# ElectroSpaceX Requirements Specification

## Version 2.0

## April 22, 2022

Use this Requirements Specification template to document the requirements for your product or service, including priority and approval. Tailor the specification to suit your project, organizing the applicable sections in a way that works best, and use the checklist to record the decisions about what is applicable and what isn't.

The format of the requirements depends on what works best for your project.

This document contains instructions and examples which are for the benefit of the person writing the document and should be removed before the document is finalized.

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

# Table of Contents

# 1. Executive Summary

## *1.1 Project Overview*

Describe this project or product and its intended audience, or provide a link or reference to the project charter.

The project consists of a web application, which resembles an e-commerce and management system of an electronic business. The web application should be user-friendly, convenient and will contain at least three users, such as the client, economist and administrator.

The services the business provides are mainly focused on the sale of electronic means and their successful delivery. In addition to the services mentioned above, the web application will enable users to give their feedback upon different products. Each user will register and login using its credentials, and then will be able to perform different orders, view its historic of purchases and provide feedbacks. On the other hand, the economist will be responsible of the financial states and the company's profits. The administrator can have much more access than a simple user or economist. He can keep track of the purchases made, be aware of the stock of the products, cancel orders and also view specific client data. He is responsible of taking care of different problems or dissatisfaction users might have.

The technologies we are going to use, include PHP programming on the back-end and HTML, CSS, JavaScript, Bootstrap on the front-end, and MySql as a database.

## *1.2 Purpose and Scope of this Specification*

Describe the purpose of this specification and its intended audience. Include a description of what is within the scope what is outside of the scope of these specifications.

The purpose of this project is to provide a proper management system of an online electronic store, which currently operates in Albania.

In scope of these requirements:

- Approaching the system as one of the three users, each of them having distinct responsibilities.
- Being in track of the services provided, as a result the business will be aware of the client feedback and manage decisions related to the improvement of its products
- Managing the supply of the electronic products and its trading.

Out of scope of these requirements:

- Delivery of the products to its final destination requires a separate management system.

# 2. Product/Service Description

In this section, describe the general factors that affect the product and its requirements. This section should contain background information, not state specific requirements (provide the reasons why certain specific requirements are later specified).

## *2.1 Product Context*

How does this product relate to other products? Is it independent and self-contained? Does it interface with a variety of related systems? Describe these relationships or use a diagram to show the major components of the larger system, interconnections, and external interfaces.

### *2.2 User Characteristics*

Create general customer profiles for each type of user who will be using the product. Profiles should include:

- Student/faculty/staff/other
- experience
- technical expertise
- other general characteristics that may influence the product

### *2.3 Assumptions*

List any assumptions that affect the requirements, for example, equipment availability, user expertise, etc. For example, a specific operating system is assumed to be available; if the operating system is not available, the Requirements Specification would then have to change accordingly.

### *2.4 Constraints*

Describe any items that will constrain the design options, including

- parallel operation with an old system
- audit functions (audit trail, log files, etc.)
- access, management and security
- criticality of the application
- system resource constraints (e.g., limits on disk space or other hardware limitations)
- other design constraints (e.g., design or other standards, such as programming language or framework)

### *2.5 Dependencies*

List dependencies that affect the requirements. Examples:

- This new product will require a daily download of data from X,
- Module X needs to be completed before this module can be built.

# 3. Requirements

- Describe all system requirements in enough detail for designers to design a system satisfying the requirements and testers to verify that the system satisfies requirements.
- Organize these requirements in a way that works best for your project. See Appendix DAppendix D, Organizing the Requirements for different ways to organize these requirements.
- Describe every input into the system, every output from the system, and every function performed by the system in response to an input or in support of an output. (Specify what functions are to be performed on what data to produce what results at what location for whom.)
- Each requirement should be numbered (or uniquely identifiable) and prioritized.
  See the sample requirements in Functional Requirements, and , as well as these example priority definitions:

  **Priority Definitions**

  The following definitions are intended as a guideline to prioritize requirements.

  - Priority 1 – The requirement is a “must have” as outlined by policy/law
  - Priority 2 – The requirement is needed for improved processing, and the fulfillment of the requirement will create immediate benefits
  - Priority 3 – The requirement is a “nice to have” which may include new functionality

  It may be helpful to phrase the requirement in terms of its priority, e.g., "The value of the employee status sent to DIS **must be** either A or I" or "It **would be nice** if the application warned the user that

the expiration date was 3 business days away". Another approach would be to group requirements by priority category.

