Requirement analysis report

Reykjavik University
Computer science

Semester project II

Arnar Björn Pálsson, Atli Egilsson, Daníel Örn Stefánsson, Eyrún Magnúsdóttir og Viðar Einarsson

Table of contents

Introduction / About the System

This system is a web-page for teachers and students which allows them to submit and review their programming assignments. The teachers define the assignments and provide a few examples of input and output. Students can then submit their solutions and will know instantly if their solution is correct or if there were any errors. Teachers can rely on this system to grade their student's performance on another platform. We put high focus on the simplicity of the system in such a way that the users don't get caught up in adjusting to the system rather than conducting to their courses.

To analyze these points we researched the competition for these online systems, interviewed students and teachers that are familiar with these methods and determined the requirements and actions to support such a system.

Evaluation of web sites

It's important that the system is easy to use and has a good UX. By evaluating other similar web sites we could possibly get a grip on the system's layout and it's functionality.

We decided to look at three web sites, Mooshak, Kattis and HackerRank. All of those web sites have a very similar functionality and structure. By evaluating these web sites we gathered their pros and cons. We can then use this information to develop and improve our system.

Mooshak

Pros

- Good compiler.
- Informative feedback.
- Good error handling.
- Simple to use.

Cons

- Outdated user interface.
- Does not allow students to group up.
- Hard to read from "wrong output" when the output is long.

Kattis

Pros

- Very user friendly.
- User can choose between uploading a file or switch to an built in editor (Ace editor).
- Tons of practice problems for different users with different programming background.

Cons

- Problem description, input and output examples are not displayed on the page where you input your code.
- There are no default templates, for example you have to write “#include <iostream>”, “using namespace std” and the main function for every single problem.

HackerRank

Pros

- Very user friendly.
- The description of the problem is on the same page as the editor
- You get templates to the problems.
- You can upload a file and the text from the file goes to the editor and then you can edit it on the web page.
- Easy to see results and grade for each problem.
- Discussion board for each problem for students.

Cons

- No noticeable cons.

User Groups

We decided to analyze our users and split them up into 4 different user groups, to be able to understand their usage of the system better, and therefore know what each user group would require of the system.

We decided to add one more user group to our evaluation so that the TA's can also view students submitted code and help them accordingly, and they need to have a special access to the system to do that.

User group	Background	System usage	Environment	Main goals
Administrators	**Age:** 22+ **Education:** Degree in Computer Science or similar fields. **Computer skills:** Above average	**Usage:** According to course schedule **Training:** Required **% of userbase:** Less than 5%	**Technical environment:** Desktop computer / laptop **Real environment:** At work / At home	• Create users • Create courses • Edit users • Edit courses • Link users to courses
Teachers	**Age:** 20+ **Education:** Teacher **Computer skills:** Above average	**Usage:** According to course schedule **Training:** Shouldn't need a lot. **% of userbase:** Less than 10%	**Technical environment:** Desktop computer / laptop **Real environment:** At work / At home	• Create assignments • Grade students solutions • Answer questions • Submit possible solutions • View students submissions
Teacher Assistants	**Age:** 18+ **Education:** College **Computer skills:** Above average	**Usage:** According to course schedule **Training:** Not required **% of userbase:** Less than 15%	**Technical environment:** Desktop computer / laptop / tablet / mobile phone **Real environment:** At work	• Answer questions • Submit possible solutions • Grade students solutions • View students submissions
Students	**Age:** 18+ **Education:** College **Computer skills:** Above average	**Usage:** According to course schedule **Training:** Not required **% of userbase:** More than 70%	**Technical environment:** Desktop computer / laptop / tablet / mobile phone **Real environment:** At school / home	• Submit solutions • Ask questions • View grades

Requirement list

This requirement list resembles the main features to be available in the upcoming system.

The requirements are sorted by priorities.

Priorities	
A	Absolutely essential
B	Useful, but not mission critical
C	Nice to have

Functional requirements

Number	Description	Use Case Number(s)	Priority
1	Users can log in to the system.	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	A
2	Admins can create users.	1	A
3	Admins can create courses.	1, 5	A
4	Admins can link users to existing courses and define their role in the course.	1, 5	A
5	Admins can edit users.	1, 5	A
6	Admins can delete users.	1, 5	A
7	Admins can edit courses.	1, 5	A
8	Admins can delete courses.	1, 5	A
9	Teachers can create assignments within a course.	1, 2, 3, 4, 6, 7, 8	A
10	Teachers can view student’s submitted solution.	2	A
11	Teachers can split up assignments into multiple parts.	4	A
12	Teachers can grade student’s assignments.	2	A

