
Software Requirements Document for “Gamification for learning”

Author: Team 26

Akzhol Bakytzhan

Dosymbetov Alimkhan

Ferdaussiuly Aibek

Sauyrkulov Serikadil

Tukpetov Raiymbet

Version	Date	Author	Change
0.1	20/02/15	Group 26	Initial Document
0.2	01/03/15	Group 26	Updated with Features

Table of Contents

Software Requirements Document for “Gamification for learning”	1
Table of Contents	2
1 Introduction.....	4
1.1 Purpose	4
1.2 Scope	4
1.3 Definitions, acronymns, abbreviations	4
1.4 References	5
1.5 Overview	5
2 Overall Description.....	6
2.1 Product Perspective	6
2.1.1 Concept of Operations.....	6
2.1.2 Major User Interfaces	6
2.1.3 Hardware Interfaces.....	6
2.1.4 Software Interfaces	6
2.1.5 Communication Interfaces.....	7
2.1.6 Memory Constraints	7
2.1.7 Operations	7
2.1.8 Site Adaptation Requirements	7
2.2 Product functions	8
2.2.1 Actors	9
2.2.2 Use cases:	9
2.2.3 Example use case.....	10
2.3 User characteristics.....	30
2.4 Constraints.....	30
2.5 Assumptions and Dependencies	30
3 Specific Requirements	31
3.1 External Interface Requirements	31
3.1.1 User Interfaces.....	31
3.1.2 Hardware Interfaces.....	31
3.1.3 Software Interfaces	31
3.1.4 Communications Interfaces	31
3.2 FEATURES.....	31
3.2.1 Registration	31
3.2.2 Login	32
3.2.3 Student play tournament	33
3.2.4 The common statistics all students.	34

3.2.5	Control tournament by a teacher.....	35
3.2.6	Added/Delete student in the groups.....	36
3.2.7	Manage of users.....	37
3.2.8	Add, edit, delete news	38
3.2.9	Admin show statistics of students.....	39
3.2.10	Students show yourself statistics and honors.....	40
3.3	Performance requirements	41
3.4	dDesign Constraints.....	41
3.5	Software System Attributes	41
3.5.1	Reliability	41
3.5.2	Availability	41
3.5.3	Security.....	42
3.5.4	Maintainability	42
3.5.5	Portability	42
3.6	Other Requirements	42

1 Introduction

1.1 PURPOSE

The purpose of this SRS document is to provide a detailed requirements information and description of the functionalities of the Gamification for learning system.

1.2 SCOPE

This document will describe the use cases and features of Gamification for learning system.

1.3 DEFINITIONS, ACRONYMS, ABBREVIATIONS

Term	Description
Availability	available useful work
Alternate flow	is a step or a sequence of steps that achieves the use case's goal following different steps than described in the main success scenario.
Features	is an intentional distinguishing characteristic of a software item (in performance, portability, or - especially - functionality)
Maintainability	is index of ability to changes something on the system
Main flow	An unconditional set of steps that describe how the use case goal can be achieved and all related stakeholder interests can be satisfied. Each step is essential to achieving the use case goal (no step can be skipped), and each step succeeds.
Portability	the portability of a piece of software to multiple platforms
Post-conditions	is a condition or predicate that must always be true just after the execution of some section
Pre-conditions	is a condition or predicate that must always be true just prior to the execution of some section
Reliability	the ability of a system or component to perform its required functions under stated conditions for a specified period of time.
Use case	is a list of steps, typically defining interactions between a role (known in

	Unified Modeling Language (UML) as an "actor") and a system, to achieve a goal.
--	---

1.4 REFERENCES

<https://code.google.com/p/gamification-sdu-en3-04/source/browse/Docs>

1.5 OVERVIEW

[OMIT]

2 Overall Description

Gamification for learning system is a useful web application for university SDU to teaching students easy and help them tighten knowledge of programming, English, mathematics with gaming system. They are can plays games for learning together, and receive points and give honors. Teachers can creates tournaments and manage their, and can see results of students. And each user will have login and password.

2.1 PRODUCT PERSPECTIVE

This application is similar to several existing gaming application such as kahoot, battlecode, blackboard etc. The product is called “Gamification for learning” of Engineer first course students SDU will be developed on the web application on domain kz. This product is based on modern technologies HTML5, CSS3, styles bootstrap. Product can run on a personal computer with minimum system requirements and even on mobile devices. And it will focus on mobile devices. Support for asynchronous transfer to both sides: client-server and server to client.

2.1.1 Concept of Operations

This application will run on the sdu.edu.kz server system.

2.1.2 Major User Interfaces

See Appendix_

2.1.2.1 Example Screenshot and description

See Appendix_

2.1.3 Hardware Interfaces

This software requires no more than standard personal computer peripherals.

2.1.4 Software Interfaces

// example: CGI-URL or function signatures etc (OMIT for now).

2.1.5 Communication Interfaces

// example: modem etc (OMIT for now)

2.1.6 Memory Constraints

// RAM, and other storage constraints (OMIT for now)

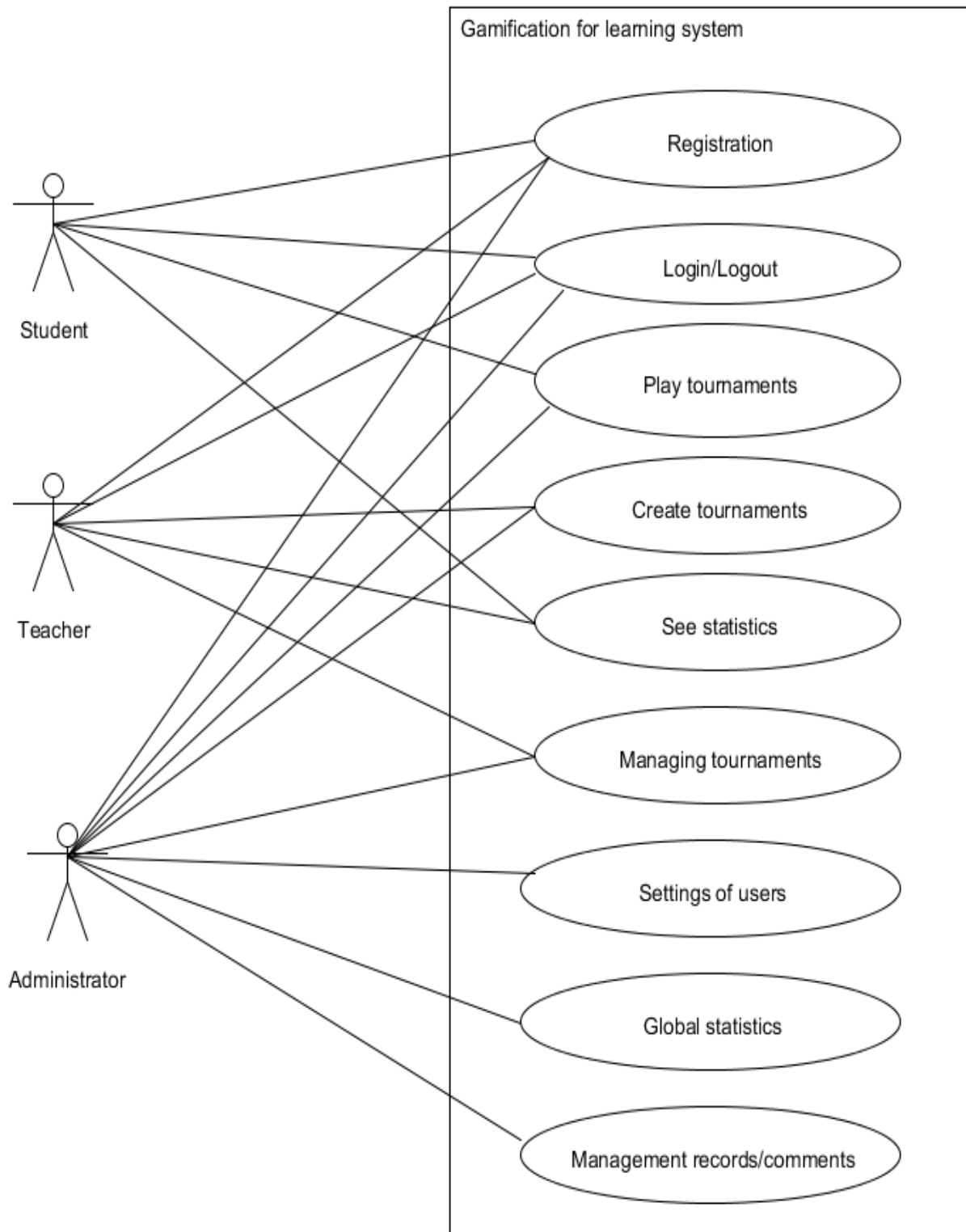
2.1.7 Operations

// special operations (if any) (OMIT for now)

2.1.8 Site Adaptation Requirements

//ex: Japanese language etc (OMIT for now)

2.2 PRODUCT FUNCTIONS



2.2.1 Actors

Student – plays tournaments and learn.