- A good requirement is:
  - Correct
  - Unambiguous (all statements have exactly one interpretation)
  - Complete (where TBDs are absolutely necessary, document why the information is unknown, who is responsible for resolution, and the deadline)
  - Consistent
  - Ranked for importance and/or stability
  - Verifiable (avoid soft descriptions like “works well”, “is user friendly”; use concrete terms and specify measurable quantities)
  - Modifiable (evolve the Requirements Specification only via a formal change process, preserving a complete audit trail of changes)
  - Does not specify any particular design
  - Traceable (cross-reference with source documents and spawned documents).

## *3.1 Functional Requirements*

In the example below, the requirement numbering has a scheme - BR_LR_0## (BR for Business Requirement, LR for Labor Relations). For small projects simply BR-## would suffice. Keep in mind that if no prefix is used, the traceability matrix may be difficult to create (e.g., no differentiation between '02' as a business requirement vs. a test case)

The following table is an example format for requirements. Choose whatever format works best for your project.

For Example:

| Req# | Requirement | Comments | Priority | Date Rvwd | SME Reviewed / Approved |
|---|---|---|---|---|---|
| BR_01 | The system is designed as a web application with three users | This will be the main platform for users | 1 | 4/14/22 | Gejsi Dervishi, Kevin Mamaj |
| BR_02 | The system's service will be based on online sales and purchases, as well as system management. | | 2 | 4/14/22 | Gejsi Dervishi, Kevin Mamaj |
| BR_03 | All privileges will be granted to the system administrator. Will get real-time access to all system and database options for clients and economists. | The admin account has full access to the system's features. | 1 | 4/14/22 | Gejsi Dervishi, Kevin Mamaj |
| BR_04 | The application must have a specific UI (user interface) for client. | | 1 | 4/14/22 | Gejsi Dervishi, Kevin Mamaj |

| Req# | Requirement | Comments | Priority | Date Rvwd | SME Reviewed / Approved |
|---|---|---|---|---|---|
| BR_05 | Every user should be able to log in and out and every client has the opportunity to register into the application | Economist and administrator are already in the database | 1 | 4/14/22 | Gejsi Dervishi, Kevin Mamaj |
| BR_06 | Every user should have access to their own section of the system. | | 1 | 4/14/22 | Gejsi Dervishi, Kevin Mamaj |
| BR_07 | The system should have profiles for each user. | | 1 | 4/14/22 | Gejsi Dervishi, Kevin Mamaj |
| BR_08 | The administrator must have the most privileges of all users. | | 1 | 4/14/22 | Gejsi Dervishi, Kevin Mamaj |
| BR_09 | Client profiles should be private | Data Protection | 1 | 4/14/22 | Gejsi Dervishi, Kevin Mamaj |
| BR_10 | The administrator must have access to CRUD operations (Create, Read, Update, Delete) categories, products, orders, users, employees. | | 1 | 4/14/22 | Gejsi Dervishi, Kevin Mamaj |
| BR_11 | The data is presented to the administrator in a table format. | This is done because tables and visuals provide a clearer picture of all the clients who have registered. | 2 | 4/14/22 | Gejsi Dervishi, Kevin Mamaj |
| BR_12 | Administrator should be able to read all orders and decide if the order is ready to be delivered or not. | | 2 | 4/14/22 | Gejsi Dervishi, Kevin Mamaj |
| BR_13 | A supply balance report should be able to be filed by administrator, and he can choose the amount of products that are needed. | | 2 | 4/14/22 | Gejsi Dervishi, Kevin Mamaj |
| BR_14 | The admin should not be able to see any of the user's personal information. | | 1 | 4/14/22 | Gejsi Dervishi, Kevin Mamaj |

