* 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** |
| --- | --- | --- | --- | --- | --- |
| LMS\_BR\_01 | Allow users (clients, workers, managers, administrators) to register and create accounts with appropriate roles and access levels. | Business Process = “User Registration” | 1 | 12/05/22 | Ari Gjerazi  Julia Marku |
| LMS\_BR\_02 | Implement a secure login mechanism to authenticate users and control access to various system functionalities. | Business Process = “User Authentication” | 1 | 12/05/22 | Ari Gjerazi  Julia Marku |
| LMS\_BR\_03 | Provide functionality for clients to check the availability of rooms, view real-time prices, and make reservations. | Business Process = “Room Availability and Booking” | 1 | 12/05/22 | Ari Gjerazi  Julia Marku |
| LMS\_BR\_04 | Allow clients to request room service directly through the app, which notifies the workers responsible for the requested task. | Business Process = “Room Service” | 3 | 12/05/22 | Ari Gjerazi  Julia Marku |
| LMS\_BR\_05 | Enable clients to provide their thoughts and reviews on the hotel, workers, and managers. | Business Process = “Reviews and Feedback” | 2 | 12/05/22 | Ari Gjerazi  Julia Marku |
| LMS\_BR\_06 | Implement a notification system to alert workers about rooms requiring service/cleaning. | Business Process = “Worker Notifications” | 3 | 12/05/22 | Ari Gjerazi  Julia Marku |
| LMS\_BR\_07 | Provide managers with a dashboard to view room availability, prices, worker performance, and manage orders. | Business Process = “Manager Dashboard” | 1 | 12/05/22 | Ari Gjerazi  Julia Marku |
| LMS\_BR\_08 | Allow managers to create and assign orders/tasks to workers, specifying the room number and required tasks. | Business Process = “Order Management” | 2 | 12/05/22 | Ari Gjerazi  Julia Marku |
| LMS\_BR\_09 | Enable managers to update room prices and availability in real-time. | Business Process = “Price and Availability Management” | 2 | 12/05/22 | Ari Gjerazi  Julia Marku |
| LMS\_BR\_10 | Provide administrators with extensive control over the website and app, allowing them to update content, manage orders, and rearrange prices and accommodations. | Business Process = “Administrator Privileges” | 1 | 12/05/22 | Ari Gjerazi  Julia Marku |

## 