Teachers –. create tournaments and shows results of students.

Administrator – control and administrate all contents.

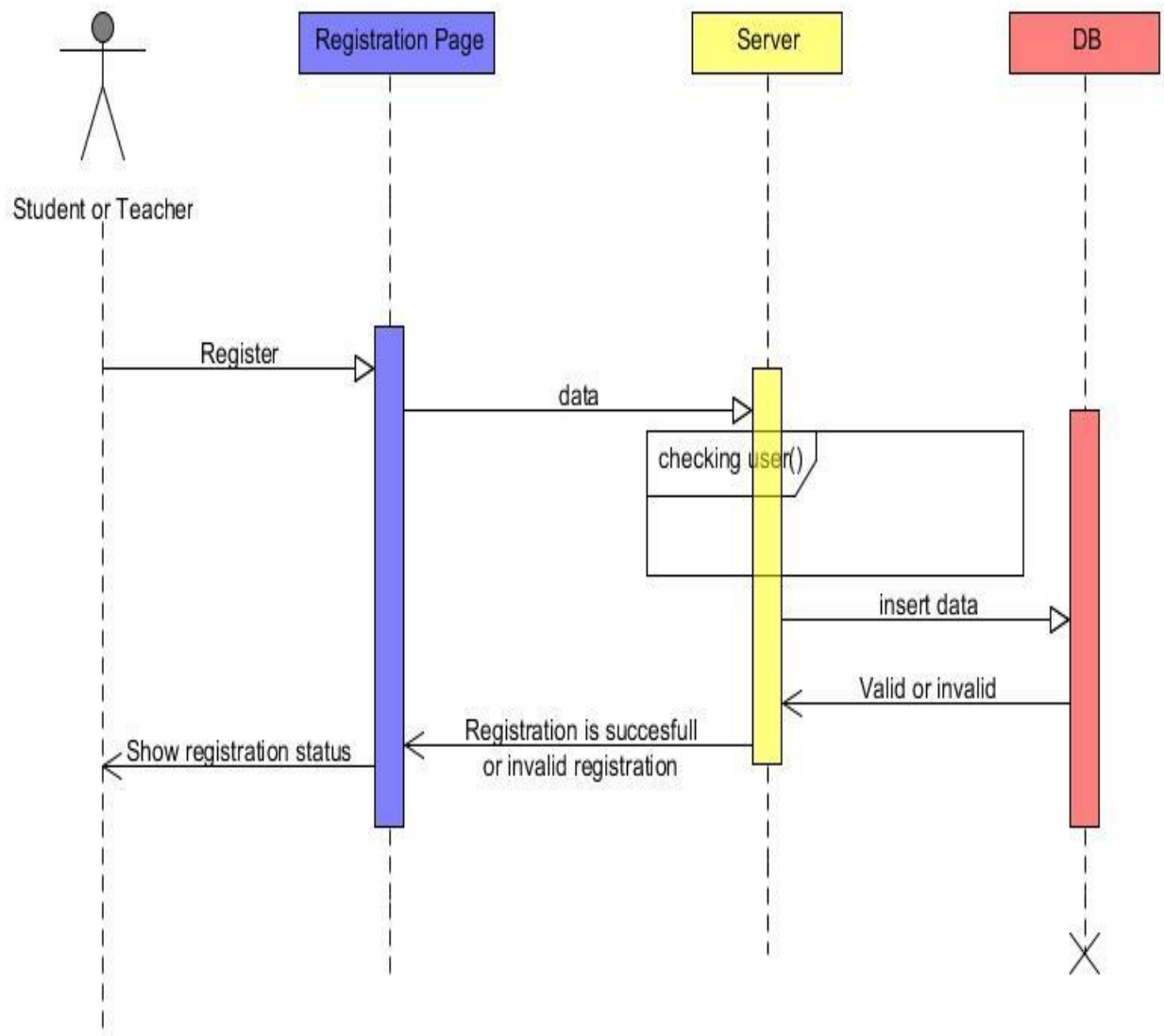
2.2.2 Use cases:

- **Student:**
 - Registration
 - Authorization
 - Play tournaments
 - Takes honors and score
- **Teacher**
 - Registration
 - Authorization
 - Create tournaments
 - Show statistics
 - Management tournaments
- **Administrator**
 - Registration
 - Authorization
 - Play tournaments
 - Create tournaments
 - Show statistics
 - Management tournaments
 - Settings of users
 - Managing comments and records

