# [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**

**VERSION 1.0 1**

**APRIL 19, 2021 1**

**1. EXECUTIVE SUMMARY 3**

1.1 Project Overview 3
1.2 Purpose and Scope of this Specification 3

**2. PRODUCT/SERVICE DESCRIPTION 3**

2.1 Product Context 3
2.2 User Characteristics 3
2.3 Assumptions 3
2.4 Constraints 3
2.5 Dependencies 4

**3. REQUIREMENTS 4**

3.1 Functional Requirements 5
3.2 Non-Functional Requirements 5
*3.2.1 Product Requirements 5*
**3.2.1.1 User Interface Requirements** 6
**3.2.1.2 Usability** 6
**3.2.1.3 Efficiency** 6
3.2.1.3.1 Performance Requirements 6
3.2.1.3.2 Space Requirements 6
**3.2.1.4 Dependability** 6
**3.2.1.5 Security** 7
*3.2.2 Organizational Requirements 7*
**3.2.2.1 Environmental Requirements** 7
**3.2.2.2 Operational Requirements** 7
**3.2.2.3 Development Requirements** 7
*3.2.3 External Requirements 7*
**3.2.3.1 Regulatory Requirements** 7
**3.2.3.2 Ethical Requirements** 7
**3.2.3.3 Legislative Requirements** 7
3.2.3.3.1 Accounting Requirements 7
3.2.3.3.2 Security Requirements 7
3.3 Domain Requirements 7

**4. USER SCENARIOS/USE CASES 7**

# 1. Executive Summary

## *1.1 Project Overview*

**The focus of this software will be to create an educational platform that caters to the needs of teachers, parents, and students. The software will be designed as a comprehensive learning management system (LMS) that provides an online platform for managing course content, assignments, and progress tracking.**

**Teachers will have the ability to upload course material, such as presentations, handouts, and assignments. They will be able to manage course content by organizing it based on subject, topic, and difficulty level. Teachers can also create and manage assessments, such as quizzes and exams, and track student progress and grades. Communication with students and parents will be facilitated through the platform's messaging system.**

**Parents will be able to monitor their child's academic performance and progress, view their child's grades, assignments, and progress reports, communicate with teachers through the platform's messaging system, and receive alerts and notifications regarding their child's performance or upcoming deadlines.**

**Students will be able to access course material, including presentations, handouts, and assignments, complete and submit assignments and assessments, track their own academic performance and progress, and communicate with teachers through the platform's messaging system.**

**The platform will be accessible through a web browser and will include mobile applications for Android and iOS devices. The application will have a three-level user system: Teacher, Parent, and Student. Each user will have different views and functionalities based on their roles.**

**The software will be designed to create a user-friendly, comprehensive, and scalable platform that will streamline the educational experience for everyone involved. Teachers will be able to easily upload and manage course material, while students will be able to access and complete assignments and exams from a single location. Parents will be able to monitor their children's progress and communicate with teachers to stay informed about their child's academic performance. Overall, the aim of the educational platform is to create a centralized location for managing coursework, assignments, and progress tracking, providing an enhanced learning experience for all users involved.**

## *1.2 Purpose and Scope of this Specification*

**The purpose of this software specification is to define the requirements and functionalities of an educational platform that serves as a learning management system for teachers, students, and parents. The intended audience for this specification includes software developers, project managers, and stakeholders involved in the design, development, and implementation of the platform.**

**In Scope:**

- **The platform must allow teachers to create and manage courses, upload and organize course content, set assignments, and monitor student progress.**

- **The platform should enable students to access course materials, complete assignments, and receive feedback from teachers.**
- **The platform must provide a dashboard for parents to monitor their children's progress, view grades and assignments, and communicate with teachers.**
- **The platform should offer features such as a calendar, notifications, and messaging to facilitate communication between teachers, students, and parents.**
- **The platform must ensure the security of data, such as student records and personal information.**
- **The platform should support various types of multimedia content, such as videos and images, for a rich learning experience.**

Out of Scope:

- **The platform will not provide any hardware or software for users to access the system.**
- **The platform will not provide any direct teaching or educational content.**
- **The platform will not include any features or functionalities that violate any laws or regulations.**

# 2. Product/Service Description

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

## *2.1 Product Context*

In today's world, education has become more important than ever. With the rapid pace of technological advancement, it is necessary for people to continuously update their skills and knowledge to stay competitive in the job market. However, traditional education systems are often not able to keep up with these changes, leaving many people struggling to find ways to access the information they need to succeed.

