

**Jordan University of Science and Technology**

Software Engineering Department

SE321: Software Requirements Engineering

First Semester 2022

Software Requirements Specification (SRS) Template

Software Requirements Specification

**Hydroponics Agriculture**

**Version: (1) Date: (1/12/2022)**

**Group Members:**

|  |  |  |
| --- | --- | --- |
| **Name** | section | ID |
| **Ala’a Jamal Abdelqader** | 1 | 145335 |
| **Shaima’a Ali Bani issa** | 1 | 145493 |
| **Miral Thair Dalayka** | 1 | 149665 |

**Table of Contents**

1. Introduction 1

1.1 Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms, and Abbreviations 1

1.4 References 1

1.5 Overview 1

2. General Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Characteristics 2

2.4 General Constraints 2

2.5 Assumptions and Dependencies 2

2.6 Apportioning of Requirements. 2

3. Specific Requirements 2

3.1 External Interface Requirements 3

3.1.1 User Interfaces 3

3.1.2 Hardware Interfaces 3

3.1.3 Software Interfaces 3

3.1.4 Communications Interfaces 3

3.2 Functional Requirements 3

3.2.1 <Functional Requirement or Feature #1> 3

3.2.2 <Functional Requirement or Feature #2> 3

3.3 Use Cases 3

3.3.1 Use Case #1 3

3.4 Classes / Objects 3

3.4.1 <Class / Object #1> 3

3.4.2 <Class / Object #2> 3

3.5 Non-Functional Requirements 4

3.5.1 Performance 4

3.5.2 Reliability 4

3.5.3 Availability 4

3.5.4 Security 4

3.5.5 Maintainability 4

3.5.6 Portability 4

3.6 Inverse Requirements 4

3.7 Design Constraints 4

3.8 Logical Database Requirements 4

3.9 Other Requirements 4

4. Analysis Models 4

4.1 Sequence Diagrams 5

4.3 Data Flow Diagrams (DFD) 5

4.2 State-Transition Diagrams (STD) 5

5. Change Management Process 5

A. Appendices 5

A.1 Appendix 1 5

A.2 Appendix 2 5

# 1. Introduction

## Purpose

*The purpose of the following SRS document (Software Requirements Specification) is to clarify the developments that we will make on the hydroponic system, through the automation of manual processes, which improves the results of the process and reduces the costs of the existing system.  
By explaining the purpose and features of the system, both farmer and AE will simplify their functionality, and rise efficiency.*

## 1.2 Scope

*The HAS will decrease the user involvement by designing a smart monitor and controlling system, which can make it easy to implement the connection of a monitoring field and to remote monitoring centers.*

*Most specifically This system can monitor the sensors feedback accurately, automatically transmit the data of humidity, light intensity,* *EC levels, water temperature, water level, nutrients contained in water and pH in real time access on a mobile application. Moreover, the farmer has detailed access to the information.*

## 1.3 Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| *SRS* | *Software Requirements Specification* |
| *HAS* | *Hydroponics Agriculture System* |
| *Fun* | *Functional requirement* |
| *N-Fun* | *NON - Functional requirement* |
| *EC* | *electrical conductivity* |
| *Cams* | *cameras* |
| *AEs* | *Agricultural Engineers* |

## 1.4 References

Requirements engineering from system goals to UML models to software specifications are obtained and worked through Conducting interviews with officials in the Faculty of Agriculture at the University of Science and Technology.

<https://makehorticulturalknowledgework.com/projects/hydroponics-agriculture-and-employment-development-project>

SRS Template

<https://press.rebus.community/requirementsengineering/back-matter/appendix-c-ieee-830-template/>

## 

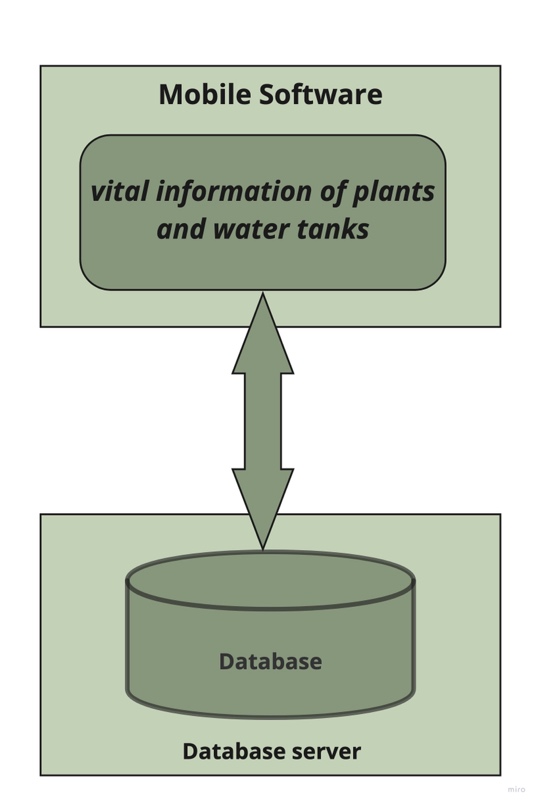
## 1.5 Overview

*In the following pages you will find two main points first one is The Overall Description which talks about the general factors that affect the hydroponic system like assumptions, constraints. And you will get a summary and general knowledge about hydroponic system requirements and functionality.*