Number	Description	Use Case Number(s)	Priority
13	Students can submit solutions to certain assignments.	2, 3, 4, 8	A
14	Students can view their assignments.	3	B
15	Students can view the expected output for their assignments.	3	B
16	Students can view their grades.	2	B
17	Students can view their submissions.	2, 9	B
18	Users can post questions about assignments.	6, 7	B
19	Teachers can limit the submission rate on a single assignment.	4	B
20	Teacher/TA can post hints about the assignment.	7	B
21	Students can view other student's submissions, anonymously.	9	C
22	Teachers can post solutions to assignments after deadline.	10	C
23	Student can group up for assignments.	4, 8	C
24	Students can create their programs inside the web application.	8, 11	C
25	Students can debug their programs inside the web application.	11	C
26	Student groups can work together simultaneously on their program.	8	C
27	Users get notifications on new material regarding their courses.	7, 10	C
28	Users can change UI language (English / Icelandic)		C
29	Users can change their information (e-mail, password..)		C

Non-functional requirements

Number	Description	Priority
1	The system supports all main web browsers (Chrome, Safari, Firefox, Opera, Internet Explorer and Edge).	A
2	The system has to adjust to the screen resolution.	A
3	The system is easy to use and does not require system training.	A
4	The system is safe to use.	A
5	The system must be usable on multiple devices, laptops, desktops, smartphones and tablets.	A
6	The system supports more than one programming language	B

Interviews

About the interviewees

Interviews were conducted with six people. Four of them had already used mooshak before, the latter two had never used a system like that before.

Their age was in the range 21-28 years old, five of them were male and one female. Two of them are currently students and use mooshak to turn in their assignments. The other two are former students who have already graduated and have used mooshak before in their studies.

Interpretation of interviews

The data of the interviews were evaluated and we will interpret the results here. The questions we gave the interviewees will be in the appendix of this document.

All of the interviewees had turned in their assignments online, at least at some time during their studies. Four of them have used mooshak before and all of them had very similar opinions on the system.

They all thought that it would be nice to get more feedback from the system, either with hints from the teacher/TA's or by being pointed to where the error is, in a simpler way then it is now.

Three of the interviewees thought that it would be nice to be able to write the code directly in an editor on the page, but they didn't think it was necessary though. One student thought that it would be pretty pointless, considering that all programming students usually have some kind of programming environment. They all thought it didn't really matter if they received their grades on mooshak because they usually have to use myschool anyways for other courses.

All of the interviewees thought that the design of mooshak needs to be heavily updated and the layout and navigation of the page could be much better.

Use cases

Name	**Admin creates a user and links it to a course**
Number	**1**
Priority	**High**
Pre-condition	**There is an existing course.**
Description	**Admin logs in to the web-page.** **Admin creates the user, assigns its role and enrols him to the desired course.**
Alternative Flow	**Admin links a user to the wrong course. He removes the user from the course list and assigns it to the correct one.** **Admin accidently gives a student the same rights as a teacher. He revokes his authorization rights and gives student his rights again.**
Post-condition	**The course is now active.** **Teachers can start providing assignments.** **Students can log on to the system and start solving assignments.**
Requirements	**1, 2, 3, 4, 5, 6, 7, 8.**
Actor(s)	**Admin, Teachers, TA's, Students.**

Name	**Teacher grades assignments**
Number	**2**
Priority	**High**
Pre-condition	**The teacher has been assigned to an active course. Students have submitted their solutions to the project.**
Description	**The teacher logs in to the web-page, selects which project to grade and starts reviewing submitted solutions. He can now see if each students submission was successful and also an overview of the students code.**
Alternative Flow	**Teacher makes a mistake when grading a student and needs to correct his grade, he can edit the student's grade and update it. A student hands an assignment in too late. TA's can upload his assignment onto the page, if late submissions are allowed, and grade it so the student can also see his grade on this platform.**
Post-condition	**Students have received their grades and can review their code.**
Requirements	**1, 9, 10, 12, 13.**
Actor(s)	**Teacher, TA's, Students.**

Name	**Student submits a solution to an assignment.**
Number	**3**
Priority	**High**
Pre-condition	**Student has been assigned to an active course and can submit his solution to the course's assignment.**
Description	**Student logs in to the web-page, selects assignment and submits his solution. Students can see if their solution was accepted by the system or not.**
Alternative Flow	**Student submits a wrong solution. He can then re-submit his solution. If the solution is not correct the system will store the most correct solution.**
Post-condition	**Student has submitted his solution and can see if it was accepted.**
Requirements	**1, 9, 13, 14, 15.**
Actor(s)	**Student**

Name	**Teacher creates an assignment and splits it up into multiple parts**
Number	**4**
Priority	**High**
Pre-condition	**Teacher has been linked to a course**
Description	**Teacher logs into the system and selects the desired course. There he can choose to add a new assignment. He uploads the assignment and gives each part of the assignment a value. There he can also configure the submission rate for this assignment.**
Alternative Flow	**Teacher uploads the assignment to the wrong course, he can delete it from the wrong course and post it to the correct one. Teacher uploads the wrong file for the assignment, he can edit the assignment and replace the wrong file with the correct one.**
Post-condition	**Students can now view and submit solutions to the new assignments.**
Requirements	**1, 9, 11, 13, 19, 23.**
Actor(s)	**Teacher, Students.**

