EXAMINATION

The exam is open book, open laptop and open Internet. But you have to provide your individual solutions – the collaboration in any form is not allowed. You can provide your solutions either on paper or electronically. In case of the electronic solutions, please email the file to <rma@ut.ee> no later than 20:05 with a subject “[RE] exam solutions”.

Scenario

(adapted from the scenario of the Software Engineering course in 2012)

Tartu-based shirt manufacturer Sangar (www.sangar.ee) is considering the possibility of expanding its online presence. Currently (in November 2017) Sangar’s Web site re-opened standard e-shop functionality. The Web site allows customers to browse the catalogue of shirts sold by Sangar, to add shirts to the shopping cart and to check-out and pay online. Shirts are shipped to the designated address via postal services. The application has received an average of 3000 visits per month since its deployment. About 20% of these visits lead to purchases. On average, each purchase is for 2 shirts and the profit margin per shirt is EUR 4.

In December 2017, Sangar started to consider the possibility of manufacturing and selling customized shirts through their e-shop. Customers would be able to design customized shirts by selecting the fabric, the type of collar, type of cuff, and the types of buttons. Customers would also be able to add a label (e.g. their full name or initials) to their shirt design. Once a customer has completed a shirt design, he/she can add one or multiple units of this shirt to their shopping cart. At each step during the customization process, the web site will display an image of the customized shirt as well as the price of the shirt. The price depends on the options that have been chosen (e.g. the type of fabric chosen). One of the advantages of selling customized shirts is that they have a higher margin (EUR 8 per customized shirt, as opposed to EUR 4 for traditional shirts).

Sangar hired a consultancy company named LeftConsultants to make a market analysis and to assess the potential benefits of selling customized shirts. The consultants concluded that if Sangar introduced into their e-shop site an option allowing customers to design their own customized shirts, the number of customers using the customization option would be about 2000 per month (in addition to the existing customer visits), and the conversion rate of customers who design customized shirts would be 50% – meaning that 50% of customers who design a customized shirt would end up buying it. Customers of customized shirts would buy on average 1.5 shirts per purchase. Each customized shirt has a profit margin of EUR 8.

Sangar’s objective: to put a system (i.e., software-intensive system) in place for “mass-customisation” of shirts.

Disclaimer: Although based on a real-world company, the above scenario is fictitious and does not reflect the current situation or plans of the company in question.
Main Examination Task

For the given scenario, prepare a requirements specification (document), which includes requirements for the new software-intensive system or its component.

When creating the specification you must pay attention to the following concerns: (20 points)

- Your requirements specification should be structured, organised, traced, and etc. and should follow any known requirements specification template (below please indicate which one).
- Requirements specification follows ____________________________ template.
- Maintain the term glossary during the whole requirements specification process.
- Define the traceability model including both the traceability artefacts and traceability relationships. Maintain the traceability in the whole requirements specification.
- Unless indicated differently, the goals, domain properties, and (all types of) requirements must be documented using the “requirements shell” or any other structured-text template.
- Explain what parts correspond to what solutions of the examination tasks. You can use the following table:

<table>
<thead>
<tr>
<th>Examination task</th>
<th>Solution given in Section, Table, Figure, etc…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td></td>
</tr>
<tr>
<td>Task 2</td>
<td></td>
</tr>
<tr>
<td>Task 3</td>
<td></td>
</tr>
<tr>
<td>Task 4</td>
<td></td>
</tr>
<tr>
<td>Task 5</td>
<td></td>
</tr>
<tr>
<td>Task 6</td>
<td></td>
</tr>
<tr>
<td>Task 7</td>
<td></td>
</tr>
<tr>
<td>Task 8</td>
<td></td>
</tr>
<tr>
<td>Task 9</td>
<td></td>
</tr>
<tr>
<td>Task 10</td>
<td></td>
</tr>
<tr>
<td>Task 11</td>
<td></td>
</tr>
</tbody>
</table>

1. Carefully choose the problem scope!!! Clarify the problem scope along the four facets of the system context (i.e., subject facet, usage facet, IT facet, and development facet). (5 points)

2. What are major stakeholders and their goals? How do these stakeholders depend on one another for their goal achievement? Explain your answer with the i* strategic dependency model (social viewpoint). (10 points)
3. Explain what will be a *software intensive system* in the given problem description. Give at least one example (not need for the requirements shell) of system requirement and software requirement. (5 points)

4. What are the *functions* of the software intensive system? Explain your answer using the *graphical use cases*. Explain how the elicited functions satisfy goals defined in solution of Task 2. (10 points)

5. Select one *use case* from the graphical use case model and refine it using the *textual use case template*. The filled textual template should be at the *right level of detail* to illustrate at least four interactions of the stakeholders and software system. All defined scenarios (i.e., main, alternative, exceptional, and other) should respect the rules of scenario documentation. (10 points)

6. What are *additional goals* identified from the scenarios defined in the textual use case template. Document at least two of these goals using the “requirements shell” or any other structured-text template. (10 points)

7. Create a *class diagram*, which would describe *conceptual* model of the system. Derive from the class diagram at least three solution-oriented requirements and document (using the “requirements shell” or any other structured-text template) at least two (solution-oriented) requirements. (10 points)

8. Create a *sequence diagram* to illustrate functionality and interactions among (at least) three objects from your class diagram. Derive from the sequence diagram at least one solution-oriented requirement and document it (using the “requirements shell” or any other structured-text template). (10 points)

9. Elicit and document *quality* (non-functional) requirements:
   - one performance requirement,
   - one reliability requirement,
   - one usability requirement.
How do the quality requirements characterise one (select from your specification) functional concern (defined in terms of the requirements artefacts, like goals, scenarios or solution-oriented requirement) of your software system? What is the traceability between the quality requirements and solution-oriented requirements? (10 points)

10. Create *traceability graph* for the selected concern of your intensive software system. Your traceability graph should include at least three types of traceability artefacts and at least two types of traceability relationships. The created traceability graph should be consistent with the traceability model. (10 points)

11. Apply the *model analysis* to verify correctness of your models (i.e., *strategic dependency model*, *use case model*, *class diagram*, and *sequence diagram*). Discuss, correctness of which models should be improved and how. (10 points)