2.2.3 Example use case

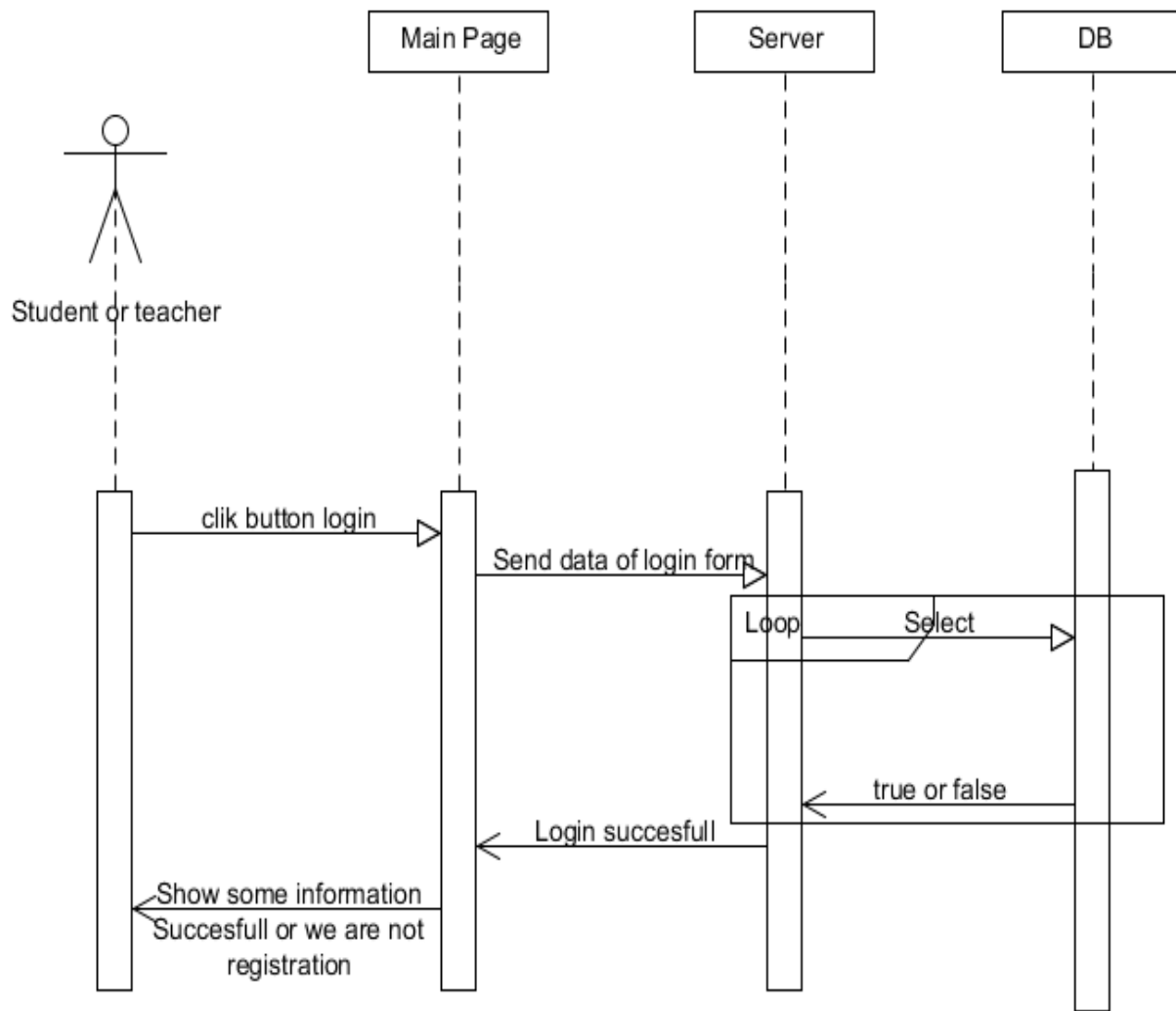
2.2.3.1 Registration

- Description: To use create a new account
- Actors: student, teacher, admin
- Main flow:
 - User fills in necessary fields(name, surname, e-mail, etc.) and presses SUBMIT button
 - Client sends server a request to register new account with provided information.
 - Server composes and runs SQL request to insert data into `users` table in the database.
 - When insertion is successful, server sends client appropriate message.
 - Client receives server message and opens Main Window.
- Alternate flow:
 - User fills in necessary fields(name, surname, e-mail, etc.) and presses SUBMIT button
 - Client sends server a request to register new account with provided information.
 - Server composes and runs SQL request to insert data into `users` table in the database.
 - When user already exist, server sends client appropriate message.
 - Client receives server message.
- Pre-conditions: User installed application on his/her computer.
- Post-conditions: User will have new account in the system.



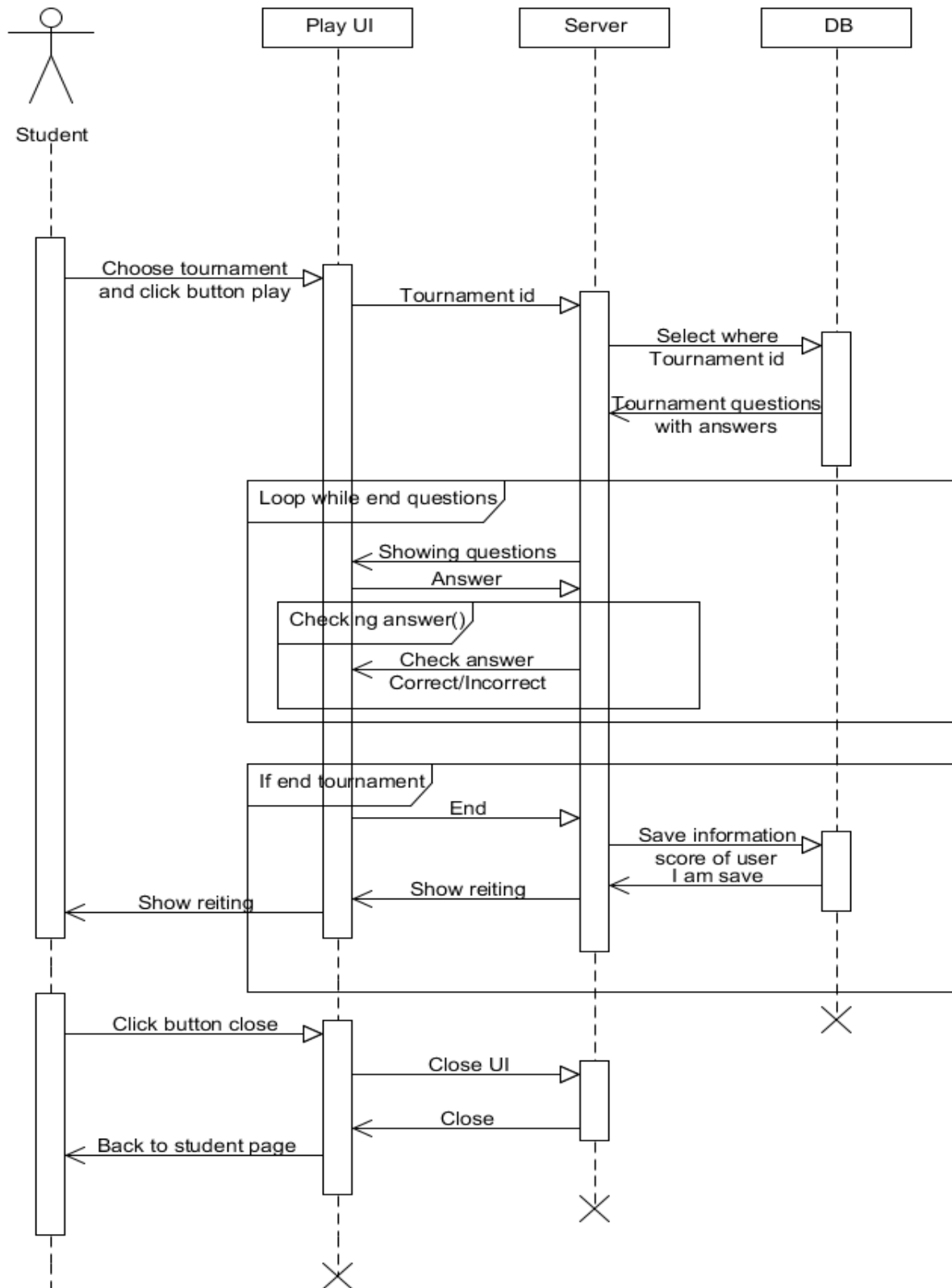
2.2.3.2 Authorization (Login)

- Description: To use login/logout to account
- Actors: student, teacher, admin
- Main flow:
 - User fills login form and click button login
 - Client sends username and password to server
 - Server select where username and password on database and checking
 - When username and password incorrect, server sends appropriate message to client
 - Client receives message
- Alternate flow:
 - User fills login form and click button login
 - Client sends username and password to server
 - Server select where username and password on database and checking
 - When username and password correct, server sends appropriate message to client
 - Client receives message and open user page
- Pre-conditions: User will be registered on server.
- Post-conditions: User login on yourself account.