| Req# | Requirement | Comments | Priority | Date Rvwd | SME Reviewed / Approved |
|---|---|---|---|---|---|
| BR_15 | The economist's report/balance should be available to the administrator at the end of each month. | | 1 | 4/14/22 | Gejsi Dervishi, Kevin Mamaj |
| BR_16 | The economist should do a monthly report on the profit or loss of the company | | 2 | 4/14/22 | Gejsi Dervishi, Kevin Mamaj |
| BR_17 | The system should have a calculation of total monthly, weekly and daily incomes and the economist has the right to `study them. | | 2 | 4/14/22 | Gejsi Dervishi, Kevin Mamaj |
| BR_18 | At the moment when the stock is from 0-3 products in the administrator page will be shown a warning in order to send an email to the supplier. | The email will have the required details for the product by the supplier. | 3 | 4/14/22 | Gejsi Dervishi, Kevin Mamaj |
| BR_19 | When a product needs to be shipped, the economist will print a receipt with buyer details and give it to the transporter | | 3 | 4/14/22 | Gejsi Dervishi, Kevin Mamaj |

## *3.2 Non-Functional Requirements*

### 3.2.1 Product Requirements

Our software will be based as a web application, consisting of three user interfaces.
The client interface, the economist interface and the administrator interface will appear corresponding to the login credentials.
The header of the website contains the logo of the business, a search bar and the menu with various options, such as products, news, services, about us and the log-in button. While the footer contains social media contact details, address details and privacy policy information.

- Client Interface:
  -When the client presses the log-in button then he will be directed to log-in window, with an extra option to register, if not yet.
  -The client interface allows the user to make ***real-time purchases,*** by adding products to the cart or deleting existing ones.
  -The client can view its ***profile details*** and also edit its information. The information includes the name, surname, the phone number, the address, the country name and the postal number.
  -The client can add products on the wish list, and also delete them.
  -View the ***purchase historic*** and also the products on the ***wish list***.
  -The client can search a product or view recommended items.

-The user can post feedback as a comment or represented as star review.
-Read flash news about technology and products on the website.
-***Log-out*** option included, in order to log out of the website.

- Admin Interface:
  -The admin logs in the website with its credentials, and owns various responsibilities.
  -Performing ***CRUD operations*** related to creating, reading, updating and deleting products, categories, economists and other workers.
  -List all the users.
  -Adding product quantity.
  -Set sales for specific products on the website.
  -Checking ***product availability***.
  -Being able ***to view orders*** and invoices.
  -***Requesting supplies*** which will be visible on the client interface.
  -View reports generated by the economist.
  -View the feedback posted by the client upon different products.
  -Log out of the site.

- Economist Interface:
  -Logging in as an economist.
  -Viewing ***client purchase invoices***.
  -Viewing total expenses.
  -Calculating the ***daily & monthly profit***.
  -***Generate reports*** related to the financial state of the business throughout the month.
  -View supply needs and its required budget.
  -Add workers and define their wages.
  -Log-out option, to leave the website.

#### 3.2.1.1 Usability Requirements

Include any specific usability requirements, for example,

Learnability Requirements

- The application is user-friendly, simple and easy to understand.
- An extensive document will be provided in order to instruct common users about the usage of the system.
- This step by step instruction will be separated into modules, related to the interfaces the system owns.
- The system is easy to remember for common users.

Flexibility Requirements

- The software is designed to minimize as much as possible user errors.
- It can be updated easily in order to meet new requirements.

Accessibility Requirements

- The software can be conveniently accessed in various times and environments, since it requires only a device and Internet connection to be used.

#### 3.2.1.2 Efficiency Requirements

The software will provide a fast and reliable way of managing an electronic shop operating in online sales.

Performance Requirements

The software is a web application, as a result it will be stored in a web server.

- The performance depends on the speed and the quality of the Internet connection.
- Its efficiency is highly related to the server hardware capability.
- Number of users accessing the website effects the performance.

Space Requirements

Since the business serves as an e-commerce website, there will be a great number of users predicted to access the website and perform purchases or different tasks. It is thought that several hundred clients can interface with the application. As for the staff it has a limited number of users and does not have a major effect in space.

#### 3.2.1.3 Dependability Requirements

Availability

- The web application will be available for use all the time.
- The web application can be accessed in any geographical area, as long as the user has a stable Internet connection.
- Error message warnings will be provided in order to prevent system fatal error.
- Different user categories can access specific parts of the web application.
- The system should behave as expected, with very few errors.
- Downtime can affect the user experience, as a result an immediate maintenance should be provided. It should be avoided as much as possible during certain times when the application is highly frequented.
- Scheduled maintenance can be performed in times when the store itself is closed, or at times when the website bears as little traffic as possible.

