SRS Outline and Evaluation for afetbilgi.com

For further explanation and details, please, turn to the referred clauses of IEEE 29148-2018, the highlighted and commented version; pay attention to the comments. Clause numbers are written in italic in the outline below.)

The outline format is based on *Figure 8* in *Clause 8.5.2*.

Title Page **[1 pt]** *(Clause 9.2.1)*
Table of Contents **[1 pt]** *(Clause 9.2.2)*
List of Figures **[1 pt]** *(Clause 9.2.2)*
List of Tables (if any) **[1 pt]** *(Clause 9.2.2)*

1. Introduction *(Clause 9.6.1)* **[15 pts total]**
    - 1.1. Purpose of the System **[1 pts]** *(Clause 9.6.2, 9.5.2, 9.4.2)*
    - 1.2. Scope **[1 pts]** *(Clause 9.6.3, 9.5.3, 9.4.3)*
    - 1.3. System Overview
        - 1.3.1. System Perspective **[4 pts]** *(Clauses 9.6.4, 9.5.4.1)*
            Context diagram and explanations of context diagram go here. Plus, other content as appropriate.
        - 1.3.2. System Functions **[3 pts]** *(Clauses 9.6.5, 9.5.4.2)*
        - 1.3.3. Stakeholder Characteristics **[3 pts]** *(Clauses 9.6.6, 9.5.4.3, 9.4.5)*
            (Reminder: Stakeholders include (classes of) users as subset(s).)
        - 1.3.4. Limitations **[2 pts]** *(Clause 9.6.7)*
    - 1.4. Definitions **[1 pts]** (including acronyms and abbreviations) *(Clauses 9.2.3* and *9.2.5)*

2. References **[1 pts]** *(Clause 9.2.4)*

3. Specific Requirements *(Clause 9.6.10)* **[40 pts total]**
    - 3.1. External Interfaces **[5 pts]** *(Clause 9.6.11, 9.5.8)*
        Interface classes go here. Plus, other content as appropriate.
    - 3.2. Functions **[20 pts]** *(Clause 9.6.12, 9.5.5, 9.5.10)*
        *Use-case diagram and its explanations go here; use-case descriptions follow. **You should aim for 10 use-cases.** Choose the three most complicated use-cases. Include **one sequence diagram, one activity diagram and one state diagram** to elaborate these three use-cases. Plus, other content as appropriate.*
    - 3.3. Usability Requirements **[1 pts]** *(Clause 9.6.13, 9.5.6)*
    - 3.4. Performance Requirements **[1 pts]** *(Clause 9.6.14, 9.5.7)*
    - 3.5. Logical Database Requirements **[8 pts]** *(Clause 9.6.15)*
        Key data objects (persistent or not) and their major attributes will be described here. Show the class diagram(s) with associations. Class dictionary can be omitted, provided that naming is understandable.
    - 3.6. Design Constraints **[2 pts]** *(Clause 9.6.16)*
        Specify constraints on the system design imposed by external factors, such as official standards, regulatory requirements, or project limitations.
    - 3.7. System Attributes **[2 pts]** *(Clause 9.6.18)*
        Reliability, Availability, Security, and others as applicable.
    - 3.8. Supporting Information **[1 pts]** *(Clause 9.6.20)*

4. Suggestions to improve the existing system. **[43 pts total]**
    - 4.1. System Perspective **[3 pts]** *(Clauses 9.6.4, 9.5.4.1)*
        Context diagram and explanations of context diagram go here for suggestions to improve the existing system. Plus, other content as appropriate.
    - 4.2. External Interfaces **[5 pts]** *(Clause 9.6.11, 9.5.8)*
        Interface classes go here for suggestions to improve the existing system. Plus, other content as appropriate.
    - 4.3. Functions **[20 pts]** *(Clause 9.6.12, 9.5.5, 9.5.10)*
        Use-case diagram and its explanations go here for suggestions to improve the existing system; use-case descriptions follow. ***You should aim for 5 use-cases.*** *Choose the three most complicated use-cases. Include* ***one sequence diagram, one activity diagram and one state diagram*** *to elaborate these three use-cases. Plus, other content as appropriate.*
    - 4.4. Usability Requirements **[1 pts]** *(Clause 9.6.13, 9.5.6)*
    - 4.5. Performance Requirements **[1 pts]** *(Clause 9.6.14, 9.5.7)*
    - 4.6. Logical Database Requirements **[8 pts]** *(Clause 9.6.15)*
        Key data objects (persistent or not) and their major attributes will be described here for suggestions to improve the existing system. Show the class diagram(s) with associations. Class dictionary can be omitted, provided that naming is understandable.
    - 4.7. Design Constraints **[2 pts]** *(Clause 9.6.16)*
        Specify constraints on the system design imposed by external standards, regulatory requirements, or project limitations for suggestions to improve the existing system.
    - 4.8. System Attributes **[2 pts]** *(Clause 9.6.18)*
        Reliability, Availability, Security, and others as applicable for the improved system.
    - 4.9. Supporting Information **[1 pts]** *(Clause 9.6.20)*

