CENG350 Software Engineering, Spring 2023-2024
SRS Outline and Evaluation v1.0 for FarmBot

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

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

Title Page **[0.5 pt]** *(Clause 9.2.1)*
Table of Contents **[0.5 pt]** *(Clause 9.2.2)*
List of Figures **[0.5 pt]** *(Clause 9.2.2)*
List of Tables (if any) **[0.5 pt]** *(Clause 9.2.2)*
Revision History **[0 pt]** *(Clause 9.2.1)*

1. Introduction *(Clause 9.6.1)* **[24 pts total]**
    - 1.1. Purpose of the System **[2 pt]** *(Clause 9.6.2, 9.5.2, 9.4.2)*
    - 1.2. Scope **[2 pt]** *(Clause 9.6.3, 9.5.3, 9.4.3)*
    - 1.3. System Overview
        - 1.3.1. System Perspective **[11 pts total]** *(Clauses 9.6.4, 9.5.4.1)*

            Context diagram and the explanations *of context go here* **[4 pts]**. *Plus, other content as appropriate. This subsection (namely 1.3.1) should be a summary; defer technical details to section 3.*

            - 1.3.1.1. System Interfaces **[1 pt]** *(Clause 9.6.4.1)*
            - 1.3.1.2. User Interfaces **[1 pt]** *(Clause 9.6.4.2)*
            - 1.3.1.3. Hardware Interfaces **[1 pt]** *(Clause 9.6.4.3)*
            - 1.3.1.4. Software Interfaces **[1 pt]** *(Clause 9.6.4.4)*
            - 1.3.1.5. Communication Interfaces **[1 pt]** *(Clause 9.6.4.5)*
            - 1.3.1.6. Memory Constraints **[1 pt]** *(Clause 9.6.4.6)*
            - 1.3.1.7. Operations **[1 pt]** *(Clause 9.6.4.7)*
        - 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)*
        - 1.3.4. Limitations **[2 pts]** *(Clause 9.6.7)*
    - 1.4. Definitions **[1 pt]** (including acronyms and abbreviations) *(Clauses 9.2.3* and *9.2.5)*

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

3. Specific Requirements *(Clause 9.6.10)* **[50 pts total]**
    - 3.1. External Interfaces **[6 pts]** *(Clause 9.6.11, 9.5.8)*

        External Interfaces Class Diagram and its explanations *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 *goes here;* ***detailed use-case descriptions in a reasonable template*** *follow. You are expected to have* ***about 10 use cases covering major system functionality.*** *Have some associations in your use-case diagram, e.g. include, extend, specialization. Choose three most complicated use cases. Construct three diagrams* ***(one sequence diagram, one activity diagram, and***

one state diagram) *to elaborate on these three use cases. Plus, other content as appropriate.*

3.3. Logical Database Requirements **[8 pts]** *(Clause 9.6.15)*

Key data objects (persistent or not) and their major attributes. Draw the ***Class Diagram*** *with associations. A class dictionary can be omitted, provided that the naming is understandable.*

3.4. Design Constraints **[4 pts]** *(Clause 9.6.16)*

Specify constraints on the system design imposed by external factors, such as official standards, regulatory requirements, or organizational/managerial limitations.

3.5. System Attributes **[10 pts]** *(Clause 9.6.18)*

Important quality attributes (Usability (Clause 9.6.13, 9.5.6), Performance (Clause 9.6.14, 9.5.7), Dependability properties, Maintainability, and so on) in the order of priority with associated requirements.

3.6. Supporting Information **[2 pts]** *(Clause 9.6.20)*

4. Suggestions to Improve the Existing System **[20 pts total]**

4.1. System Perspective **[2 pts]** *(Clauses 9.6.4, 9.5.4.1)*

Context diagram and explanations *of context go here for suggestions to improve the existing system. Plus, other content as appropriate.*

4.2. External Interfaces **[2 pts]** *(Clause 9.6.11, 9.5.8)*

External Interfaces Class Diagram and its explanations *go here for suggestions to improve the existing system. Plus, other content as appropriate.*

4.3. Functions **[8 pts]** *(Clause 9.6.12, 9.5.5, 9.5.10)*

Use-case diagram *for suggestions to improve the existing system goes here;* ***detailed use-case descriptions in a reasonable template*** *follow. You are expected to have* ***about 4 use cases covering suggestions to improve the existing system.*** *Have some associations in your use-case diagram**,** e.g. include, extend, specialization. Choose three most complicated use cases. Construct three diagrams* ***(one sequence diagram, one activity diagram, and one state diagram)*** *to elaborate on these three use cases. Plus, other content as appropriate.*

4.4. Logical Database Requirements **[3 pts]** *(Clause 9.6.15)*

Key data objects (persistent or not) and their major attributes for suggestions to improve the existing system. Draw the ***Class Diagram*** *with associations. A class dictionary can be omitted, provided that the naming is understandable.*

4.5. Design Constraints **[1 pt]** *(Clause 9.6.16)*

Specify constraints on the system design imposed by external factors, such as official standards, regulatory requirements, or organizational/managerial limitations for suggestions to improve the existing system.

