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.

All specified goals, domain properties, requirements, scenarios, and etc. must respect the “good requirements” criteria.

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.

In the introduction of your requirements specification, 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></td>
<td></td>
</tr>
</tbody>
</table>
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.

1. What is the problem scope, which you have defined for your solution? Your answer should be given along four context facets (i.e., subject facet, usage facet, IT facet, and development facet).

2. What are major stakeholders and their goals? How do these stakeholders depend one on another for their goal achievement? Explain your answer with the i* strategic dependency model (social viewpoint).

3. What should the stakeholders do to satisfy the dependency relationships and to achieve their individual goals? Explain your answers with (at least one) strategic rationale model.

4. What should the software system do to satisfy the stakeholders' dependency relationships? Explain your answer with the i* strategic dependency model (technical viewpoint).

5. What should the software system do to help stakeholders to achieve their goals? Explain your answer with the i* strategic rationale model (technical viewpoint).

6. Document at least two selected goals (using the “requirements shell” or any other structured-text template).

7. What are the functions of the software system? Explain your answer using the graphical use cases. Explain how the elicited functions satisfy goals defined in solution of Task 2.

8. 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.

9. 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. Select one use case from the graphical use case model and refine it using the sequence diagram. The sequence diagram should be at the right level of detail to illustrate at least four interactions of the stakeholder and software system objects. Explain what are type, preconditions and post-conditions of the scenario documented in your sequence diagram.
11. What are additional goals identified from the sequence diagram defined in the sequence diagram. Document at least two of these goals using the “requirements shell” or any other structured-text template.

12. Take one goal described in Task 6. Build the goal hierarchy using the KAOS goal modelling language. The goal hierarchy should be at least of 4 levels. It should describe at least 2 alternative design solutions. The lower lever goals (of at least two branches) should be assigned to agents.

13. From the goal model defined in Task 12 derive at least 2 assumptions/expectations and document them using the “requirements shell” or any other structured-text template.

14. Operationalise at least 2 requirements defined in the goal model defined in Task 12. Describe what objects these requirements concern. What are the inputs and outputs created in the operation models?

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

16. Create a class diagram, which would describe a model of the data used in the software system. Derive from the class diagram at least two solution-oriented requirements and document them (using the “requirements shell” or any other structured-text template).

17. Create a state diagram for two objects of different classes from your class diagram. Derive from the state diagram at least two solution-oriented requirements and document them (using the “requirements shell” or any other structured-text template).

18. 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 two solution-oriented requirements and document them (using the “requirements shell” or any other structured-text template).

19. Elicit and document quality (non-functional) requirements:
   - two performance requirements,
   - one reliability requirement,
   - one usability requirement,
   - one security requirement, and
   - one portability 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?

20. Create traceability matrix for the selected traceability artefacts. Your matrix should include at least three types of traceability artefacts and at least three types of traceability relationships.

21. Create traceability graph for the selected functional concern of your software system.

22. Explain which requirements and why:
   - must be included to software system realisation;
   - must be excluded from the software system realisation;
   - are nice to have in the software system realisation.

23. Explain (e.g., by grouping requirements to the hierarchy structure) which requirements can be selected for prioritisation. Perform requirements prioritisation for the selected set of requirements. Use the AHP method. Your solution must include at least one illustration of the priority calculations.
24. Apply the model analysis to verify correctness of your models.

25. What is the quality of your requirements specification? Which quality properties should be improved? Discuss your specification along various quality types as listed in the “requirements documentation” lecture.

26. Validate your created requirements specification by checking whether the outputs of activities fulfill defined quality criteria. Explain your answer.

27. Validate your created requirements specification by checking whether the inputs of activities fulfill defined quality criteria. Explain your answer.

28. Validate your created requirements specification by checking whether the execution of activities adheres to process definitions and activity guidelines. Explain your answer.

29 Build a prototype to illustrate the major functionality of your software system. The prototype should check the feasibility and validity of the requirements. The prototype should support (a part of) the requirements specified in the requirements specification. Explain what requirements are implemented in the prototype, what additional requirements should be defined in the specification. You can do either the paper prototype or use the prototype software (e.g., proto.io). Draw-based mockups are not accepted as solution.