

# **Farm Management System Requirements Specification**

**Version 1.1**

**May 28, 2021**

To regenerate the TOC, select all (CTL-A) and press F9.

# Table of Contents

<b>1. EXECUTIVE SUMMARY .....</b>	<b>3</b>
1.1 PROJECT OVERVIEW .....	3
1.2 PURPOSE AND SCOPE OF THIS SPECIFICATION .....	3
<b>2. PRODUCT/SERVICE DESCRIPTION.....</b>	<b>3</b>
2.1 PRODUCT CONTEXT .....	3
2.2 USER CHARACTERISTICS .....	3
2.3 ASSUMPTIONS .....	4
2.4 CONSTRAINTS .....	4
2.5 DEPENDENCIES.....	ERROR! BOOKMARK NOT DEFINED.
<b>3. REQUIREMENTS .....</b>	<b>4</b>
3.1 FUNCTIONAL REQUIREMENTS .....	4
3.2 NON-FUNCTIONAL REQUIREMENTS .....	8
3.2.1 User Interface Requirements.....	8
3.2.2 Usability.....	8
3.2.3 Performance.....	9
3.2.4 Manageability/Maintainability .....	9
3.2.5 System Interface/Integration .....	10
3.2.6 Security.....	11
3.2.7 Data Management .....	11
3.2.8 Standards Compliance .....	Error! Bookmark not defined.
3.2.9 Portability .....	Error! Bookmark not defined.
3.2.10 Other Non-Functional Requirements .....	11
3.3 DOMAIN REQUIREMENTS.....	11
<b>4. USER SCENARIOS/USE CASES .....</b>	<b>11</b>
<b>APPENDIX .....</b>	<b>13</b>
APPENDIX A. DEFINITIONS, ACRONYMS, AND ABBREVIATIONS .....	15
APPENDIX B. REFERENCES.....	15
APPENDIX C. REQUIREMENTS TRACEABILITY MATRIX .....	15
APPENDIX D. ORGANIZING THE REQUIREMENTS .....	18

# 1. Executive Summary

## 1.1 Project Overview

In the last 10 years farms in Albania had a noticeable development after years of forced collectivization of agricultural cooperatives. According to official statistics, today in Albania, more than 90% of farms are mainly based on family farms that have an economic and social dimension; that is around 300 000 farms. Our goal is to provide the local farmers with an efficient solution based on their needs. Our project is centered on providing a management system for farmers in order to ease the management of their daily activity.

The Farming Management System will help to plan, monitor and analyze all activities on the farm easily. Tillage, planting, crop protection, fertilization, irrigation, harvesting and all other activities are managed with a few clicks. Plus, you can track input usage quantities, costs and work hours for every activity. The system will be also centered around livestock management: that include buying livestock, selling, and observing or calculating profit. Whether you are a cattleman or an agriculturist our product will provide useful facilities for you to control your business.

## 1.2 Purpose and Scope of this Specification

The purpose of this project is to implement technology in order to improve and facilitate the everyday operations of farms. This document will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external actions.

This document is intended for both the stakeholders and the developers of the system.

This software system will be a Farming Management System for local farm owners. This system will be designed to support production management and to comply with actual farming standards. By maximizing the farmer's work efficiency and production, the system will meet the farmer's needs while remaining easy to understand and use.

# 2. Product/Service Description

According to INSTAT, in Albania, as of 2011 there were 353'341 farms. Out of these farms, 299'077 were farms that practiced both livestock farming and agriculture, 54'086 were farms that practiced only agriculture, and 178 practiced only livestock breeding and farming.

## 2.1 Product Context

This product is independent and self-contained.

## 2.2 User Characteristics

Users that will use the program are:

- Farmers/Owners: the main users for which this program is built, they are often inexperienced with use of such software so we must make it easy for them
- Farm economists: these users will use the system heavily in calculating the profit and the spending of a farm. They expertise in the economic aspect of the product and to them will come in handy graphs that we will provide
- Employees: are the workforce of the farm, they will have a simple UI where mostly they will be assigned a task and they will report on it.
- Farm managers: in big farms the role of the owner and the role of farm manager may be done by different people. Farm managers are concerned with the economic aspect as well as with the organizational aspect of a farm.