This is where an education platform comes in. An education platform is an online platform that provides access to a wide range of educational resources, including courses, tutorials, and other learning materials. The platform can be accessed from anywhere in the world, at any time, making it easy for people to fit learning into their busy lives.

The platform is designed to be user-friendly and intuitive, making it easy for people of all ages and backgrounds to use. It features a variety of tools and resources to help learners succeed, including interactive quizzes, discussion forums, and personalized learning plans.

One of the key advantages of an education platform is that it can be customized to meet the needs of different types of learners. For example, some people may prefer to learn at their own pace, while others may prefer a more structured approach. The platform can accommodate both types of learners, providing a personalized learning experience that is tailored to their individual needs.

Overall, an education platform is a powerful tool for anyone looking to enhance their skills and knowledge. Whether you are a student, a working professional, or simply someone who wants to learn something new, an education platform can provide the resources and support you need to achieve your goals.

## 2.2 User Characteristics

Teacher:

Age range: 25-65 years old
Education level: Tertiary
Experience: Extensive experience in teaching, may have used various online learning platforms in the past
Technical expertise: Proficient in using computers and technology, may have expertise in using various educational software
Characteristics: May have different teaching styles and preferences, may have varying levels of comfort with technology, may have limited time for managing the platform due to teaching and administrative responsibilities.
The teacher is the primary user of the platform and has various responsibilities such as creating and managing courses, uploading and organizing course content, setting assignments, monitoring student progress, and communicating with students and parents. The teacher is expected to be proficient in using the platform and must be able to troubleshoot any issues that arise. The teacher may have different teaching styles and preferences that require customization and flexibility of the platform. The teacher may also have limited time to manage the platform due to their teaching and administrative responsibilities.

Student:

Age range: 6-24 years old
Education level: Primary, secondary, tertiary
Experience: May have some experience with online learning platforms but may not be proficient in using technology
Technical expertise: Basic computer skills, familiarity with social media and messaging apps
Characteristics: Diverse backgrounds, learning styles, and interests. May have limited access to technology or internet connectivity. May have varying levels of engagement with the platform.
The student is the primary user of the platform and must be able to access course materials, complete assignments, and receive feedback from teachers. The student's learning styles and interests must be considered when designing the platform to ensure a personalized and engaging learning experience. The student may have varying levels of engagement with the platform, and it is essential to provide features such as gamification, multimedia content, and social interaction to promote engagement. The student may have limited access to technology or internet connectivity, and it is necessary to provide offline access to course materials and assignments.

Parent/Guardian:

Age range: 25-65 years old
Education level: Varies
Experience: May have experience using online platforms for personal or professional use
Technical expertise: Basic computer skills, familiarity with social media and messaging apps
Characteristics: May have varying levels of involvement in their children's education. May have limited access to technology or internet connectivity. May have concerns regarding their children's

privacy and security. May be interested in tracking their children's progress and communicating with teachers.

The parent is a secondary user of the platform and has a role in monitoring their children's progress, viewing grades and assignments, and communicating with teachers. The parent may have varying levels of involvement in their children's education, and it is necessary to provide features such as progress tracking, notifications, and communication channels to promote their engagement. The parent may have limited access to technology or internet connectivity, and it is essential to provide user-friendly and accessible interfaces to facilitate their use of the platform. The parent may also have concerns regarding their children's privacy and security, and it is necessary to ensure the platform's compliance with relevant regulations and policies.

## *2.3 Assumptions*

Here are some assumptions that may affect the requirements:

Availability of equipment: It is assumed that all users have access to the necessary equipment to use the platform, including a computer or mobile device and a stable internet connection.

User expertise: It is assumed that teachers, students, and parents have basic computer skills and are familiar with using online platforms for communication and learning.

Platform compatibility: It is assumed that the platform is compatible with commonly used web browsers, operating systems, and mobile devices.