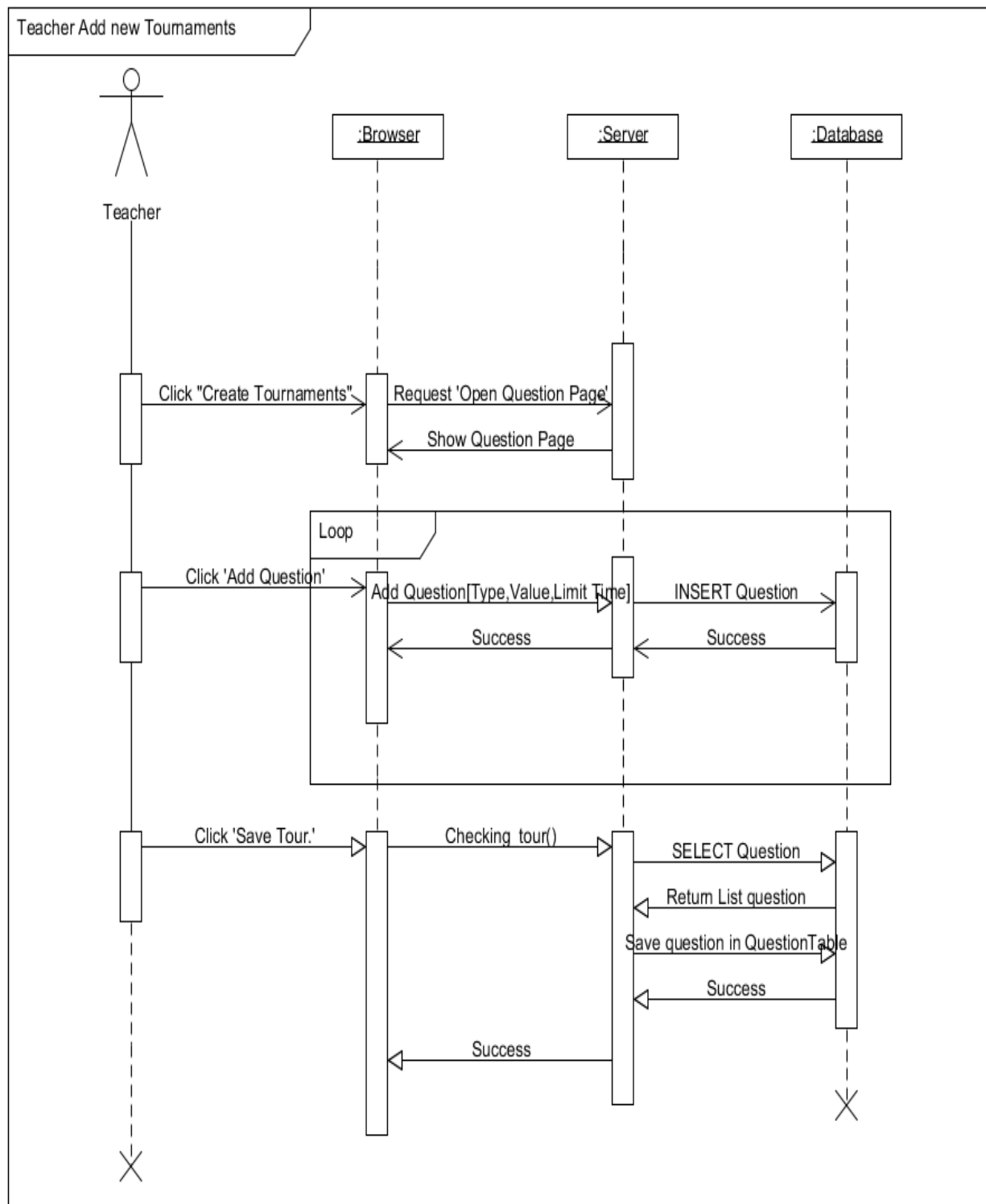
2.2.3.3 Play tournaments

- Description: To use play tournaments on system and take scores and honors
- Actors: students, admin
- Main flow:
 - User choose tournament and click button play sends id tournament to server
 - Server receive id and select questions and answers on database
 - While all questions answered server sends questions to client and client sends answers to server , so server check answer and send result to client
 - When ends server save data of clients and send appropriate message
 - Client receive message and open user page
- Alternate flow: no
- Pre-conditions: User will be registered student
- Post-conditions: User take score and honors(if merit)



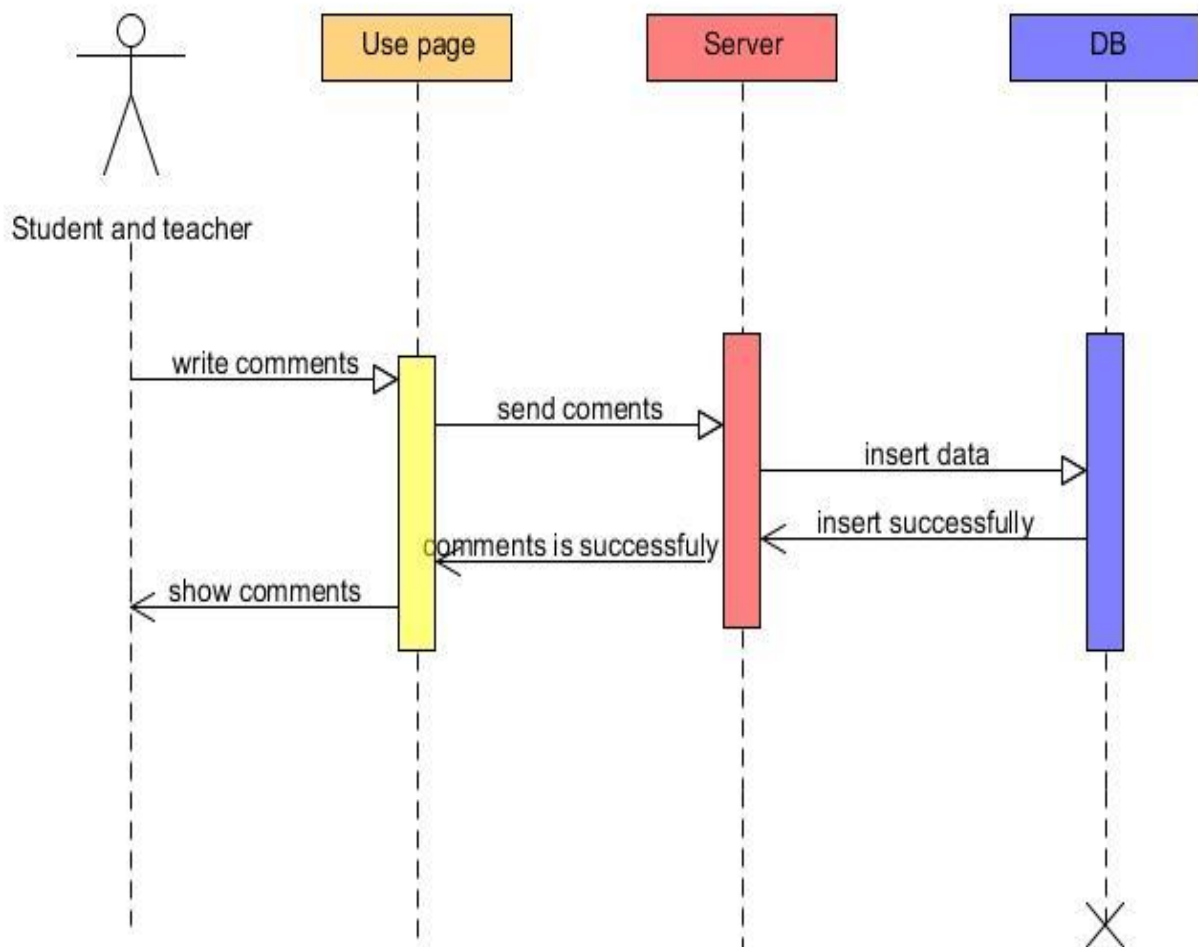
2.2.3.4 Create tournaments

- Description: To use create a new tournament
- Actors: teacher, admin
- Main flow:
 - User presses button Create tournament and send request to server
 - Server receive and send question page with forms
 - Client fill forms and can add questions to tournament ,so presses button save client send request to server
 - Server save tournament and send appropriate message
 - Client receive message and open user page
- Alternate flow: no
- Pre-conditions: User will be registered teacher
- Post-conditions: User creates new tournament and must be viewed on my tournaments



