

# **Software Requirements Specification**

**For**

**“On-line Form Management System”**

**Version 1.0**

**Prepared by**

**Team-The Web**

**Organization: PESIT**

**Dated: 20-03-2015**

## Table of Contents

### Introduction

#### Purpose

#### Intended Audience

### Overall Description

#### Product Perspective

#### Product Features

#### User Classes and Characteristics

#### Operating Environment

#### Design and Implementation Constraints

#### Assumptions and Dependencies

### System Features

#### User Login/Registration Feature.

#### Description and Priority

#### Priority :- HIGH

#### Support for different User Classes.

#### Description and Priority

#### Priority :- HIGH

### External Interface Requirements

#### User Interfaces

#### Hardware Interfaces

#### Software Interfaces

#### Communications Interfaces

### Other Nonfunctional Requirements

#### Performance Requirements

#### Safety Requirements

#### Security Requirements

#### Software Quality Attributes

### Other Requirements

# 1. Introduction

## 1.1 Purpose

*The software product named “**Form Management System**” is intended for the Administration department of any organization to help the users(administrators, evaluators and record-keepers) to easily manage (create, edit and save) the application forms online.*

## 1.2 Intended Audience

*This document is intended for the system analysts, system designers, developers and clients to understand the behaviour of the software. This document provides the information about the environment in which the software product will be used, software tools and technologies to be used for development and the features that the software provides for end-users.*

## 2. Overall Description

### 2.1 Product Perspective

*In any organization, any of the transactions will involve the most frequent task of filling up of (paper) forms and submitting them. For ex:- A Bank requires the users to fill up of forms for some or the other reason, An Educational Institution requires the staff to fill up attendance forms of students etc.. It is difficult to manage all the filled forms and requires a lot of human effort. To make it easier to manage forms, a software product that automates and manages this process is required. This software product is intended for the same purpose.*

### 2.2 Product Features

*The major features that the software must provide are :-*

- *Create (By Administrator(s))*
- *Edit (By Evaluators)*
- *Save (By Evaluators)*
- *Retrieve (By Record-keepers)*
- *Delete (By Evaluators)*

### 2.3 User Classes and Characteristics

*Different categories of users who use the product include :-*

- *Administrators .*
- *Evaluators.*
- *Record-Keepers.*

### 2.4 Operating Environment

*This online web application where any user who has browser installed on his work-station can use the product via internet. ex:- Firefox 3.5 and above, Google Chrome.*

*Web Server components must run on machines with Windows / Linux Operating systems.*

### 2.5 Design and Implementation Constraints

*The product shall be developed for and on LINUX environment.*

- *Languages and tools to be used for development include HTML, CSS, Javascript, Python, MySQL DB, cherryPy Server. // Aren't we supposed to code it in java?*

- *Communication Interfaces :- HTTP N/w Protocol with SSL/TLS connection.*

## **2.6 Assumptions and Dependencies**

- *Third Party components which might be used include cryptographic libraries, web servers (may be Apache or Turbogears etc..)*
- *Performance of the system may be affected by the limited Internet connectivity speed.*

## **3. System Features**

### **3.1 User Login/Registration Feature.**

#### **3.1.1 Description and Priority**

*The system must allow registered users to login to the system before usage and the administrator can add new users and specify the class to which the user belongs.*

#### **3.1.2 Priority :- HIGH**

### **3.2 Support for different User Classes.**

#### **3.2.1 Description and Priority**

*The system must support 3 types of users.*

- *Administrators :-  
Who manage creating forms, add new users, authorize user permissions, modify users, remove users.*
- *Evaluators :-  
Who can edit and save/submit form(s). An evaluator cannot edit and sign other evaluators forms.*
- *Record-Keepers :-  
Who can retrieve the saved forms and print the forms whenever required.*

#### **3.2.2 Priority :- HIGH**

#### **3.2.3 Requirements List:-**

### **3.2.3.1 Administrator Module Operations List.**

- Admin should be able to create modify and remove users.
- Admin should be able to specify access privileges to the user.
- Admin should send notification to the user based on a new form filling.
- Admin should submit the records to a database for easy access and retrieval.
- Admin should ensure that all details in the form is filled and no data field is left empty.
- Admin should provide random key while user gets register.
- Admin should use Asymmetric algorithm for encrypting digital signature.

### **3.2.3.2 Evaluator Module Requirements List.**

- Upon notification from the administrator, the user should be able to open, edit and save the form
- It should display the form of intended user and not others' forms
- It should allow the user to insert digital signature to prove the authenticity of the user
- It should submit the form upon successful completion

### **3.2.3.3 Record-Keeper Module Requirements List.**

- The user should be able retrieve the data by querying the system whenever required.
- Should be able to print the forms submitted/saved by user.
- User can only visualize the user application forms but can't edit that.
- User(Record-keepers) should be able to notify to users(Evaluators) on pending requests.
- Users shall retrieve the documents based on :-
  - ❖ User or Author id.
  - ❖ Departments.
  - ❖ Submitted date or time.

## **4. External Interface Requirements**

### **4.1 User Interfaces**

*<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.>*

### **4.2 Hardware Interfaces**

*<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.>*

### **4.3 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.>*

### **4.4 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.>*

## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

*<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.>*

### 5.2 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.>*

### 5.3 Security Requirements

*The client and server must be communicate securely over the Internet.*

*The users must be identified and authenticated before accessing the system.*

*The evaluators signature must be kept confidential and secure, no one should be able to access the digital signature.*

*The stored application forms access must be provided only for authenticated users.*

### 5.4 Software Quality Attributes

*The software must be flexible enough to support different kinds of application forms.*

*The software must be available always for the users to access.*

*The software must be easy to maintain.*