TDT4242 Spring 2021: Exercise 1 – Requirement Engineering (10 points)

Purpose of the exercise

This exercise is designed to let you practice requirement elicitation methods, boilerplates methods, and requirement quality evaluation.

Context of the exercise

We have developed a web application https://secfit-v22.herokuapp.com/ with only very high-level requirements specification (see Appendix).

Note: This web application is the same one we have previously used in another course, TDT4237 (software security). We inserted several security vulnerabilities in this web application for exercises of TDT4237. So, if you run the web application and find that it is insecure, you do not need to fee~~l~~l surprised.

The existing web application is in this URL: https://gitlab.stud.idi.ntnu.no/aasmuha/tdt4242-base

You should fork this repository in later exercises.

Exercise design and tasks

We will organize groups into pairs. One group will behave as the customers, and another group will behave as developers. Then, the roles will be swapped. The customer group needs to propose some new features to be added to the existing web application. The needs will be documented by the developer group using proper requirement engineering approaches.

Task 1: Elicit and describe requirements of new features of the web application required by your peer group (8 points)

- Sit together with your peer group, brainstorm to generate ideas of the new features and elicit your peer group's functional requirements based on actual demands you can think of.
- The requirements should come from reasonable and actual needs. Use Goal-oriented requirement language to describe the goals, sub-goals and possible related rules/constraints/regulations regarding the requirement.
- Describe the requirements by drawing up to 3 use cases. You need to use the use case template presented in the lecture or the "software requirement book (uploaded in Blackboard) chapter 8."
- Convert the use cases into functional requirements using boilerplates.
- Elicit and describe two non-functional requirements of the new features (two per new feature).

Task 1 should be written and delivered in the Requirement document. Limit length for Task 1 is 5 A4 pages.

Task 2: Cross-evaluation of quality of the functional requirements (2 points)

- Exchange the requirements documents you create and let your peer group (i.e., your customer) evaluate and comment on the quality of the requirements you document with respect to completeness, unambiguity, consistency, and correctness. (Note that the quality of your Requirement document will also be evaluated and reported in your peer group's report)
- Hold a meeting between both groups to discuss the requirement evaluation. As you are your peer group's customer, you should know if your peer group really catch your ideas and document your idea correctly.

Task 2 should be written and added to the Requirement document. Limit length for Task 1 is 2 A4 pages.

Kommentar [HG1]: Maybe this limit should be a word count instead of page count? Then we don't risk students delivering two pages with a very small font size to include as much text as possible.

Kommentar [HG2]: Same goes for limit of task 1

Evaluation criteria

- Proper use of goal-oriented approaches to document the needs of the new features
- Proper use of the use case methods to describe the requirements
- Proper formulation of the functional requirements using boilerplates
- Proper formulation of non-functional requirements
- A proper and comprehensive evaluation of the requirements documentation of your peer group
- It is mandatory to attend the brainstorming to generate ideas of the new features of your group and to attend the requirements evaluation discussion of your peer group (Note: Each group leader should report the attendance of the group members in these events in the report. No show up can lead to up 2 points punishment)

Appendix: High-level requirements of the existing web app

ID	Type	Role	Description
FR1	Sign up and sign in	All	A user must be able to sign in to an authorized session and gain rights to log workouts and meals, create exercises, leave comments, and upload files.
FR2	Create/log workout	Athlete	The athlete should be able to create/log a workout consisting of a date, name, zero or more exercises, and notes. A visibility level (Public, Coach, or Private) must also be set. Each exercise will have one or more sets of some number of units (e.g., seconds, km, reps).
FR3	Upload images and video to workout	Athlete	The athlete should be able to be able to upload images and/or videos to their workouts.
FR4	Edit workout	Athlete	The athlete should be able to edit all details of their workouts after they have been created and saved.
FR5	View workout	All	The user should be able to view all of the details, files, and comments on workouts of sufficient visibility. For athletes, this means that the workout needs to be either their own or public. For coaches, this means that the workout is at least one of their athletes' non-private workouts OR the workout is public. For visitors, this means that the workout needs to be public.
FR6	Create exercise	All	The user should be able to create a new exercise both inside and outside of workouts. An exercise consists of a name, description, unit, duration, calories

			burned per minute and used muscle group.
FR7	Upload documents	Coach	The coach should be able to upload documents for each of his athletes.
FR8	View coach documents	Athlete	The athlete should be able to view all of the documents that the coach has uploaded for them
FR9	Leave comment	All	The athlete should be able to comment on their own workouts. The coach should be able to comment on their athletes' nonprivate workouts.
FR10	View list of own workouts	Athlete	The athlete should be able to view a list of all of their own work outs.
FR11	View list of athletes' workouts	Coach	The coach should be able to view a list of all of their athletes' workouts.
FR12	View all public workouts	All	The user should be able to view a list of all public workouts.
FR13	Filter workouts	All	The user should be able to filter out workouts from each of the aforementioned lists.
FR14	Sort workouts	All	The user should be able to sort workouts (e.g., by date or name).
FR15	View athletes	Coach	The coach should be able to view a list of all of their athletes.
FR16	Add athlete	Coach	The coach should be able to add another athlete by sending a request to that user.
FR17	Remove athlete	Coach	The coach should be able to remove athletes.
FR18	Accept/Decline coach offer	Athlete	The athlete should be able to accept/decline a coach's offer.
FR19	Set coach	Athlete	The athlete should be able to specify a coach.
FR20	Remove coach	Athlete	The athlete should be able to remove a user as their coach.
FR21	Register new user	All	The user should be able to register a new account with personal information, a username, e-mail address, and password.
FR22	Workout image gallery	All	The user should be able to press to enter a gallery view from the workout and see a gallery of the image files from the workout.
FR23	Delete gallery files	Athlete	The owner of the workout should be able to delete images from the gallery.
FR24	Create/log meal	Athlete	The athlete should be able to create/log a meal consisting of a date, name, calories and notes.
FR25	Upload files to meal	Athlete	The athlete should be able to be able to upload images and/or videos to their meals.
FR26	Edit meal	Athlete	The athlete should be able to edit all details of their meal after creating and saving it.
FR27	View meal	All	The user should be able to view all of the details and files on the meals.
FR28	Sort meals	All	The user should be able to sort meals by date or name