Bookstore Requirements Specification

Version 2.5

April 11, 2023

**Table of Contents**

[**BOOKSTORE REQUIREMENTS SPECIFICATION 1**](#_heading=h.19c6y18)

[**VERSION 1.0 1**](#_heading=h.3tbugp1)

[**APRIL 10, 2023**](#_heading=h.28h4qwu)

[**1.**](#_heading=h.nmf14n) **EXECUTIVE SUMMARY 3**

[1.1](#_heading=h.37m2jsg) Project Overview 3

[1.2](#_heading=h.1mrcu09) Purpose and Scope of this Specification 3

[**2.**](#_heading=h.46r0co2) **PRODUCT/SERVICE DESCRIPTION 3**

[2.1](#_heading=h.2lwamvv) Product Context 3

[2.2](#_heading=h.111kx3o) User Characteristics 3

[2.3](#_heading=h.3l18frh) Assumptions 3

[2.4](#_heading=h.206ipza) Constraints 3

[2.5](#_heading=h.4k668n3) Dependencies 4

[**3.**](#_heading=h.2zbgiuw) **REQUIREMENTS 4**

[3.1](#_heading=h.1egqt2p) Functional Requirements 5

[3.2](#_heading=h.3ygebqi) Non-Functional Requirements 5

[*3.2.1*](#_heading=h.2dlolyb) *Product Requirements 5*

[**3.2.1.1**](#_heading=h.44sinio) **User Interface Requirements** 6

[**3.2.1.2**](#_heading=h.2jxsxqh) **Usability** 6

[**3.2.1.3**](#_heading=h.z337ya) **Efficiency** 6

[3.2.1.3.1](#_heading=h.3j2qqm3) Performance Requirements 6

[3.2.1.3.2](#_heading=h.1y810tw) Space Requirements 6

[**3.2.1.4**](#_heading=h.4i7ojhp) **Dependability** 6

[**3.2.1.5**](#_heading=h.2xcytpi) **Security** 7

[*3.2.2*](#_heading=h.sqyw64) *Organizational Requirements 7*

[**3.2.2.1**](#_heading=h.3whwml4) **Environmental Requirements** 7

[**3.2.2.2**](#_heading=h.2bn6wsx) **Operational Requirements** 7

[**3.2.2.3**](#_heading=h.qsh70q) **Development Requirements** 7

[*3.2.3*](#_heading=h.3cqmetx) *External Requirements 7*

[**3.2.3.1**](#_heading=h.1pxezwc) **Regulatory Requirements** 7

[**3.2.3.2**](#_heading=h.49x2ik5) **Ethical Requirements** 7

[**3.2.3.3**](#_heading=h.2p2csry) **Legislative Requirements** 7

[3.2.3.3.1](#_heading=h.147n2zr) Accounting Requirements 7

[3.2.3.3.2](#_heading=h.3o7alnk) Security Requirements 7

[3.3](#_heading=h.1rvwp1q) Domain Requirements 7

[**4.**](#_heading=h.4bvk7pj) **USER SCENARIOS/USE CASES 7**

# Executive Summary

## Project Overview

The focus of this software will be to manage all important steps of a bookstore. The software can keep data about all the books in the bookstore, such us ISBN of the book, title, category of the book, selling price, author, stock. At the same time, it keeps track of the sold products by bill number, their sold quantities, prices, and date of transaction.

The application will have a three-level user system: Librarian, Manager and Administrator. Each will have different views and usage of the software. Each of the users has a username and a password to enter in the software.

**Librarian** -> Has the right to check out books that a customer may need from the bookstore. The librarian creates a bill and enters the data of the bought books, such as ISBN of the book and its quantity. If the book is out of stock or does not exist, the librarian is alerted. The software provides them the total amount of the bill, which is in printable format. The librarian can also take book requests from the store’s clients.

**Manager** ->The manager has the right to supply the bookstore with the needed books. So, they can enter in the stock the new book category, and/or add books of the same category to the stock of the bookstore. The manager is also informed if there are few items of a book in the bookstore stock. The manager can also check the performance of the librarians by checking their total number of bills, books sold, and the total amount of money made for a certain date or between a certain period of time.