Privacy and security: It is assumed that the platform will comply with relevant privacy and security regulations to protect user data and information.

Content creation: It is assumed that teachers have the necessary skills and expertise to create and upload course content to the platform.

Student engagement: It is assumed that students will be motivated and engaged in using the platform, and that the platform will provide features to promote engagement and participation.

Technical support: It is assumed that technical support will be available to users to address any technical issues that may arise when using the platform.

These assumptions may impact the design and development of the platform, as well as the training and support provided to users. If any of these assumptions are not met, the requirements specification may need to be revised accordingly to ensure that the platform can still meet the needs of its users.

## *2.4 Constraints*

Describe any items that will constrain the design options, including

- parallel operation with an old system

**The application must operate in parallel with an old system, there are several items that can constrain design options. Some of these items include:**
**Compatibility, Capacity, Interference, Maintenance and Cost**

- audit functions (audit trail, log files, etc.)

**By creating the new application with audit functions, such as an audit trail or log files, there are several items that can constrain the design options. Some of these items include:**
**Storage capacity, Processing power, Data security, Compliance requirements and the User interface**

- access, management and security

**By creating the new application with access, management, and security requirements, there are several items that can constrain the design options. Some of these items include:**
**Authentication and authorization, Role-based access control, Data privacy and confidentiality, System monitoring and auditing and Compliance requirements**

- criticality of the application

**When designing the application for criticality, there are several items that can constrain the design options. Some of these items include:**
**Redundancy and fault tolerance, Performance, Security, Maintenance and support also Compliance requirements**

- system resource constraints (e.g., limits on disk space or other hardware limitations)

**When designing the application that is subject to resource constraints, there are several items that can constrain the design options. Some of these items include:**
**Hardware limitations, Scalability, Performance requirements, Cost and the Compatibility with existing infrastructure**

- other design constraints (e.g., design or other standards, such as programming language or framework)

**When designing the application, there are often additional design constraints that can constrain the design options. Some of these constraints include:**
**Design standards, Compatibility with existing systems, User interface requirements, Performance requirements, Maintenance and support requirements**

## *2.5 Dependencies*

- **Students must submit their assignment before the deadline. There is no submission after the required data and time, or else they will get a 0 for that task.**
- **There will be no more than 4 course/subject evaluation methods per day. More than this means more workload for students and reciprocally more work for teachers or parents.**
- **The application is not designed for an unlimited number of students. The capacity of the system holds no more than 20 teachers and 100 students.**
- **With each student there will also be his/her respective parent/s, so that should not be more than 120.**
- **The subject will have a limited number of assignments/quizzes/projects/exams that should not exceed 100% altogether.**
- **To make the communication process between the students and the teachers/parents possible, notifications need to be sent to all these three to ensure that everything is being notified.**

- **Teachers need to have a graphical description regarding the child's academic performance/activity progress so that they can make their assessments more accurately and objectively.**
- **A forum for the communication between the students is mandatory, so that they can share any difficulties they have about certain questions together.**
- **A syllabus system is strictly requested too; in this way, the course can be divided into sections, topics, assessment methods, learning objectives, and the weight in percentage for each assessment method.**
- **Students need to be enrolled by a teacher to be part of a certain course, and the lack of a login system (so that the parents/students/teachers can enter the system) is prohibited.**

# 3. Requirements

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

## *3.1 Functional Requirements*

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

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

For Example:

| Req# | Requirement | Comments | Priority | Date Rvwd | SME Reviewed / Approved |
|---|---|---|---|---|---|
| LMS_BR_01 | The system should allow instructors to create and publish courses. | Business Process = “Course Creation and Publication” | 3 | | |
| LMS_BR_02 | The system should provide a dashboard for students to view their progress. | Business Process = “Student Progress Tracking” | 2 | | |
| LMS_BR_03 | The system should allow students to view course materials and assignments. | Business Process = “Course Materials and Assignments” | 1 | | |
| LMS_BR_04 | The system should allow instructors to grade and provide feedback on student assignments. | Business Process = “Grading and Feedback” | 2 | | |
| LMS_BR_05 | The system should provide a discussion forum for students to communicate with each other and with instructors. | Business Process = “Discussion Forum” | 3 | | |
| LMS_BR_06 | The system should allow instructors to schedule and conduct virtual classroom sessions. | Business Process = “Virtual Classroom” | 1 | | |
| LMS_BR_07 | The system should allow students to take quizzes and exams. | Business Process = “Quizzes and Exams” | 2 | | |
| LMS_BR_08 | The system should allow instructors to generate reports on student performance. | Business Process = “Reporting” | 1 | | |
| LMS_BR_09 | The system should allow administrators to manage user accounts and system settings. | Business Process = “Administration” | 1 | | |
| LMS_BR_10 | The system should support integration with external learning tools and systems. | Business Process = “Integration” | 3 | | |