## **2.3 Assumptions**

Assumptions that affect the requirements are:

- 1) The system must be available at all times
- 2) The user must be able to change and manipulate the data they enter in real time
- 3) There will be multiple users
- 4) There are 3 different scenarios in which the system may be used: a farm that practices both livestock farming and agriculture, a farm that only practices agriculture and a farm that practices only livestock breeding and farming.

## **2.4 Constraints**

User constrains

One user will only be able to see its own data and no other users' data.

Framework constrains

Laravel is easy to learn and fast to develop but some features are clearly better-suited for rapid prototyping and should be replaced as an application grows.

Symfony on the other hand seems to be a better option to consider in the future.

# **3. Requirements**

## **3.1 Functional Requirements**

BR => Business Requirement

BR\_A => Agriculture only farm

BR\_L => Livestock only farm

BR\_AL => Agriculture and Livestock

Note: BR\_A and BR\_L apply to BR\_AL as well. BR applies to all of them.

<b>Req#</b>	<b>Requirement</b>	<b>Comments</b>	<b>Priority</b>	<b>Date Rvwd</b>	<b>SME Reviewed / Approved</b>
BR_01	The system should store and show the data in an organized and consistent way.	Users must have access to all their data all of the time	1	06-Jun-21	Kejdi Domi
BR_02	The system must provide notifications between users.	Users can send messages to each other.	1	06-Jun-21	Kejdi Domi
BR_03	The system must prompt the user for registering its data when they are firstly using the system; registering a new account.	This should be handled with care and must be a multistep process so the user will not get bored and tired.	2	06-Jun-21	Kejdi Domi
BR_04	The only user which can be registered to the system autonomously will be the manager.	He will add other users as the registering phase continues.	1	06-Jun-21	Kejdi Domi

### **Farm Management System Requirements Specification**

<b>Req#</b>	<b>Requirement</b>	<b>Comments</b>	<b>Priority</b>	<b>Date Rvwd</b>	<b>SME Reviewed / Approved</b>
BR_05	After the manager creates a new user which may be an Owner, an Economist, or an Employee, the system must inform that user by email, including a temporary password in the email.	Logical conclusion of BR_04.	1	06-Jun-21	Kejdi Domi
BR_AL_06	The system must keep a list of Livestock information and Crop information that the farm sells/manages.	This will be shown in the dashboard like Crops, Livestock.	1	06-Jun-21	Kejdi Domi
BR_A_07	The system must provide curated information about the crops divided in these subsections: When to plant, How to take care, and Economic aspect	Research to be done for the types of plants and agricultural aspect. Information accessible in learn more section of Crops view in dashboard.	2	06-Jun-21	Kejdi Domi
BR_L_08	The system must provide an interface to manage veterinarians, where a user can view information like availability, address, phone number, name, and email.	We must account for multiple veterinarians.	2	06-Jun-21	Kejdi Domi
BR_L_09	The system must notify the user for upcoming vaccines or medical treatments that Livestock must go through their lifetime.	This is important to the certification of the institution and must be handled professionally.	1	06-Jun-21	Kejdi Domi
BR_AL_10	The system must keep a list of all buyers and suppliers of Livestock and Crops.	They will collectively be called The Market, and will be included in dashboard.	2	06-Jun-21	Kejdi Domi
BR_AL_11	The system must show best buyers; this will be calculated as a weighted average of the amount they buy, the cost they are willing to buy and the frequency.	This information together with BR_AL_10 is important to users: Manager, Economist and Owner.	3	06-Jun-21	Kejdi Domi
BR_AL_12	The system must show the most profitable suppliers; this will be calculated as a weighted average of the price they are willing to sell, the distance from the farm and the prosperity of the calves bought.	We will prompt the manager to add the distance in km.	3	06-Jun-21	Kejdi Domi

