**System Requirements Specification**

**Project Validator**

**03/10/2012**

**Table of Contents**

**1. INTRODUCTION 3**

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, acronyms, and abbreviations 3

1.4 References 4

1.5 Overview 4

**2. CurrENT SYSTEM 5**

2.1 HISTORY 5

2.2 COMPARISON OF EXISTING SYSTEM 5

**3. PROPOSED SYSTEM 6**

3.1 PROBLEM STATEMENT 6

3.1.1PRODUCT POSITION STATEMENT 6

3.1.2 PRODUCT OVERVIEW 6

3.1.2.1 PRODUCT PERSPECTIVE 7

3.1.2.2 SUMMARY OF CAPABILITIES 7

3.1.2.1 ASSUMPTIONS & DEPENDENCIES 7

3.1.3 FUNCTIONAL REQUIREMENT 8

3.1.3 NON-FUNCTIONAL REQUIREMENT 8

3.1.3 STAKEHOLDER & USER DESCRIPTION 8

3.1.3.1 STAKEHOLDER SUMMARY 8

3.1.3.2 USER SUMMARY 9

3.1.3.3 USER ENVIRONMENT 9

3.1.3.4 KEY STAKEHOLDER/USER NEEDS 10

3.2HARDWARE CONSIDERATIONS 10

3.2.1 SYSTEM REQUIREMENTS 10

3.2.2 ENVIRONMENT REQUIREMENTS 11

3.3ERROR HANDLING & SYSTEM MODIFICATIONS 11

3.4SECURITY ISSUES 11

**System Requirement Specification Outline**

**Introduction**

The purpose of this document is to collect, analyze and define high-level needs and features of the Project Validator application. It focuses on the capabilities needed by the stakeholders and the target users, and why these needs exist. The details of how the Project Validator software fulfills these needs are detailed in the use-case and supplementary specifications.

**Purpose**

Project Validator is used by educational institutes to validate the topics of projects and seminars. This application would avoid the growing problem of plagiarism existing within the institutes related to the seminar or project topic selection or submission. Project validator would also provide adequate measure for storing all the seminar as well as project topics into the institute’s database for future reference as well as for overcoming the problem of plagiarism. It would also provide a Graphical User Interface which would make the software user friendly even for the lame man to use.

**Scope**

This application can be used to store as well as cross-check the repetition of ideas for Project as well as the Seminar topics. The scope of this application is only limited to Computer Science Engineering. It can be used to store and check mini-project plagiarism as well as T.E. seminar topic repetition and the most important of them all: the B.E. final year plagiarism. Limitation of this software is that it would be useful only for the Computer Science Engineering department of the educational institutes.

**Definitions, Acronyms and Abbreviations**

**Definitions**:

1. Plagiarism : an act or instance of using or closely imitating the language and thoughts of another author without authorization and the representation of that author's work as one's own, as by not crediting the original author
2. Database Server : A database server is a [computer program](http://en.wikipedia.org/wiki/Computer_program) that provides [database](http://en.wikipedia.org/wiki/Database) services to other computer programs or [computers](http://en.wikipedia.org/wiki/Computer), as defined by the [client–server](http://en.wikipedia.org/wiki/Client%E2%80%93server) [model](http://en.wikipedia.org/wiki/Software_modeling)
3. Terminal: (Computers) any device for entering information into a computer or receiving information from it, as a keyboard with video display unit, either adjoining the computer or at some distance from it.

**Abbreviations:**

1. DBMS: Database Management Systems.
2. JSP: Java Server Pages.
3. HTML: Hyper Text Markup Language.
4. CSS: Cascading Style Sheets.

**References**

1. “A Handbook for Deterring Plagiarism in Higher Education**”** by Jude Carroll, Oxford: Oxford Centre for Staff and Learning Development, 2002, 96pp. ISBN: 1-873-57656-0.
2. “Jsp 2.0: The Complete Reference” , Tata McGraw-Hill Education, 01-Jan-2003
3. “Advanced Java ”, Document Version 1.2, June 2006, University of Durham, Information Technology Service.
4. “Database System Concepts”, Fifth Edition by [Avi Silberschatz](http://www.cs.yale.edu/homes/avi), [Henry F. Korth](http://www.lehigh.edu/%7Ehfk2/hfk2.html) and [S. Sudarshan](http://www.cse.iitb.ac.in/%7Esudarsha)

**Overview**

This SRS document provides us with a complete information about the proposed Project Validator application with respect to the current system, how the proposed application would be helpful, summary of capabilities of the application, assumptions and dependencies for the target system. This document majorly focuses on the functional and the non-functional requirements. This document also provides us with information about the stakeholder and user description as well as requirements, hardware requirements & error handling & various security issues associated with the project.