Latency

Latency is the delay between a user's action and the response of the web application, referred as the round trip time it takes a packet of data to travel through the network.

- Connection based latency is not covered by the software, since it is related to the speed and quality of the Internet connection.
- The latency of the web application depends mainly on the strength of the Internet connection, the size of the database etc.
- The linking between interfaces should be optimal, in order to avoid latency.

#### 3.2.1.4 Security Requirements

Security is a property that defines the ability of the system to protect itself from attacks. When it comes to security, the software should provide:

- Confidentiality: Access should be granted only to authorized users.
- Integrity: Only the authorized user can modify the data depending on its role.
- Availability: The data in the system is accessible all the time, online.

3.2.1.4.1 Protection

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

- Each user is required to log in using its credentials.
- User input filtering is provided, in case the user enters incorrect data.
- The encryption of customer personal data, as well as inventories, company financial information etc.
- SQL filtering, in case of SQL injections.
- Refreshing the session, in order to prevent Session ID hijacking attempts.
- Data integrity is represented by a series of integrity constraints used in the database system. These constraints include entity integrity, which states that the table must have a primary key, defined to be unique. Referential integrity which states the concept of a foreign key. The foreign key of one table refers to the primary key of another. Domain Integrity restricts the attributes or values of a column upon a defined domain. User-defined integrity which are specified according to the needs of specific users.

3.2.1.4.2 Authorization and Authentication

Authorization:

- The process of giving permission to a user to access their information or website.
- Users of a specific role should be able to view a particular part of the web application.
- Authorization works through settings that are implemented and maintained by the organization.
- It is combined with authentication to give a brief information to the server of the client that is requesting access.

Authentication:

- The process of determining the exact information of a user, by comparing the user's credential with the ones in the database of authorized user or in the data authentication server.
- The user has to prove its identity to the server.
- Identifies who the person or the system that intends to access the software is.
- The server can authenticate users by the input of username and password.

### 3.2.2 Organizational Requirements

Requirements that are a consequence of organizational policies and procedures.

#### 3.2.2.1 Environmental Requirements

- Users should have access to Internet connection, in order to use the web application and perform various operations.
- Users should be able to access the system by entering their credentials and perform tasks related to their role. If the user is a client, he can purchase items, put products on the wish list, view their purchases etc.
- The users induce changes into the database, by performing operations, such as purchases, modifications of personal information etc. For example the administrator can add/delete products, categories and users. Thus the database is modified.

#### 3.2.2.2 Operational Requirements

- Users should firstly access the web application by entering the correct credentials. As a result a user can approach the website as a client, an economist or an administrator, depending on their role.
- Users that are authorized by the system can proceed with other operations.
- The administrator is informed of the purchases, the clients that access the website, the stock of the products and the company's finances and takes actions according to these reports.

#### 3.2.2.3 Development Requirements

- Communicating with the business representatives, in order to have a preview of their requirements and services the business will provide. This phase consists in gathering information and analysis of the project.
- Making a design of how the product will resemble on its final state.
- Starting the implementation by creating the ER diagrams and then the database of the system.
- Building the web application step by step using front-end and back-end technologies.
- Creating each interface that has a different target group of users and linking them with each other.
- Testing the product in a group of users and checking for minor errors, which need improvements. Final product is then launched.

### 3.2.3 External Requirements

Requirements that arise from factors which are external to the system and its development process.

#### 3.2.3.1 Regulatory Requirements

3.2.3.1.1 Network and Hardware Interfaces

- Persistent TCP connection is required to enable our web application to exchange messages over the network.
- Operating system is not as important, as long as it supports web browsers.
- A server should be up all the time to host the system, except on cases of event maintenance.

3.2.3.1.2 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 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)
- XYG Number
- Element Code
- Amount
- Amount Sign
- Year Flag
- Total Amount
- Total Amt Sign

### 3.2.3.2 Ethical Requirements

The web application cannot make decisions on its own in regard to:

- Add or remove a user from the system.
- Modify specific categories of the database.
- Share personal information of the users.
- Make purchases of the products without the approval of the client user.

