[YourProject] Requirements Specification

Version 1.0

April 19, 2021

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**

[[YOURPROJECT] REQUIREMENTS SPECIFICATION 1](#bookmark)

[VERSION 1.0 1](#bookmark1)

[APRIL 19, 2021 1](#bookmark2)

[1.](#bookmark3)EXECUTIVE SUMMARY 3

[1.1](#bookmark5)Project Overview 3

[1.2](#bookmark7)Purpose and Scope of this Specification 3

[2.](#bookmark9)PRODUCT/SERVICE DESCRIPTION 3

[2.1](#bookmark11)Product Context 3

[2.2](#bookmark13)User Characteristics 3

[2.3](#bookmark15)Assumptions 3

[2.4](#bookmark17)Constraints 3

[2.5](#bookmark19)Dependencies 4

[3.](#bookmark21)REQUIREMENTS 4

[3.1](#bookmark23)Functional Requirements 5

[3.2](#bookmark25)Non-Functional Requirements 5

[3.2.1](#bookmark27)Product Requirements 5

[3.2.1.1](#headingh.44sinio) User Interface Requirements 6

[3.2.1.2](#headingh.2jxsxqh) Usability 6

[3.2.1.3](#bookmark29)Efficiency 6

[3.2.1.3.1](#headingh.3j2qqm3) Performance Requirements 6

[3.2.1.3.2](#bookmark31)Space Requirements 6

[3.2.1.4](#headingh.4i7ojhp) Dependability 6

[3.2.1.5](#headingh.2xcytpi) Security 7

[3.2.2](#bookmark33)Organizational Requirements 7

[3.2.2.1](#headingh.3whwml4) Environmental Requirements 7

[3.2.2.2](#headingh.2bn6wsx) Operational Requirements 7

[3.2.2.3](#headingh.qsh70q) Development Requirements 7

[3.2.3](#bookmark35)External Requirements 7

[3.2.3.1](#headingh.1pxezwc) Regulatory Requirements 7

[3.2.3.2](#headingh.49x2ik5) Ethical Requirements 7

[3.2.3.3](#headingh.2p2csry) Legislative Requirements 7

[3.2.3.3.1](#headingh.147n2zr) Accounting Requirements 7

[3.2.3.3.2](#headingh.3o7alnk) Security Requirements 7

[3.3](#bookmark37)Domain Requirements 7

[4.](#bookmark39)USER SCENARIOS/USE CASES 7

# 

# Executive Summary

## Project Overview

The Coffee Shop Franchise project aims to create a system that allows a coffee shop franchise to efficiently manage its employees and incentivize customers to use the service. The project will rely on a web application that enables online ordering and in-person ordering via a special menu accessible through a QR code. The system will also incorporate a point system for customers, allowing for special offers on various items

## Purpose and Scope of this Specification

The purpose of the Coffee Shop Franchise project is to improve the efficiency of the coffee shop franchise's operations and increase customer satisfaction. The project scope includes the development of a web application that enables online and in-person ordering, a multi-level user system, and a central database for evaluating store and item performances. The system will also include a point system for customers and will enable recording of data for each individual store using a database.

For example:

**In scope**

This document addresses requirements related to phase 2 of Project A:

* modification of Classification Processing to meet legislative mandate ABC.
* modification of Labor Relations Processing to meet legislative mandate ABC.

**Out of Scope**

The following items in phase 3 of Project A are out of scope:

* modification of Classification Processing to meet legislative mandate XYZ.
* modification of Labor Relations Processing to meet legislative mandate XYZ.

(Phase 3 will be considered in the development of the requirements for Phase 2, but the Phase 3 requirements will be documented separately.)

# 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).

## Product Context

The Coffee Shop Franchise project is a digital solution that aims to streamline the coffee shop franchise's operations and improve the customer experience. The project is part of the wider trend of digital transformation and aims to take advantage of the latest technologies to provide a better service.

## User Characteristics

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

The system accounts for 4 types of users

**Customer:**

* Register to the system using a phone number.
* Place orders online.
* Cancel an order inside a limited time window.
* Check their own order history.
* Use points to get select coupons and special offers.

**Shop Staff:**

* Place orders made by customers in person.
* Receive orders made by customers online.
* Mark in person orders as complete.
* Send online orders to delivery staff.
* Delete orders.

**Delivery Staff:**

* Receive orders from shop staff.
* Mark order as complete.
* Cancel orders.
* Check order list.
* Check own daily delivery history.

**Manager:**

* Add shop or delivery staff to the system.
* Remove shop or delivery staff from the system.
* Edit staff information.
* Check item list.
* Check item sales.
* Check overall store performance and profitability.
* Check item inventory status.
* Update item inventory.