| Req# | Requirement | Comments | Priority | Date Rvwd | SME 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”<br>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”<br>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.<br>5/11/2005 – Priority changed from 2 to 3. | ~~2~~<br>3 | | |

## *3.2 Non-Functional Requirements*

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

### 3.2.1 Product Requirements

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

#### 3.2.1.1 **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).

#### 3.2.1.2 **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/)

### 3.2.1.3 **Efficiency**

#### ***3.2.1.3.1 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”.

#### ***3.2.1.3.2 Space Requirements***

### 3.2.1.4 **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**

### 3.2.1.5 **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.

### 3.2.2 Organizational Requirements

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

3.2.2.1 **Environmental Requirements**

3.2.2.2 **Operational Requirements**

3.2.2.3 **Development Requirements**

### 3.2.3 External Requirements

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

3.2.3.1 **Regulatory Requirements**

3.2.3.2 **Ethical Requirements**

3.2.3.3 **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

**3.2.3.3.1 Accounting Requirements**

**3.2.3.3.2 Security Requirements**

## *3.3 Domain Requirements*

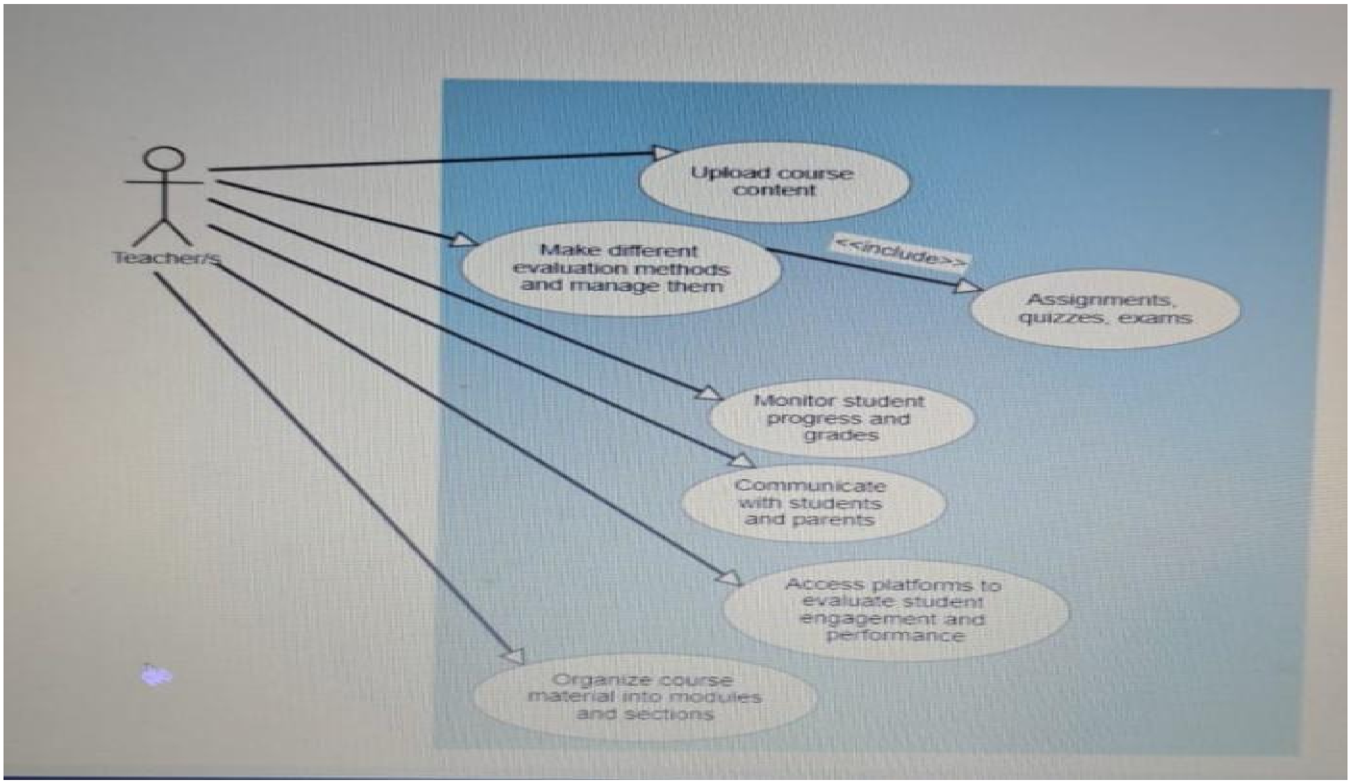
Everything related to the domain that might be needed in the project shall be mentioned here. Sometimes the domain Requirements might be thought of as part of either functional or non-functional requirements.