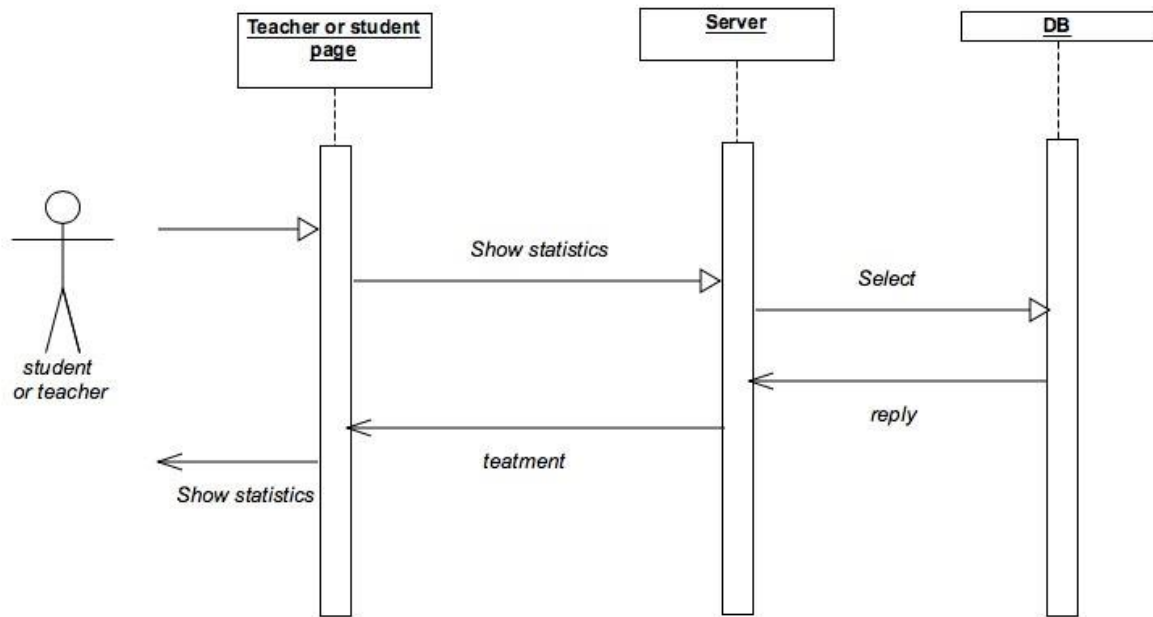
2.2.3.5 Write comment

- Description: To use write comments to tournaments
- Actors: student, teacher, admin
- Main flow:
 - User write comment and pressed button submit
 - Server receive request and run SQL transaction and send appropriate message to client
 - Client receive message
- Alternate flow: no
- Pre-conditions: Users must be logged.
- Post-conditions: User write their thinks about tournament.



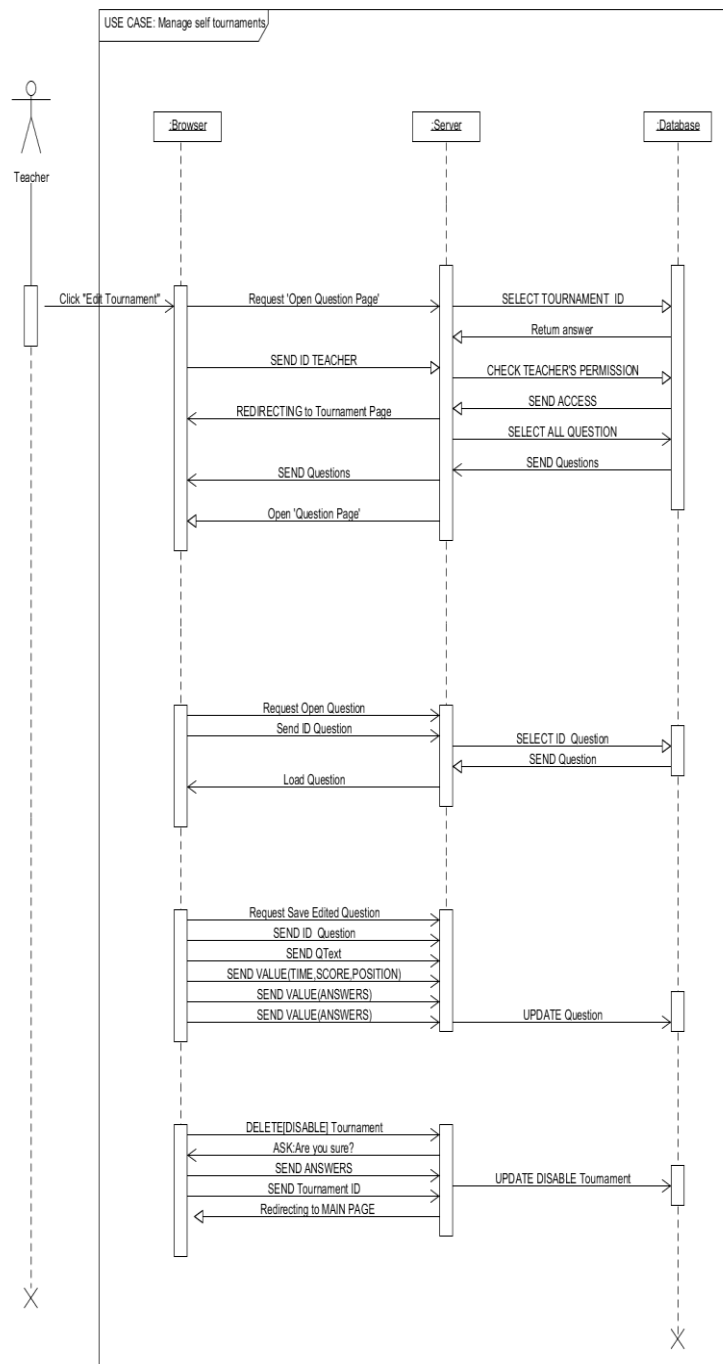
2.2.3.6 **Show statistics**

- Description: To use show statistics of the tournament
- Actors: student, teacher, admin
- Main flow:
 - User pressed button show statistics
 - Server receive request and run select statistics and send data to client
 - Client receive information
- Alternate flow:
- Pre-conditions: User must be logged and student must be take part, teacher must be author of tournament
- Post-conditions: User take statistic and show it



