**SOFTWARE ENGINEERING (IT300)**

**Software Requirement Specification**

# School Management System

**Submitted by :**

AVP Sahithi - 16IT203
Ch Jamithra - 16IT213
Bachina Sony - 16IT208


**Faculty:** Thanmayee S
Department of Information Technology,
National Institute of Technology-Karnataka
Mangalore, India.

Session 2018-2019

# Table of Contents

# Revision History

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

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to present a detailed description of School Management System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli.This is an interface between Admin, Teacher and Parents of each student.

## 1.2 Document Conventions

*Main Section Titles*

- Font: Times New Roman
  - Face: Bold
  - Size: 14

*Sub Section Titles*

- Font: Times New Roman
- Face: Bold
- Size: 12

*Other Text Explanations*

- Font: Times New Roman
- Face: Normal
- Size: 12

*<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>*

## 1.3 Intended Audience and Reading Suggestions

The SRS is intended for several audience, including the customer, as welll as the project managers,designers, developers and testers.

- The customer will us this SRS to verify the developer team has created a product that is acceptable to the customer.
- The projects managers of the developer team will use this SRS to plan milestones and delivery date and ensure that the developing team is on track during development of system.

- The designers will use this SRS as a basis for creating the system's design. The designers will continually refer back to this SRS to ensure that the system they are designing will fulfill the customer's needs.
- The developers will use this SRS as a basis for developing the system's functionality. The developers will link the requirements defined in this SRS to the software they create to ensure that they have created software that will fulfill all the customer's documented requirements.
- The testers will us this SRS to derive test plans and test cases for each documented requirement. When portions of the software are complete, the testers will run their tests on that software to ensure that the software fulfills the requirements documented on this SRS. The testers will again run their tests on the entire system when it is complete and ensure that all requirements documented in this SRS have been fulfilled.

## 1.4 Product Scope

It is a well established fact that it takes much and effort to accomplish the task of managing the profiles of all teachers and students manually. Considering the fact that number of students are increasing annually, an automated system becomes essential to meet this need.The System will provide an online interface to the parent and teacher where they can view information and upload content. The authority concerned can use this system to reduce his workload and process the application in a speedy manner.Provides a communication platform between the teacher, parent and the administrator.

## 1.5 References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

Books:

Websites:

*<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>*

# 2. Overall Description

## 2.1 Product Perspective

*<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>*

## 2.2 Product Functions

*<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>*

Functionalities of an Admin:

| S.No | Function | Description |
|---|---|---|
| 1 | Add Student | The admin can add student information(details) by editing or updating database |
| 2 | Add Teacher | The admin can add teacher information(details) by editing or updating database |
| 3 | Add Course | The admin can add course information by editing or updating database |
| 4 | View Details | The admin can view all student, course, teacher details by collecting information from database |
| 5 | View Feedback | The admin can view all teacher feedback given by parents by collecting information from database |
| 6 | View Suggestions | The admin can view all suggestions given by parents and teachers details by collecting information from database |
| 7 | Upload notices/events | The admin can upload events or notices regarding school which all parents and teachers details by collecting information from database |

| | | |
|---|---|---|
| 8 | Login/Logout | Admin can login and logout of the application. |

Functionalities of a Teacher:

| S.No | Function | Description |
|---|---|---|
| 1 | Mark attendance | The teacher can upload students attendance % which will be uploaded in database |
| 2 | Mark grades | The teacher can upload students grades which will be uploaded in database |
| 3 | View Notices | The teacher should be able view notices/events uploaded by admin by getting information from database |
| 4 | Give complaints | The teacher can give complaints about their students which will be uploaded in database |
| 5 | Give Suggestions | The teacher can give suggestions about school which will be uploaded in database |
| 6 | Login/Logout | The teacher can login and logout from his session |

Functionalities of a Parent:

| S.No | Function | Description |
|---|---|---|
| 1 | View attendance | The parent should be able to view his child's attendance % by collecting information from database |
| 2 | View grades | The parent should be able to view his child's grade card by collecting information from database |
| 3 | Give feedback | The parent should be able to give feedback about his child's teacher which will be uploaded in database |
| 4 | View complaints | The parent should be able to view complaints about his child given by teacher by collecting information from database |
| 5 | View notices/events | The parent should be able view notices/events uploaded by admin by getting information from database |
| 6 | Give suggestions | The parent should be able to give suggestions about school which will be uploaded in database |

| | | |
|---|---|---|
| 7 | Login/Logout | The parent can login and logout from his session |

## 2.3 User Classes and Characteristics

The

*<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.>*

## 2.4 Operating Environment

Operating environment for this school management system can be like this:

- Operating system: Windows, Ubuntu
- Database: MySQL
- Distributed database
- Platform: XAMPP

*<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.>*

## 2.5 Design and Implementation Constraints

*<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).>*

## 2.6 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.>*

## 2.7 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).>*

# 3. External Interface Requirements

## 3.1 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.>*

## 3.2 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.>*

## 3.3 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.>*

## 3.4 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.>*

# 4. 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.>*

## 4.1 System Feature 1

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

### 4.1.1 Description and Priority

*<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

*<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

*<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:

### 4.2 System Feature 2 (and so on)

# 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

*<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.>*

## 5.4 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.>*

## 5.5 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.>*

# 6. 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.>*