LEARN4FUN

Gamification theory used in the learning process

Technical University of Cluj-Napoca
User Interface Design

Table of Contents

1. Project Specification 3
2. Audience 3
3. Task analysis 5

Task #1: Creating quizzes; 5
Task #2: Solving quizzes; 5
Task #3: Posting fun facts; 6
Task #4: Fun fact of the day; 6
Task #5: Reviewing fun facts; 7
Task #6: Posting tutorials; 7
Task #7: Viewing and reviewing tutorials; 8
Task #8: Accessing the global leaderboard/ranking; 8
Task #9: Access to personal profile and viewing statistics; 9
Task #10: Contact-the-developer; 9
Task #11: Posting a question on the QA page; 10
Task #12: Answering a question from the QA page; 10

4. User scenarios 11

Scenario #1: Creating quizzes: 11
Scenario #2: Solving quizzes; 11
Scenario #3: Posting the fun fact; 12
Scenario #4: Fun fact of the day; 12
Scenario #5: Reviewing fun facts; 13
Scenario #6: Posting tutorials; 13
Scenario #7: Viewing and reviewing tutorials; 14
Scenario #8: Accessing the global leaderboard/ranking; 14
Scenario #9: Access to personal profile and viewing statistics; 14
Scenario #10: Contact-the-developer; 15
Scenario #11: Posting a question on the QA page; 15
Scenario #12: Answering a question from the QA page; 15
Alternative Scenario #1: 16
Alternative Scenario #2 16

5. Walkthrough Evaluation Report 17

Scenario #2: Solving quizzes; 17
Scenario #3: Posting the fun fact; 18
Scenario #4: Fun fact of the day; 22
Scenario #5: Reviewing fun facts 23

Scenario #8: Accessing the global leaderboard/ranking; 25
Scenario #9: Access to personal profile and viewing statistics 26
Scenario #10: Contact-the-developer; 27
Scenario #12: Answering a question from the QA page; 29
6. Heuristic Evaluation 30
Group 1: Create Quiz, Solve Quiz, Access to personal profile; 30
1. Visibility of system status 30
2. Match between system and the real world 31
3. User control and freedom 31
4. Consistency and standards 32
5. Error prevention 32
6. Recognition rather than recall 33
7. Flexibility and efficiency of use 33
8. Aesthetics and minimalist design 33
9. Help users recognize, diagnose, and recover from errors 34
10. Help and documentation 34
Group 2: Posting a fun fact, Fun fact of the day, Reviewing fun facts 35
1. Visibility of system status 35
2. Match between system and the real world 35
3. User control and freedom 36
4. Consistency and standards 36
5. Error prevention 37
6. Recognition rather than recall 37
7. Flexibility and efficiency of use 38
8. Aesthetics and minimalist design 38
9. Help users recognize, diagnose, and recover from errors 39
10. Help and documentation 39
Group 3: Posting a tutorial, Viewing and reviewing tutorials, Leaderboard page 40
1. Visibility of system status 40
2. Match between system and the real world 40
3. User control and freedom 41
4. Consistency and standards 41
5. Error prevention 42
6. Recognition rather than recall 42
7. Flexibility and efficiency of use 43
8. Aesthetics and minimalist design 43

9. Help users recognize, diagnose, and recover from errors 44
10. Help and documentation 44
Group 4: Posting a question, Answering a question, Contact the developer 45
1. Visibility of system status 45
2. Match between system and the real world 45
3. User control and freedom 46
4. Consistency and standards 46
5. Error prevention 47
6. Recognition rather than recall 47
7. Flexibility and efficiency of use 48
8. Aesthetics and minimalist design 48
9. Help users recognize, diagnose, and recover from errors 49
10. Help and documentation 49
Heuristics evaluation conclusion 50

1. Project Specification

This project is aimed at stimulating the participation of students to the educational process and at proposing solutions for an increased awareness on their part. The aims of the project are approached by applying principles from the gamification theory in the recurring teaching activities and by proposing simple and efficient management instruments to the teaching staff and the students as well.

Regardless of the game type, be it a shooter, sports, roleplaying, etc. some features always stand out: a challenging competitive environment and a compelling reward system. The online platform that we envision plans to use these features in order to stimulate various audiences, through the appeal of games, to use their energy into a more productive way, that is, a learning environment.

2. Audience

- Q: Who is going to use the application?
  A: Both students (at university level) and professors will be using the application for educational purposes.
- Q: What technologies do users need to be familiar with?
  A: The regular Personal Computer (PC) user will be able to use the platform with ease, no additional requirements needed.
- Q: What would users be able to do with the application and why would they use it?

A: Users of the application would be able to reinforce the things that they learn during classes as well as discover many other interesting facts that they did not know beforehand, everything in a more pleasant and interactive environment.

- Q: Where and how can users use the application?
  A: Wherever a PC/laptop with internet connection is available.
- Q: I do not like this product! Do I have any alternatives?
  A: No other platform would provide the exact services as this one, but for viable alternatives, Coursera and Moodle can be used.
- Q: Sounds complicated! Is any other tool needed in order to use the application?
  A: No! The regular (updated) Google Chrome, Mozilla Firefox or Microsoft Edge is enough.

3. Task analysis

Task #1: Creating quizzes;
Type of users: Mainly teachers, limited student ability;
Activity: Creating quizzes on various topics, creating combined quizzes, creating quizzes on tutorials;
Goal: The main purpose is to create a competitive environment from which students can learn as much as possible.
Description: Quizzes can be created by both teachers and students, with some specifications. Competitive quizzes (will contribute to the leaderboard, can be used in tournaments, etc.) can only be created by teachers. If a student wishes to create such a quiz, he will need the approval of a teacher (the teacher will still have to post it). On the other hand, students will be able to create quizzes on tutorials that they post, but these will only contribute to achievements (e.g. Solve 1/5/10 tutorial quizzes) or lesser rewards.
Steps:

1) Visit the "Create quiz" page;
2) Give it a title a brief description and add related tags;
3) Select Add Questions;
4) Select the type of question you wish to add: Multiple/single choice, fill-in the blanks or free-answer questions;
5) The previous step can be repeated as many times as needed;
6) Review the quiz and submit it.

Task #2: Solving quizzes;
Type of user: Students;
Activity: Solving quizzes that can have multiple questions and various rewards;
Goal: The main goal is to learn from the quizzes and to collect points and achievements;
Description: The website will contain quizzes with interesting and tricky questions for the students to solve. After each answer, the student receives a feedback. At the end of the quiz, a score is computed and added to the user's profile. Also, users receive achievements in case of completing different milestones in these quizzes (for example top score in Java quiz, 10/50/100 completed quizzes).
Steps:

1) Select a quiz from the dedicated quizzes page;
2) Answer the questions from the quiz;
3) Submit the quiz and receive your score and rewards.

Task #3: Posting fun facts;
Type of users: Both students and teachers;
Activity: Creating and posting fun facts;
Goal: The main goal is to share interesting facts, on the website, with other users;
Description: Both teachers and students can post fun facts. A fun fact can contain text, images, videos and links. On the website, a page is dedicated for these types of posts, where both students and teachers can share, view and react to fun facts with comments and other reinforcement tools (e.g. smiley faces, likes, thumbs-up, etc.).
Steps:

1) Click on the "Fun fact page";
2) Click the "Post a fun fact button";
3) Insert title in a text field input;
4) Insert a short description in a similar field;
5) Insert the content in another text area;
6) Add external links and tags;
7) Add images/videos by using either the dropdown feature or the "Browse" button;
8) User is able to either go back ("Back" button), preview his created fun fact ("Preview" button) or post the fun fact ("Post" button).

Task #4: Fun fact of the day;
Type of users: Students, teachers, guests;
Activity: The ability to view the fun fact of the day;
Goal: To randomly select a fun fact and display it on the main page;
Description: From the pool of created (and approved) fun facts, one will be chosen randomly, each day, and it will be displayed on the main page. The section will contain a video, image, some text or a combination of these, the ability to click it and be redirected to the fun fact archive where users will be able to access extra information on the fun fact.
Steps:

1) Visit the main page;
2) Wait for the fun-fact of the day to appear;
3) Read the fun fact and click on it for more information.

Task #5: Reviewing fun facts;
Type of users: Students, teachers;
Activity: Creating a review on fun facts;
Goal: The main purpose of this activity is to offer reinforcement (whether positive or negative) and feedback on fun facts, that other users can see;
Description: All users can review fun facts. A review can consist of either or multiple of the following: reinforcement though a like/dislike, a comment or the ability to share it on different other media. Getting positive or negative reviews will influence the user's standings on the leaderboards and will be reflected on his account.
Steps:

1) Select a fun fact either from their specific page or from the "Fun fact of the day" section;
2) Choose to leave a like/dislike or a comment.

Task #6: Posting tutorials;
Type of users: Students, teachers;
Activity: Creating and posting various tutorials;
Goal: The main purpose of this activity is to share various how-to-s in order to have a centralized information hub;
Description: Both teachers and students can post tutorials. These tutorials can either be in written format or video format. These tutorials don't have to be created by the person who posts them, but they can be. Tutorials will be posted in a special, separate section of the website. The tutorials may also come in an e-book format.
Steps:

1) Visit the "Add a tutorial";
2) Add a title and a brief description;
3) Add the content of the tutorial which can be a combination of: text, images, videos, hyperlinks or an e-book;
4) Submit the tutorial.