Overall presentation and document quality **[3 pts]**

What is expected for SRS part-1:
The structure and format of your SRS document must be complete. All (sub)section titles must be present. The sections can be empty except Sections 1.3.1 and 3.2. Section 1.3.1 will have the context diagram and its explanation; Section 3.2 will have the use-case model (Note that a use-case model consists of a use-case diagram and the descriptions of the selected use-cases). You should aim to find 10 use-cases. For the Section 3.2, you don't need to create state diagram, activity diagram, and sequence diagram for the three most complicated use-cases for SRS part-1; though you are responsible to create these diagrams for SRS final. This is the minimum requirement, you can do more for SRS part-1.

Individual feedback will not be provided for part-1. However, common obvious mistakes will be summarized within two days of the deadline.

Use-case description format: Adopt a good format from the use case literature. A properly extended form of the textbook's format is ok.

Suggestions to improve the existing system (Section 4):
Section 3 is based on "afetbilgi.com" as is. Section 4 presents your own suggestions to improve existing "afetbilgi.com" project. Section 4 exists to model your suggestions to improve "afetbilgi.com". You can make useful and realistic suggestions to improve "afetbilgi.com" in this Section. Keep in mind that you will account for your suggestions later in Software Architecture Description. Section 4 will have the same organization as Section 3. In some cases you may need to reproduce some material from Section 3 for the sake of clarity; keep it at minimum. In other words, avoid unnecessary duplication between section 3 and section 4.

Section 4.1: System perspective resulting from your suggestion should be given here. System context diagram after your suggestions should be created here. If your suggestions provide any external entity different than existing external entities or any interactions on the existing external entities, these changes should be drawn using different color **-color yellow is a good choice-** for system context diagram after your suggestions. To sum up, system context diagram should include both existing project perspective and project perspective after your suggestions **using different color** so that we can easily understand your suggestions on system perspective.

Section 4.2: External Interfaces after your suggestions go here, also external interfaces class diagram should be created. If your suggestions provide any external interfaces different than existing external interfaces, these changes should be drawn using different color **-color yellow is a good choice-** for external interfaces class diagram. External interfaces class diagram should include both existing external interfaces and external interfaces after your suggestions **using different color.** We can easily understand your external interfaces suggestions thanks to different color on the diagram.

Section 4.3: You should aim to find 5 use-cases according to your suggestions. These suggested use-cases should be shown on the use-case diagram using different color, also existing use-cases should be included in the use-case diagram. **Color yellow** is again a good choice for these suggested use-cases of your suggestions. Suggested use-case descriptions should be created using same description format with existing use-case descriptions. Note that you don't need to include existing use-case descriptions in this Section. You should choose three most complicated use-cases among suggested use-cases. After that, you should include **one sequence diagram, one activity diagram and one state diagram** to elaborate these three use-cases.

Section 4.6: Logical database requirements after your suggestions go here, also logical database requirements class diagram should be created. If your suggestions lead to any database requirements different from existing database of "afetbilgi.com", these changes should be indicated using different color **-color yellow is a good choice-** for logical database class diagram. Logical database requirements class diagram should include existing logical database requirements (as much as needed) and logical database requirements specific to your suggestions **using different color.**

What is to be submitted:
One submission per group.

A zip file submitted to ODTUClass and named as **group#** including:

1) SRS document (including diagrams) named as **srs.pdf**.

2) UML diagrams' project files as separate files, which must be able to be opened by StarUML.

Grading (15% of overall course grade):

SRS part-1 → 1.5%

Document Structure (All Sections are present) → **16 pts**
1.3.1 → **24 pts**
3.2 → **60 pts**

SRS final → 13.5%

The grading rubric above is for SRS final.
Total collectable points = **106**.
Awarded points = **min (points collected, 100)**