The software and its services are not in use for anyone under the age of 18. The website does not collect information from anyone under the age of 18, or allow anyone under that age to use the services provided.

### 3.2.3.3 Legislative Requirements

Data Protection
Enshrined in Article 35 of the Constitution of the Republic of Albania (only available in Albanian here) ('the Constitution'), the protection of personal data constitutes a fundamental right.
User personal information should be confidential and private. Data integrity is quite important, and the safety of the users has to be aimed. The system should be safe and as a result the user experience will be pleasant.

ElectroSpaceX website collects immediately the customers' personal data when it is required to sign up for buying online. It protects all the personal data from its customers and only use them for contractual and informative purposes. Using the website, the customer has to give his confirmation for registering himself as a customer, for using his personal data for further procedures and to guarantee that the given information is correct, true and updated.

### 3.2.4 Manageability/Maintainability

#### 3.2.4.1 Monitoring

- The application user interface will be easy to comprehend and it will not be prone to system crashes.
- The log-in interface needs a username and password as input, if not then the client user needs to register and have an active account.
- The user will enter the system in case he has typed the correct credentials, if not an error message will appear. Filtering the data is required for this step.

#### 3.2.4.2 Maintenance

- The software will be developed using PHP scripting language as a back-end technology. As front-end technologies HTML, CSS and JavaScript will be used. The database will be built using MySQL.
- PHP is quite flexible, and enables the developer to make changes even after starting a project. It is compatible with the majority of operating systems, as a result it can easily run on different platforms.
- The web application will be created in a way that new features can be added in the future.

#### 3.2.4.3 Operations

Some operations that can be taken by the users:

- Users can log-in using their credentials.
- Client user can purchase a product.
- Client user can update their profile.
- Client user can post a feedback about a specific product.
- CRUD functionality for admin.
- Admin can add and remove other users.
- Admin can send a request when a product is low in stock.
- Economist can check the financial state of the company.

### 3.2.5 Data Management

The database will contain some of these tables and attributes:

- User(id, username, name, surname, email, phone, password, created_at, active)
- Role(role_id, role_desc, user_id)
- Product(product_id, product_name, product_desc, category_id, category_photo, product_status, product_price, product_discount, quantity, brand_id, date_created)
- Cart(id, product_id, brand_id, user_id, quantity, adress_id)

### 3.2.6 Standards Compliance

The web application is developed in such a way that it will respect the legislative laws related to the user data safety and guarantee a reliable service on the e-commerce field of electric means.

### 3.2.7 Portability

Portability does not affect the maintenance of our software. Since it is a web application, it can be accessed anywhere, as long as the user has a stable Internet connection and an electronic device, such as a smartphone, computer or laptop.

### 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*

- The ElectroSpaceX is a web application which covers three categories of users, such as economists, clients and admin.
- It provides different services, from the online trading of products to the management of an electronic store.
- Providing a menu of products, which can be modified by the admin.
- The system data is safe, considering the fact that the users need to be authenticated and authorized to enter their profile.

# 4. User Scenarios/Use Cases

Provide a summary of the major functions that the product will perform. Organize the functions to be understandable to the customer or a first time reader. Include use cases and business scenarios, or provide a link to a separate document (or documents). A business scenario:

- Describes a significant business need
- Identifies, documents, and ranks the problem that is driving the scenario
- Describes the business and technical environment that will resolve the problem
- States the desired objectives
- Shows the “Actors” and where they fit in the business model
- Is specific, and measurable, and uses clear metrics for success

# 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):

| Business Requirement | Area | Deliverables | Status |
|---|---|---|---|
| BR_LR_01<br>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<br>Assign BU Conceptual Design | Accepted |
| | | BUA-PF-01<br>Derive Bargaining Unit-Process Flow Diagram | Accepted |
| | | BUA-PF-01<br>Derive Bargaining Unit-Process Flow Diagram | Accepted |
| BR_LR_09<br>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<br>Assign BU Conceptual Design | Accepted |
| | | BUA-PF-02<br>BU Assignment Rules Maint Process Flow Diagram | ReadyForReview |

For example (2):

| 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 |

| BizReqID | Pri | Major Area | DevTstItems DelivID | Deliv Name | Status |
|---|---|---|---|---|---|
| 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 | |
| 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.