1. **Current System**

In the present scenario this system is not available at the educational institute level. Currently repetition of topics is avoided through manual system in which all the topics are saved in the institute database and validation is done manually. Some high level softwares do exist on an industrial level but no such implementation of a software has so far been found on an educational institute level.

**History**

The main source of idea behind this project has been the plagiarism of the modern European art forms. This has led to a lot of ambiguity among the art appreciators and has led to severe investigations into finding the true original art form, thus wasting a lot of efforts as well as finance. This project has the same basis and hence avoids the plagiarism among the college projects by adopting certain standard procedures.

**2.2Comparison of Existing System**

Currently repetition of topics is avoided through manual system in which all the topics are saved in document and validation is done manually. For the large number of topics this system is inadequate to use. The scope of the application is limited as compared to the similar system already in use which can be applied to any trade of engineering. Project Validator application will overcome all this drawbacks through a complete automated system.

1. **Proposed System**
   1. **Problem Statement**

Many educational institute currently face the problem of validating the topics due to large number of topics and no effective provision for maintaining complete data . Also it’s difficult to check and compare all the topics in the file. A successful solution to this problem would be to provide a software which would accept the topics from the user and check whether topic is repeated or not by comparing them with the topics present in the database.

**Product Position Statement**

|  |  |
| --- | --- |
| For | Educational Institutes with annual project and/or seminar submission by its students. |
| Who | Provide projects or seminar to its students within a pre-specified range of domain topics. |
| The Project Validator | Is a server based application which will deployed on a server and can be accessed through any terminal within the local network of the institute through any web browser. |
| That | Will make it easier for the staff within the institute to register the seminar and project topics efficiently as well as to retrieve them whenever required. The main aim of the application would be to avoid plagiarism within the institute in accordance to seminar and projects. |
| Unlike | The normal approach of filling up and submitting the project/seminar registration form manually by the students as well as the staff. The application would allow online registration and would indicate any repetition if it exists. |
| Our product | Project Validator would provide online registration of topics through an interactive GUI and would indicate the staff about the repetition before submitting the topic on the spot itself. |

**Product Overview**

The Project validator will have an interactive graphical interface. It will be a menu-driven application which will contain the list for the online registration of the mini-project, seminar as well as B.E. final year project topics. It will indicate the user about the repetition in the idea of the project being implemented at the time of submitting the topic itself. This would allow the students to think over their suggested topics and make any changes necessary in the project to avoid plagiarism.

**Product perspective**

The software being built is a self-contained application. It depends on the Server applications like The Apache Server for its functioning. Initially while setting up the software into the institutes ERP it needs to be updated or integrated with the previous seminar/project topics present within the institutes database.

**Summary of Capabilities**

**Project Validator**

|  |  |
| --- | --- |
| Customer Benefit | Supporting Features |
| Students do not waste time and energy | Project Validator offers online registration for the seminar/project topic thereby saving students’ efforts of getting the registration and filling it manually. |
| Institute staff need not to worry about plagiarism | Project Validator validates the students’ topics before submitting the application itself. This avoids the staff’s burden of manually cross-checking the database for repetition of topics. |
| Students need to worry about plagiarism | Once the topic gets submitted by the application/software, then the students can be sure of its approval and thus can concentrate on their projects/seminars right from the time of submission. |
| Maintainers at ease | Project Validator would provide a separate user account with certain privileges for the system maintainers so as to simplify their work. |
| Administrators at ease | Similar to the maintenance, the application would provide a separate user account for the admins of the system. This classification of the user accounts ensures the security of the system. |

**Assumptions & Dependencies**

**Assumptions**:

The institute would have a local area network setup within the entire infrastructure so that the application can be accessed from any part of the campus.

The institute would a centralized server which would be backed up by subordinate servers.

The centralized server would have power supply for 24 hrs.

The maintenance team would update & maintain the application whenever required.

**Dependencies**:

All the other terminals within the institute should an up-to-date web browser (IE/Mozilla).

The centralized server should have atleast 17” display.

The centralized server should be installed with the supported operating systems.

**Functional requirement**

1. The application should be menu-driven.
2. It should provide online registration for seminar, mini-project as well as final year project topics.
3. It should indicate about plagiarism before submitting the student application.
4. It should be sustainable to any academic changes within the complex.
5. It should also provide instructions on how to use the application.
6. It should make a provision for searching any desired seminar/mini-project/final year project topic.
7. It should make a provision for modifying any desired seminar/mini-project/final year project topic.
8. It should make a provision for displaying all seminar/mini-project/final year project topics for a particular academic year.
9. It should abide by the institute rules concerned with the seminar/project topic submission.
10. It should contain some games.
11. It should display important notices regarding seminar/projects.

**Nonfunctional requirement**

* + 1. It should have a HD resolution.
    2. It should have a response time of less than 1 second.
    3. It should sustain a load of about 200 simultaneous user logins.
    4. It should be touch-sensitive.
    5. It should be user-friendly.