*Second point is Specific Requirements in this point you will find functional requirements; requirements that define the fundamental actions the system will provide, then you will find External Interfaces; contains a detailed description of all inputs and outputs from the software system, followed by Performance Requirements “non functional requirements” , then Design Constraints which talks about design constraints that can be imposed by other standards, hardware limitations, etc.. , at the end you will find Software System Attributes; a number of attributes of software that can serve as requirements.*

# 2. General Description

## 2.1 Product Perspective BACK, PIC



*This system will consist of a mobile application. The mobile application will be used to monitor and know the vital information of plants and water tanks and get alerts in the event of any problem.*

*The farmer and the agricultural engineer can edit the vital information of the plant and water tanks (knowing that the engineer has more access in editing plant and water info)*

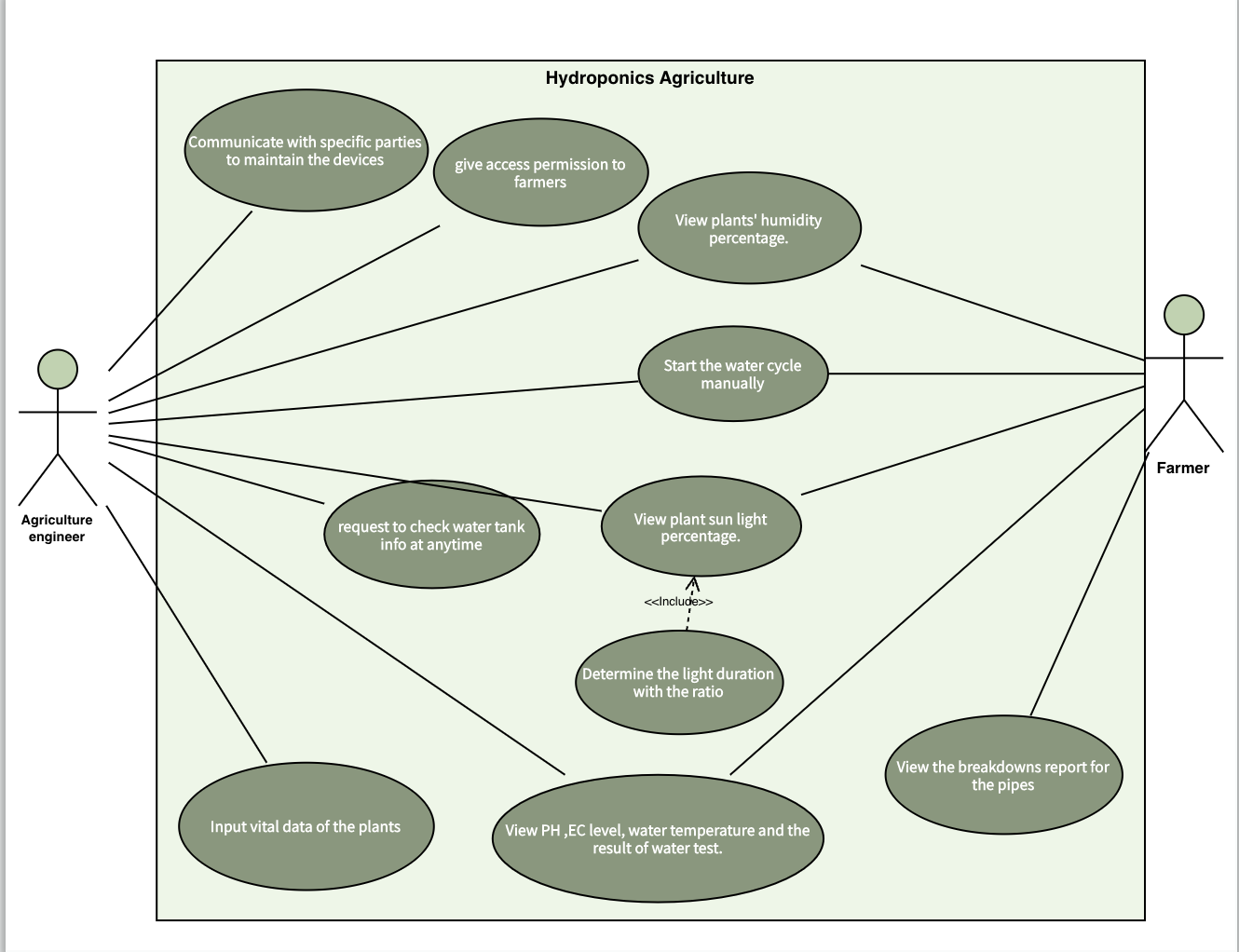
*Data from current plant information will be stored and organized in a database system. farmer and AE will be notified via email*