**Admin:**

* Add managers to specific store locations.
* Remove managers from store locations.
* Change store location of manager.
* Add managers to system.
* Remove managers from system.
* Check item performance in specific locations.
* Check overall item performance.
* Check a specific location’s performance.
* Add items to specific store locations.
* Remove items from specific store locations.
* Add items to all store locations.
* Remove items from all store locations.

## 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.

1. Users are able to operate a smartphone at a basic level.
2. Online users have internet access.
3. Item inventories in the database are accurate to the real item inventory of the store.
4. Stores have a register that is logged on to a shop staff account, meaning that stores have internet access.

## 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)

1. Since the system cannot physically check the store’s inventory, item inventories need to be manually updated by managers when new supplies come in.
2. Customers need to be logged in in order to make online orders and to access customer-level permissions. Customers who are not logged in and who make in-person orders will not receive points and their purchase will not be recorded to a specific user’s purchase history.

## Dependencies

List dependencies that affect the requirements. Examples:

1. Users need to provide their phone numbers in order to register to the service.
2. Customers cannot place orders on items which rely on items that are out of stock.
3. Orders cannot be placed without the customer’s payment first.
4. Once an order is marked as complete, it cannot be edited or deleted.

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

# 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

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\_SR\_01 | The system database should be updated whenever an order is marked as complete. |  | 1 | 14/05/23 |  |
| BR\_SR\_02 | The system should not allow customers to make orders for items which are out of stock. |  | 1 | 14/05/23 |  |
| BR\_SR\_03 | Customers should be allowed to use points for special discounts if they have the necessary points to do so. |  | 2 | 14/05/23 |  |
| BR\_SR\_03 | The project should have a website that shows information to visitors. |  | 2 | 14/05/23 |  |
| BR\_SR\_04 | The system should have an online ordering service for customers. |  | **1** | 14/05/23 |  |
| BR\_SR\_05 | The system should store information on different store locations | Central database with information on each store location | 1 | 14/05/23 |  |
| BR\_SR\_06 | The system should allow for in-person order placement either through a QR code. |  | 1 | 14/05/23 |  |
| BR\_SR\_07 | The system should allow for in-person order placement through a cashier register. |  | 1 | 14/05/23 |  |
| BR\_SR\_08 | The system should have a multi-level user system. | 5 different user roles: customer, delivery, store staff, manager and admin. | 1 | 14/05/23 |  |
| BR\_SR\_09 | Customers should be able to access information on their previous orders. |  | 2 | 27/05/23 |  |
| BR\_SR\_10 | Users should be allowed to cancel their orders within a limited time period |  | 1 | 27/05/23 |  |
| BR\_SR\_11 | Shop staff should be able to send orders to delivery once they are complete |  | 1 | 27/05/23 |  |
| BR\_SR\_12 | Delivery should be able to mark an order as done once the order has been delivered. |  | 1 | 27/05/23 |  |
| BR\_SR\_13 | Customers should be able to choose whether they want to pay with card or with cash. |  | 1 | 27/05/23 |  |
| BR\_SR\_14 | Customers should not be able to place orders by card if the card cannot transfer the payment to the store account. |  | 1 | 27/05/23 |  |
| BR\_SR\_15 | Shop staff should be allowed to enter an order by a user who has no account in the register. | Users with no account can be marked in the register as a default “Customer” | 2 | 27/05/23 |  |
| BR\_SR\_16 | Managers and admins should be able to manage the employees of their store |  | 1 | 27/05/23 |  |
| BR\_SR\_16 | The system should update store information when an order is completed. | Upon order completion, the database should be update to show the items sold and the profit earned | 2 | 27/05/23 |  |
| BR\_SR\_17 | Admins should be able to add or remove items from any store |  | 2 | 27/05/23 |  |
| BR\_SR\_18 | Managers should be able to add or remove items from their assigned store |  | 2 | 27/05/23 |  |
| BR\_SR\_19 | Admins should be able to manage all other employees |  | 1 | 27/05/23 |  |
| BR\_SR\_20 | The system should notify staff and managers when items are running low |  | 3 | 27/05/23 |  |
| BR\_SR\_21 | The system should generate reports on sales, revenue and popular menu items for Admin and Staff to see |  | 3 | 28/05/23 |  |
| BR\_SR\_22 | The system should support multiple different languages to cater to a diverse customer base |  | 2 | 28/05/23 |  |
| BR\_SR\_23 | The system should have robust security measures to protect customer data, payment information, and system integrity |  | 2 | 28/05/23 |  |
| BR\_SR\_24 | The system should have cross-platform compatibility with multiple web browsers and operating systems for a wider customer reach |  | 2 | 28/05/23 |  |
| BR\_SR\_25 | The system should provide customers with the ability to customize their orders for enhanced personalization. |  | 3 | 28/05/23 |  |
| BR\_SR\_26 | The system should provide managers and admins predictive analytics to forecast future sales and assist with inventory planning |  | 3 | 28/05/23 |  |
| BR\_SR\_27 | The system should provide the ability to create and distribute customer surveys to gather feedback and measure customer satisfaction levels |  | 3 | 28/05/23 |  |
| BR\_SR\_28 | The system should suggest personalized menu items based on customer preferences and past orders. |  | 3 | 28/05/23 |  |
| BR\_SR\_29 | The system should have optimized performance to handle high volumes of orders and scale as the business grows |  | 1 | 28/05/23 |  |
| BR\_SR\_30 | The system should be integrated with popular third-party delivery platforms for expanded delivery options. |  | 2 | 28/05/23 |  |