**Administrator** ->The administrator has the right to manage almost everything that librarian and manager does. Beside them they have the right to manage the employees (librarian and manager), by registering, modifying, and deleting them. The administrator keeps info about employees such as: name, birthday, phone, email, salary, access level. The software also provides statistics about books sold the same way it does for managers.

## Purpose and Scope of this Specification

The purpose of this software specification is to define the scope of the project and clarify which requirements will be addressed in the development of the software. The scope of this specification includes defining the features and functionalities of the software, as well as outlining the three-level user system (Librarian, Manager, and Administrator) and the specific tasks that each user is able to perform.

In Scope:

* Definition of the functional and non-functional requirements for the Bookstore Management Software
* Identification of the user roles and associated permissions for the software
* Definition of user scenarios and use cases to illustrate the software's intended usage
* Documentation of regulatory, ethical, and legislative requirements related to the software
* Identification of any constraints or dependencies that may affect the development or implementation of the software
* Identification of assumptions made during the requirements gathering process
* Identification of any organizational, environmental, operational, or development requirements that may affect the software's implementation or use

Out of Scope:

* Detailed technical specifications or implementation details for the software
* User interface design or graphical elements of the software
* Hardware or infrastructure requirements for the software
* Marketing or branding strategies for the software
* Customization or configuration of the software for specific clients or use cases

# Product/Service Description

The main focus of this software is to help librarians and manager exchange data in order to keep the transactions and stocks organized and transparent at all times. The manager will first supply the bookstore with new books and enter their information in the bookstore stock organized by their genres. Afterward, the librarian can search for a certain book title and check its stock status and then perform the sales transactions.

The librarian will be easily informed if the book is out of stock or not found at all. Simultaneously the manager will be alerted of what books are running out of stock and supply the bookstore with the needed books. Based on how many books are sold overall by each librarian, the manager can determine their performance and check the most demanded books by the public and what categories are preferred.

## Product Context

The Bookstore Management Software is a mostly independent and self-contained product designed to manage the operations of a bookstore. It does not interface with other systems or products, and is designed to be a standalone solution for bookstores.

This software is a helpful tool for every bookstore to keep track of their transactions, measure employee productivity, form a revenue strategy and organize the overall stock efficiently. Our application only focuses on this institution as a for-profit business, meaning it will only be available to the employed staff of the bookstore, not open to the public.

Without disrupting the app’s effectiveness, its priority is to be easily operable by every employee regardless of their qualifications or knowledge of computer use. It also provides great data backup and organization, avoiding the use of on-paper documentation of numerous book ISBN (book serial numbers). This way data is reached faster and easier in case of a transaction.

To store this data, the app is operating on cloud storage, thought as the best option appropriate to the size of this business. This storage type assures cost-effectiveness, accessibility for authorized personnel, security and protection of data, and automatic backup/recovery of data.

## User Characteristics

Librarian:

* Checks out information about a book that a customer is interested in
* Creates a bill with the data of the book of interest
* Checks for book availability
* Gets notified if the book is out of stock or does not exist
* Prints the bill for the customer if the book is in stock
* Bills are stored to provide statistics regarding books sold whenever requested by manager
* Basic computer skills required to navigate GUI and print bills

Manager:

* Supplies the bookstore with the needed books
* Enters book information to the bookstore stock
* Categorizes books and groups them together accordingly
* Gets notified when there are fewer than 5 available copies per book in bookstore stock
* Checks librarian’s performance based on the:

1. Number of books sold
2. Amount of money made

All are reflected in the created bills over time

* Basic computer skills required

Administrator:

* Can manage everything that the employees can
* Manages the employees by registering, modifying, and deleting them
* Keeps data about the aforementioned such as name, birthday, phone number, salary, access level, etc.
* Receives data about the total income, meaning the total of books sold and the total cost, being the total of items bought and staff salaries in a given period
* Revoke’s permission access from employees
* Basic computer skills required

