**#1 Objective:**

Student Management System (S.M.S.) is Application Software design to Introduce Structured Information exchange Environment for Integrating Students, Teachers, and Administration Staff of college.

Major Objectives of Student Management Systems

Student information systems must meet the following objectives:

* Enable self - service for students to perform basic functions and tasks in a “one - stop” service and access environment.
* Integrate data sources and process them through a single function that supports onetime entry of student data.
* Support the retention and recruitment of students and encourage a strong and positive relationship with the College.

**#2 Introduction**

The Student Management System can handle all the details about Students. The details include college details, Course details, Academic details, and Students personal details etc.

The existing system provides some basic facilities to the users in all. However, the proposed system proves to be more reliable in the context that the SMS system provides many other facilities like correction in database, pending dues and other account management, generation of mark sheet, student's private access to view grades, etc. Also the proposed system provides all these facilities under one roof so as to ensure better college management.

So, this system can be a better substitute for the pre-existing system.

**2.1 Purpose**

This software package is developed from scratch exclusively for The Shri G. S. Institute of technology and Science, Indore in order to,

* To maintain/process complete details of the student about their personal and academics Domains.
* To enable student to view their marks and grades.
* To enable professors to upload marks and submit grades.
* To enable Administration staffs to Create and Maintain Information of Students and Faculties.
* To enable Administration Office to Update Fee, Dues, Scholarships etc and related information.

**2.2 Intended User**

* End Users of the Software Developed will be Administration Staff, Faculties, and Students.
* The Software development team for their use in analyzing the requirements.
* The Instructors of the course 'Software Engineering' for their review and monitoring progress of the project.

**2.3 Project Scope**

The scope of the to be developed 'Student management System' software package to provide all types of Courses offered by the administration of SGSITS to its students. It provides need of suitable interface for all students and instructors of an offered course, and enables Administration to view, create, and update information of student and faculties.

Hence the scope of the project is limited at college level only.

Future Extension: Project can be extended to University Level and can be accessed through World Wide Web.

**2.4 Limitations**

* 1. Project can only accessed through college servers.
  2. Limited number of data is handled/managed.
  3. Lack of interactivity and attractive GUI.
  4. Limited number of functionality.
  5. End Users are limited

**#3 Overall Description**

**3.1 Product Prospective**

Student Management System is meant to serve as a common platform where management of everyday academic tasks can be carried out conveniently. Our goal is to develop a replacement to the academic portal used at SGSITS, Indore to much extent making it more users friendly and to promote academic networking among the users.

**3.2 Product Features**

**Modules**

**Registration:**

This module helps to register a new user into the Software Database, and check is made for duplicate entry and generates ID & Password for future access.

**Login / Logout:**

This module take Input as user Id & password for logging into the system and user can easily logout as well.

**Display Details:**

This module will display the Personal and Other details to the logged user.

**Updates / Correction:**

It helps to make some correction in previously stored details that were entered incorrectly during registration. As well also provide necessary updates in database.

**Maintenance:**

It provides each faculty to upload Attendance, Marks of student enrolled.

**Connection to Database:**

Connection to database for storing data which has been used during runtime. All input and output are maintained.

**3.3 Operating Environment**

This software package is expected to work in the following atmosphere:

* OS -Windows, Linux etc.
* Latest Web Browser Support such as Firefox, Chrome.
* MYSQL database and Hosting Apache Server.
* Language Supports such as: PHP, HTML, CSS, JS etc.

**Minimum Specifications :**

1. 512 MB DDR3 Ram

2. Processor @1.6 Ghz

3. Web Browser that supports Html 3+

4. Operating System Window XP or above, Linux etc.

**Recommended Specification**

1. 1GB DDR3 Ram

2. Processor 2.5+ Ghz

3. Latest Web Browser like Firefox 40+ etc.

4. Operating System Windows 7 or above,

Linux latest (Ubuntu 13+) etc.

**3.4 User Classes**

Users of the Student management System are of three categories

* + **Administration Staff :** This is the main user of the Student Management System can create other users ID for Student management System.
  + **Teacher Staff :** This user can handle students records related to marks and attendance.
  + ***Student :*** This user can view their academics records.

**3.5 Design Constraints**

* The communication between the portal software and the database will be in SQL.
* The portal layout will be produced with HTML/CSS.
* The product will be written in PHP.
* System administrators must have access to comprehensive documentation.
* Is designed in Procedural Oriented Programming way.

**#4 System Features**

**4.1 Functional Requirement**

**Registration :**

To register a unique user with their user id and password key along with category (Admin, Faculty, or Student). It takes general information of user like personal details to provide with all features of system independently.

Input: User's details.

Output: Registered, unique ID-password generated.

**Login/Logout :**

This application can only be accessed by those who are registered with the system so Login and Logout facility has to be maintained.

Login: user ID and password / Logout.

Output: Logged in/ Logged out.

**Updates :**

This system can facilitate to update and make changes accordingly.

Input: User ID-password, incorrect data fields and requested corrections.

Output: Updated and corrected data field in the database.

**Generate Mark sheet :**

There must be a facility to generate marks of students as updated by their respective faculties.