A.1.1.1.

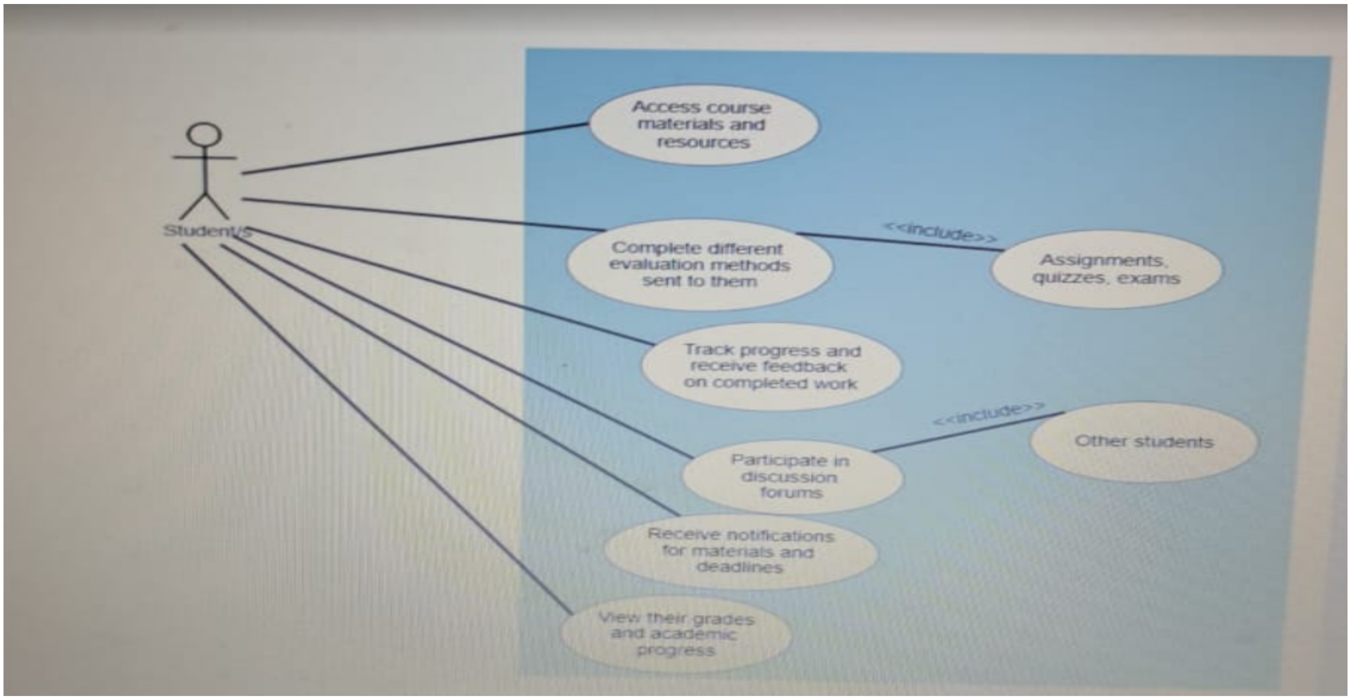
Please provide all necessary non-functional requirements, similar to the requirements explained in the lesson slides or in the textbook.