Task #7: Viewing and reviewing tutorials;
Type of users: Students, teachers (view and review) + guests (view only);
Activity: Viewing, reviewing and answering to available reinforcement quizzes;
Goal: This activity entails the ability of viewing tutorials, participating in various reinforcement-learning activities and offering feedback on them.
Description: Any user is able to view the available tutorials. This is mainly targeted at the students, but teachers can benefit from this ability as well. A review can consist of either or multiple of the following: reinforcement though a like/dislike, a comment or the ability to share it on different other media. Quizzes may also be related to tutorials, thus helping the users assimilate the information in an easier way.
Steps:

1) Visit the tutorials page;
2) Select a tutorial;
3) Enjoy what the tutorial has to offer;
4) Leave feedback in the form of either a comment or clicking a like/dislike button;
5) Optional step: solve a reinforcement quiz related to the tutorial.

Task #8: Accessing the global leaderboard/ranking;
Type of users: Students, teachers, guests;
Activity: Viewing the total score for any of the users as well as the rewards accumulated over time;
Goal: This activity allows users to see both their overall ranking and that of other users, ranked by total score and various rewards;
Description: Anyone is able to access the ranking. This is an important part of the application because it stimulates the users to complete as many quizzes as possible and gather badges in order to be at the top of the leaderboard. It also offers an overview of all the students and professors, thus one can find the most suitable person for when he requires help.
Steps:

1) Visit the leaderboard page;
2) Scroll through the page;

Task #9: Access to personal profile and viewing statistics;
Type of users: Students, teachers, guests;
Activity: Viewing, in a more detailed manner, a user's rankings, strengths and other user-related data;
Goal: This activity allows users to see the overall ranking of the targeted profile as well as additional information, such as: personal preferences, interests, all the completed achievements and level of expertise in different technologies/fields;
Description: All profiles are public, thus anyone can see them. A user can only fill in personal data about himself. All the other statistics (scores for different columns etc.) are automatically computed based on the user's activity and results on the platform. This functionality can be very helpful for choosing partners for team projects, finding the right professor for a license thesis or other personal projects.
Steps:

1) Access either the leaderboard page or the personal profile;
2) See all the statistics related to the user (global score, number of positive/negative reviews on various website features).

Task #10: Contact-the-developer;
Type of users: Students, teachers;
Activity: Submitting feedback and/or requests to the development team;
Goal: Keeping the development team in contact with the users, in order to fix bugs and improve the overall experience;
Description: Both students and teachers can submit feedback about their experience with the platform, report bugs or make suggestions about features they would like to be added. Once submitted, the information is stored in a database and is only accessible to the development team.
Steps:

1) Go to the "Provide feedback" page;
2) Complete the form and submit it;

Task #11: Posting a question on the QA page;
Type of users: Students, teachers;
Activity: When one has a question related to computer science and needs help with finding the answer, he or she may ask the community;
Goal: Helping students with their projects and trying to overcome any hurdles that may arise in the learning process;
Description: A student in need of help can post a question and add several tags so that others can find it easier. If someone encounters a problem with a project, chances are somebody else will have or has already solved it. This way it would be easier to find a right or better answer for the same question, even better: a TA or professor would be able to provide a correct solution.
Steps:

1) Visit the QA page;
2) Provide a title and a description to the problem;
3) Select the appropriate question-related tags;
4) Submit the question.

Task #12: Answering a question from the QA page;
Type of users: Students, teachers;
Activity: Help others solve errors/bugs or even find a right solution for a problem;
Goal: Increase the cooperation between users and at the same time share valuable knowledge;
Description: Any logged user can answer a question on the QA page. However, the comment may be deemed valuable or not by the rest of the community by using “helpful” or “not helpful” buttons. This contributes to a user’s total score and depending on the tags, it increases or decreases the score for particular skills.
Steps:

1) Visit the QA page;
2) Select a question that is to be answered;
3) Provide an answer/solution;
4) Submit the answer/solution;
5) Answers can be rated as either being helpful or not helpful.

4. User scenarios

Scenario #1: Creating quizzes:
Steps:

1) Visit the “Create quiz” page;
2) Give it the “Java Master” name;
3) Give it a brief description: This is a tutorial for those who wish to test their knowledge about the more advanced parts of Java;
4) Add related tags: OOP, Java, master, design-patterns;
5) Select question type: free-answer question;
6) Type your question: “Describe the Observer design pattern”;
7) Select question type: multiple-choice;
8) Type your question: “Java is a(an)”;
9) Type the options (“Scripting Language”, “OOP Language”, “Procedural Language”) and select the correct answer (“OOP Language”);
10) Steps 5-6 can be repeated as many times as needed by pressing the “Add another question” button;
11) Review the quiz and submit it by pressing the “Done” button;
12) In case the quiz has no name or doesn’t have at least 1 question the user won’t be able to create the quiz and he will be prompted to fill the required fields.

Motivation: The main motivation for the chosen user interface design is that it is easy to understand, reflected through the fact that the user can choose from a certain amount of options and is guided through various steps, until the process of creating a quiz is done. Also, this design resembles the one used on Moodle websites.

Scenario #2: Solving quizzes;
Steps:

1) From any page, click on the Quizzes tab;
2) Select the Java Master quiz;
3) Answer the questions from the quiz by fulfilling their various requirements;
4) Click on the Submit button and you will be redirected to the Results page for that quiz;
5) After viewing your statistics, you can Check the Global Rankings, view your Profile or go back to solve another quiz;
6) In case the user quits the quiz before submitting it, the progress will not be saved, and in case of a tournament it will result in disqualification.

Motivation: The main motivation for the chosen user interface design is that the user is guided throughout the process of solving a quiz through steps with intuitive explications. Furthermore, this design is applied also on Moodle websites.

Scenario #3: Posting the fun fact;

Steps:

1) From any page, visit the "Fun-facts" page;
2) From the "Fun-facts" page you can click on any fun-fact, then click on the "Post a fun-fact" button OR click the "Post a fun-fact" button directly from the fun-fact page;
3) Add a title: "Did you know? – Firefox";
4) Add a brief description: "This fun fact explains the origin of the browser's name and logo";
5) Add the content of the fun fact: "The English word for red panda is "Firefox" which is where the browser gets its name from – this means the Firefox logo is actually a red panda, not a fox!";
6) Add related images: "Firefox logo in comparison with a red panda";
7) Add links: https://www.thefactsite.com/2013/02/top-100-technology-facts.html;
8) Add related tags: "browser", "Firefox", "did-you-know";
9) Submit the fun fact by pressing the "Post" button;
10) In case the user doesn't enter a title or content, he will not be able to submit the fun fact and will be prompted to fill those fields.

Motivation: The motivation for the chosen interface design is that the user is guided throughout the process through intuitive steps. Furthermore, it is also used on Facebook when posting activities, making it easy to be understood by teachers or students.

Scenario #4: Fun fact of the day;

Steps:

1) Visit the home page by either typing the URL in the browser or clicking the "Home" button from any page on the website;
2) Click on the bullets to find the "Fun fact of the day" (e.g. "Did you know? – Firefox") or wait for it to auto-scroll into view;
3) View what the fun fact has to offer (Firefox logo – red panda comparison image, shortened content) and/or click on it for more details;
4) Clicking on the fun-fact will redirect the user to that specific fun-fact's page.

Motivation: The main motivation for the chosen design is that it is similar to viewing high rated posts on Facebook, which makes it more familiar for students, teachers or guests.

Scenario #5: Reviewing fun facts;

Steps:

1) Click on the “Fun fact page”;
2) Click on the “Did you know? – Firefox” fun fact;
3) Click on either the “Like”, “Dislike” or “Share” button;
4) Fun facts can also be “Liked”, “Disliked” or “Shared” from the main fun-fact page or the “Fun fact of the day” section;
5) Insert a comment in the “Comment” field: “I really find this fun fact interesting, good job for sharing it!”;
6) Either press “Enter” to submit a comment or click the “Submit” button;
7) In case the comment is blank, the user won’t be able to submit it.

Motivation: The main motivation for the chosen design is that it is similar with the one of reviewing posts on Facebook.

Scenario #6: Posting tutorials;

Steps:

1) Click the "Tutorials" button that will redirect you to the Tutorials page;
2) Write a title for the tutorial you are going to submit: "Angular 7";
3) Write a short description related to the tutorial you are going to add: " Angular 7 is here and it's not spooky at all! This post highlights the latest features and improvements in Angular 7";
4) Write a long description related to the topic of the tutorial you want to add: "Angular is a TypeScript-based open-source front-end web application platform led by the Angular Team at Google and by a community of individuals and corporations. Angular is a complete rewrite from the same team that built AngularJS.";
5) If you know a link related to this tutorial you can add it in the Video section: "https://www.youtube.com/watch?v=5wtnKulcquA";
6) Choose a tag that fits best to your question: "angular";
7) Submit the tutorial by pressing the "DONE" button.