### **Farm Management System Requirements Specification**

<b>Req#</b>	<b>Requirement</b>	<b>Comments</b>	<b>Priority</b>	<b>Date Rvwd</b>	<b>SME Reviewed / Approved</b>
BR_13	The system must provide a comprehensive and detailed description of the legal requirements a farm must have to function properly in Albania.	We must learn and get used to all the legislation concerning farms in our country. This will be included in Legal Information in the dashboard.	1	06-Jun-21	Kejdi Domi
BR_14	At the end of every month the Economist must complete a Monthly Report form detailing the transactions and concluding in profit calculation.	This requirement is purely economic and is relied entirely on Economist actor.	1	06-Jun-21	Kejdi Domi
BR_A_15	Show a tracking page for plants that the user planted in form of a loading bar which will show the user the right time for seed planting/transplant and the right time for harvest.	The loading bar should have a gradient to it from green to red. The page will be accessed through Track button at Crop interface in the dashboard.	3	06-Jun-21	Kejdi Domi
BR_16	The system must keep a list of all users: Owners, Managers, Economists and Employees.	Certain users have privileges over others, for example: Owner can delete, add, modify Manager and Employee, Manager can delete, add, modify Economist and Employee but can only add Owner (connected to BR_04), Economist has only modify privileges over certain aspects of Employee user like wage. Employees can be considered least privileged users.	1	06-Jun-21	Kejdi Domi
BR_17	The system must provide an interface for the Manager where they can assign tasks to Employees.	This can be implemented as a button in the Employee entry in the Employee table (Employees Dashboard) which redirects to the task form. Keep in mind tasks can be repeated through the day, week, or even months. (Example => Time - 21:00, Repeat – every day, Task – clean the calves. Deniable - no)	2	06-Jun-21	Kejdi Domi
BR_18	The system must notify Employees for new tasks that were assigned to them.	In the Tasks section in employee dashboard.	2	06-Jun-21	Kejdi Domi

### **Farm Management System Requirements Specification**

<b>Req#</b>	<b>Requirement</b>	<b>Comments</b>	<b>Priority</b>	<b>Date Rvwd</b>	<b>SME Reviewed / Approved</b>
BR_19	Employees must be prompted to complete a short report on the tasks they accomplish.	If the task is deniable, they can explain why they denied it.	3	06-Jun-21	Kejdi Domi
BR_A_20	Suggestions about other uses for the crops the user is harvesting should be shown on loading screens and on the bottom of the homepage.	For example, the user might be selling grapes our system should be able to identify that the user is selling grapes (entered by the user) and suggest that the user sell also wine that comes from grapes.	3	06-Jun-21	Kejdi Domi
BR_AL_21	The system must provide a Share feature: a way to share information in json format between different farms. This information includes Market information, Crops information, Vets information and Livestock information.	The idea is that different farms can communicate with each other, especially exchanging Buyers and Suppliers information. Our system will work with JSON format to create files that will be saved locally on the user's machine and shared (Note that the system itself doesn't need a way to share information, the sharing is done by the user and may be done by email.)	3	06-Jun-21	Kejdi Domi
BR_22	The system must handle Deletions of data with caution; for highly important information like deleting a manager the system should ask for the current user's password.	This is done in order to not lose critical information.	1	06-Jun-21	Kejdi Domi
BR_A_21	Create a database separate from the whole project where we will store all data necessary for the growth and maintenance of different plants	This requirement is very important and needs further specifications. It will be divided into other requirements that will need their own research.	1	19-Apr-21	Kejdi Domi
BR_AL_22	Create a tab where the user can create, manage, and view the space it needs to store the crops harvested. The user can add warehouses, haystacks, and barrels.	For example, the user will add a haystack to his/her farm, we will prompt him to enter a value signaling the capacity of the haystack and after this another value that will tell how much of the haystack is occupied/free in percentage.	2	19-Apr-21	Kejdi Domi

