Software Requirements Specification

for

Student Information System

Prepared by

AMIT(16IT106)

ARUNABHA(16IT109)

HIMABINDU(16IT117)

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This project is to create a technical solution that satisfies the functional requirements for the students.

1.2 Document Conventions

<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

Developers for third party services and government officials.

1.4 Product Scope

The main scope of the system design is :

- *Organize the system into modules*
- *Organize sub-modules for each module*
- *Allocate tasks to processors*
- *Choose an approach to manage data store*
- *Handle access to global resources*
- *Choose implementation logic*

1.5 References

<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

The Student Information System records basic student information, Examination information, education information regarding student. Student management function involves :

- Manage student records
- Student Basic Information
- Manage course and specialty
- Manage semester and year
- Result management
- Subject management

In Student Management System, administrator has a Login ID and Password. Also all the users have different permission rights to access the applications.
There are two main roles in the system. Admin and operator. Admin has complete access to the whole system, while operator is the role that is responsible for the use of the system.

2.2 Product Functions

Functional component 1: Login

- Input: Admin name and password.
- Output: Allows administrator to enter into a new page.
- File I/O interface: Functional components login when clicked after giving the username and password correctly it takes us to a new page which is the home page, with other functional components.

Functional component 2: Student details

- Input: reg.no, first name, last name, dob, fathers name, sex, address, contact no, course, semester.
- Output: Details of each and every students.
- File I/O interface: This module is not related to other functional components.

Functional component 3:Course details

- Input: course id, course name, comment, course key.
- Output: All the entered data will be stored in respective database and will be displayed in the grid.
- File I/O interface: This module is not related to other functional components.

Functional component 4:Subject details

- Input: subject id, subject name, comment, course, subject type, semester.

- Output: All the entered data will be stored in respective database and will be displayed in the grid.
- File I/O interface: This module is not related to other functional components.

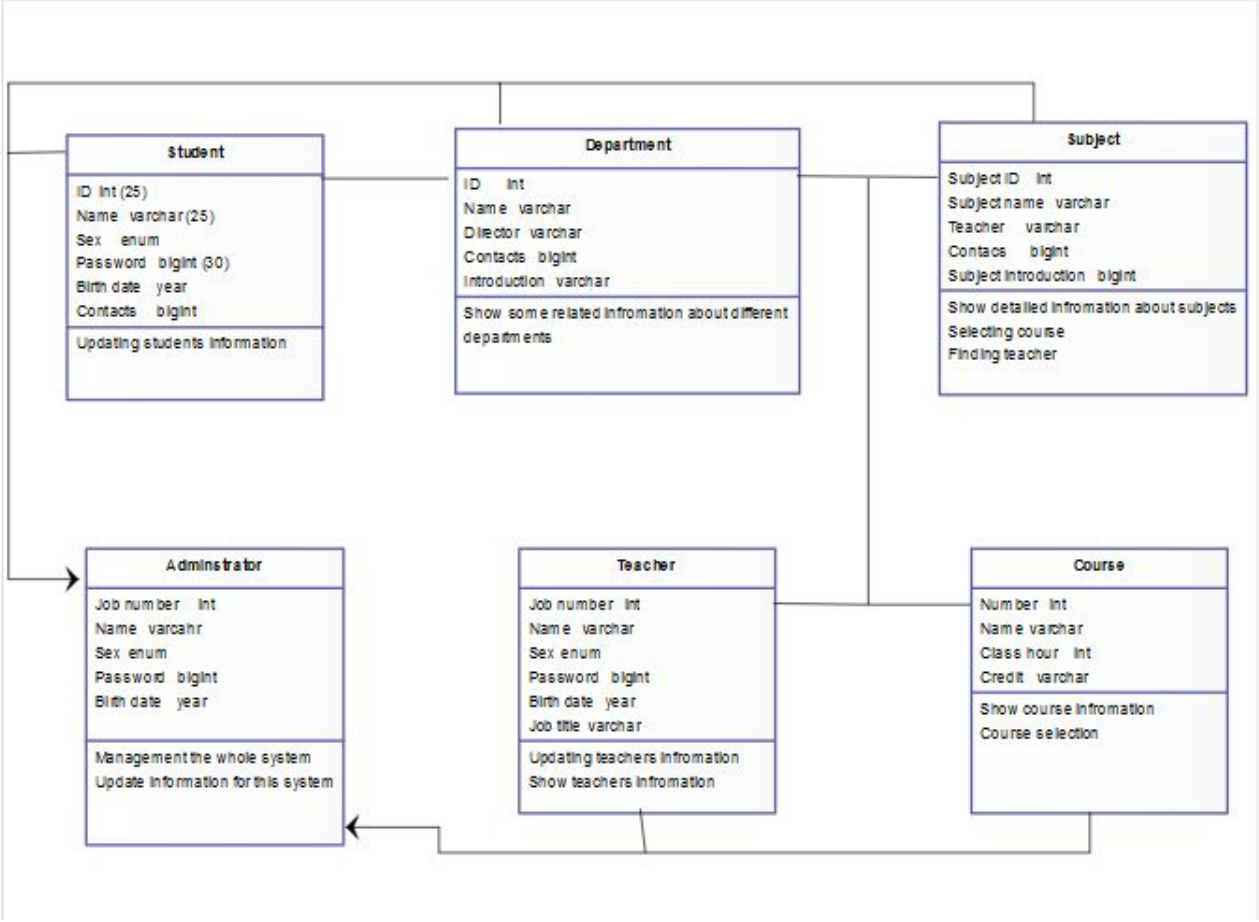
Functional component 5:Examination details