Motivation: The motivation for the chosen design is the fact that it resembles the design of creating a post on Facebook, making it more familiar to the users. Furthermore, the user is guided through intuitive steps throughout the process.

Scenario #7: Viewing and reviewing tutorials;
Steps:

1) Click the "Tutorials" button that will redirect you to the Tutorials page;
2) See a list of tutorials you can watch or read;
3) Select a tutorial for example "Angular 7";
4) Enjoy what the tutorial has to offer and learn the most from it;
5) Leave feedback in the form: "What an amazing tutorial. It helped me a lot with my work";
6) Click the like or dislike button depending on your opinion about how useful was the tutorial you just watched;
7) Moreover, you can solve a reinforcement quiz related to the tutorial. E.g.: questions related to differences between different versions on angular or how it is different from other JS frameworks.

Motivation: The motivation for the chosen design is the fact that it resembles the design of reviewing posts on Facebook, making it more familiar to the users.

Scenario #8: Accessing the global leaderboard/ranking;
Steps:

1) From either the “Home” page or any other website page, click on the “Rankings” button on the navigation bar;
2) On the top of the page you will see your standings (e.g. User: Mihai, Total Score: 125, Rewards: “Completed 10 quizzes”, “Posted 7 Fun-Facts”, “Achieved “Master” in Java”, Position: 17);
3) On the rest of the page you will see a list containing the leaderboard (e.g. Position: 1: User: Andrei, Score: 250, Position 2: User Andreea, Score 225, etc.);
4) Click on any user to view his statistics and achievements (badges).

Motivation: The motivation for the chosen design is the fact that this kind of design is seen throughout multiple websites in the case of showing a leaderboard, being simple through representing the rank of each user through a list.

Scenario #9: Access to personal profile and viewing statistics;
Steps:

1) Click on either the “Rankings” button, or the “My Account” button;
2) If on the “Rankings” page, click on the “View My Profile” button;
3) See your name: “John Doe”, university: “College Cork”, your role: “computer science student”, age (optional): 21, score: 15872 and standing: 8 on the top of the page;
4) View a graph on the acquired skills: Technologies, Concepts, Mathematics, Algorithms, Hardware, Software, etc.;
5) View personal badges: “Math-Master”, “Hardware Hero”, “VHDL Geek”, etc.;

Motivation: The motivation for this design is the fact that it can be seen throughout multiple websites (e.g. GitHub), making it more familiar for students, teachers or guests.

Scenario #10: Contact-the-developer;
Steps:

1) Click on the “Provide feedback” button in order to visit the respective page;
2) Read the “How important is your opinion to us” text;
3) Write the suggestion: “I believe you should add more badges and achievements for users so that we would feel more engaged in the activities on the website”;
4) Press the “Submit” button to provide your feedback;
5) In case the suggestion field is empty, the user will be prompted to fill it, but the submit will not take place; The user is required to enter a minimum of 75 and a maximum number of 500 characters for the feedback to be considered valid.

Motivation: The design is chosen because it is the most common thing seen on websites, being a simple form, easy to understand, through which one can take contact with the developers or owners.

Scenario #11: Posting a question on the QA page;
Steps:

1) Click the "QA" button that will redirect you to the QA page;
2) Write a title: "JS Frameworks";
3) Write a short description related to the title you entered above: "Which JS frameworks are you aware of?";
4) Write a long description related to the topic of the quiz you want to post: "A JavaScript framework is an application framework written in JavaScript. It differs from a JavaScript library in its control flow: a library offers functions to be called by its parent code, whereas a framework defines the entire application design".
5) Choose a tag that fits best to your question: "JavaScript".
6) Submit the question by pressing the "ADD QUESTION" button.

Motivation: This design can be seen also on Stack Overflow, so it is familiar for students and teachers.

Scenario #12: Answering a question from the QA page;
Steps:

1) Click the "QA" button that will redirect you to the QA page;
2) See a list of questions that can be answered;
3) Choose from the list of questions and answer it. E.g.: choose the question "JS frameworks", and in the answer box enter your opinion: "Some JS frameworks that I am aware of are: Angular, Ember, React, Meteor.";
4) Press the submit button to send your answer to that question;
5) Your answers can be rated as either being helpful or not.

Motivation: The design can be seen also on Stack Overflow, making it more familiar for students and teachers.

Alternative Scenario #1: Reviewing a fun fact
Description:

- A fun fact can be reviewed on the fun fact page (by navigating to the “Fun facts” page, scrolling over the fun facts, selecting one and clicking the “Like” / “Dislike” button).
- The fun fact of the day can also be reviewed (by navigating to the “Main page”, where the fun fact of the day appears, and similarly clicking on the “Like” / “Dislike” button).
- The difference between the two possible ways of reviewing a fun fact is the starting page (Main page / Fun facts page), whereas the reviewing mechanism is the same.

Alternative Scenario #2: Accessing personal statistics and rankings
Description:

- If the user clicks on the “View My Profile” button, the website will redirect him to his profile, where he/she can see his/her score, together with graphs on acquired skills and badges.
- If the user clicks the “Rankings” button on the navigation bar, he/she will be redirected to the leaderboard page, where it is displayed the user’s standings: score, badges, position (together with other user’s standings).
- Both actions give common information to the user (score, standing, badges), but in a different format and style.

5. Walkthrough Evaluation Report

Scenario #2: Solving quizzes**;**

1) From any page, click on the Quizzes tab;
    - The user is trying to get on the Quizzes tab because he wants to solve a quiz;
    - The user will see the correct control because the Quizzes button will have an on-click effect, indicating that it has been selected;
    - The user will see that the control produces the desired effect because the Quizzes page will pop up after the button from the tab has been clicked;
    - There are many controls around the Quizzes button from the tab that the user might click and will take the user to another page than the intended one, but will have no harmful effect;
    - The user will understand that he got the desired effect since he will find himself on either a desired page or an undesired one.
2) Select the “Java Master” quiz;
    - The user is trying to select the “Java Master” quiz because he wants to solve that specific quiz;
    - The user will see the correct control because the “Java Master” quiz panel will have an on-click effect, indicating that it has been selected;
    - The user will see that the control produces the desired effect because the “Java Master” quiz page will pop up after the panel has been clicked;
    - All the quizzes on the page will have similar functionality, so that the user might click on another quiz and will be taken another page than the intended one, but will have no harmful effect;
    - The user will understand that he got the desired effect since he will find himself on either a desired page or an undesired one.
3) Answer the questions from the quiz by fulfilling their various requirements;
    - The user is trying to answer the questions from the quiz because that is his purpose for visiting the page;
    - The user will see the correct control because he is able to edit and create his own answer to the questions;
    - The user will see that the control produces the desired effect because: in case of “multiple/single choice” the currently selected answer(s) will have their buttons filled, in case of “fill-in the gaps”, the answer will appear in the gaps and in case of “free-answer” the answer will appear in the answer box;
    - Unless the user steps out of the question-solving area, there are no other controls to be selected other than the correct ones;
    - In case something goes wrong in answering a question (e.g. incomplete question) the user will receive warnings in text form so that he is able to understand what he did wrong.

4) Click on the “Submit” button and you will be redirected to the “Results” page for that quiz;
    - The user is trying to click the “Submit” button in order to finish answering a quiz;
    - The user will see the correct control because the “Submit” button will have an on-click effect, indicating that it has been selected;
    - The user will see that the control produces the desired effect because he will be redirected to the quiz’s “Results” page;
    - Unless the user steps out of the question-solving area, there is no other control to be selected other than the correct one;
    - In case questions have not been answered, the user will be asked if he wants to proceed with unanswered questions.
5) View your statistics;
    - The user is trying to view his statistics because he wants to know his performance on the quiz;
    - The user will see the correct control because text and icons will appear in front of him;
    - The user will see that the control produces the desired effect because he will be able to see his score for the respective quiz and his earned badges;
    - There is no control that can be selected to provide an undesired effect since the statistics are only to be viewed and interpreted by the user;
    - The user will be able to understand the feedback since it will be presented in both text and graphical form.

Conclusion: After evaluating the design, it has been decided that the user will be able to easily perform the desired action, thus no changes to the design are required.

Scenario #3: Posting the fun fact;

1) From any page, visit the “Fun-facts” page;
    - The user is trying to get on the “Fun-facts” page because he wants to post a fun-fact;
    - The user will see the correct control because the “Fun-facts” button will have an on-click effect, indicating that it has been selected;
    - The user will see that the control produces the desired effect because the “Fun-facts” page will pop up after the button from the tab has been clicked;
    - There are many controls around the “Fun-facts” button from the tab that the user might click and will take the user to another page than the intended one, but will have no harmful effect;
    - The user will understand that he got the desired effect since he will find himself on either a desired page or an undesired one.