### ***Farm Management System Requirements Specification***

<b>Req#</b>	<b>Requirement</b>	<b>Comments</b>	<b>Priority</b>	<b>Date Rvwd</b>	<b>SME Reviewed / Approved</b>
BR_AL_23	On startup and on loading pages, suggest crops (like soy) that livestock eats or show curiosities around the world related to livestock farms which can improve the performance of the farm.	A little touch to detail.	3	19-Apr-21	Kejdi Domi
BR_AL_24	Alert the user for the best time to invest in new livestock which will be correlated with the time that the crops are ready for harvest, and the time when there are more weddings.	If you are buying bred cows, the period from about May 15 to August 1 might be a good time, although the volume of sales is typically low.	3	19-Apr-21	Kejdi Domi
BR_25	The system must allow the user to choose its currencies and store the conversion rates between them.	Rates can be entered manually or chosen from dropdown menu.	2	06-Jun-21	Kejdi Domi
BR_26	The system must provide updated converting rates between currencies. Rates are to be updated every 24 hours.	Can be implemented through an API.	2	06-Jun-21	Kejdi Domi
BR_AL_27	The system must provide weather information.	This will be implemented through an API.	2	06-Jun-21	Kejdi Domi
BR_AL_28	The system must update weather information every 24 hours.	Weather will be shown in the Weather section in the dashboard.	2	06-Jun-21	Kejdi Domi

## **3.2 Non-Functional Requirements**

### **3.2.1 User Interface Requirements**

- The system must be able to run full screen on a screen no smaller than 800-600 px
- Menus should be divided into logical sections and we are to use Hierarchical Menus.
- Error messages are to be shown as popups above the current screen, except when there are errors connected to form completion, in that case the cursor must move to the entity which caused the error and a simple red highlight should be applied to the text area.

### **3.2.2 Usability**

- The system should be easy to learn and explain.
- The users must be prompted with messages displaying and explaining how to use the system.
- The system should be intuitive.



### **3.2.3 Performance**

#### **3.2.3.1 Capacity**

The solution should be capable of serving 1000 concurrent users for 70% of the time and up to 3000 concurrent users during the peak times. The solution should be capable of handling a load of 30,000 queries per minute.

#### **3.2.3.2 Availability**

- Maximum permitted number of failures per hour = 2
- The maintenance should be scheduled and we must notify the user 1 week prior.
- The system should be available 24/7
- Each user will have his own authentication and session
- There will be a backup database and a backup server so that if ever any crash happens there always is a backup and when work is being done the application will never stop being live.

#### **3.2.3.3 Latency**

Response Time: For 90% for user interactions the application response times should be <2 seconds. For any transaction having a response time greater than 0.5 sec a suitable progress indicator should be shown to the user. Any user interaction response time should not exceed 10 seconds. In cases of unforeseen delays in responses due to network latency etc. a timeout message should be shown after 15 seconds.

### **\*TO BE DECIDED\***

### **3.2.4 Manageability/Maintainability**

#### **3.2.4.1 Monitoring**

We are to use Sentry (a listener/handler for errors that asynchronously sends out the error/event to Sentry.io) with Laravel for monitoring errors.

#### **3.2.4.2 Maintenance**

If a system crash happens the core application is going to restart. While being in this process we are either going to use the other backup server or show to the page the message that for the moment our page is in maintenance until the moment that the server is up and running again. If there is a problem with the restart and backup does not happen correctly the server will be restarted. Application performance is dependent on:

- Database transaction speed
- User's internet bandwidth
- System specifications like RAM, processor speed, etc. dependent on user's machine
- Number of users on the application on the same time
- The Laravel framework capabilities

#### **3.2.4.3 Operations**

- Every user is expected to manage its own information, log in and of course log out.

## ***Farm Management System Requirements Specification***

- Managers are expected to add quantitative and relative data to the farm (Crops, Livestock, Market, Vets), as well as assign tasks to other employees
- Economists are expected to manage information concerned with the visualization of profit and expenses
- Owners are expected to deal with the legal technicalities of the farm and communicate decisions to managers and economists
- Employees are expected to perform tasks.
- CRUD functionalities affect all users as well as relationship classes like Crops or Livestock

### **3.2.5 System Interface/Integration**

Specify the use of other required products (e.g., a database or operating system), and interfaces with other systems (e.g., UWHires package interfaces with PubCookie and ODS, HEPPS system interfaces with Budget system). For each interface, define the interface in terms of message format and content. For well-documented interfaces, simply provide a reference to the documentation.

Outline each interface between the product and the hardware or network components of the system. This includes configuration characteristics (e.g., number of ports, instruction sets), what devices are to be supported, and protocols (e.g., signal handshake protocols).