## Assumptions

* It is assumed that this software will be available only to the employed staff
* In order to use the app, it is crucial to have a good internet connection at all times
* It is assumed that the staff members understand English and know how a computer works
* It is expected that the computers have an operating system that supports Java
* The employees have to be equipped with the essential knowledge on how to use the app and, if required, update its data
* A user browses and/or validates, depending on his or her permissions
* The administrator will presumably have unrestricted access to all the relevant data of this bookstore
* User maintenance: adds, updates, and deletes system users and their associated information
* Book maintenance: adds, updates, and deletes books that will be used by the organization using the system

## Constraints

The purpose of constraints is to help prevent the bookstore app from becoming complex and confusing.

* Resource constraints are given by the relation between the demands for processing the data and the capacity of the computing machinery. The resources are runtime, memory, facilities, communication and energy. Hence, the system is optimized with regard to minimal resource consumption.
* The system is strictly used by the authorized individuals (administrator, manager and librarian). The software will need to have access control mechanisms to ensure that only authorized personnel can access and modify sensitive data. Management of user accounts and privileges will also need to be included. Additionally, the software will need to be designed with security best practices in mind to ensure that customer data is protected.
* The software will be a critical component of the bookstore's operations. Any downtime or issues with the software could result in lost sales and revenue. As such, the software will need to be designed with high availability and reliability in mind.
* The users must have an active internet connection in order for the app to work properly.

## Dependencies

* Integration with the store's existing database system to ensure seamless data transfer and avoid any discrepancies or data duplication.
* Availability of reliable and stable internet connection to ensure cloud storage is always accessible and up-to-date.
* Compatibility with the store's existing hardware such as computer devices or printers, that may be required to use the software.
* Completion of any necessary training or educational programs for the store employees to learn how to use the software effectively.
* Timely delivery of hardware, software, and other components required to implement the software in a timely and efficient manner.
* Compliance with any regulatory or legal requirements, such as data protection laws and regulations, and software licensing agreements.
* The software requires Java and JavaFX to be installed on the user's system in order to run properly.

# 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 System Interface/Integration, 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).