2) From the “Fun-facts” page you can click on any fun-fact, then click on the “Post a fun-fact” button OR click the “Post a fun-fact” button directly from the fun-fact page;
    - The user is trying to click on the “Post a fun-fact” button because he wants to share a new fun-fact with the community;
    - The user will see the correct control because the “Post a fun-fact” button will have an on-click effect, indicating that it has been selected;
    - The user will see that the control produces the desired effect because the “Post a fun-fact” page will pop up after the button has been clicked;
    - There is no other control around this button that may impede the user to select the correct one;
    - The user will understand that he got the desired effect since he will find himself on either a desired page or an undesired one.
3) Add a title: “Did you know? – Firefox”;
    - The user is trying to add a title because he wants the fun fact to have a name;
    - The user will see the correct control because the user is able to insert text in the title box;
    - The user will see that the control produces the desired effect because the inserted title will appear in the title box;
    - There is no other control around this field that may impede the user to select the correct one;
    - The user will understand the provided feedback because if the user clicks the “Preview” button, the fun fact will have a title;
4) Add a brief description: “This fun fact explains the origin of the browser’s name and logo”;
    - The user is trying to add a short description to the fun fact because he wants the fun fact to have one;
    - The user will see the correct control because the user is able to insert text in the “brief description” box;
    - The user will see that the control produces the desired effect because the inserted description will appear in its box;
    - There is no other control around this field that may impede the user to select the correct one;
    - The user will understand the provided feedback because if the user clicks the “Preview” button, the fun fact will have a short description;

5) Add the content of the fun fact: “The English word for red panda is “Firefox” which is where the browser gets its name from – this means the Firefox logo is actually a red panda, not a fox!”;
    - The user is trying to add content to the fun fact because he wants to share that information with the community;
    - The user will see the correct control because the user is able to insert text in the content box;
    - The user will see that the control produces the desired effect because the inserted content will appear in its box;
    - There is no other control around this field that may impede the user to select the correct one;
    - The user will understand the provided feedback because if the user clicks the “Preview” button, the fun fact will have content;
6) Add related images: “Firefox logo in comparison with a red panda”;
    - The user is trying to add images to the fun fact because he wants to provide a graphical representation of the information he wants to share with the community;
    - The user will see the correct control because the user is able to either click the “Browse” button or drag and drop the image in its box;
    - The user will see that the control produces the desired effect because the inserted image will appear in the box;
    - There is no other control around this field that may impede the user to select the correct one;
    - The user will understand the provided feedback because if the user clicks the “Preview” button, the fun fact will have an image added;
7) Add links: https://www.thefactsite.com/2013/02/top-100-technology-facts.html;
    - The user is trying to add links to the fun fact because he wants to provide the users means to finding more information on the fun fact;
    - The user will see the correct control because the user is able to edit the box and insert links;
    - The user will see that the control produces the desired effect because the inserted links will appear in the box;
    - There is no other control around this field that may impede the user to select the correct one;
    - The user will understand the provided feedback because if the user clicks the “Preview” button, the fun fact will have links added;

8) Add related tags: “browser”, “Firefox”, “did-you-know”;
    - The user is trying to add tags to the fun fact because he wants to provide the users means to finding similar information to the one presented on this fun fact;
    - The user will see the correct control because the user is able to edit the box and insert tags;
    - The user will see that the control produces the desired effect because the inserted tags will appear in the box;
    - There is no other control around this field that may impede the user to select the correct one;
    - The user will understand the provided feedback because if the user clicks the “Preview” button, the fun fact will have tags added;
9) Submit the fun fact by pressing the “Post” button;
    - The user is trying to click the “Post” button in order to finish editing a fun-fact and submit it;
    - The user will see the correct control because the “Post” button will have an on-click effect, indicating that it has been selected;
    - The user will see that the control produces the desired effect because he will receive a “Your fun fact has been submitted! Thank you for sharing this information with us!” message and the fun fact can be seen on the “Fun facts” page;
    - The user might also be able to click the “Preview” or “Back” button, though they will have enough space between them, so they won’t be easily clicked. If the user will leave the title and/or the content blank, he/she will not see the desired effect of posting a fun fact;
    - In case the fun fact doesn’t have a title or content, the user will get a message indicating the lack of those (e.g. “Title missing: Adding a title is compulsory for posting the fun fact!”) and if the user presses the “Back” button he will be informed that his progress will not be saved and will be asked if he still wished to leave.

Conclusion: After evaluating the design, it has been decided that the user will be able to easily perform the desired action, thus no changes to the design are required.

Scenario #4: Fun fact of the day;

1) Visit the home page by either typing the URL in the browser or clicking the "Home" button from any page on the website;
    - The user wants to view the fun fact of the day. The desired effect is to get to "Home page", by clicking the "Home" button or typing the URL in browser;
    - The user will see the correct control, because the "Home" button in the navigation bar will have an on-click effect, indicating that it has been selected;
    - The user will see that the control produces the desired effect because it will lead him to the "Home page";
    - The user could click on another navigation button, but in that case, it will redirect him to an unwanted page;
    - The user will understand the feedback, because the "Home page" will appear, as a result of his choice.
2) Click on the bullets to find the "Fun fact of the day" (e.g. "Did you know? – Firefox") or wait for it to auto-scroll into view;
    - The user will want to find the "Fun fact of the day", by clicking on the bullets;
    - The user will see the correct control of the fun fact of the day;
    - The control produces the effect desired by the user, showing him the fun fact of the day;
    - The user might wait until it pops-up into the view;
    - The feedback of this step is showing the fun fact of the day.
3) View what the fun fact has to offer (Firefox logo – red panda comparison image, shortened content) and/or click on it for more details;
    - The user will want to see what the "Fun fact of the day" offers to him, and if it stirs up his interest;
    - The user will see the correct control of the fun fact of the day, given the fact that it will pop-up into his view;
    - The control produces the effect desired by the user, showing him details about the Fun fact of the day;
    - The user could click on the Fun fact of the day, if he wants to be redirected to the fun fact complete page;
    - The feedback of this step is showing details of the fun fact of the day.
4) Clicking on the fun-fact will redirect the user to that specific fun-fact's page.
    - The user will want to see the "Fun fact of the day" page, if he is interested;
    - The user will see the correct control, because by pressing on the fun fact, he will be redirected to its page;
    - The control produces the effect desired by the user, redirecting him to the fun fact page;
    - If the user is not interested, he could scroll down or close the fun fact of the day;

- The user will understand the feedback, because he will either be redirected to the fun fact of the day page, or simply the Fun fact of the day pop-up will be closed.

Conclusion: After evaluating the design, it has been decided that the user will be able to easily perform the desired action, thus no changes to the design are required.

Scenario #5: Reviewing fun facts

1) Click on the “Fun fact page”;
    - The user wants to view fun facts. The desired effect is to get to the “Fun fact page”, by clicking the corresponding button in the navigation bar;
    - The user will see the correct control, because the “Fun fact” button in the navigation bar will have an on-click effect, indicating that it has been selected;
    - The control produces the desired effect as long as the user clicks on the right element in the navigation bar;
    - The user could click on another navigation button, which will redirect him to a completely different page. In that case he/she will see the mistake;
    - The user will understand the feedback, because if he/she clicks on the “Fun fact page”, a list of fun facts will appear (desired page), and the button will have a clicked effect. If instead, the user got to another page, the content and the selected button will be different.
2) Click on the “Did you know? – Firefox” fun fact;
    - The user wants to select a fun fact from the list of the fun facts by clicking on the region of the fact, in order to like, dislike, share and/or comment;
    - The user will see the correct control of the fun fact being selected;
    - The control produces the desired effect of showing the fun fact to the user, given he/she clicked on its region;
    - There is no other control that the user might select;
    - The feedback of this step is showing that the fun fact is selected, by emphasizing its borders and making a selected style for its region. The user will automatically see the fun fact being selected after he/she clicks on it.

3) Click on either the “Like”, “Dislike” or “Share” button;
    - The user wants to produce the effect of giving feedback to a fun fact (like dislike) or to share it. There are 3 buttons available for these actions. The user has to select one of them in order to perform the desired action;
    - The control produces the desired effect of liking/disliking/sharing the fun fact;
    - The user will see the correct control, as the corresponding button will have an “on hover” style before clicking, and a permanently selected style after clicking;
    - The user might click on another close button (e.g. dislike instead of like), in case of which he can undo the action by clicking on the button again;
    - The user will understand the feedback: e.g. in case of pressing the like button, the button will have a selected style (another color, thicker border), and the like count will increase by one. In case the user wants to undo the action, another press on the button will result in the button having a deselected style and the like/dislike count decreasing by one.
4) Insert a comment in the “Comment” field: “I really find this fun fact interesting, good job for sharing it!”;
    - The user wants to fill the comment field with a relevant comment;
    - The user will see the correct control of the comment appearing in the temporary comment field (not sent until the “Send comment” button is clicked);
    - The control produces the desired effect of filling the comment field;
    - An incorrect control would be the insertion of invalid characters which is not supported by the website, in this case the invalid characters will not appear;
    - The user will see immediate feedback, as the comment field will contain the text inserted by the user. The user can edit this as long as he/she wishes.
5) Either press “Enter” to submit a comment or click the “Send comment” button;
    - The user wants to produce the effect of leaving a comment to the selected fun fact, by completing the comment field and pressing the “Send comment” button or by hitting ENTER;
    - The user will see the correct control of the message being sent and appearing at the last position;
    - The control produces the desired effect of posting a comment, the comment being attached to the fun fact, and appearing as the last submitted comment, offering information about the user (poster of the comment) and the content of the comment;
    - The user can leave the comment field empty, in case of which the comment will not be sent and a warning message will appear;
    - The user will understand the feedback. In case of a valid comment, after hitting ENTER, pressing “Send comment”, the user will see his/her comment in the list of comments. In case the comment is empty, the comment field will have a warning style, and the “Cannot post empty comment!” error message will appear.

