**Software Requirements Specification**

**for**

Online Course Management System

**Version 1.0 approved**

**Prepared by**

**Meenakshi Ashokan**

**Musafir KY**

**N Niranjan**

**Adi Shankara Institute of Engineering and Technology**

**<date created>**

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1. Introduction 1**

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

**2. Overall Description 2**

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

**3. External Interface Requirements 3**

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

**4. System Features 4**

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

**5. Other Nonfunctional Requirements 4**

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

**6. Other Requirements 5**

**Appendix A: Glossary 5**

**Appendix B: Analysis Models 5**

**Appendix C: To Be Determined List 6**

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

# **Introduction**

## **Purpose**

*A course management system is a collection of software tools providing an online environment for course interactions. It typically includes a variety of online tools and environments, such as: An area for faculty posting of class materials such as course syllabus and handouts. Video description along with the corresponding link can be uploaded.*

## **Document Conventions**

Times New Roman font for the entire report

Chapter/Section Title – Times New Roman18, Bold;

Heading 2 – Times New Roman16, Bold;

Heading 3 – Times New Roman14, Bold;

Body- Times New Roman 12, Normal.

## **Intended Audience and Reading Suggestions**

*The document is intended for the people of following profession:*

* *Project managers-Project managers are those who supervise the entire project.*
* *Implementers or coding expertise-This category of professionals implements the design stated by the developers using programming languages. They are responsible for all the application modules and graphical user interfaces.*
* *Tester- This class of people test the developed system with the help of certain test cases and determine the efficiency and estimates the performance of the system.*
* *Documentation writers-Documentation writers prepare the user manuals and other necessary documents for proper setting of the system in a certain operating environment.*
* *Users-The people who want the system in their Educational Institutes. They are responsible for the quality of software requirement specification documents through their valuable comments on the initial requirement documents.*

## **Product Scope**

*An Online Course Management System is a software application for the administration, documentation, tracking, reporting and delivery of educational courses or training programs. LMS tools were created for making knowledge-sharing easier. The whole system was created to work across different web-based platforms.*

## **References**

*www.softwaresuggest.com*

# **Overall Description**

## **Product Perspective**

*This requires an internet connection to display content and allow users to interact with it. All system information is stored in a database, which is located on a web-server. This requires an internet browser (Chrome, Firefox, or Edge) to be accessed*

## **Product Functions**

***Super Admin:***

* *He/She can add institutions*
* *He/She can add an admin for an institution*

***Admin:***

* *He/She can add Faculty/Professor*
* *He/She can add courses*

***Faculty:***

* *He/She can Upload course materials*
* *He/She can add students*
* *He/She can Upload assignment/Tests*

***Students:***

* *He/She can view materials*
* *He/She can do assignments*
* *He/She can view attendance report*

## **User Classes and Characteristics**

*The student is expected to be internet literate once they can log in the system and navigate between webpages they can use the basic functionality of the system.*

*Faculty expected to be internet literate and to be able to use more complex functionality of the system.*

*<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>*

## **Operating Environment**

*<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>*

## **Design and Implementation Constraints**

1. *The system must run in the Windows operating system environment.*

1. *Login and password is used for the identification of users.*

1. *Only registered users, corporate users and admin will be authorized to use the services.*

*<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>*

## **User Documentation**

*<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>*

## **Assumptions and Dependencies**

*<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>*

# **External Interface Requirements**

## **User Interfaces**

*Interface between the software product and the users include screen layout which provides standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on.*

*<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>*

## **Hardware Interfaces**

*Minimum requirements:*

*client side :*

*server side :*

*Recommended requirements :*

*client side :*

*server side :*

*<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>*

## **Software Interfaces**

*<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>*

## **Communications Interfaces**

*<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>*

# **System Features**

*<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>*

## C**reating courses**

*<Don’t really say “System Feature 1.” State the feature name in just a few words.>*

4.1.1 Description and Priority

Integration with registration shall periodically upload the latest registrar's classes list to determine courses that are offered in the current semester.

*<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>*

4.1.2 Stimulus/Response Sequences

The system shall generate courses for each class that registered and determine the current set of students that enrolled in that class.

*<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>*

4.1.3 Functional Requirements

*The system shall allow course instructor to update course content*

*<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>*

*<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>*

REQ-1:

REQ-2:

## Grade Management

Allow grades to be entered online: The system shall allow instructors to enter and modify grades online.

Allow students to access their grades online: The

system shall allow students to log in their account and check their grades at any time.

The system shall provide statistical information such as averages, standard deviation, median about students grades.

3.2.2.4 Track and Handle Re-grade Requests: The system

shall be able to track and handle requests for regrades, and all information about re-grades shall be available to the student, and the course instructor.

4.3 Homework Submissions

Accept submissions in multiple formats: The system shall accept submissions in multiple formats, including .zip, .cpp, .txt, .doc, etc.

Support for late submissions: The system shall provide information about late submissions, and also disallow submissions after a certain period of time.

4.4 online quizzes

The system shall let instructors upload quizzes.

The system shall allow instructor to upload answer

key to the system.

The system shall allow students to answer quizzes.

The system shall compare answer key with

student answer.

5.5 Create Accounts

The system shall automatically create accounts for each class.

Create one account for course instructor regardless to the number of classes that he/she teach.

The account username is course name and its number.

The account password is set by the faculty .

Create one account for each student that registered in this class.

The account username is course name and its number

Faculty can modify student grades from their account.

# **Other Nonfunctional Requirements**

## **Performance Requirements**

*Response Time*

*Average response time shall be less than 2 seconds.*

*Throughput*

*The system shall accommodate 1000 booked per minute.*

*Recovery Time*

*In case of a system failure, redundant system shall resume operations within 30 seconds. Average repair time shall be less than 1 hour.*

*Start-up/Shutdown Time*

*The system shall be operational within 1 minute of starting-up.*

*Capacity*

*The system accommodates 40 concurrent users.*

*Utilization of Resources*

*The system shall store in the database no more than ten thousand transactions. If the database grows over this limit, old transactions shall be backed up and deleted from the operational database.*

*<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>*

## **Safety Requirements**

*<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>*

## **Security Requirements**

*<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>*

## **Software Quality Attributes**

*<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>*

## **Business Rules**

*<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>*

# **Other Requirements**

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

**Appendix A: Glossary**

*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>*

**Appendix B: Analysis Models**

*<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams*.>

**Appendix C: To Be Determined List**

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*