Human Resource Requirements Specification

Version 1.0

August 28, 2018

Table of Contents

[1. Executive Summary 3](#_Toc191724230)

[1.1 Project Overview 3](#_Toc191724231)

[1.2 Purpose and Scope of this Specification 3](#_Toc191724232)

[2. Product/Service Description 3](#_Toc191724233)

[2.1 Product Context 3](#_Toc191724234)

[2.2 User Characteristics 3](#_Toc191724235)

[2.3 Assumptions 3](#_Toc191724236)

[2.4 Constraints 3](#_Toc191724237)

[2.5 Dependencies 4](#_Toc191724238)

[3. Requirements 4](#_Toc191724239)

[3.1 Functional Requirements 5](#_Toc191724240)

[3.2 User Interface Requirements 5](#_Toc191724241)

[3.3 Usability 5](#_Toc191724242)

[3.4 Performance 6](#_Toc191724243)

[3.4.1 Capacity 6](#_Toc191724244)

[3.4.2 Availability 6](#_Toc191724245)

[3.4.3 Latency 6](#_Toc191724246)

[3.5 Manageability/Maintainability 6](#_Toc191724247)

[3.5.1 Monitoring 6](#_Toc191724248)

[3.5.2 Maintenance 6](#_Toc191724249)

[3.5.3 Operations 6](#_Toc191724250)

[3.6 System Interface/Integration 7](#_Toc191724251)

[3.6.1 Network and Hardware Interfaces 7](#_Toc191724252)

[3.6.2 Systems Interfaces 7](#_Toc191724253)

[3.7 Security 8](#_Toc191724254)

[3.7.1 Protection 8](#_Toc191724255)

[3.7.2 Authorization and Authentication 8](#_Toc191724256)

[3.8 Data Management 8](#_Toc191724257)

[3.9 Standards Compliance 8](#_Toc191724258)

[3.10 Portability 8](#_Toc191724259)

[4. User Scenarios/Use Cases 9](#_Toc191724260)

[5. Deleted or Deferred Requirements 9](#_Toc191724261)

[6. Requirements Confirmation/Stakeholder sign-off 10](#_Toc191724262)

[APPENDIX 11](#_Toc191724263)

[Appendix A. Definitions, Acronyms, and Abbreviations 11](#_Toc191724264)

[Appendix B. References 11](#_Toc191724265)

[Appendix C. Requirements Traceability Matrix 11](#_Toc191724266)

[Appendix D. Organizing the Requirements 13](#_Toc191724267)

# Executive Summary

## Project Overview

Human Resource manage are responsible for recruiting, interviewing and placing workers. They also handle employee relations and benefits training also handle attendance of the employees. It also preparing or updating employment records related to hiring, transferring, promoting and terminating. The Human resource has payroll it consists of the information about the employee salary details like the deductions if the Employee is absent. It will easy for the Human Resource to compute the Salary of the employees.

## Purpose and Scope of this Specification

***Include a description of what is within the scope what is outside of the scope of these specifications. For example:***

In scope

* Attendance of employees
* Post of Job hiring
* Employee Details

Out of Scope

* Payroll

# Product/Service Description

The general factor of this system is to facilitate the work of Human Resource.The system can post job hiring and the information on each employee is also on the system it consists of the information about the employee name and the position and it also hold the attendance of the employee attendace if it is late or absent. In payroll it consists of the information about the employee saraly details like the deductions if the Employee is absent. In this system, it will easy for the Human Resource to compute the Salary of the employees.

## Product Context

The application like this is the same to other product that the user wants to learn. This application is self-contained **(search more).** 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. (martisano)

## User Characteristics - skip

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

## Assumptions - skip?

This is available for the entire user that is interested to the discrete mathematics, showed that providing feedback on progress toward learning goals increases performance and formative assessment provides information to identify gaps between current and desired performance, which a learner may then strive to narrow to maximize performance.

**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.**

## Constraints

Describe any items that will constrain the design options, including

* parallel operation with an old system - skip
* audit functions (audit trail, log files, etc.)- skip
* No security, it is open to all user :access, management and security
* criticality of the application- skip

## Dependencies

List dependencies that affect the requirements. Examples:

* This new system required internet to access the needed of system. This new product will require a daily download of data from X,
* This system is depended to android studio. Module X needs to be completed before this module can be built.