## Non-Functional Requirements

User Interface Responsiveness:

* The web application should load quickly and respond to user interactions within 2 seconds, ensuring a smooth and responsive user experience.
* The user interface should be compatible with different screen sizes and resolutions, providing consistent functionality and usability across devices.

### Product Requirements

* Security: The system should be secure and protect sensitive data, including customer information and payment details. It should comply with industry-standard security protocols and regulations.
* Environmental requirements: The system should be designed to minimize environmental impact and energy usage. This can include using energy-efficient hardware and software, as well as minimizing waste and promoting sustainable practices.
* Operational requirements: The system should be designed to integrate smoothly with existing store operations and workflows. It should be easy to maintain and update, with minimal downtime for maintenance or updates.
* Ethical requirements: The system should comply with ethical standards and regulations, including data privacy laws and fair labor practices. It should prioritize customer safety and well-being, as well as employee satisfaction and fair compensation.
* Legislative requirements: The system should comply with all relevant legislation and regulations, including food safety laws, labor laws, and data privacy laws. It should also adhere to local and national regulations regarding franchising and business operations.

#### **User Interface Requirements**

* The user interface should be visually appealing, easy to navigate, and provide clear instructions to the user. It should be accessible on both desktop and mobile devices.

#### **Usability**

* The system should be easy to use for all user roles and require minimal training. The online ordering process should be simple and intuitive, with the ability to customize orders and view past orders.

#### **Efficiency**

* The system should be fast and responsive, with quick loading times for all pages and minimal lag in processing orders. The database should be able to handle a large volume of data and be scalable for future growth.
* The system should be optimized for efficient resource utilization, minimizing processing time and reducing network bandwidth consumption.
* Database queries and operations should be optimized to ensure quick response times and smooth performance, even with large volumes of data.

##### Performance Requirements

* The system should be reliable and perform well under heavy usage. It should have a high availability and be able to handle multiple simultaneous users without downtime or errors.

##### Space Requirements

* The system should be designed to take up minimal physical space in the stores, with all necessary hardware and equipment being compact and efficient.

#### **Dependability**

* The system should be dependable and maintain data integrity. It should have backup systems in place in case of hardware failure or other disruptions.

**Availability**

* The system should have a high availability, aiming for a minimum uptime of 99.9% to ensure uninterrupted access for customers and employees.
* Regular backups of the system's data should be performed to prevent data loss in case of hardware failures or other unforeseen events.

**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**

* User data, including personal information, order history, and payment details, should be securely stored and encrypted to protect against unauthorized access or breaches.
* The system should comply with data privacy regulations, such as the General Data Protection Regulation (GDPR), ensuring user consent for data collection and providing options for data deletion.

### Organizational Requirements

* The system should provide centralized management capabilities for franchise owners and administrators to oversee multiple stores within the franchise.
* Franchise owners should have access to consolidated data and reports across all stores to monitor overall performance and make informed business decisions.
* The system should support different user roles with varying levels of access and permissions, such as franchise owners, managers, store workers, and administrators.
* Each user role should have defined responsibilities and access rights based on their position within the organization.
* The system should generate comprehensive reports and analytics to provide insights into sales performance, customer behavior, and inventory management.
* Franchise owners and managers should be able to access real-time and historical data for decision-making and performance evaluation.

#### **Environmental Requirements**

* The application should aim to minimize the use of paper by providing digital alternatives for processes such as order taking, invoicing, and reporting.
* Features like e-receipts and digital menus can be implemented to reduce paper waste.
* The application should be designed to operate efficiently and minimize energy consumption on user devices, such as smartphones, tablets, and computers.
* Optimized code and efficient use of resources, such as CPU and battery, should be prioritized during development.
* The application can highlight and promote environmentally friendly products or initiatives offered by the coffee shop franchise, such as organic or fair-trade coffee, reusable cups, or sustainable packaging options.
* It can provide information about the franchise's efforts towards sustainability and encourage users to make environmentally conscious choices.