# 4. User Scenarios/Use Cases

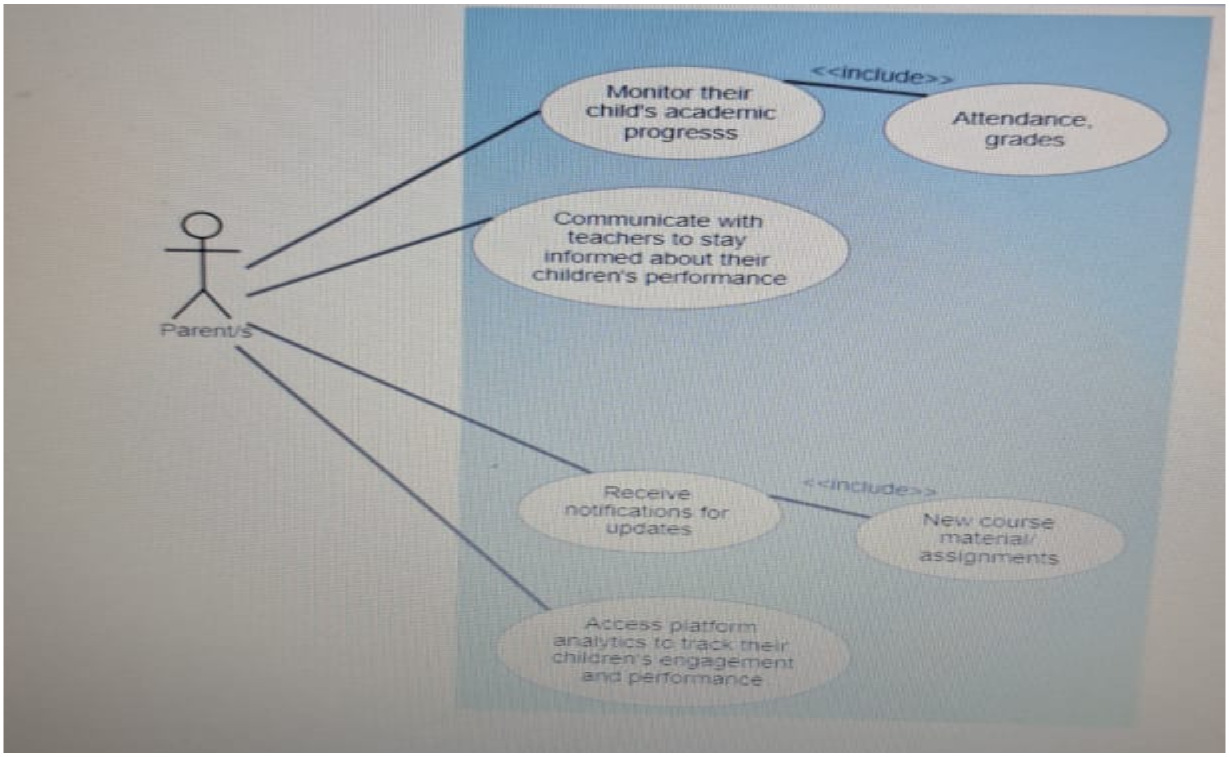
- ***Use Case 1 (Teacher Use Case)***


Teacher/s
Upload course content
Make different evaluation methods and manage them
<<include>>
Assignments, quizzes, exams
Monitor student progress and grades
Communicate with students and parents
Access platforms to evaluate student engagement and performance
Organize course material into modules and sections