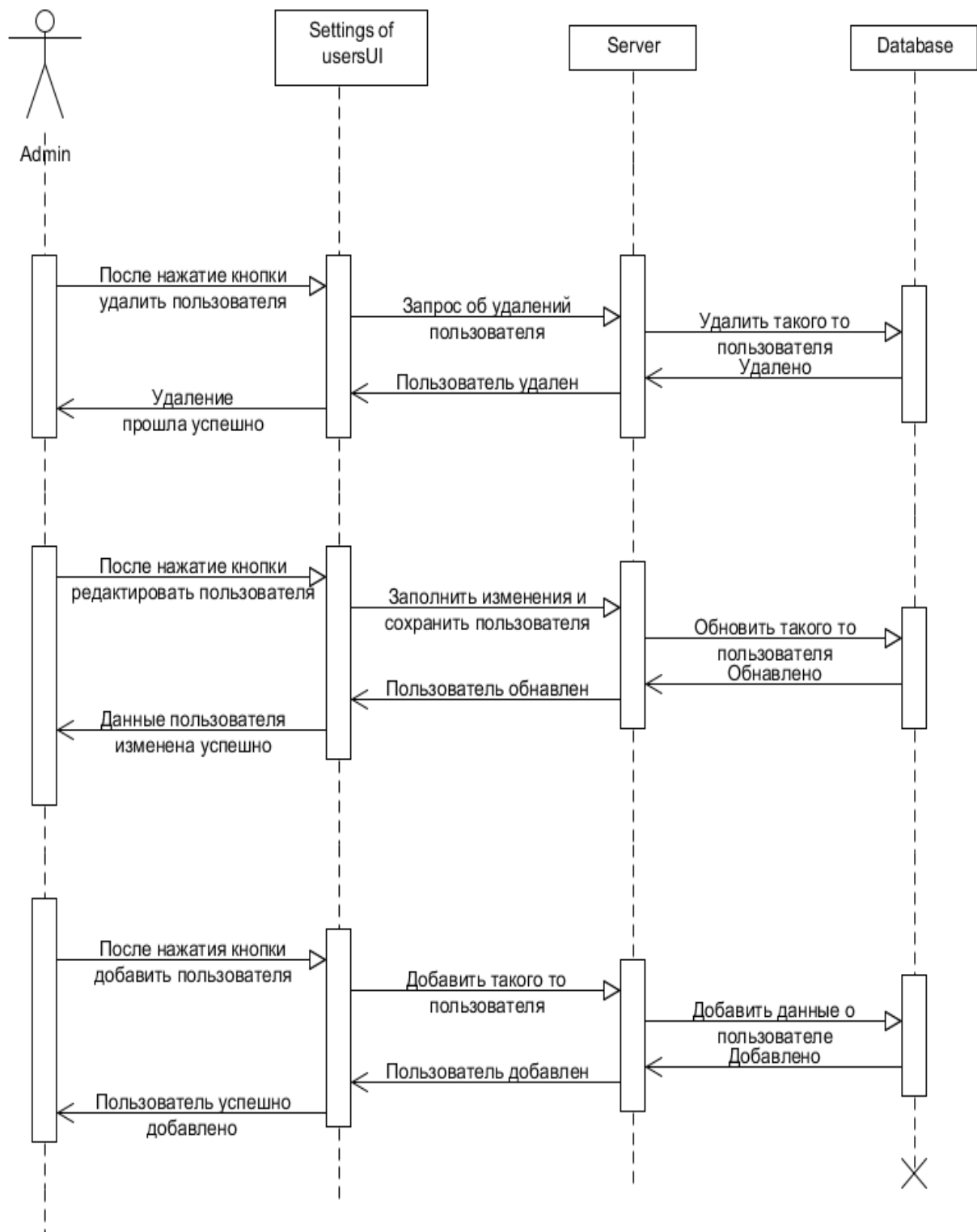
2.2.3.7 Managing tournaments

- Description: To use managing tournament and does actions such delete, update, change status
- Actors: teacher, admin
- Main flow:
 - User pressed button edit tournament
 - Server receive request and send tournament form
 - User fill form and send data to server
 - Server receive data and run update transaction and send appropriate message
 - Client receive message
 -
 - User pressed delete tournament
 - Server receive request and run delete transaction and send appropriate message
 - Client receive message
 - User pressed change status OPEN or CLOSE tournament
 - Server receive request and run change status transaction and send appropriate message
 - Client receive message
- Alternate flow: no
- Pre-conditions: User must be teacher or admin and choose tournament which want to change
- Post-conditions: Tournament will be deleted, updated, changed status.



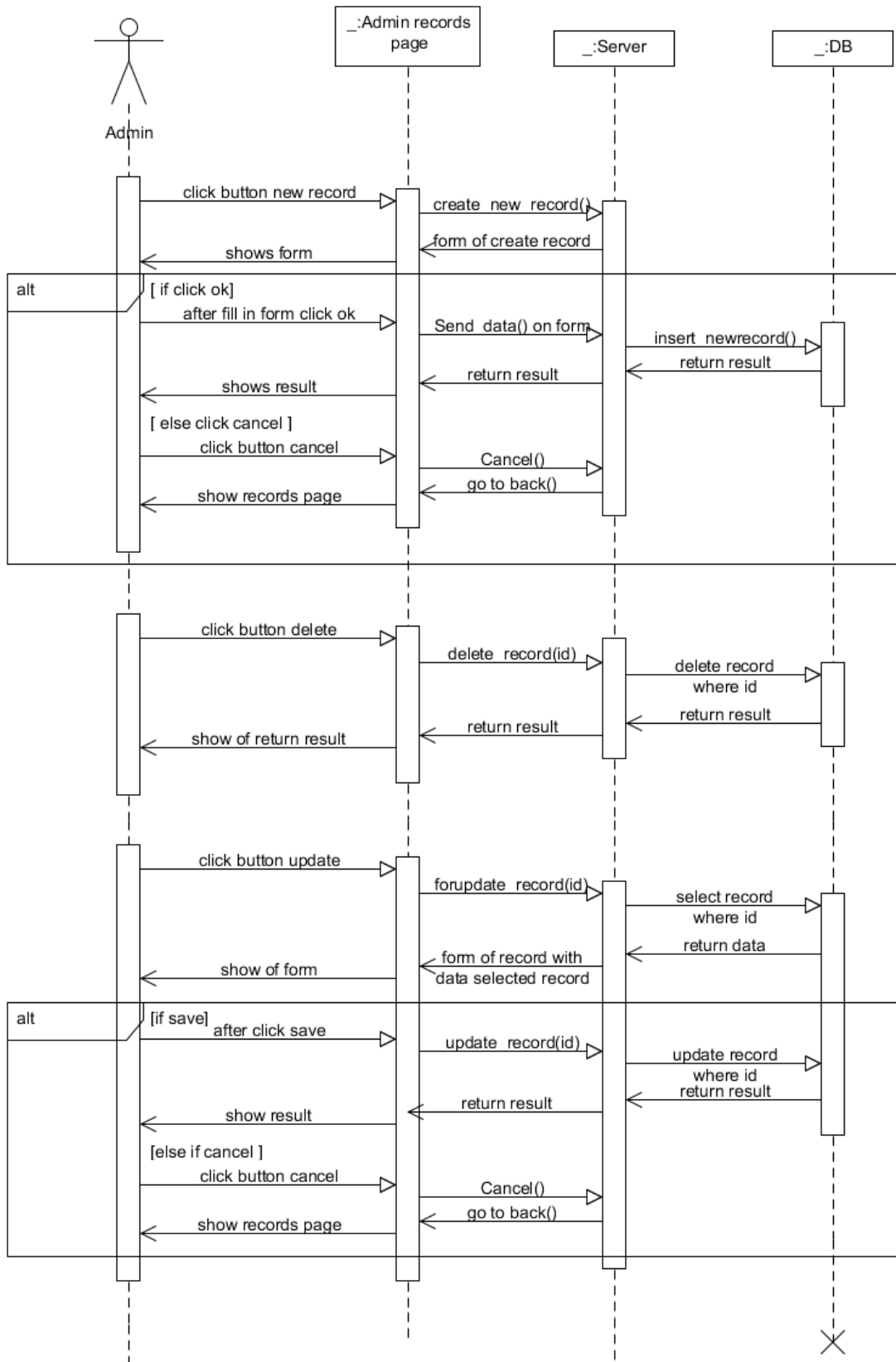
2.2.3.8 Settings of users

- Description: Use administrator for administrate of users (delete, update, add)
- Actors: admin
- Main flow:
 - When pressed delete user and send request to server
 - Server receive request and run SQL transaction and send appropriate message to client
 - Client receive message
 - When pressed add user and send request to server
 - Server receive request and run SQL transaction and send appropriate message to client
 - Client receive message
 - When pressed update user and send request to server
 - Server receive request and send user form
 - Client fill form and send data to server
 - Server receive data and run SQL transaction and send appropriate message to client
 - Client receive message
- Alternate flow:
- Pre-conditions: User must be admin and logged on yourself admin page
- Post-conditions: Admin does delete, add or update users