Conclusion: After evaluating the design, it has been decided that the user will be able to easily perform the desired action, thus no changes to the design are required.

Scenario #8: Accessing the global leaderboard/ranking;

1) From either the "Home" page or any other website page, click on the "Rankings" button on the navigation bar;
    - The user wants to view the global leaderboard, by clicking the corresponding button from the navigation bar;
    - The user will see the correct control, as the button will have an on-click style;
    - The control will produce the desired effect, redirecting the user to the corresponding page;
    - The user could do a mistake by pressing another button, leading him to another page, which will make him realize this;
    - The user will understand the feedback, because the button will have an on-click style, leading him to the corresponding page of the rankings.
2) On the top of the page you will see your standings (e.g. User: Mihai, Total Score: 125, Rewards: "Completed 10 quizzes", "Posted 7 Fun-Facts", "Achieved "Master" in Java", Position: 17);
    - The user would like to view his standing and statistics, in the top part of the page;
    - The user will see all the information wanted, i.e. his username, total score, rewards and position;
    - The control will produce the effect of showing the user the leaderboard and his standing;
    - There is no other control which would produce an undesired effect;
    - The user will see as feedback his standing compared to others.
3) On the rest of the page you will see a list containing the leaderboard (e.g. Position: 1: User: Andrei, Score: 250, Position 2: User Andreea, Score 225, etc.);
    - The user would like to see how he is positioned compared to other users;
    - The user will see all the information about all users;
    - The control will produce the effect of showing the user other rankings;
    - There is no other control which would produce an undesired effect;
    - The user will see as feedback the leaderboard.
4) Click on any user to view his statistics and achievements (badges).
    - The user will want to view other users' statistics and achievements;
    - The user will see the correct control, as the button will have an on-click style;
    - The control will produce the desired effect, which will pop-up information about other users;
    - There is no other control which could produce undesired effect;
    - The user will see as feedback other users' achievements.

Conclusion: After evaluating the design, it has been decided that the user will be able to easily perform the desired action, thus no changes to the design are required.

Scenario #9: Access to personal profile and viewing statistics

1) Click on either the “Rankings” button, or the “My Account” button;
    - The user wants to view his/her personal profile. The desired effect is to get to the “My Account page”, by clicking the corresponding button in the navigation bar;
    - The user will see the correct control, as the button will have an on-select style;
    - The control will produce the desired effect of navigating to the “My Account” page, as long as the corresponding button is clicked;
    - The user could do the mistake of clicking by mistake on another button in the navigation bar, in case of which he/she will be redirected on another page and will see immediately the mistake, in case of which he can go back or click on the desired navigation button;
    - The user will understand the feedback: the button corresponding to the current page will have a selected style, and the content of the page will be the user profile containing personal information, statistics and badges.
2) If on the “Rankings” page, click on the “View My Profile” button;
    - The user wants to produce the effect of going to the “My Account page”, from the “Rankings page”. In the part where the user sees his/her personal standings, there will be a button “View my profile”;
    - The button will have an “on hover” style, indicating that the user will be redirected after the button has been pressed;
    - The control will produce the desired effect of reaching the “My Account page”;
    - There is no other control that the user might select which would produce an error;
    - Similarly, the user will understand the feedback (page content and button having a selected style).
3) See your name: “John Doe”, university: “College Cork”, your role: “computer science student”, age (optional): 21, score: 15872 and standing: 8 on the top of the page;
    - The user would like to view his/her personal information and score, by scrolling on the “My Account page”;
    - The user will see part of the information, and a scrolling is necessary to display more elements;
    - The control will produce the desired effect of viewing different parts of the “My Account page”;
    - There is no other control which would produce an undesired effect;
    - The user will see as feedback the parts of his/her profile.

4) View a graph on the acquired skills: Technologies, Concepts, Mathematics, Algorithms, Hardware, Software, etc.;
    - The user will like to view acquired skills in different technologies. This can be done by scrolling to the skills section and hovering with the mouse on different statistics in order to get more details;
    - The user will see the correct control (detailed information appears when hovering with the mouse on different graph parts);
    - The control will produce the desired effect of viewing the statistics and acquiring details;
    - The user will understand the feedback, as he will see parts of the statistics, elements of the graphs, details, numerical values characterizing his/her performance.
5) View personal badges: “Math-Master”, “Hardware Hero”, “VHDL Geek”, etc.;
    - The user wants to view his personal badge. In order to produce this effect, he/she needs to scroll to the badges section;
    - The user will see the desired section, if he/she scrolls on the page. In case of hovering over a badge, relevant information will appear;
    - The control will produce the desired effect of having a list with all the acquired badges, in a nice format;
    - There is no other control that the user might select instead of the current one in this case;
    - The user will understand the feedback: badges are shown in a designated section on the page.

Conclusion: After evaluating the design, it has been decided that the user will be able to easily perform the desired action, thus no changes to the design are required.

Scenario #10: Contact-the-developer;

1) Click on the “Feedback” button in order to visit the respective page;
    - The user wants to provide some feedback related to his or hers experience or, in the worst case, make a complaint;
    - It will be very easy, even for a first-time user to find the “Feedback” button, as it is located at the top of the page in the navigation bar, always available;
    - After the button is pressed, the user is redirected the “Contact the developer page”, clearly indicating that the action completed successfully;
    - The user may select another button from the navigation tab, or another button from the page he is currently at;
    - If the user chooses another action, no feedback is provided regarding where to find the “Feedback” button. At this stage the application is unaware of a user’s intention, so it is considered his or her responsibility to make the right call.

2) Read the “How important is your opinion to us” text;
    - Many users may skip this step, in which the type of feedback expected and the way in which is handled by the development team is described. However, giving that any type of written input is accepted, this will not hurt the flow of the execution. In order to compel users to read the message, it is the first thing displayed on the page;
    - The users will clearly see the message, as it is the first thing displayed on the page;
    - This step does not impact the flow of execution. There is no way to verify that the right control has been used;
    - The user may directly go to the input area, or press the submit button;
    - Because the flow of execution is not disturbed by skipping this step no corrective feedback is required;
3) Write the suggestion: “I believe you should add more badges and achievements for users so that we would feel more engaged in the activities on the website”;
    - The user wants to input his or her thoughts into a filed, in order to send them to the developer team;
    - The input area for the feedback is the only text input available on this page, there is no danger of confusing it with some other field;
    - As the users type in, the message is also echoed on the screen, thus confirming the normal flow of the operation;
    - No other means of inputting text are available on the page;
    - There is no danger of filling in a different form, so no feedback is provided. In the worst case, the users did not select the input text and all the keystrokes are ignored. Anyone who has used a computer before will interpret this in the right way and select the input area.
4) Press the “Submit” button to provide your feedback;
    - After completing the message, the user will want to send it;
    - This button is the only one available on the screen, right after the input area, making it a very logical next step in the control flow of the application;
    - After the button is pressed, the user is redirected to a new page, confirming that his or her message has been successfully sent;
    - No other button is available on the page so there is no risk of erroneous control flow;
    - In case the user does not type in any message and the button is pressed, an error message will inform him about the correction, namely to type in some form of text.

Conclusion: After evaluating the design, it has been decided that the user will be able to easily perform the desired action, thus no changes to the design are required.

Scenario #12: Answering a question from the QA page;

1) Click the "QA" button that will redirect you to the QA page;
    - The user wants to answer a question posted by another user, in order to help others in need or for the more selfish goal of increasing his or hers score/ranking;
    - In order to access the page, the user must click “QA” in the navbar which is always available;
    - If the right tab is selected, the user will be redirected to the QA page;
    - The user may select a different tab, thus arriving at a different page of the application;
    - There is no way at this point to categorize any of the user’s actions as erroneous, as his or her intentions are not yet known to the application. In consequence, no feedback can be provided.
2) See a list of questions that can be answered;
    - The users browse through all the questions in order to choose one he or she can solve;
    - Upon entering the page, all questions are displayed, without any additional effort (apart from scrolling) on the user’s behalf;
    - As he or she scrolls, the page displays the questions accordingly;
    - The user may select accidently a question in which case he or she will be sent to that questions dedicated page;
    - If that is the case, by being redirected to a new page, any user with a minimum experience of surfing the web, will understand that he or she accidently clicked the question and needs to use the browsers “Back” button;
3) Choose from the list of questions and answer it. E.g.: choose the question "JS frameworks", and in the answer box enter your opinion: "Some JS frameworks that I am aware of are: Angular, Ember, React, Meteor.";
    - After selecting a question, the user wants to fill in his or her answer;
    - On the question page, right after the question’s detailed presentation, there is an input area available (only one on the page) that the user must click;
    - As the user types, his or her words are echoed on the screen thus confirming that the flow of action is normal;
    - There is no other input area available on the page, so the user cannot select a different form to write into;
    - There is no danger of filling in a different form, so no feedback is provided. In the worst case, the users did not select the input text and all the keystrokes are ignored. Anyone who has used a computer before will interpret this in the right way and select the input area.