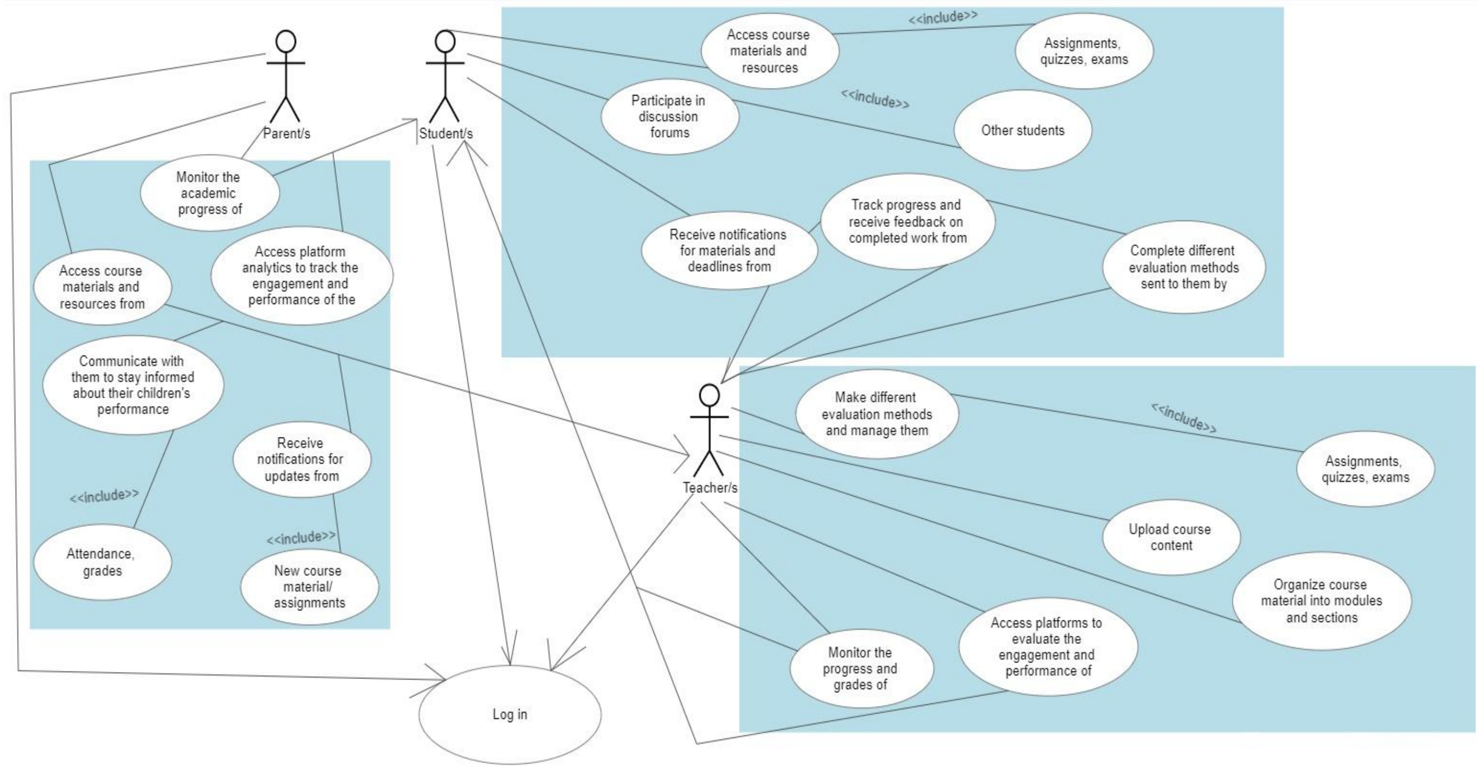
### *Use Case 2(Student Use Case)*


Student/s
Access course materials and resources
Complete different evaluation methods sent to them
<<include>>
Assignments, quizzes, exams
Track progress and receive feedback on completed work
Participate in discussion forums
<<include>>
Other students
Receive notifications for materials and deadlines
View their grades and academic progress

### *Use Case 3 (Parent Use Case)*


Parent/s
Monitor their child's academic progresss
<<include>>
Attendance, grades
Communicate with teachers to stay informed about their children's performance
Receive notifications for updates
<<include>>
New course material/ assignments
Access platform analytics to track their children's engagement and performance

### *Use Case 4 (All combined into one)*


Parent/s
Student/s
Teacher/s
Monitor the academic progress of
Access platform analytics to track the engagement and performance of the
Access course materials and resources from
Communicate with them to stay informed about their children's performance
Receive notifications for updates from
<<include>>
Attendance, grades
<<include>>
New course material/ assignments
Access course materials and resources
<<include>>
Assignments, quizzes, exams
Participate in discussion forums
<<include>>
Other students
Receive notifications for materials and deadlines from
Track progress and receive feedback on completed work from
Complete different evaluation methods sent to them by
Make different evaluation methods and manage them
<<include>>
Assignments, quizzes, exams
Upload course content
Organize course material into modules and sections
Access platforms to evaluate the engagement and performance of
Monitor the progress and grades of
Log in