## 

## 2.2 Product Functions

*The hydroponic system consists of several features that make the process more successful in easier way such as:*

* *Independent access for all users to manage the system.*
* *Input data from the AE.*
* *Viewing data in simple ways to analyze, understand, and track by all users.*
* *Although the system is automatically works, but agriculture Eng has full control at any time.*
* *Improve maintaining processes by view breakdowns reports.*



## 2.3 User Characteristics

*The System contain Two main members, farmer and AE , the* ***AE*** *selects the Hydroponics field for a specific type of plant and it determines the vital information and values ​​of this plant in addition to preparing the water tank from EC levels, water temperature, water level and adding water cycle period depending on the studies conducted for this plant*

*(the system will notify the AE if there is any problem in the system) ،In addition, the agricultural engineer will communicate with specific parties in case any equipment needs maintenance or renewal,* ***Farmer*** *can monitor all plant information and obtain all updates continuously in addition to controlling some characteristics such as changing the time of the water cycle peroid in addition to controlling the duration and percentage of lighting. (the system will notify the farmer if there is any problem with plants)*

## 2.4 General Constraints

**Performance Requirements**

* **Static**

1. System shall *monitor all the vital information of plants field* at a same time.
2. System shall not allow farmer to sign in if he does not have access to the system form the AE.
3. System shall not allow farmer to enter any information for the plants.
4. System shall contain several specific information for each group of plants.
5. System shall show plant information to the maintenance company.

* **Dynamic**

1. The process of notifying the farmer and sending the report if any problem happened shall be processed in less than 10 seconds.
2. The process of updating plant infographic shall processed in less than 1 second.

**Design Constraints**

1. System shall be compatible on android and IOS operating systems.
2. System shall support the last 4 versions of mobile OS as a minimum.
3. System shall run via Internet Connection.
4. System shall run on any 1 GHZ single core processor or better.

**System attributes**

* **Reliability**

1. System shall not crash during usage .

* **Availability**
  1. System shall work all the time.
  2. A notification is sent to the farmer If any problem happened

* **Portability**

1. System should be written in Swift using X Code.
2. System should be written in Java or Kotlin using android studio SDK for android applications.

## 2.5 Assumptions and Dependencies

* *HAS is a mobile application that can be accessed from anywhere with an internet connection.*
* *A farmer can access the application if and only if the AE give him a permission.*
* *A farmer and AE both can monitor multiple fields at the same time.*
* Plant information should be updated continuously.
* *It is assumed that farmer and AE are familiar with how to use a mobile application before using HAS.*

## 2.6 Apportioning of Requirements

* *Add a new plant field (with its information).*
* *Add a new feature like live monitor plant via cams.*
* *Add a new feature such as chatting between farmer and AE.*

# 3. Specific Requirements

*This section contains all the software requirements at a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements.*

## 3.1 External Interface Requirements

*This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.*

### 3.1.1 User Interfaces

### 3.1.2 Hardware Interfaces

### 3.1.3 Software Interfaces

### 3.1.4 Communications Interfaces

## 3.2 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

### 3.2.1 <Functional Requirement or Feature #1>

3.2.1.1 Introduction

3.2.1.2 Inputs

3.2.1.3 Processing

3.2.1.4 Outputs

3.2.1.5 Error Handling

### 3.2.2 <Functional Requirement or Feature #2>

…

## 3.3 Use Cases

### 3.3.1 Use Case #1

### 3.3.2 Use Case #2

…

## 3.4 Classes / Objects

### 3.4.1 <Class / Object #1>

3.4.1.1 Attributes

3.4.1.2 Functions

<Reference to functional requirements and/or use cases>

### 3.4.2 <Class / Object #2>

…

## 3.5 Non-Functional Requirements

Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc).

### 3.5.1 Performance

### 3.5.2 Reliability

### 3.5.3 Availability

### 3.5.4 Security

### 3.5.5 Maintainability

### 3.5.6 Portability

## 3.6 Inverse Requirements

State any \*useful\* inverse requirements.

## 3.7 Design Constraints

Specify design constrains imposed by other standards, company policies, hardware limitation, etc. that will impact this software project.

## 3.8 Logical Database Requirements

Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc.

## 3.9 Other Requirements

Catchall section for any additional requirements.

# 4. Analysis Models

List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS’s requirements.

## 4.1 Sequence Diagrams

## 4.3 Data Flow Diagrams (DFD)

## 4.2 State-Transition Diagrams (STD)

# 5. Change Management Process

Identify and describe the process that will be used to update the SRS, as needed, when project scope or requirements change. Who can submit changes and by what means, and how will these changes be approved.

# A. Appendices

Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS’s overall set of requirements.

*Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.*

## A.1 Appendix 1

## A.2 Appendix 2