4) Press the submit button to send your answer to that question;
    - After providing the answer to a question, the user will want to submit it;
    - The submit button is right after the input box. However, buttons for liking or disliking the question or any of the previous answers are also available;
    - After pressing the submit button, the input area will be cleared, and the newly added answer will be right at the top, after the input field;
    - The user may press one of the like or dislike buttons;
    - In case the user presses one of the like/dislike buttons instead of the submit one, he or she may notice a change in the counts for likes, but most obviously, the input area will still contain his or her message, and no new answer will be added to the page. Any user with minimum experience surfing this web will interpret this correctly and look for a submit button (which is conveniently situated right after the input field).

Conclusion: After evaluating the design, it has been decided that the user will be able to easily perform the desired action, thus no changes to the design are required.

6. Heuristic Evaluation

This section describes the evaluation of Jakob Nielsen's 10 usability heuristics on the design of the 12 tasks associated as groups of 3.

Group 1: Create Quiz, Solve Quiz, Access to personal profile;

1. Visibility of system status

Compliance:

Each field/component has a label describing what it displays/must be filled in. Considering that our website does not have any functionality that requires intensive computation we did not have to add any message of the type "waiting" or" processing".

Example:

Both pages related to quiz have labels near the fields of interest such as "Title" or "Description". The personal profile page only displays data about the user. It shows personal details of the person next to its picture. The statistics all have labels describing what they refer to.

Problems:

When creating a quiz, a user might make a typo which the system will not detect. This is more dangerous in the "tags" section as they are used for computing a user's statistics. In consequence if a user creates a quiz with tag "htnl" instead of "html" it will not contribute to the "technology" statistic as there is no "htnl" technology. In addition to that when solving a quiz, the same problem will be encountered by the user who solves it, as he will not be rewarded for the misspelled tag.

Possible solution:

Add a combo box for tags instead of a fill-in box.

User Experience Improvements:

Besides adding the combo box, we could make the labels more colorful to better attract attention.

2. Match between system and the real world

Compliance:

The quizzes follow the same structure as real-world quizzes (question, types of questions used, elements required when creating a quiz etc.) The personal page is in accordance to what we think is the mind-model most people have on how statistics/results should be displayed.

Example:

We are using 3 types of question which are often encountered in the real world such as multiple choice, fill-in-the-blanks or free answer. The personal profile displays way in a manner often encountered in real world (first the name, then occupation and optionally age).

Problems:

There may be some issues regarding the personal page. Some people may find other designs/ordering of items more intuitive. As far as quizzes go, we believe they are a perfect match to what a university student encounters in everyday life.

Possible solution:

In order the make the personal page more intuitive we could make a survey and ask about what different people would prefer. Then, we could change the page accordingly.

User Experience Improvements:

Provide a more interactive/engaging design with more visual elements.

3. User control and freedom

Compliance:

As long as the user does not press the “submit” or “add” button, nothing is saved. If he or she enters a page by mistake, it can easily navigate back using the browser’s “back” button, or by selecting another item from the navbar at the top of the page.

Example:

If a user accidently clicks on “Java Master” quiz he or she can either press the back button or select “Quizzes” from the navbar. This will go back to the select page without affecting the user in any way shape or form.

Problems:

We do not have any “undo” “redo” button which in some cases (solving a quiz) could be helpful. We do not view them as necessary, as a user may select a new answer with the same ease as pressing the “undo” button. Only the free answer type of question may have a slight benefit from this type of functionality.

Possible solution:

Add this functionality to the app, which would increase the implementation's difficulty by quite a margin.

User Experience Improvements:

Add autocorrect function for the type-in type of fields.

4. Consistency and standards

Compliance:

The same keywords have been used across the design as well as the same fonts, colors and placement of buttons.

Example:

Both create and solve quiz pages have their buttons at the bottom of the page in a green rectangular container.

Problems:

There may be a few slip-ups in terms of the vocabulary chosen and some user may be confused using the same word in different contexts.

Possible solution:

Make a survey and ask possible users about that and how they would change.

User Experience Improvements:

Use fonts, word and colors which are pleasant to the user, intuitive and attract attention without overwhelming the user.

5. Error prevention

Compliance:

As there is not a lot of functionality and in order not to reduce a user's freedom, we have not taken a lot of steps as far as errors go. That being said, we try to involve the user only when it is essential. As a result, a user can only enter personal information when editing his/her profile. All the statistics are computed automatically to avoid any involuntary or voluntary (cheating) typos made by a user when describing his or her skills.

Example:

A user can only select an answer for question where only one is correct. If another answer is checked the old one is deselected. If a question has multiple correct answer this functionality is not available for obvious reasons.

Problems:

Regarding tags, we do not check whether the word entered is a valid tag.

Possible solution:

Add combo boxes.

User Experience Improvements:

Use more interactive components.

6. Recognition rather than recall

Compliance:

The users do not have to remember how to use the platform as it has been designed to be as intuitive as possible. For steps which may need some explanations we have labels with instructions of what is expected of the user to do.

Example:

When creating a quiz, the user does not have to remember what each type of question is, as it is explained, in addition to having the same name as in the paper-based case.

Problems:

Again, regarding the tags, someone may not be aware of what are the most relevant tags used to describe a certain concept.

Possible solution:

Add a combo box.

User Experience Improvements:

Add more visual cues (images).

7. Flexibility and efficiency of use

Compliance:

There is no distinction between an experienced user and a novice one. The only difference comes in the difficulty of the quiz they are able to solve. The instructions/requirements for each task are designed in order to make the experience as pleasant as possible.

Example:

All users are able to create and solve quizzes. The process is the same and all options (types of question) are available to everyone.

Problems:

Some user may prefer some key shortcuts for the functionality of some buttons.

Possible solution:

Add some shortcuts (such as “submit”) to the keyboard.

User Experience Improvements:

Add some shortcuts (such as “submit”) to the keyboard.

8. Aesthetics and minimalist design

Compliance:

All design elements serve a purpose. They are meant to attract users’ attention to certain parts of a page which accomplish different functionalities.

Example:

We use white as the back-ground color in order not to tire the eyes. We only encapsulate in different color containers only what is essential to be easily distinguishable from the surroundings.

Problems:

Some users may prefer a more colorful design with more visual effects when pressing a button or navigating between pages.

Possible solution:

Perform a survey and ask users about what design elements they would change.

User Experience Improvements:

Add effects for when a button is pressed or slowly fade-in elements when entering a page.

9. Help users recognize, diagnose, and recover from errors

Compliance:

Whenever a user's input is not in accordance with an accepted one, he or she is notified with an error message on top of the field with the error and the measures needed in order to pass the check.

Example:

For example, if a user forgets to fill in a name when editing his or hers profile a message "Name cannot be empty" is shown above the name field.

Problems:

Some users may prefer the message to be shown in a more "aggressive" manner maybe via a dialog box.

Possible solution:

Show more serious errors using dialog-boxes.

User Experience Improvements:

Use more details when displaying an error.

10. Help and documentation

Compliance:

As there is no complex functionality in using our platform, we did not create a special help section to describe how the user can use the platform. However, we did specify instructions for those tasks that we considered may need a bit more clarification as in the case of creating a "fill-in-the-blanks" question.

Example:

When creating a "fill-in-the-blank" question, the user has specific clarification on where to fill in the text showed on the page, how to include blanks, and how to associate each blank to a particular question.

Problems:

After creating a few quizzed a user may find the instruction present on the page as unnecessary and prefer they would not be present all the time.

Possible solution:

Create a special "help" page or even add tutorials on how to solve/add quizzes on out page dedicated to tutorials.

User Experience Improvements:

Make the instruction, which are now visible at all times, only appear when the user hovers a certain element or clicks on an input field.

Group 2: Posting a fun fact, Fun fact of the day, Reviewing fun facts

1. Visibility of system status

Compliance:

Every component that is present on these pages has an appropriate label or icon describing its purpose.

Example:

Every fun fact has its title and content present and labeled accordingly. The “Fun fact of the day” does not have such labels for presentability purposes.

Problems:

Not all typos will be detected when posting a fun fact, this especially being visible at the “tags” section.

Possible solution:

Adding a dropdown-box with most common tags might be a solution to the aforementioned problem.

User Experience Improvements:

In order to improve the user experience, we would like to add more constraints to the input and more informative messages regarding those constraints. What is more, we would like to add a dropdown-box with most common tags so that the user would be less inclined to create his own.

2. Match between system and the real world

Compliance:

The fun facts follow a real-life pattern: a piece of information accompanied with visual content (pictures/videos). The fun fact of the day is nothing more than a more condensed version of the aforementioned piece of information, which is easier to digest.

Example:

The reviews follow a natural, top-down flow, indicating the order in which they have been added.

Problems:

Some users might find that the fun-fact format has an overwhelming amount of information.

Possible solution:

Reducing the amount of information and moving the full content of the fun-fact to another page.