# Requirements

* The system was made through the android studio, In this process you will know if the program was working or not. Structured Query Language is standard computer language for relational database management and data manipulation**. 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.
* The system input will come from user’s, the user need to input the answer to each question contained in each level then its output will show the user the list of answer each level, the function of each output will eventually be seen by the last part of level. **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.)**
* The priority of the system is the user that having difficulty on how to understand the discrete math. Nowadays most of the user use website and application that why this application may help those users to understand more on how solve the problem in discrete mathematics. **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 | Days Completed | Date Rvwd | Reviewed / Approved |
| --- | --- | --- | --- | --- | --- | --- |
| BR\_LR\_05 | The system should associate a supervisor indicator with each job class. | Business Process = “Maintenance | 3 |  | 7/13/04 | Bob Dylan, Mick Jagger |
| BR\_LR\_08 | The system should handle any number of fees (existing and new) associated with unions. | Business Process = “Changing Dues in the System”  An example of a new fee is an initiation fee. | 2 |  | 7/13/04 | Bob Dylan, Mick Jagger |
| BR\_LR\_10 | The system should capture and maintain job class status (i.e., active or inactive) | Business Process = “Maintenance”  Some job classes are old and are no longer used. However, they still need to be maintained for legal, contract and historical purposes. | 2 |  | 7/13/04 | Bob Dylan, Mick Jagger |
| BR\_LR\_16 | The system should assign the Supervisor Code based on the value in the Job Class table and additional criteria as specified by the clients. | April 2005 – New requirement. It is one of three new requirements from BR\_LR\_03. | 2 |  |  |  |
| BR\_LR\_18 | The system should provide the Labor Relations office with the ability to override the system-derived Bargaining Unit code and the Union Code for to-be-determined employee types, including hourly appointments. | April 2005 – New requirement. It is one of three new requirements from BR\_LR\_04.  5/11/2005 – Priority changed from 2 to 3. | ~~2~~  3 |  |  |  |

## 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/>)

## Performance

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

* The system numerical requirement is the number of users of a system; we make it for those interested in this system. It’s easy to use if the user is new. **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**.
* The number of system transaction will depending on the two options: single player and multiplayer, single player: each level has ten seconds to answer every question, if the question is not answered automatically the answer will be wrong. Then in multiplayer you can choose you want to play with you, there’s a encrypt and decrypt this is the type of play in multiplayer**. 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”.**

### Capacity

Include measurable capacity requirements (e.g., the number of simultaneous users to be supported, the maximum simultaneous user load, per-user memory requirements, expected application throughput)

### 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).

### Latency

Include explicit latency requirements, e.g., the maximum acceptable time (or average time) for a service request.

## Manageability/Maintainability -SKIP

### 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.

### Operations

Specify any normal and special operations required by the user, including:

* periods of interactive operations and periods of unattended operations
* data processing support functions
* backup and recovery operations
* safety considerations and requirements
* disaster recovery and business resumption

## System Interface/Integration -SKIP

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

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

### Network and Hardware Interfaces

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

### Systems Interfaces

Example systems interface requirements:

1. System1-to-System2 Interface

The <external party> will create and send a fixed length text file as an email attachment to [System2mail@u.washington.edu](mailto:heppsmai@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

* 1. The FileName file is a fixed length text file.
  2. The FileName file is an unformatted ASCII file (text-only).
  3. The FileName file contains a batch totals record and several detail records.

File Description: Batch Totals Record

* 1. The batch totals record can be placed at the beginning, in the middle, or at the end of the file.
  2. 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

* 1. The FileName file contains a row for each record meeting xxx criteria.
  2. 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

## Security

### Protection

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

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

### Authorization and Authentication

Specify the Authorization and Authentication factors. Consider using standard tools such as PubCookie. Using Security Assertion Markup Language (SAML). How valid users can access the application, and avoid alteration of codes

## Data Management

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

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

## Standards Compliance

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.

## Portability

If portability is a requirement, specify attributes of the system that relate to the ease of porting the system to other host machines and/or operating systems. For example,

* Percentage of components with host-dependent code;
* Percentage of code that is host dependent;
* Use of a proven portable language;
* Use of a particular compiler or language subset;
* Use of a particular operating system;
* The need for environment-independence - the product must operate the same regardless of operating systems, networks, development or production environments.