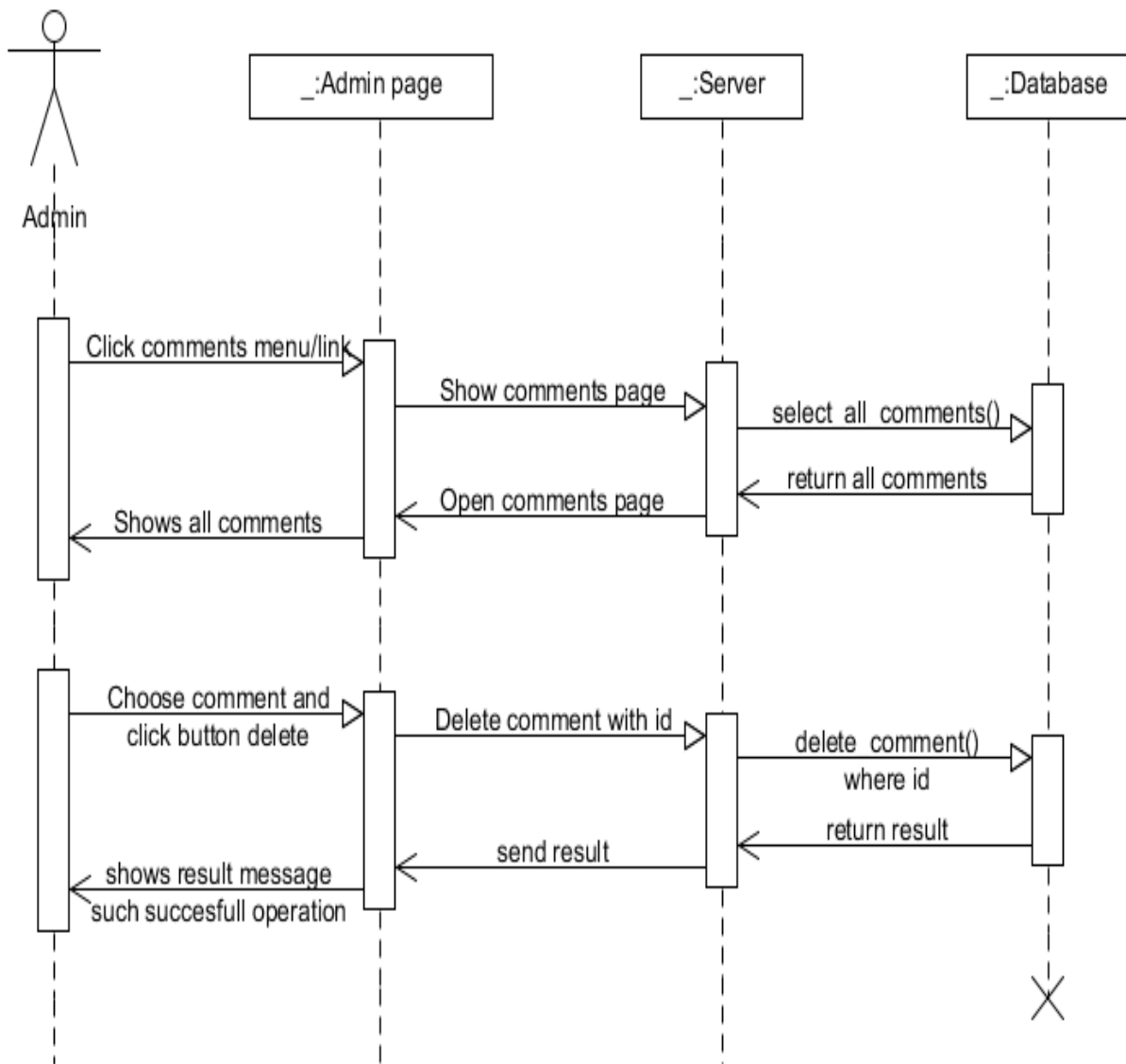
2.2.3.9 Managing records

- Description: To use administrate records such add new record, update, remove
- Actors: admin
- Main flow:
 - When client pressed button new record and send request to server
 - Server receive request and run record form
 - Client fill record form and press Ok and send data of form to server
 - Server check data and run SQL transaction insert
 - When pressed delete record
 - Server run SQL transaction delete and sent appropriate message
 - Client receive message
 - When pressed update record
 - Server receive request and run record form
 - Client fill record form and press Ok and send data of form to server
 - Server check data and run SQL transaction update
- Alternate flow:
 - If client fills form incorrect and send data to server
 - Server check data and send error message to client
 - Client receive message
- Pre-conditions: User installed application on his/her computer.
- Post-conditions: User will have new account in the system.



2.2.3.10 Managing comments

- Description: To use administrate comments such delete, add to reserve.
- Actors: admin
- Main flow:
 - Client press menu comments
 - Server open comments page
 - Client choose comments and press button delete
 - UI send appropriate message to server
 - Server do this request on database
 - Return appropriate message to client
 - Client receive message and open comments page
- Alternate flow: no
- Pre-conditions: User must be admin and logged on admin page
- Post-conditions: Admin does something with comments



2.3 USER CHARACTERISTICS

- **Students:** A men who are studying on first course of engineering faculty in the university of SDU and use the system to learn English, programming, mathematics easy and obvious with support gaming system.
- **Teachers:** Teachers of SDU which creates yourself tournaments and controlling it(ex. Delete, update, open, close...) and see statistics of students on the tournament.
- **Administrator:** PhD Meirambek Zhaparov who administrate all the contents, users of the web site and managing all data.

2.4 CONSTRAINTS

This system is used mainly by students of first course engineers and instructors of university SDU.

This is an application for university SDU.

This web application is used only for support to learning students.

2.5 ASSUMPTIONS AND DEPENDENCIES

Network dependencies

Assumed this product will be used on PC, and mobiles devices(Android, IOS)

Assumed that server will supported around 2000 users

3 Specific Requirements

3.1 EXTERNAL INTERFACE REQUIREMENTS

3.1.1 User Interfaces

See-assignment1

<https://gamification-sdu-en3-04.googlecode.com/svn/Screenshots>

3.1.2 Hardware Interfaces

none

3.1.3 Software Interfaces

none

3.1.4 Communications Interfaces

3.2 FEATURES

3.2.1 Registration

- 3.2.1.1.1 The Gamification for learning shall allow the users to register on server.
- 3.2.1.1.2 The user shall press button “Registration” for register on server
- 3.2.1.1.3 The Gamification for learning shall allow to users enter user’s data for register.
- 3.2.1.1.4 The user shall type his/her unique username and not unique password into register form.
- 3.2.1.1.5 On registration form shall must have username, password, telephone, e-mail fields
- 3.2.1.1.6 The user shall press button “Save” to save data in registration form on server database

3.2.1.2 Inverse Requirements for registration

- 3.2.1.2.1 The Gamification for learning shall not register those users who didn’t fill in all mandatory fields
- 3.2.1.2.2 The Gamification for learning shall not register those users who are already existed

3.2.1.3 Error handling requirements for registration

- 3.2.1.3.1 The Gamification for learning shall show those fields which didn’t filled(show near error text on red color)
- 3.2.1.3.2 The Gamification for learning shall show those fields which filled with some errors
- 3.2.1.3.3 The Gamification for learning shall show information about user already existed if registered user is existed

3.2.2 Login

- 3.2.2.1.1 The user shall type login form to enter into system
- 3.2.2.1.2 The system shall check existence of user in database
- 3.2.2.1.3 The system shall give access to users who are existed in database
- 3.2.2.1.4 The system redirect user to user main page

3.2.2.2 Forgot password

3.2.2.2.1 User shall press link/button “Forgot password” (if user forgot password)

3.2.2.2.2 User shall fill in fields of forgot password

3.2.2.2.3 User shall press button “Send” to send password to e-mail

3.2.2.2.4 System shall close “Forgot password” page and redirect to main page for perform login

3.2.2.3 Inverse requirements for Login

3.2.2.3.1 The system shall not log to user page for doesn’t existing or users

3.2.2.3.2 The system shall not log to user page if user fill wrong username or password

3.2.2.3.3 The system shall not show forgot password page if doesn’t pressed forgot password
link/button

3.2.2.3.4 The system shall not send password message to user if her/his entered doesn’t existing
username and e-mail

3.2.2.4 Error handling requirements for Login

3.2.2.4.1 The system shall show error about wrong password or username if they are written
incorrect on login and forgot password page

3.2.3 Student play tournament

3.2.3.1.1 User(only registered students) shall be can play on tournaments

3.2.3.1.2 The system shall generate play tournament page in 1 minute with tournament question

3.2.3.1.3 The user shall give answer to questions of tournament

3.2.3.1.4 The system shall compute time limit and when finished time limit show result of user

3.2.3.1.5 The system shall receive user answer and check to show correct or incorrect

3.2.3.1.6 The system shall compute score and time of user

3.2.3.2 Inverse requirements for play

3.2.3.2.1 The system shall not give access to play for closed/already playing tournaments

3.2.3.2.2 The system shall not receive multiple answer for one answer questions

3.2.3.2.3 The system shall not show answers of questions

3.2.3.3 Error handling requirements for play

3.2.3.3.1 If user disconnected on internet system shall continue compute time limit

3.2.4 The common statistics all students.

3.2.4.1.1 The system shall collecting all action of user's(*Teacher and Students)

3.2.4.1.2 The system shall collecting this data: user score, how many enter in the tournament, how many user finish successful and fail tournament, how many user answered give answers and which is correct and incorrect.

