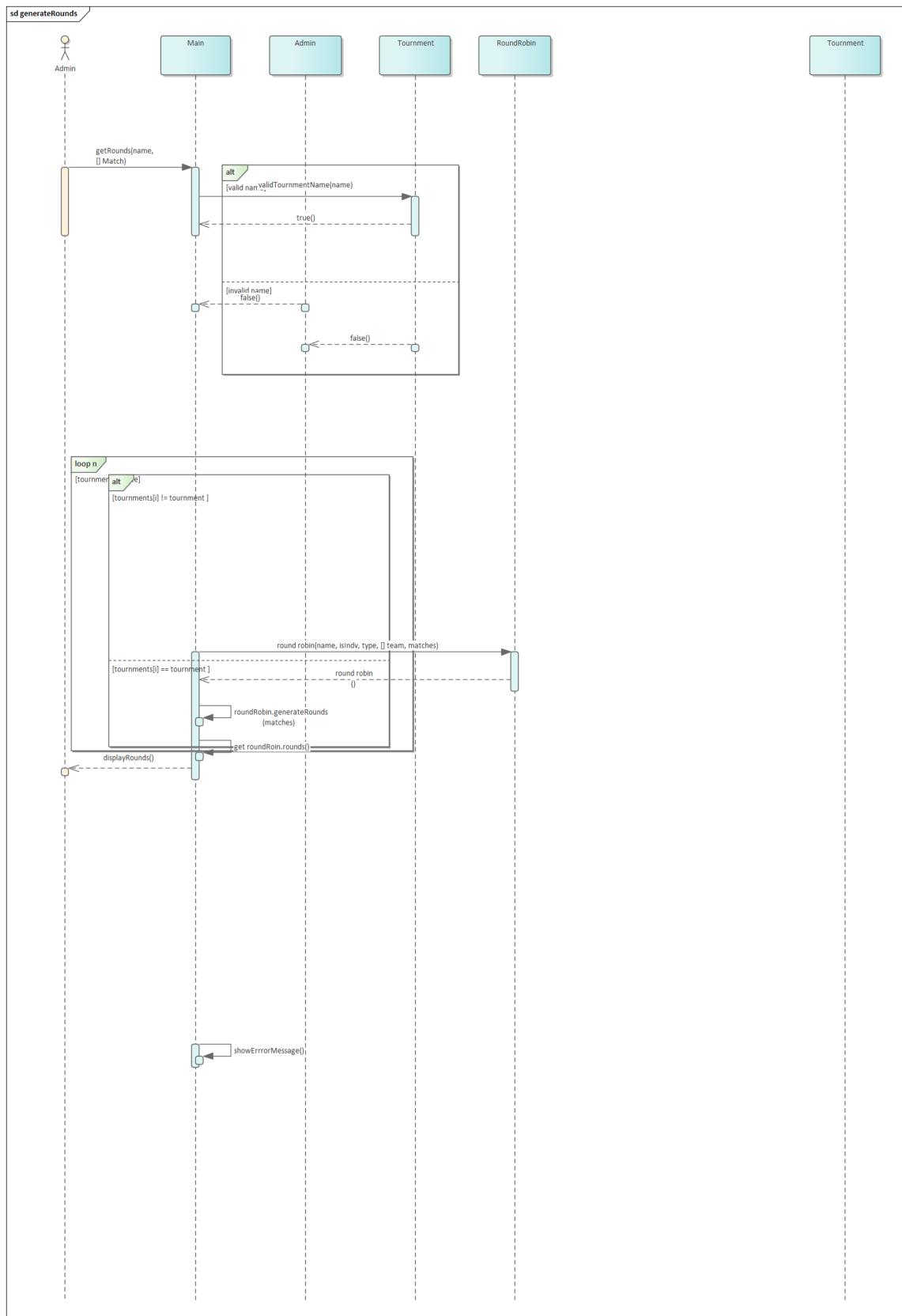
Student creating a profile



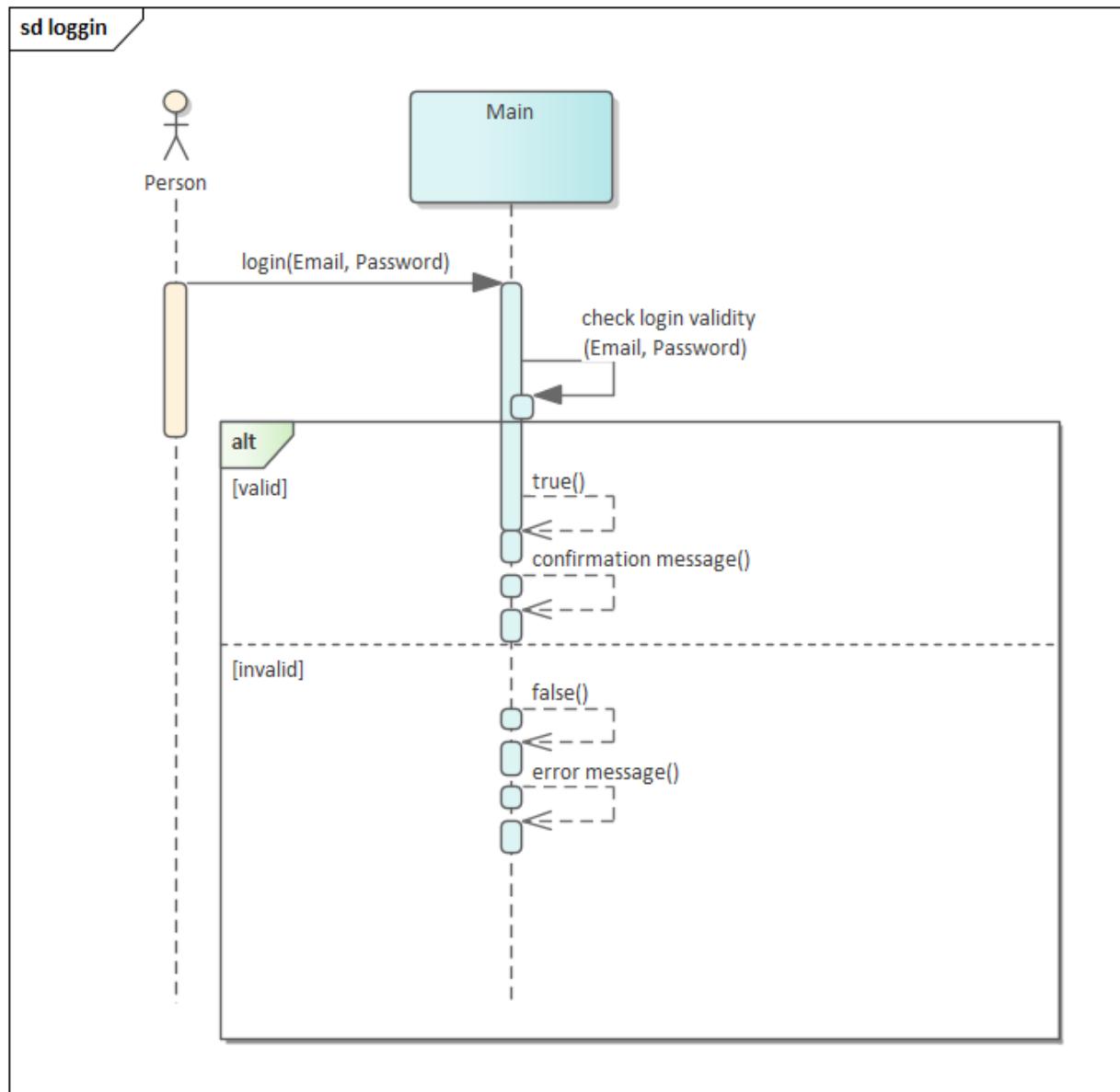
Admin generating table of matches



Admin generating rounds



Loggin in



Adding a team