#### **3.2.5.1 Network and Hardware Interfaces**

Specify the logical characteristics of each interface between the product and the hardware or network components of the system. This includes configuration characteristics (e.g., number of ports, instruction sets), what devices are to be supported, and protocols (e.g., signal handshake protocols).

#### **3.2.5.2 Systems Interfaces**

Example systems interface requirements:

##### ***A. System1-to-System2 Interface***

The <external party> will create and send a fixed length text file as an email attachment to [System2mail@u.washington.edu](mailto:System2mail@u.washington.edu) to be imported into the System2 system for payroll calculation. This file must be received on EDIT day by 4:00 PM in order to be processed in the EDIT night run. The requirements below document the file specifications, data transfer process, and specific schedule. This file is referred to as "FileName" in this document.

##### ***File Structure and Format***

A1. The FileName file is a fixed length text file.

A2. The FileName file is an unformatted ASCII file (text-only).

A3. The FileName file contains a batch totals record and several detail records.

##### ***File Description: Batch Totals Record***

A4. The batch totals record can be placed at the beginning, in the middle, or at the end of the file.

A5. The batch totals record contains the following:

- Record Type (value: XA)
- Process Type (value: A)
- Batch Number (3 digit number assigned by Payroll Dept)
- Origin Code (AIG)
- Total number of detail records
- Total deduction amount

##### ***File Description: Detail Records***

A6. The FileName file contains a row for each record meeting xxx criteria.

A7. Each row in the FileName file contains the following fields, comma-delimited and encased in double-quotes where the data includes commas or spaces:

- Employee Id
- Record Type
- Process Date (MMDDYY)

## ***Farm Management System Requirements Specification***

- XYG Number
- Element Code
- Amount
- Amount Sign
- Year Flag
- Total Amount
- Total Amt Sign

### **3.2.6 Security**

#### **3.2.6.1 Protection**

Specify the factors that will protect the system from malicious or accidental access, modification, disclosure, destruction, or misuse. For example:

- encryption
- activity logging, historical data sets
- restrictions on intermodule communications
- data integrity checks

#### **3.2.6.2 Authorization and Authentication**

Specify the Authorization and Authentication factors. Consider using standard tools such as PubCookie.

### **3.2.7 Data Management**

Specify the requirements for any information that is to be placed into a database, including

- types of information used by various functions
- frequency of use
- data access rules
- data entities and relationships
- integrity constraints
- data retention
- valid range, accuracy, and/or tolerance
- units of measure
- data formats
- default or initial values

### **3.2.8 Other Non-Functional Requirements**

Please provide all necessary non-functional requirements, similar to the requirements explained in the lesson slides or in the textbook.

## **3.3 Domain Requirements**

Everything related to the domain that might be needed in the project shall be mentioned in here. Sometimes the domain Requirements might be thought as part of either functional or non-functional requirements.

## **4. User Scenarios/Use Cases**

No.	User Case	Description
-----	-----------	-------------

### ***Farm Management System Requirements Specification***

1	Successful login	User provides correct email and password and logs in the system.
2	Failed login	User provides incorrect or incomplete credentials.
3	Create account	User provides email and password, retypes password and chooses its role. User is redirected to Login page to log in the system.
4	Incorrect password	User retypes incorrect password or tries to register with incomplete credentials.
5	User reads information about the system	There is detailed information in the Home page about the functionalities that the system offers.
6	User enters the dashboard	User can check information about crops and livestock, manage the inventory, check the market and weather updated in real-time and the profile.
7	Crops	User can access a table with detailed information for tracking plants; seed planting/transplant and harvesting.
8	Managing crops information	User can insert, edit, and delete data about space needed to store harvested crops.
9	Livestock	User can access detailed information about the livestock available in the database in a tabular format.
10	User registers livestock	User inserts information of a new livestock in the database.
11	User deletes livestock	User deletes livestock information from the database.
12	Crops inventory	User inserts and manages data about price and quantity of plants, profits and expenses.
13	Livestock inventory	User has access to profits and expenses from the livestock management.
14	Market	Users can access a list of all sellers of calves along with detailed information about livestock being sold and a list of most profitable sellers. User can access information for the best and most frequent buyers.
15	Search livestock	User can optionally search livestock by filtering the full list.
16	List of suppliers	User can access a full list of suppliers and buyers together with their addresses.
17	Weather	User is notified in real-time about climate changes.
18	User reminder	User is provided with a reminder functionality which can be used for anything.
19	Notifications	User is notified for upcoming vaccines or medical treatments that should be provided to the calves, as well as for best timing to invest in new livestock or planting/harvesting crops.
20	User profile	User can enter personal details or edit existing credentials.
21	User logout	User logs out of the system and is redirected to the login page.
22	User contacts the veterinarian	Any form of communication (e.g. phone, email) with the veterinarian must be provided to the user.