Name	**Admin creates a course**
Number	**5**
Priority	**High**
Pre-condition	**None**
Description	**Admin logs in to the web-page. Admin creates the course on the web-page and assigns teachers and TA's to it accordingly.**
Alternative Flow	**Admin assigns the wrong teacher/TA's to the course, he removes them from the course and assigns the correct teacher/ TA's to the course.**
Post-condition	**Now the course is accessible to the teacher/TA's and they can add assignments to the course.**
Requirements	**1, 3, 4, 5, 6, 7, 8.**
Actor(s)	**Admin, Teachers, TA's, Students.**

Name	**Students can post questions about assignments.**
Number	**6**
Priority	**Medium**
Pre-condition	**Student is linked to a course.**
Description	**Student logs in to the web-page and chooses his course. He can now post questions regarding his current assignment.**
Alternative Flow	**Student chooses the wrong course and posts his question there. He deletes his question from that course and posts it in the right course.**
Post-condition	**Student has posted a question about the assignment and everybody linked to the course can now answer it.**
Requirements	**1, 9, 18.**
Actor(s)	**Teacher, TA's, Students.**

Name	**Teacher posts a hint for an assignment. (Also available for TA's)**
Number	**7**
Priority	**Medium**
Pre-condition	**Teacher is linked to a course, an active assignment is ongoing.**
Description	**Teacher logs in to the web-page, he opens up the desired assignment and chooses to post a hint for the assignment. He writes the hint and submits it.**
Alternative Flow	**Teacher posts the hint on a wrong assignment, he deletes the hint and recreates it on the correct one.**
Post-condition	**The hint is now visible on the assignment page, users can now view the hints and will get a notification about the new material.**
Requirements	**1, 9, 18, 20.**
Actor(s)	**Teacher, TA's.**

Name	**Three students group up for an assignment**
Number	**8**
Priority	**Low**
Pre-condition	**Students are assigned to a course and the course has an active assignment that allows groups of people to work together.**
Description	**One student makes a group and invites others, then anyone in the group can submit a solution.**
Alternative Flow	**A student invites a wrong student to the group. He removes the student and reinvites the correct one.**
Post-condition	**The students have now formed a group, and can all contribute to the assignment, and submit solutions. They can now all work simultaneously on the assignment.**
Requirements	**1, 9, 13, 23, 24, 26.**
Actor(s)	**Students**

Name	**Student views the list of submissions for a single assignment**
Number	**9**
Priority	**Medium**
Pre-condition	**This assignment is either currently active or submissions can no longer be received.**
Description	**The student logs in to the web-page. The student selects the assignment and proceeds to view other people's submissions.**
Alternative Flow	**-**
Post-condition	**-**
Requirements	**1, 17, 21.**
Actor(s)	**Students**

Name	**Teacher posts the correct solution to an assignment after the deadline**
Number	**10**
Priority	**Low**
Pre-condition	**Assignment has passed the deadline**
Description	**Teacher logs into the web-page, selects an assignment and submits the correct solution, all the students in the course get a notification about the solution.**
Alternative Flow	**Teacher submits the solution of the wrong assignment. He then removes the solution from the assignment and submits the correct one.**
Post-condition	**The student's in the course can now see the solution for the selected assignment.**
Requirements	**1, 22, 27.**
Actor(s)	**Teacher, TA's, Students**

Name	**User submits a solution from code written on the web-page**
Number	**11**
Priority	**Low**
Pre-condition	**There is an open assignment.**
Description	**User starts the assignment and opens it in the editor on the page, selects one of the allowed programing languages and gets the template from the teacher, writes the required code and submits it.**
Alternative Flow	**User submits the code but gets a compile error. The user opens the editor again and finds lines that are marked red, fixes the issue and submits again.**
Post-condition	**The user has submitted a working solution**
Requirements	**1, 24, 25.**
Actor(s)	**Teacher, TA's, Student**

Rules

The general rules that apply for the system will be described below. Note that this is not a complete list and that it will probably change and expand during the development process.

Each student can be enrolled in as many courses as they desire.
Assignment grades are on the scale of 0-10.
User can not be a teacher and a student in a single course.
Users have to be logged in to be able to view any course pages.
The most qualified submission to an assignment is always used for grading.

Glossary

Word / Phrase	Meaning
TA	Teachers Assistant
User	Teachers, TA's and Students
UX	User experience

Appendix

Interview questions

1. Eru rafræn skil á verkefnum í þínu námi?
2. Hvernig finnst þér fyrirkomulagið á verkefnaskilum hjá þínum skóla vera?
3. Hvað finns þér gott við þetta fyrirkomulag?
4. Hvað finnst þér slæmt við þetta fyrirkomulag?

Extra questions for interviewees experienced with Mooshak

1. Hvernig finnst þér núverandi fyrirkomulag á verkefnaskilum vera á mooshak?
1. Myndir þú vilja geta séð lausnir eða vísbendingar frá kennaranum inn á mooshak?
2. Myndir þú vilja að það væri hægt að skrifa kóða beint inn á mooshak?
3. Myndir þú vilja fá einkunnir fyrir verkefnin þín beint inn á mooshak ?
4. Er eitthvað sem þú myndir vilja hafa inn á mooshak sem er ekki til staðar núna?
5. Er eitthvað sem þú vilt bæta við?