User Experience Improvements:

In order to improve the users’ experience, we could move the full content of the fun-fact to another page and only keep the title and short description on the fun-fact page.

3. User control and freedom

Compliance:

The user is not able to commit to any change to the system without being asked if he wishes to do that. What is more, the user is free to roam between functionalities without any repercussions.

Example:

The user finds himself on the “Post a fun fact” page, completes some of the fields, but meanwhile he finds out that the fun fact has already been posted. He, then, can leave the page without anything being saved.

Problems:

The review system does not do any check-up when the user wishes to post a review, so he might end up posting an unwanted review.

Possible solution:

Add a delete/edit button to the posted review.

User Experience Improvements:

We can add a delete/edit button to the posted review, thus the user may have more control over what he posts.

4. Consistency and standards

Compliance:

The design has been made in such a way so that posting a fun fact is consistent to other types of posts on the website. What is more, same color codes and fonts have been used across the entire platform.

Example:

On the “Post a fun fact” page the “Back”, “Reset” and “Post fun-fact” buttons are placed at the bottom of the page, with the same color and font as other “Post” pages.

Problems:

Some unintentional consistency issues may appear during the development.

Possible solution:

Encourage users to leave feedback on the design and usability of the pages.

User Experience Improvements:

Take as much of the feedback into consideration as possible and build the design in such a way that as many changes as possible can be made without breaking the application.

5. Error prevention

Compliance:

Users are prompted with error messages whenever they wish to post fun-facts and the information that was entered is invalid.

Example:

If the “Title” field of a fun fact is empty, the user will receive a “Title of the Fun Fact is missing” message upon submitting the fact.

Problems:

The user only receives error messages upon posting the fun-fact.

Possible solution:

The error messages could be displayed when the field is out of focus rather than on submission.

User Experience Improvements:

We can improve the feedback the user receives both before and after submitting a fun fact so that the creation process may go smoother, with as few errors as possible.

6. Recognition rather than recall

Compliance:

The platform has been built in such a way that the user always knows what he is supposed to do. If not, labels are present, with possible options, in front of him.

Example:

When posting a fun fact, every field the user must complete has appropriate labels indicating the content they are expected to contain.

Problems:

The user might not be able to remember which tags the current piece of content belongs to or which tags already exist.

Possible solution:

Adding a dropdown-box with most common tags might be a solution to the aforementioned problem.

User Experience Improvements:

In order to improve the user experience, we would like to add more constraints to the input and more informative messages regarding those constraints. What is more, we would like to add a dropdown-box with most common tags so that the user would be less inclined to create his own.

7. Flexibility and efficiency of use

Compliance:

In the matter of flexibility and efficiency of use the users are not being differentiated.

Example:

Both experienced and inexperienced users have the same options when interacting with the members of this group.

Problems:

More experienced users might want to improve their efficiency when posting a fun fact.

Possible solution:

Add navigation/submission shortcuts to the keyboard.

User Experience Improvements:

Add navigation/submission shortcuts to the keyboard.

8. Aesthetics and minimalist design

Compliance:

Every piece of information on pages belonging to this group serves a purpose and is relevant to the user.

Example:

The users are interested in the title, short description, content and reviews of the fun facts.

Problems:

While the users are generally interested on the aforementioned properties, they usually are interested in the properties of one of the fun facts, not all of them.

Possible solution:

Make the fun fact expand/collapse on click, hiding the irrelevant content for the user.

User Experience Improvements:

We can add an expansion/collapse functionality on all the fun facts and comments on the fun-fact page.

9. Help users recognize, diagnose, and recover from errors

Compliance:

All errors that appear on the platform contain no error codes and are intuitive to the user.

Example:

If a user tries to access an invalid page of the application, a “Page not found” message will be displayed.

Problems:

Not all possible errors are dealt with accordingly or have messages that are intuitive enough.

Possible solution:

Ask feedback from the users then conduct a survey on the most frequent issues that appear during the feedback.

User Experience Improvements:

Make changes according to the feedback received from the users/testers.

10. Help and documentation

Compliance:

This group contains no functionality as complex as to require additional help or documentation, thus we didn’t create one.

Example: -

Problems:

First-time users might need help in posting a fun fact.

Possible solution:

A “Post a fun fact” tutorial, using pictures with highlighted buttons on pictures can be made on the tutorials page.

User Experience Improvements:

If we create the aforementioned tutorial, new users might feel less overwhelmed by the process of creating a fun fact.

Group 3: Posting a tutorial, Viewing and reviewing tutorials, Leaderboard page

1. Visibility of system status

Compliance:

Every component that is present on these pages has an appropriate label or icon describing its purpose.

Example:

Every tutorial has its title and content present and labeled accordingly. The "Leaderboard" has no functionality that needs special description.

Problems:

Not all typos will be detected when posting a tutorial, this especially being visible at the "tags" section.

Possible solution:

Adding a dropdown-box with most common tags might be a solution to the aforementioned problem.

User Experience Improvements:

In order to improve the user experience, we would like to add more constraints to the input and more informative messages regarding those constraints. What is more, we would like to add a dropdown-box with most common tags so that the user would be less inclined to create his own.

2. Match between system and the real world

Compliance:

Both the tutorials and the leaderboard follow a real-life pattern.

Example:

The tutorials contain a step-by-step top-down approach to solve the problem posed by the tutorial and the leaderboard contains a top-down list containing the best "players", their stats as well as the user's standing.

Problems:

The leaderboard does not contain all the users making it impossible to track one's progress in relation to the others.

Possible solution:

Add more slots on the leaderboard and maybe use a recycler-like view as well as leave to users the ability to visit other people's statistics.

User Experience Improvements:

In the future we plan on adding more slots on the leaderboard page as well as the ability to visit other users' statistics so that the comparison between them can be made in an easier way.

3. User control and freedom

Compliance:

The user is not able to commit to any change to the system without being asked if he wishes to do that. What is more, the user is free to roam between functionalities without any repercussions.

Example:

The user wishes to post a tutorial on the respective page, completes some of the fields but he is not pleased with the work that he's done so far. He, then, can leave the page without anything being saved. Same goes for visiting the "Leaderboard" page since read operations are traditionally harmless operations.

Problems:

The review system does not do any check-up when the user wishes to post a review, so he might end up posting an unwanted review.

Possible solution:

Add a delete/edit button to the posted review.

User Experience Improvements:

We can add a delete/edit button to the posted review, thus the user may have more control over what he posts.

4. Consistency and standards

Compliance:

The design has been made in such a way so that posting a fun fact is consistent to other types of posts on the website. What is more, same color codes and fonts have been used across the entire platform.

Example:

On the "Post a tutorial" page the "Cancel", "Add Steps" and "Done" buttons are placed at the bottom of the page, with the same color and font as other "Post" pages.

Problems:

Some unintentional consistency issues may appear during the development.

Possible solution:

Encourage users to leave feedback on the design and usability of the pages.

User Experience Improvements:

Take as much of the feedback into consideration as possible and build the design in such a way that as many changes as possible can be made without breaking the application.

5. Error prevention

Compliance:

Users are prompted with error messages whenever they wish to post tutorials and the information that was entered is invalid.

Example:

If the “Title” field of a tutorial is empty, the user will receive a “Title of the Tutorial is missing” message upon submitting the tutorial.

Problems:

The user only receives error messages upon submitting the tutorial.

Possible solution:

Error messages could be displayed when the field is out of focus rather than on submission.

User Experience Improvements:

We can improve the feedback the user receives both before and after submitting a tutorial so that the creation process goes smoother, with as few errors as possible.

6. Recognition rather than recall

Compliance:

The platform has been built in such a way that the user always knows what he is supposed to do. If not, labels are present, with possible options, in front of him.

Example:

When posting a fun fact, every field the user must complete has appropriate labels indicating the content they are expected to contain.

Problems:

The user might not be able to remember the tags the current piece of content belongs to, which tags already exist or the content of the tutorial before adding the steps.

Possible solution:

Adding a dropdown-box with most common tags might be a solution to the tags problem and forgetting the steps could be improved by adding them on the same page as the tutorial.

User Experience Improvements:

In order to improve the user experience, we would like to add more constraints to the input and more informative messages regarding those constraints. What is more, we would like to add a dropdown-box with most common tags so that the user would be less inclined to create his own. Finally, we would like to place the steps inside the “Add a tutorial” page by using a dynamic form.

7. Flexibility and efficiency of use

Compliance:

In the matter of flexibility and efficiency of use the users are not being differentiated.

Example:

Both experienced and inexperienced users have the same options when interacting with the members of this group.

Problems:

More experienced users might want to improve their efficiency when posting a tutorial.

Possible solution:

Add navigation, submission and step-adding shortcuts to the keyboard.

User Experience Improvements:

In order to improve the user's experience, we would like to add navigation, submission and step-adding shortcuts to the keyboard.

8. Aesthetics and minimalist design

Compliance:

Every piece of information on pages belonging to this group serves a purpose and is relevant to the user.

Example:

The users are interested in the title, short description, content and reviews of the tutorial.

Problems:

The users might not want to see the reviews of the tutorial alongside its content or they might want to hide some steps or piece of content.