## Functional Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. Req# | Requirement | Comments | Priority | Data reviewed | SME reviewed/  approved |
| R\_01 | Users should be able to log in with a username and password. | This ensure secure access to the system. | 1 | 13.05.2023 | Elisa Gjuraj  Nevina Dervishi |
| R\_02 | The app will offer different views for different user level(librarian,manager and administrator). | Each user will have a unique login with corresponding access level ,the app will provide different views. | 1 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_03 | The app shall allow librarians to check out books for customers . | To fulfill this requirement, the app should provide a checkout feature where librarians can select the books requested by customers and complete the checkout process. | 1 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_04 | The app shall allow librarians to create bills with book details and prices. | To fulfill this requirement, the app should provide a user-friendly interface where librarians can input the necessary book information, such as ISBN, title, author, and quantity. The system should then calculate the total price based on the entered book prices and quantitie | 1 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_05 | The app should alert the librarian when books are out of stock or unavailable for purchase. | Timely alerts help the librarian manage stock and inform customers about book availability. | 2 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_06 | The app shall provide a total amount for bills that is in printable format for librarians. | The bill should include all relevant information such as book details, prices and total amount due | 3 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_07 | The app shall allow librarians to take book requests from customers. | The librarian should be able to record the book request and contact the customer when the book becomes available. | 2 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_08 | The app shall allow managers to add ,update and delete book categories in the system. | Proper categorization helps organize books and improve search capabilities. | 1 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_09 | The manager should be able to update the stock quantity of books in the bookstore. | Accurate stock management ensures timely replenishment and prevents overselling or stockouts. | 1 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_10 | The app shall alert managers when the stock of a book is low. | The alert should be triggered when the stock of a book falls below a specified threshold. | 2 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_11 | The app shall allow managers to check the performance of librarians by checking their total number of bills, book sold, and total amount of money made for a certain date or between a certain period of time. | The manager should be able to filter the performance data by data or time period. | 3 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_12 | The app shall allow administrator to register,modify,and delete librarian and manager accounts. | Efficient employee management allows for proper role assignment and maintenance of employee records. | 1 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_13 | The app shall allow the administrator to store employee information such as name, birthday,  phone, email, salary, and access level. | The administrator should be able to view and edit the employee information at any time. | 1 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_14 | The app shall provide statistics about books sold for administrators in the same way it does for managers. | The administrators should be able to view book sales data by date or time period. | 3 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_15 | The app shall allow administrator to generate sales report for certain date or between a certain period of time. | The administrator should be able to generate sales reports for specific dates or periods of time, providing valuable information for decision-making purposes. | 2 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_16 | The app shall provide the ability for librarians and managers a search function to find books based on ISBN, title, category or author. | This requirements will provide librarians and managers the ability to search for books based on various criteria, helping them to locate books quickly and efficiently. | 1 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_17 | The app shall keep track of the total stock of each book and alter managers when the stock of a book falls below a certain threshold | This requirement will ensure that managers are alerted when the stock of a book falls below a certain threshold, allowing them to restock before running out of stock**.** | 2 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_18 | The app shall allow managers to update book details such as title, author, category and price. | The managers will be able to update book details such as title, author, category and price ensuring that the data is accurate and up-to-date. | 2 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_19 | The app shall allow administrator to set access level for each employee. | The administrator can set appropriate levels for each employee ,ensuring that employees only have access to the necessary features and data. | 1 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_20 | The app should generate reports on various aspects, such as sales, inventory and employee performance ,in printable formats. | Reports assist in monitoring bookstore operations and evaluating performance. | 2 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_21 | The app should allow the librarian to register and update customer information, including name, contact details, and purchase history. | The requirement emphasizes the importance of maintaining a purchase history for each customer. This means that the system should record and track the books that each customer has purchased in the past. This purchase history data can be useful for several purposes, including personalized services, such as recommending books based on the customer's preferences and purchase history. | 2 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_22 | The app should provide personalized book recommendations to customers based on their purchase history and preferences. | This requirements allows books to be recommended based on the customer's purchase history | 2 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_23 | The app should track sales by recording the bill number, sold quantities, prices and transaction dates. | This requirement enables accurate sales tracking is essential for effective inventory management, enabling bookstores to monitor stock levels, identify popular books, and make informed purchasing decisions. | 1 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_24 | The app shall allow managers to create and manage purchase orders for restocking books. | This requirement can be fulfilled by providing the manager with a user-friendly interface with the system. The system should allow the manager to generate and send purchase orders to suppliers electronically. | 2 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_25 | The app should store supplier information, including contact details and payment terms. | This requirement ensure effective supplier management by providing a centralized database for storing supplier information. | 2 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_26 | The system should send notifications and reminders to librarians managers, and administrators for pending tasks and low stock. | This requirement plays a vital role in keeping stakeholders informed and facilitating timely actions. By sending notifications and reminders, the system ensures that librarians, managers, and administrators stay updated on pending tasks. Additionally, notifications about low stock. | 2 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_27 | The system should allow the manager to create and manage discount,  promotions and special offers for books. | This requirement empowers the manager to effectively implement marketing strategies by offering discounts and promotions on selected books. | 2 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_28 | The system should provide an online platform for customers to place book orders and make payments. | This requirement aims to offer customers the convenience of ordering books online. By providing a user-friendly online platform, the system allows customers to browse and select books, place orders, and securely complete payments. | 1 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_29 | The app should enable customers to track the status of their online book orders. | This requirement focuses on providing customers with updates on their orders. By including order tracking, the system lets customers know where their books are during the shipping process. This increases customer satisfaction, reduces support inquiries, and builds trust. | 2 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_30 | The app should handle returns and refunds ,including tracking return requests, processing refunds and updating inventory accordingly. | This requirement emphasizes the importance of managing returns and refunds effectively. By having a system in place to handle customer returns, process refunds, and update inventory, the bookstore can provide a seamless experience for customers who need to return or exchange books. | 2 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_31 | This app shall allow customers to make payments using their credit or debit cards. | This requirements enables customers to complete transactions seamlessly, expanding their payment choices and enhancing overall user experience. | 1 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_32 | The app should enable customers to reserve books for specific duration before making a purchase. | This requirements can provide a feature where customers can select the desired book and choose a reservation duration . | 2 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_33 | The app should allow customers to provide reviews and ratings for books they have purchased. | The app should include a feature where customers can provide their feedback, write reviews, and rate the books they have purchased. | 2 | 13.05.2023 | E.Gjuraj  N.Dervishi |
| R\_34 | The app should provide a feature to identify and display the bestseller book(s). | This requirements enables book recommendations for librarians assist in book ordering and stocking decision. | 2 | 13.05.2023 | E.Gjuraj  N.Dervishi |