## **4.1 User Scenarios**

### **Employee Scenarios**

#### *1. Employee Login*

- a. Employee enters email and password
- b. If email and password are correct and match the database, they are redirected to the employee homepage
- c. Homepage displays Tasks Assigned table and can display Accomplished Tasks table with a push of a button

#### *2. Employee Login fails*

- a. Employee enters email and password
- b. If they do not match in the database an error will be shown letting them know something is incorrect

#### *3. Employee Views Personal Info*

- a. Employee clicks the profile picture or the username or the Profile link in dropdown menu
- b. Employee is redirected to a page showing his/her credentials like name, surname, age, phone number, wage, SSN, etc

#### *4. Employee Edits Personal Info*

- a. Employee has the right to change some personal information
- b. They click the settings icon in the dropdown menu
- c. Enters new info (password, username, phone number)
- d. They hit Save button and are required to enter their password as a security measurement
- e. Employees cannot change salary and SSN
- f. The data is updated and saved in database
- g. A message is shown to user to confirm that the data was changed and new data is being displayed in the profile page.

#### *5. Employee Views Tasks*

- a. Tasks are shown in user homepage, if no tasks a message informing the user there are no tasks required from him at this moment will be displayed
- b. If a task is clicked it shows a popup window displaying additional information with a "Mark as Done" button at the bottom
- c. Employee can sort tasks and choose to display up to 100 tasks at a time
- d. Tasks are shown in a table

#### *6. Employee Marks Tasks as Done*

- a. Employee clicks "Mark as Done" button in the task
- b. A task report is shown to be filled with a save button
- c. Task is put in "Accomplished Tasks" table

## ***Farm Management System Requirements Specification***

- d. A congratulation message is shown (not as a pop-up)

### ***7. Employee Logout***

- a. Employee clicks the Logout button in their dropdown menu
- b. Employee can freely logout from account

## **Owner Scenarios**

### ***8. Owner Logs In***

- a. Owner enters email and password
- b. He/she is redirected to Dashboard page

### ***9. Owner Views Personal Information***

- a. By clicking the profile picture, username or the profile link in the dropdown menu, owner is redirected to Owner Information Page
- b. Owner views his/her information

### ***10. Owner Edits Personal Info***

- a. Owner clicks the Settings icon in dropdown menu
- b. Owner is redirected to Edit Information page
- c. Owner changes his/her info (password, farm name, username, etc.,)
- d. Owner clicks save
- e. For security purposes we ask for confirmation and display a message
- f. Database is updated after owner clicks Yes in the dialog box
- g. Owner is redirected to the same page now showing new information

### ***11. Owner Adds a Manager / Economist***

- a. Owner clicks the “Manage Farm” button in the dropdown menu
- b. Redirection to Manage page where 2 tables may be shown, Manager table and Economist table
- c. Owner clicks “Add New Manager” button to be redirected in a form page
- d. Owner fills the necessary data and confirms the new manager
- e. A random password (may be left as default) is generated and shown together with a “Send Mail” button
- f. Database is updated simultaneously

### ***12. Owner Removes a Manager / Economist***

- a. Owner goes to Manage page.
- b. Owner clicks Delete on a manager or economist table entry
- c. A message showing a warning is displayed as a dialog box
- d. If owner chooses yes, the entry is erased from the database