Possible solution:

Every section of the tutorial could be hidden/expanded with a mouse click.

User Experience Improvements:

We can add an expansion/collapse functionality on all the components of the tutorial.

9. Help users recognize, diagnose, and recover from errors

Compliance:

All errors that appear on the platform contain no error codes and are intuitive to the user.

Example:

If a user tries to access an invalid page of the application, a “Page not found” message will be displayed. Furthermore, on submission, all errors are in plain text and are expressive enough for a user.

Problems:

Not all possible errors are dealt with accordingly or have messages that are intuitive enough.

Possible solution:

Ask feedback from the users then conduct a survey on the most frequent issues that appear during the feedback.

User Experience Improvements:

Make changes according to the feedback received from the users/testers.

10. Help and documentation

Compliance:

This group contains no functionality as complex as to require additional help or documentation, thus we didn’t create one.

Example: -

Problems:

First-time users might need help in posting a tutorial or making their way through the leaderboard and the meaning of the badges.

Possible solution:

A “Post a tutorial” tutorial, using pictures with highlighted sections on pictures can be made on the tutorials page. A “Leaderboard” tutorial could also be made in order to explain the user the competitive system of the platform.

User Experience Improvements:

Building the aforementioned tutorials might result in new users feeling less overwhelmed by the process of creating a tutorial or the depth of our competitive system.

Group 4: Posting a question, Answering a question, Contact the developer

1. Visibility of system status

Compliance:

The content of the page is presented in a clear way. Every input field has an associated label and in some cases explanations below the label. In case of errors in the inputs, the system will respond with a corresponding error message and highlight.

Example:

In case of Asking a Question, the first input field is labeled with the “Title” label followed by a requirement (“Each question must have a title”). Similarly, the “Body of the question” is a text area with an explanation (“Please explain the exact context and add a code snippet if possible”).

Problems:

Typos are not detected. The Contact us form has a text area with a limit of 500 characters. The user is not warned that he/she exceeded the maximum allowed characters, only after pressing the Submit! Button and not before.

Possible solution:

Add a warning message in the moment in which the 500 character limit is exceeded.

User Experience Improvements:

More informative messages appearing immediately in case of an error. Add animations when completing an input field.

2. Match between system and the real world

Compliance:

The forms in the QA and Contact sections are similar with real world forms, specifying the item details which need completion and a field where the user enters the necessary data.

Example:

Ask a Question form is a top-down form with labels (“Title”, “Body of the question”) and input fields.

Problems:

The user is not limited in the amount of characters/ content he/she can ask. This could lead to a very long question spanning on the entire page. There is not quite a match between the system and the real world in this case, as the text area can extend.

Possible solution:

Limit the amount of characters in the question text area.

User Experience Improvements:

Limit the text area expansion. Highlight selected input fields. Preview the question/ contact message before posting.

3. User control and freedom

Compliance:

The user should be able to press any button and complete any form at any time. There should not be restrictions as the order of the completed forms or a button being invisible in case the input is not valid.

Example:

The user can press the Submit question button even if the question does not have a title. A warning message will be displayed in order to help the user recover.

Problems:

There is no preview and undo functionality.

Possible solution:

A possible solution to increase user control and freedom would be to add the undo feature when competing a QA form, or answering a question (maybe the answer contains mistakes and the user would like to correct / delete.

User Experience Improvements:

Add the “Undo posting an Answer” and “Edit an Answer” functionalities in order to increase user control and freedom in case of a mistake.

4. Consistency and standards

Compliance:

In the QA list, every QA has the same format and the same information presented in exactly the same way and style. The forms that need completion use the same font, input types and submit buttons.

Example:

QA elements have the same structure (question, answer button, likes, dislikes and tags). The same color codes, fonts and images are used. Everything is consistent. In the case that a new question will be added, it will be also consistent.

Problems:

The forms in these sections have a unique style which is not necessarily consistent with the forms of the other pages. This is not a major problem, but for some users it can represent a problem.

Possible solution:

Change the style of the forms in order to look like a preview of the final version and also make it similar with the forms of the other pages.

User Experience Improvements:

Change color code palette in order to be more pleasing for the users and make this color palette consistent.

5. Error prevention

Compliance:

The user should be informed if a specific field contains errors/ the data entered is not valid, before submitting the form.

Example:

In the Contact the developer section, the user is informed in case the feedback message is too short or too long (max 500 characters). The system will not allow the user to post and invalid feedback. The user needs to edit the text area and try again.

Problems:

The error messages are displayed only after pressing the submit button. There is no warning in case the field is invalid and the user moves to the next/ another field.

Possible solution:

Add warning messages in case the user moves to the next/another field immediately. This way the user will get an immediate feedback and can correct his/her mistake at the same time.

User Experience Improvements:

In order to prevent the user from making errors, the system will not let the user enter additional characters if the max allowed character limit is reached and an immediate warning message will show up. Similarly, in the Ask a Question form.

6. Recognition rather than recall

Compliance:

The users can intuitively ask a question or post a feedback. They do not need to remember special steps or complex methods in order to reach their goals.

Example:

In case of answering a question, the user will see the Answer question button, which pops out because of its desing. Pressing that button, a simple user friendly form will appear, explaining each step of the form completion. At the bottom of the page, the submit button is located, which also has a meaningful textual representation.

Problems:

There are no recognition rather than recall problems, since the user interface is fairly simple and even a new user can easily accommodate with this interface.

Possible solution:

No problems were identified at the previous step.

User Experience Improvements:

Suggesting the user possible tags when posting a question. Another improvement to be to detect typos and suggest alternative words that are grammatically correct.

7. Flexibility and efficiency of use

Compliance:

Since the users will be mostly students and some teachers, the type of users are not differentiated, everybody works with the same interface.

Example: None

Problems:

More experienced user could want to improve efficiency of posting an answer to a question.

Possible solution:

One solution would be to create a small contextual menu in case a question is selected. An advanced user could complete the form in the contextual menu, and leave a quick reply to the question. The user will not need to go on a separate page to complete this action.

User Experience Improvements:

The user experience could be improved by adding shortcuts, special commands for advanced users such as leave a fast reply to a question or post a question by pressing the ENTER key.

8. Aesthetics and minimalist design

Compliance:

The design of the QA and Contact page should be minimal but also aesthetic. There should not be too much information presented, only the necessary fields. The font and style used should be aesthetic, pleasing and simple.

Example:

The “Answer a question” form has a minimalist desing of only 3 labels and 3 input fields, in a top-bottom design. The font is consistent and there is not an overwhelming amount of text and explanations.

Problems:

Some user would prefer a more colorful form and QA element, while still maintaining its original design. Unanswered questions could have another background color in order to signal that the question still needs some answers.

Possible solution:

Review design and change it to be more colorful, with a better color palette, while not changing the textual information provided, which is enough.

User Experience Improvements:

Ask users by surveys to see if the design is aesthetic and minimalist and take into consideration the feedback.

9. Help users recognize, diagnose, and recover from errors

Compliance:

Every form is validated, adding meaningful validations and displaying warning messages which will help the users to recognize and correct the errors.

Example:

In the case the user wants to post a new question and leaves the title empty, the system will help the user recognize and diagnose the mistake by printing an error message ("Empty title not allowed. Please add a title to your question") which is a simple but yet exact message explaining what the problem is. The user then can recover by entering a title.

Problems:

There could be errors / special cases which are not covered by the validators.

Possible solution:

Analyze all possible scenarios and verify if there are errors which are not covered by the validators of the forms. If there is, implement them and add them to the forms.

User Experience Improvements:

Besides the primary warning message, a tip could also be added to further help users which are still unsure of how to solve the error. In case the mouse will hover on the error message, a tip / idea will pop up, explaining the user in detail why the error occurred and how can he/she fix it.

10. Help and documentation

Compliance:

This group contains no functionality as complex as to require additional help or documentation, thus we didn't create one.

Example: None.

Problems:

Some users could still be in need of documentation, especially when posting a new question. Besides the functionality, the users should know the correct format of the question (title of a few words asking the main question, followed by a longer section, containing the body of the question).

Possible solution:

Create a documentation illustrating examples of completed forms and steps needed to reach each user goal.

User Experience Improvements:

Adding images and videos of people using the website. Another improvement would be to add examples of completing a QA, Contact form. A special tutorial dedicated for using the website on the tutorials page.

Heuristics evaluation conclusion

We believe we have addressed most of the issues that an online learning platform may pose. There are a few possible difficulties a user may face when using the application that we have only identified after the implementation. These hurdles are mostly related to how the user input is expected (particularly the user should be able to choose from a predefined set in some places rather than be allowed to type in), the ways in which bad input is handled (maybe errors should be pointed out as they happen rather than at the very end), the way in which our application is organized and visually represented. For the last two points we have selected the design which we believe is the best. Unfortunately, there is no guarantee that other users will feel the same. The only way possible to discover possible future amends (if required), is by interviewing the users of the platform. These interviews should take place right after a user starts its activity on the platform (to discover first impression, ease of learning) and after a prolonged period of time (to see whether the user maintains its points, or they see the benefits of our design). Another feature we believe could be added to our application is spell-checking for certain fields.