# APPENDIX

The appendixes are not always considered part of the actual Requirements Specification and are not always necessary. They may include

- Sample input/output formats, descriptions of cost analysis studies, or results of user surveys;
- Supporting or background information that can help the readers of the Requirements Specification;
- A description of the problems to be solved by the system;
- Special packaging instructions for the code and the media to meet security, export, initial loading, or other requirements.

When appendixes are included, the Requirements Specification should explicitly state whether or not the appendixes are to be considered part of the requirements.

## Appendix B. Definitions, Acronyms, and Abbreviations

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

## Appendix C. References

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

## Appendix D. Requirements Traceability Matrix

The following trace matrix examples show one possible use of naming standards for deliverables (FunctionalArea-DocType-NN). The number has no other meaning than to keep the documents unique. For example, the Bargaining Unit Assignment Process Flow would be BUA-PF-01.

For example (1):

| Business Requirement | Area | Deliverables | Status |
|---|---|---|---|
| BR_LR_01<br>The system should validate the relationship between Bargaining Unit/Location and Job Class.---Comments: Business Process = "Assigning a Bargaining Unit to an Appointment" (Priority 1) | BUA | BUA-CD-01<br>Assign BU Conceptual Design | Accepted |
| | | BUA-PF-01<br>Derive Bargaining Unit-Process Flow Diagram | Accepted |
| | | BUA-PF-01<br>Derive Bargaining Unit-Process Flow Diagram | Accepted |
| BR_LR_09<br>The system should provide the capability for the Labor Relations Office to maintain the job class/union relationship.---Comments: Business Process = "Maintenance" (Priority 1) | BUA | BUA-CD-01<br>Assign BU Conceptual Design | Accepted |
| | | BUA-PF-02<br>BU Assignment Rules Maint Process Flow Diagram | ReadyForReview |

For example (2):

| BizReqID | Pri | Major Area | DevTstItems DelivID | Deliv Name | Status |
|---|---|---|---|---|---|
| BR_LR_01 | 1 | BUA | BUA-CD-01 | Assign BU Conceptual Design | Accepted |
| BR_LR_01 | 1 | BUA | BUA-DS-02 | Bargaining Unit Assignment DB Modification Description | Accepted |
| BR_LR_01 | 1 | BUA | BUA-PF-01 | Derive Bargaining Unit-Process Flow Diagram | Accepted |
| 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 |

| BizReqID | Pri | Major Area | DevTstItems DelivID | Deliv Name | Status |
|---|---|---|---|---|---|
| BR_LR_09 | 1 | BUA | BUA-TC-021 | BU Assignment Rules Maint TestCase: Add New Rule (Associated Job Class Does Not Exist) - Success | ReadyForReview |
| BR_LR_09 | 1 | BUA | BUA-TC-027 | BU Assignment Rules Maint TestCase: Modify Rule - Success | ReadyForReview |
| BR_LR_09 | 1 | BUA | BUA-TC-035 | BU Assignment Rules Maint TestCase: Add New Rule (Associated Job Class Does Not Exist) - Error Condition | ReadyForReview |
| BR_LR_09 | 1 | BUA | BUA-TC-049 | BU Assignment Rules Maint TestCase: Modify Rule - Error Condition | ReadyForReview |

For example (3):

| BizReqID | CD01 | CD02 | CD03 | CD04 | UI01 | UI02 | UCT01 | UCT02 | UCT03 | TC01 | TC02 | TC03 | TC04 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BR_LR_01 |  |  | X |  | X |  | X |  |  | X |  | X |  |
| BR_LR_09 | X |  |  | X |  | X |  |  | X |  | X |  | X |
| BR_LR_10 | X |  |  | X |  |  |  |  | X |  | X |  |  |
| BR_LR_11 |  | X |  |  |  |  |  |  |  |  |  |  |  |

## Appendix E. 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.