### ***13. Owner Views Dashboard***

- a. After login Owner is redirected to owner homepage showing the Dashboard
- b. Owner can choose between Crops, Livestock, Market, Vets, Weather and Economic Report
- c. Owner has only read-only rights

### ***14. Owner Edits Farm Information***

- a. Owner clicks Edit Farm Info in dropdown menu

## ***Farm Management System Requirements Specification***

- b. A page where all information that can be editable by owner in the farm is shown
- c. Owner makes changes and clicks Save
- d. A message is shown asking for conformation
- e. Database is updated
- f. New info is shown

### **15. Owner Logs Out**

- a. Owner logs out from his account

## **APPENDIX**

The appendixes are not always considered part of the actual Requirements Specification and are not always necessary. They may include

- Sample input/output formats, descriptions of cost analysis studies, or results of user surveys;
- Supporting or background information that can help the readers of the Requirements Specification;
- A description of the problems to be solved by the system;
- Special packaging instructions for the code and the media to meet security, export, initial loading, or other requirements.

When appendixes are included, the Requirements Specification should explicitly state whether or not the appendixes are to be considered part of the requirements.

### **Appendix A. Definitions, Acronyms, and Abbreviations**

Define all terms, acronyms, and abbreviations used in this document.

### **Appendix B. References**

List all the documents and other materials referenced in this document.

### **Appendix C. Requirements Traceability Matrix**

The following trace matrix examples show one possible use of naming standards for deliverables (FunctionalArea-DocType-NN). The number has no other meaning than to keep the documents unique. For example, the Bargaining Unit Assignment Process Flow would be BUA-PF-01.

For example (1):

<b>Business Requirement</b>	<b>Area</b>	<b>Deliverables</b>	<b>Status</b>
BR_LR_01 The system should validate the relationship between Bargaining Unit/Location and Job Class.---Comments: Business Process = "Assigning a Bargaining Unit to an Appointment" (Priority 1)	BUA	BUA-CD-01 Assign BU Conceptual Design	Accepted
		BUA-PF-01 Derive Bargaining Unit-Process Flow Diagram	Accepted
		BUA-PF-01 Derive Bargaining Unit-Process Flow Diagram	Accepted
BR_LR_09 The system should provide the capability for the Labor Relations Office to maintain the job class/union relationship.---Comments: Business Process = "Maintenance" (Priority 1)	BUA	BUA-CD-01 Assign BU Conceptual Design	Accepted
		BUA-PF-02 BU Assignment Rules Maint Process Flow Diagram	ReadyForReview

For example (2):

### Farm Management System Requirements Specification

BizReqID	Pri	Major Area	DevTstItems DelivID	Deliv Name	Status
BR_LR_01	1	BUA	BUA-CD-01	Assign BU Conceptual Design	Accepted
BR_LR_01	1	BUA	BUA-DS-02	Bargaining Unit Assignment DB Modification Description	Accepted
BR_LR_01	1	BUA	BUA-PF-01	Derive Bargaining Unit-Process Flow Diagram	Accepted
BR_LR_01	1	BUA	BUA-UCD-01	BU Assign LR UseCase Diagram	ReadyForReview
BR_LR_01	1	BUA	BUA-UCT-001	BU Assignment by PC UseCase - Add Appointment and Derive UBU	Reviewed
BR_LR_01	1	BUA	BUA-UCT-002	BU Assignment by PC UseCase - Add Appointment (UBU Not Found)	Reviewed
BR_LR_01	1	BUA	BUA-UCT-006	BU Assignment by PC UseCase - Modify Appointment (Removed UBU)	Reviewed
BR_LR_09	1	BUA	BUA-CD-01	Assign BU Conceptual Design	Accepted
BR_LR_09	1	BUA	BUA-DS-02	Bargaining Unit Assignment DB Modification Description	Accepted
BR_LR_09	1	BUA	BUA-PF-02	BU Assignment Rules Maint Process Flow Diagram	Accepted
BR_LR_09	1	BUA	BUA-UCD-03	BU Assign Rules Maint UseCase Diagram	Reviewed
BR_LR_09	1	BUA	BUA-UCT-045	BU Assignment Rules Maint: Successfully Add New Assignment Rule	Reviewed
BR_LR_09	1	BUA	BUA-UCT-051	BU Assignment Rules MaintUseCase: Modify Rule	Reviewed
BR_LR_09	1	BUA	BUA-UCT-053	BU Assignment Rules MaintUseCase - Review Assignment Rules	Reviewed
BR_LR_09	1	BUA	BUA-UCT-057	BU Assignment Rules MaintUseCase: Inactivate Last Rule for a BU	Reviewed
BR_LR_09	1	BUA	BUA-UI-02	BU AssignRules Maint UI Mockups	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-021	BU Assignment Rules Maint TestCase: Add New Rule (Associated Job Class Does Not Exist) - Success	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-027	BU Assignment Rules Maint TestCase: Modify Rule - Success	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-035	BU Assignment Rules Maint TestCase: Add New Rule (Associated Job Class Does Not Exist) - Error Condition	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-049	BU Assignment Rules Maint TestCase: Modify Rule - Error Condition	ReadyForReview