4.6. System Attributes **[3 pt]** *(Clause 9.6.18)*

Important quality attributes (Usability (Clause 9.6.13, 9.5.6), Performance (Clause 9.6.14, 9.5.7), Dependability properties, Maintainability, and so on) in the order of priority with associated requirements for the improved system.

4.7. Supporting Information **[1 pt]** *(Clause 9.6.20)*

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

5. SysML Diagrams for Existing System and Suggestions **[30 pts total]**

(*This section should only be **completed by three-person groups**. Three-person groups are expected to **model SysML diagrams** in addition to UML diagrams. Three-person group scores (out of 130) will be normalized (out of 100).*)

5.1. SysML Requirement Diagram for Existing System **[5 pts]**

SysML Requirement Diagram *for the **existing** system goes here.*

5.2. SysML Behavior Diagrams for Existing System **[10 pts]**

Three *SysML Behavior Diagrams (one **SysML Activity Diagram,** one **SysML Sequence Diagram, and** one **SysML State Machine Diagram**) for the **existing** system elaborating on the Requirement Diagram (section 5.1) go here.*

5.3. SysML Requirement Diagram for Suggestions **[5 pts]**

SysML Requirement Diagram *for **suggestions** to improve the existing system goes here.*

5.4. SysML Behavior Diagram for Suggestions **[10 pts]**

Three *SysML Behavior Diagrams (one **SysML Activity Diagram,** one **SysML Sequence Diagram, and** one **SysML State Machine Diagram)** for **suggestions** to improve the existing system elaborating on the Requirement Diagram (section 5.3) go here.*

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 only**. Section 3.2 will have the use-case model only. (Note that a use-case model consists of **a use-case diagram and the descriptions of the use cases**). You should aim to **find 10 use cases**. For 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 for creating these diagrams for the SRS final. This is the minimum expectation, 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 the ***FarmBot Express* software as is**. Section 4 presents your suggestions to improve the existing FarmBot project. Section 4 exists to model your suggestions to improve FarmBot. You can make useful and realistic suggestions to improve FarmBot 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 the minimum level. In other words, avoid unnecessary duplication between section 3 and section 4.

Section 4.1: The system perspective resulting from your suggestion should be given here. **A 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 a different color **-the color yellow is a good choice-** for the system context diagram after your suggestions. To sum up, the system context diagram should include both the existing project perspective and project perspective after your suggestions **using a different color** so that we can easily understand your suggestions on the 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 from existing external interfaces, these changes should be drawn using a different color **-the color yellow is a good choice-** for the external interfaces class diagram. The external interfaces class diagram should include both existing external interfaces and external interfaces after your suggestions **using a different color.** We can easily understand your external interface suggestions thanks to the different color on the diagram.

Section 4.3: You should aim to **find 4 use cases** according to your suggestions. These suggested use cases should be shown on the **use-case diagram using a different color**, also existing use cases should be included in the use-case diagram. **The color yellow** is again a good choice for these suggested use cases of your suggestions. Suggested use-case descriptions should be created using the same description format as existing use-case descriptions. Note that you don't need to include existing use case descriptions in this Section. You should choose the 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 on these three use cases.

Section 4.4: 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 the existing database of FarmBot, these changes should be indicated using a different color **-the color yellow is a good choice-** for the logical database class diagram. The logical database requirements class diagram should include existing logical database requirements (as much as needed) and logical database requirements specific to your suggestions **using a different color.**

Note that both SRS part-1 and SRS final documents must include the **group number and names of members in the group**.

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 as figures) named as **SRS.pdf**.
2) UML diagrams' project files as separate files, which must be able to be opened by StarUML.

Grading (16% of overall course grade):

SRS part-1 → 1.6%

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

SRS final → 14.4%

The grading rubric above is for the SRS final.
Total collectable points = **100**.

For the three-person groups, the **SRS part-1 expectation is the same** as for other groups. However, **three-person groups should complete Section 5** in addition to other sections (from Section 1 to Section 4) for the **SRS final**. In other words, one-person groups and two-person groups shouldn't complete Section 5. Section 5 includes SysML models for both the existing system and the improved system.

SRS final grading for three-person groups:
Total collectable points for three-person groups = **130**
Overall grade for three-person groups = **Total collectable points *100/130**
Overall grade for three-person groups is **total collectable points scaled to 100.**