Input: Marks of students (by faculties).

Output: Generated mark sheet.

**Attendance :**

Faculty can maintain regular attendance and students can view it easily.

Input: Daily entry of student's attendance.

Output: View attendance sheet by faculty and student.

**Alerts :**

Students are provided messages for their short attendance

and dues if any.

Input: Attendance record list, pending deus list.

Output: Provide alert message to the students.

**Maintain Dues and other details :**

There must be a facility to maintain dues and other details and thus facilitate the updation of the student data.

Input: Due list and other. details.

Output: Updated data of student.

**4.2 Non Functional Requirement**

**1. Security:-**

The system has the facility to provide security of data since login-password is maintained. It also ensures the integrity of system from accidental or malicious damage.

**2. Usability:-**

The system to be developed provides usability of certain data

so that an effective memory management is achieved.

**3. Reliability:-**

The reliability is maintained in the to be developed software,

so that the data as well as all other included operation provide the correct calculations in a reliable form .

**4. Supportability:-**

The system will be supported on various platform and thus ensure that it is platform independent.

**5. Adaptability:**

The system welcomes any new module which provides any new facility to the existing system.

**6. Familiar Interface:-**

The new system will have to be user friendly and familiar to users.

**7. Identification: -** The new system will be able to determine the class of user as Students, Faculty, or Academics.

**Software Development Process Model-**

**Model Used:**

We will use Spiral Model for development of our software.

**Spiral Model Why Incremental Model not used:**

Incremental Model was not used because:

**:**

The spiral model is an evolutionary software process model that couples the iterative nature of prototyping with the controlled and systematic aspects of the waterfall model. In other words, this model is a risk driven process model, having two main features :

One is cyclic approach for incrementally growing a system’s degree of definition and implementation while decreasing its degree of risk. The other is a set of anchor points milestones for ensuring stakeholder commitment to feasible and mutually satisfactory system solutions.

Diagram:

It has four phases of development namely:

1. Planning Phase: This phase involves the feasibility study and requirement gathering in order to ensure the planning of the system.

2. Risk Analysis Phase: This phase involves the risk analysis and corrections, however which is not yet considered in this software.

3. Engineering Phase: This is the phase which includes all the levels that are Design, Coding, Testing, Deployment and Maintenance.

4. Evaluation Phase: In this phase the software is evaluated as per user requirements and other deciding factors.

Each spiral in the model signify each requirement of the user. The spiral (loop) ends when the customer is satisfied, and the software is developed completely.

**Benefits of Spiral model:**

Spiral model has the following benefits:

1. Risk involvement and correction is more.

2. More suitable for large and complex projects.

3. It is used when customer requirements are not fixed.

4. This model can be adopted to apply throughout the entire life cycle of an application, from concept development to maintenance.

5. This model proves to be more reliable for real time softwares where risks are more. For example, all time-bounded softwares like traffic signal sensors, etc.

**Why Spiral Model Used:**

This system uses the spiral model because of certain advantage of this model over other which include:

1. Since the requirements were not fixed, so it was opted to choose this model.

2. As per this model, each requirement (which represent each spiral) can be handled more efficiently and in an organized manner.

3. Since there was limited team members, this model seemed to be more effective.

4. This model enables the developer to apply the prototyping approach at any stage in the evolution of the product.

5. Since this model can be adopted for entire life cycle from concept development to maintenance it proves to be more useful.

**Why Waterfall Model not used:**

Waterfall Model was not used because:

1. This model is more rigid and obslead.

2. It is difficult to respond to changing customer requirements.

3. Back-tracing will mean restarting from feasibility study every time when an error will be encountered.

4. Time calculation estimation is very difficult.

5. The customer must have patience.

6. There is no risk management policy.

**Why Incremental Waterfall Model not used:**

Incremental Waterfall Model was not used because:

1. The error detection and correction will correspond to the change in preceding and succeeding stages.

2. Since there are many dependent tasks (modules), it result in “blocking states”, which in turn affect the overall development of software in terms of cost, time, efficiency, etc.

3. Risks are not addressed in this model.

**Why Prototyping Model not used:**

Prototyping Model was not used because:

1. Requires a rapid prototyping tool.

2. Experts are also required for the prototype development.

3. This model thereby include more cost for the development organization.

4. It provides a look-alike working version but it is not.

5. Risk management is not involved.

**Why Incremental Model not used:**

Incremental Model was not used because:

1. Each phase of iteration is rigid and does not overlap each other.

2. Initial increment can be implemented with fewer people, additional staff are required for next increment.

3. The initial upfront cost is required for parallel development of various modules.

4. Coordination between iteration is required for better quality product.

**Why RAD Model not used:**

RAD Model was not used because:

1. For large, but scalable projects, RAD model requires sufficient human resources for parallel working teams.

2. Automated tools and pre-existing software components are required for automatic code generation.

3. It require dedicated developers for completing the project within the proposed time frame.

4. If a system cannot be properly modularized, building components for RAD will be problematic.

5. RAD may not be appropriate when technical risks are high (e.g., when a new application makes heavy use of new technology).