For example (3):

BizReqID	CD01	CD02	CD03	CD04	UI01	UI02	UCT01	UCT02	UCT03	TC01	TC02	TC03	TC04
BR_LR_01			X		X		X			X		X	



***Farm Management System Requirements Specification***

BizReqID	CD01	CD02	CD03	CD04	UI01	UI02	UCT01	UCT02	UCT03	TC01	TC02	TC03	TC04
BR_LR_09	X			X		X			X		X		X
BR_LR_10	X			X					X		X		
BR_LR_11		X											

## **Appendix D. Organizing the Requirements**

This section is for information only as an aid in preparing the requirements document.

Detailed requirements tend to be extensive. Give careful consideration to your organization scheme. Some examples of organization schemes are described below:

### **By System Mode**

Some systems behave quite differently depending on the mode of operation. For example, a control system may have different sets of functions depending on its mode: training, normal, or emergency.

### **By User Class**

Some systems provide different sets of functions to different classes of users. For example, an elevator control system presents different capabilities to passengers, maintenance workers, and fire fighters.

### **By Objects**

Objects are real-world entities that have a counterpart within the system. For example, in a patient monitoring system, objects include patients, sensors, nurses, rooms, physicians, medicines, etc. Associated with each object is a set of attributes (of that object) and functions (performed by that object). These functions are also called services, methods, or processes. Note that sets of objects may share attributes and services. These are grouped together as classes.

### **By Feature**

A feature is an externally desired service by the system that may require a sequence of inputs to affect the desired result. For example, in a telephone system, features include local call, call forwarding, and conference call. Each feature is generally described in a sequence of stimulus-response pairs, and may include validity checks on inputs, exact sequencing of operations, responses to abnormal situations, including error handling and recovery, effects of parameters, relationships of inputs to outputs, including input/output sequences and formulas for input to output.

### **By Stimulus**

Some systems can be best organized by describing their functions in terms of stimuli. For example, the functions of an automatic aircraft landing system may be organized into sections for loss of power, wind shear, sudden change in roll, vertical velocity excessive, etc.

### **By Response**

Some systems can be best organized by describing all the functions in support of the generation of a response. For example, the functions of a personnel system may be organized into sections corresponding to all functions associated with generating paychecks, all functions associated with generating a current list of employees, etc.

### **By Functional Hierarchy**

When none of the above organizational schemes prove helpful, the overall functionality can be organized into a hierarchy of functions organized by common inputs, common outputs, or common internal data access. Data flow diagrams and data dictionaries can be used to show the relationships between and among the functions and data.

### **Additional Comments**

Whenever a new Requirements Specification is contemplated, more than one of the organizational techniques given above may be appropriate. In such cases, organize the specific requirements for multiple hierarchies tailored to the specific needs of the system under specification.

There are many notations, methods, and automated support tools available to aid in the documentation of requirements. For the most part, their usefulness is a function of organization. For example, when organizing by mode, finite state machines or state charts may prove helpful; when organizing by object, object-oriented analysis may prove helpful; when organizing by feature, stimulus-response sequences may prove helpful; and when organizing by functional hierarchy, data flow diagrams and data dictionaries may prove helpful.