## Non-Functional Requirements

**In here try to use the Structure given at slide 13 in Requirements Engineering Lecture Slides, with main categories of:**

### Product Requirements

* + Requirements which specify that the delivered product must behave in a particular way e.g. execution speed, reliability, etc.

#### **User Interface Requirements**

In addition to functions required, describe the characteristics of each interface between the product and its users (e.g., required screen formats/organization, report layouts, menu structures, error and other messages, or function keys).

#### **Usability**

Include any specific usability requirements, for example,

Learnability

* The user documentation and help should be complete
* The help should be context sensitive and explain how to achieve common tasks
* The system should be easy to learn

(See <http://www.usabilitynet.org/>)

#### **Efficiency**

##### Performance Requirements

Specify static and dynamic numerical requirements placed on the system or on human interaction with the system:

* Static numerical requirements may include the number of terminals to be supported, the number of simultaneous users to be supported, and the amount and type of information to be handled.
* Dynamic numerical requirements may include the number of transactions and tasks and the amount of data to be processed within certain time period for both normal and peak workload conditions.

All of these requirements should be stated in measurable form. For example, "95% of the transactions shall be processed in less than 1 second" rather than “an operator shall not have to wait for the transaction to complete”.

##### Space Requirements

#### **Dependability**

**Availability**

Include specific and measurable requirements for:

* Hours of operation
* Level of availability required
* Coverage for geographic areas
* Impact of downtime on users and business operations
* Impact of scheduled and unscheduled maintenance on uptime and maintenance communications procedures
* reliability (e.g., acceptable mean time between failures (MTBF), or the maximum permitted number of failures per hour).

**Reliability**

**Monitoring**

Include any requirements for product or service health monitoring, failure conditions, error detection, logging, and correction.

**Maintenance**

Specify attributes of the system that relate to ease of maintenance. These requirements may relate to modularity, complexity, or interface design. Requirements should not be placed here simply because they are thought to be good design practices.

**Integrity**

#### **Security**

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

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

### Organizational Requirements

Requirements which are a consequence of organisational policies and procedures e.g. process standards used, implementation requirements, etc

#### **Environmental Requirements**

#### **Operational Requirements**

#### **Development Requirements**

### External Requirements

* + Requirements which arise from factors which are external to the system and its development process e.g. interoperability requirements, legislative requirements, etc.

#### **Regulatory Requirements**

#### **Ethical Requirements**

#### **Legislative Requirements**

Specify the requirements derived from existing standards, policies, regulations, or laws (e.g., report format, data naming, accounting procedures, audit tracing). For example, this could specify the requirement for software to trace processing activity. Such traces are needed for some applications to meet minimum regulatory or financial standards. An audit trace requirement may, for example, state that all changes to a payroll database must be recorded in a trace file with before and after values

##### Accounting Requirements

##### Security Requirements

## Domain Requirements

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

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

# 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.

1. **Definitions, Acronyms, and Abbreviations**

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

1. **References**

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

1. **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  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):

| **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 |  |
| BR\_LR\_09 | X |  |  | X |  | X |  |  | X |  | X |  | X |
| BR\_LR\_10 | X |  |  | X |  |  |  |  | X |  | X |  |  |
| BR\_LR\_11 |  | X |  |  |  |  |  |  |  |  |  |  |  |

1. **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.