#### **Operational Requirements**

* The application should be compatible with popular operating systems, including iOS and Android, to ensure broad accessibility for users.
* It should support the latest versions of these operating systems and provide backward compatibility where feasible.
* The application should be responsive and perform efficiently, even during peak usage times.
* It should have quick response times for tasks such as loading menus, placing orders, and accessing customer information.
* The application should have offline capabilities to allow users to browse menus, view previous orders, and access certain features even when an internet connection is not available.
* Offline data synchronization should be supported to update information once the connection is restored.
* The application should adhere to accessibility guidelines to ensure that users with disabilities can access and use its features effectively.
* Support for features such as screen readers, adjustable font sizes, and color contrast options should be provided.
* The application should support multiple languages to cater to a diverse user base.
* Language selection should be user-configurable within the application.

#### **Development Requirements**

* The development team should have expertise in relevant programming languages such as HTML, CSS and JavaScript for the front-end development. For the back-end, languages like Python, Java, or PHP.
* A suitable database management system like MySQL or PostgreSQL, should be selected to efficiently store and manage data related to customers, employees, menu items, orders, and loyalty points.
* Integration with necessary APIs should be implemented to enable seamless functionality and data exchange.
* Thorough testing should be conducted to ensure the system functions correctly and meets the specified requirements.
* Comprehensive documentation should be created to facilitate future maintenance and support.
* The development process should follow established project management methodologies, such as Agile or Scrum, with clear milestones, regular communication, and feedback loops to ensure efficient development and timely delivery of the system.

### 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**

* The system should comply with relevant data protection and privacy regulations, such as the General Data Protection Regulation (GDPR) or other applicable regional data protection laws.
* If the system incorporates food ordering or delivery, compliance with local food safety and handling regulations is essential. This includes adherence to food hygiene standards, proper labeling of allergens and ingredients, and maintaining records for traceability purposes.
* The system should adhere to consumer protection laws and regulations, which include accurate pricing and descriptions of menu items, clear terms and conditions for loyalty programs and offers, and proper handling of customer complaints and refunds.
* The system should comply with relevant employment laws and labor regulations, such as those related to employee contracts, working hours, minimum wage, and fair employment practices.

#### **Ethical Requirements**

* The system should provide accurate and transparent information to customers, including menu item descriptions, pricing, and availability. It should avoid deceptive practices, misleading advertising, or hidden charges.
* The point system and special offers should be designed and implemented in a fair and equitable manner. All customers should have equal opportunities to earn and redeem points, and special offers should be transparently communicated and accessible to all eligible customers.
* The system should not discriminate against any individuals or groups based on characteristics such as race, ethnicity, gender, age, religion, or disability.
* The system should use technology responsibly, avoiding the misuse or unethical application of data.

##### Accounting Requirements

* The system should accurately record all sales transactions, including cash, credit card, and digital payments.
* The system should provide inventory management features, such as tracking stock levels and monitoring ingredient usage.
* The system should generate financial reports, including income statements, balance sheets, and cash flow statements.
* The system should support tax calculations and generate reports required for tax filing purposes. It should adhere to applicable tax regulations and guidelines, including sales tax, payroll tax, and income tax.
* The system should have payroll functionality to accurately calculate employee wages, deductions, and taxes.

##### Security Requirements

* The system should implement secure user authentication mechanisms, such as username and password and two-factor authentication.
* The system should incorporate role-based access to assign specific access rights and permissions to different user roles. This ensures that users can only access the functionalities and data necessary for their roles.
* Sensitive data, including customer information, employee data, and financial records, should be encrypted both during transit and at rest to prevent unauthorized access or data breaches.
* The system should be regularly updated with the latest security patches and updates to address any known vulnerabilities and protect against emerging threats.
* Regular backups of the system's data should be performed and securely stored.

## 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

* As a customer, I want to be able to place orders online and in-person, and receive rewards for frequent purchases.
* As a delivery driver, I want to be able to receive and complete orders efficiently, with accurate information on delivery locations and special requests.
* As a store worker, I want to be able to manage orders and inventory effectively, with easy access to customer data and item performance.
* As a manager, I want to be able to evaluate store and item performance across the franchise, and make data-driven decisions to improve operations.
* As an admin, I want to be able to manage user roles and access levels, as well as ensure compliance with relevant legislation and regulations.

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.