3.2.4.1.3 The system shall be from the collection of data to analyze the activity among users and show in the table of ratings on rating page

3.2.4.1.4 The system shall give access to rating page all user(student, teacher) in the system on shorter formate. The system shall not give access unauthorized users.

3.2.4.1.5 The system shall give special access to rating page Administration in the system on full formate.

3.2.4.1.6 The system shall give special

3.2.4.1.7 The rating page shall show all data as table, colorful graphics

3.2.5 Control tournament by a teacher.

3.2.5.1.1 The system shall give access created tournament by teacher only owner and administration.

3.2.5.1.2 Teacher can control tournament. Teacher can edit the tournament title, limited time, score and tournament's question or task until starting tournament. It means, after the started of the tournament and during the tournament the teacher shall not change the tournament title, limited time, questions and task time.

3.2.5.1.3 The system shall give common statistics of tournament to teacher.

3.2.5.1.4 The teacher can give bonus score to only three students . If student successfully completed tournament.

3.2.5.1.5 If teacher has suspected cheating, spying ready answers of the students, teacher may disqualify the student from the game. But, teacher must explain the reason for the disqualification of the student and provide facts about the violation .

3.2.5.1.6 Disputes about ineligibility between students and the teacher can not require administrator intervention in the dispute. Administrator neutral person.

3.2.6 Added/Delete student in the groups.

- 3.2.6.1.1 The teacher can collecting students in the groups for tournament. Maximum number of students should not exceed 200.
- 3.2.6.1.2 The student shall make query for added student's group for join tournament
- 3.2.6.1.3 Adding a student in the group should be automatic or manual mode.
- 3.2.6.1.4 The teacher may limit the time to add to the students in the group
- 3.2.6.1.5 If the student does not have time to add to the group, then only with the permission of the teacher can go to the group.
- 3.2.6.1.6 If a student will be twice put the status of "cheating", the student automatically kicked out of the group.
- 3.2.6.1.7 The teacher can the student kicked out of the group.

3.2.7 Manage of users

- 3.2.7.1.1 System shall allow managing users to admin (delete, update)
- 3.2.7.1.2 System shall delete user on the server when admin select user and click button delete
- 3.2.7.1.3 System shall allow to admin update information data of users when select user and click button update
 - 3.2.7.1.3.1 System shall show form of update with information data of user
 - 3.2.7.1.3.2 System shall allow to admin fill form and change information of user
 - 3.2.7.1.3.3 System shall save changes information of user when admin click button save
 - 3.2.7.1.4 System shall show information about under operations result
- 3.2.7.2 **Inverse requirements**
 - 3.2.7.2.1 System shall not add, delete, update when admin not logged

3.2.7.3 Error handling requirements

3.2.7.3.1 System shall show about errors of under operations if operation operates with some errors

3.2.8 Add, edit, delete news

3.2.8.1 Add, edit, delete news powers of has only its creator.

3.2.8.1.1 If the content of the news violates the laws of the Republic of Kazakhstan. The site administrator has the right to remove the news and can report the violation to the Ministry of Internal Affairs of Kazakhstan.

3.2.8.2 Form to add news:

3.2.8.2.1 Headline news. In summary, the report on the news content

3.2.8.2.2 The text of the news.

3.2.8.2.3 The author's name

3.2.8.2.4 Attaching multimedia, documentary files.

- MP3 Audio files with a standard of no more than 15 megabytes
- Digital image standard .JPEG, .JPG, TIFF, BMP, PNG.
- Video. See 5.6
- Documentary file standard .DOC, .DOCX, .PDF, .PPTP, .PPTPX, .XLS, .XLSX, .ODT, .OPT.

3.2.8.2.5 The following files can not be affixed to the news.

- Executable file(s)
- Archived file(s)
- File larger than the maximum of volume

3.2.8.2.6 Commenting on the news.

1.1.1. Anyone can leave a review about News.

3.2.8.3 Handling error requirements.

3.2.8.3.1 If the user logs out of the window to add news and forgetting to press the button "Save".

The system will automatically inform the user about the right button click "Save" or go out of the window to add the news.

3.2.8.3.2 If the user mistakenly downloaded the file in the window to add news, the user can delete the file.

3.2.8.3.3 If the user noticed the error in the headlines or news content, it may, within 72 hours from the time change news Adding news

3.2.8.4 Inverse requirements

3.2.8.4.1 The text of the news has a length of no more than 100 characters.

3.2.8.4.2 News author must stand on the text of the news.

3.2.8.4.3 Audio file, Image file, Documentary file available for downloads from server.

3.2.8.4.4 Image will be available as a preview.

3.2.8.4.5 Video file attach from external links (Youtube, V Kontakte, Rutube)

3.2.8.4.6 All file no larger than 30 megabyts.

3.2.9 Admin show statistics of students

3.2.9.1 When user click menu link statistics, system shall show a list of the last tournaments with date, clock, and number of students.

3.2.9.2 When user can select tournament on the list, and click, system shall show list statistics with score by student and honors.

3.2.9.3 System shall allow the user to go back. (ex. Back button).

3.2.9.4 Handling error requirements

none

3.2.9.5 Inverse requirements

3.2.9.5.1 System shall not show tournaments which is not played

3.2.9.5.2 System shall not allow the user change students information on this page

3.2.10 Students show yourself statistics and honors

3.2.10.1.1 System shall show statistics of student, when student click menu link statistics on student page.

3.2.10.1.2 After system shall show statistics diagram of student(tournament date, score).

3.2.10.1.3 System shall show a list of the tournaments with student score, date play, etc.

3.2.10.1.4 System shall show honors of student like slideshow on the top.

3.2.10.1.5 System shall show list of student ratings of the tournament when user click on list tournaments on statistics page

3.2.10.2 Handling error requirements

none

3.2.10.3 Inverse requirements

3.2.10.3.1 System shall not show tournaments which user don't play tournament.

3.2.10.3.2 System shall not allow the user to change data. (Ex, score, honors)

3.3 PERFORMANCE REQUIREMENTS

3.3.1.1 Scalable to at least thousands or two thousands of users. Must support up to thousands users at one time

3.3.1.2 Web application must supported on mobile devices (Android, IOS, etc..)

3.3.1.3 All of pages must be generated within 10 seconds and shown content to user in two minute

3.3.1.4 Database should be normalized on level above 3NF

3.4 DDESIGN CONSTRAINTS

None. Can use all design software of web application.

3.5 SOFTWARE SYSTEM ATTRIBUTES

3.5.1 Reliability

Application is thin-client so main load will be on server side. Back-end side will be developed on PHP.

3.5.2 Availability

Gamification will be installed on hosting web-server. So it will be available through local network(intranet) and also through internet. It means that server will be setted to working on extranet.

3.5.3 Security

The software will be added two-factor protection from hackers. Each action in the system will be twins checking.

3.5.4 Maintainability

All of parts of application will be developed by modules. So all parts of application are extremely maintainable by adding/updating or even deleting some parts of application.

3.5.5 Portability

This software is web based; thus, it is extremely portable. Software will be show correctly on PC, Mobile devices

3.6 OTHER REQUIREMENTS

On the system must be three theme lessons such English language, Programming language and Mathematics.

The system should automatically assume scores of students and show them the results when the tournament is finished

The system should sort students by scores when the tournament is finished or students want see rating or teacher want see rating of her/his tournament or admin want see rating of students

The system games base should be for freshmen of Engineering faculty