**Stakeholder and User Descriptions**

**Stakeholder Summary**

|  |  |  |
| --- | --- | --- |
| Name | Represents | Role |
| Institute Trustees | These people provide the contract for the application development. They are the main investors in the project. | They specify the requirements or the expectations form the project. They also provide the capital and/or the resources required for the project. |
| Chairman & CEO of the contracted company | These people are the heads or the representatives of the company which is responsible for developing the project. | They officiate as well as monitor the entire project development process. |
| Project Developers | These people actually develop the project. | They communicate with the users. They also plan, model, construct and deploy the project. |

**User Summary**

|  |  |  |
| --- | --- | --- |
| Name | Description | Stakeholder |
| Institute Staff | These are the people who will be using the software for the registration purposes. | As the institute trustees are responsible for the convenience of the staff, these people are represented by the institute trustees, that is, stakeholder 1. |

**User environment**

• The main tasks associated with the Project Validator application are its use for finding the repetition of ideas I seminars/projects, its maintenance and its administration.

• The number of users vary but the number of maintainers and administrators remains fixed for their respective tasks.

• The time required for filling up the online registration form depends upon the user. The time for calculating the keywords associated with the project and then validating is usually very less. The maintenance time depends on the type of the maintenance to be carried out. Usually the administrative work is the quite complicated and time consuming of all the tasks.

• Mainly the hardware requirements are the major environmental constraints for the deployment of the application.

• As of today, the systems quite similar to our application are: Online Plagiarism Checker, Viper-Plagiarism Checker, Plagtracker,etc.

• The software being built is a self-contained application. It depends on the Server applications and needs to be updated or integrated with the previous seminar/project topics present within the institutes database.

.

**Key Stakeholder / User Needs**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Need | Priority | Concerns | Current Solution | Proposed Solutions |
| Checking Repetition of Ideas | High | Users | Manual searching of the topics by obtaining a list of the entire topics from the database. | Project Validator would check the repetition of ideas through the standard methods for plagiarism checking. |
| Displaying previous year’s seminar topics | High | User | Manual searching of the topics by obtaining a list of the entire topics from the database | The application would provide the entire list of the previous year’s seminar topics just on the single click of a button. |
| Important Dates | Low | Users/Indirect Users(Students) | Notices are put up on the notice boards regarding the important dates and deadlines | There will be a special option/marquee in the proposed application regarding important dates. |

* 1. **Hardware Considerations**
     1. **System requirements**:

1. The institute should have a local area network accessible anywhere within the institute.

2. The centralized server should be installed with the recommend Server applications..

3. The centralized server should have either Windows, Linux(Red Hat) or Macintosh OS.

4. The centralized server should be backed-up by 24 hrs power supply.

5. The centralized server should have atleast 4GB RAM.

6. The centralized server should have atleast a dual core processor working atleast on 2.2GHz.

7. The centralized server should have atleast 20K cache memory.

8. The terminals from where the application is going to be accessed must have an up-to-date web browser(IE/Mozilla) with flash support.

* + 1. **Environment requirements**:

1. To use the software it is mandatory for the individual to register himself/herself with the application’s database.

2. There should be regular maintenance to avoid any harm to the software.

3. There should be atleast one system administrator so that the system can be modified in case of changes.

4. The application should be robust, simple to use and should be user friendly.

* + 1. **Error Handling & System modifications**

At the developer side the system would be made error free by repetitive testing. During the time of development the system would be thoroughly checked for all the possible correct as well as incorrect inputs. The system would be made self-sustainable to these kinds of inputs from the user in the future. It would be made sure that the system does not fail or crash due to any unacceptable inputs. Instead the system would prompt the user in case of any invalid input. The untimely shutting down of the terminal wouldn’t affect the software & the process would be restored on turning the application ON.

The system would be easily upgradable or modifiable. There will be maintenance as well as administration teams available in case of any system modification. If the load on the server increases then system would retain the admin and password.

**3.4** **Security Issues**

For the sake of security of the application, all the external ports on the terminal will be disabled. These ports can be enabled only by the maintainers as well as the system administrators for any maintenance or administration work. The terminals would provide the following three login accounts:

* + - 1. **User Login**: The daily visitors as well as the application owner can use this account. This account allows its users to run the Project validator application. It does not allow any modifications to be done to the software or the terminal.
      2. **Maintainer Login**: This account is accessible to the system maintainers as well as the system administrator. This account allows its users to run the application, modify it upto a certain extent as well as to enable the external ports on the terminal. It prohibits any change in the configuration files of the application as well as the soft wares on which the application is dependent.
      3. **Administrator Login**: This account allows complete access to the application as well as the terminal to its user. It is only accessible by the system administrator.