- Input: course, semester, internal type,marks.
- Output: All the entered data will be stored in respective database and will be displayed in the grid.
- File I/O interface: This module is not related to other functional components.

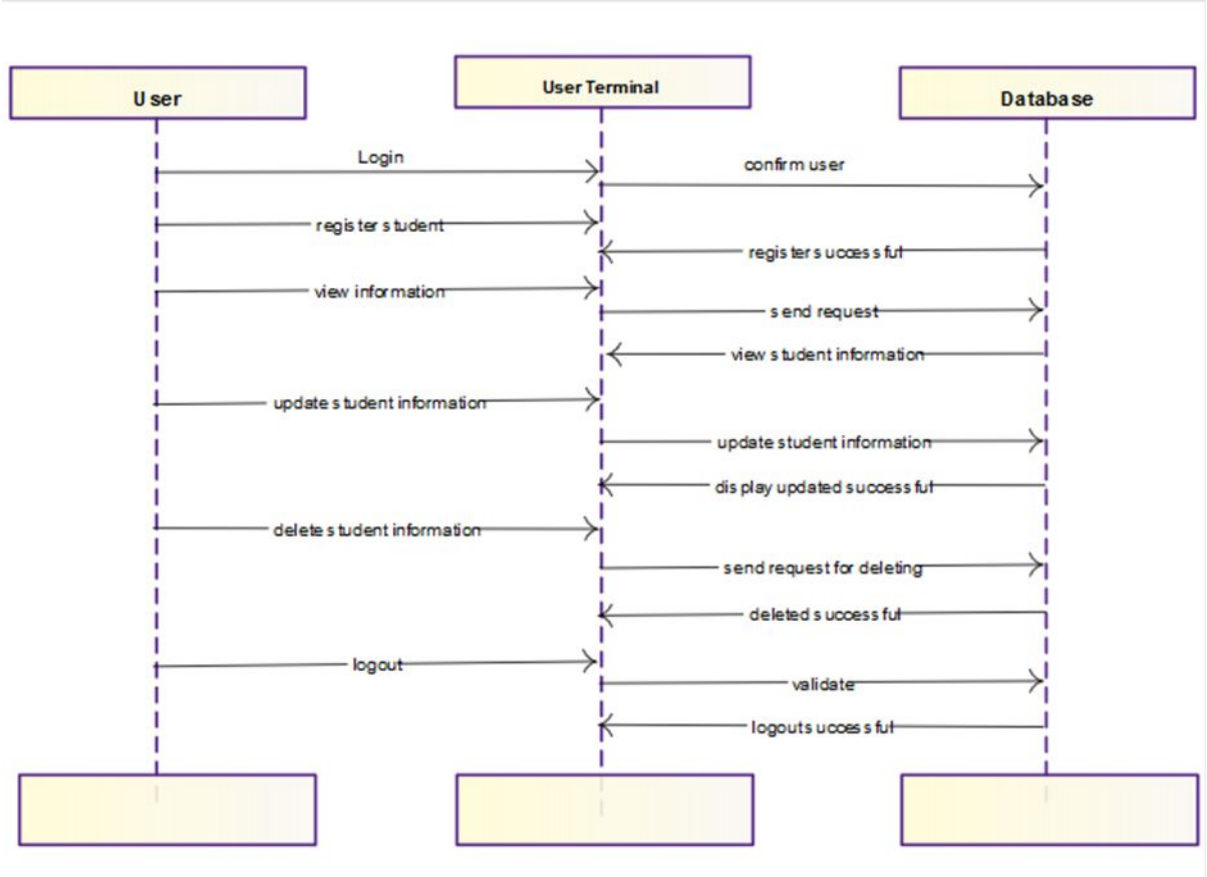
Functional component 6:Attendance details

Input: course, semester, subject, total classes.
Output: All the entered data will be stored in respective database and will be displayed in the grid.
File I/O interface: This module is not related to other functional components.

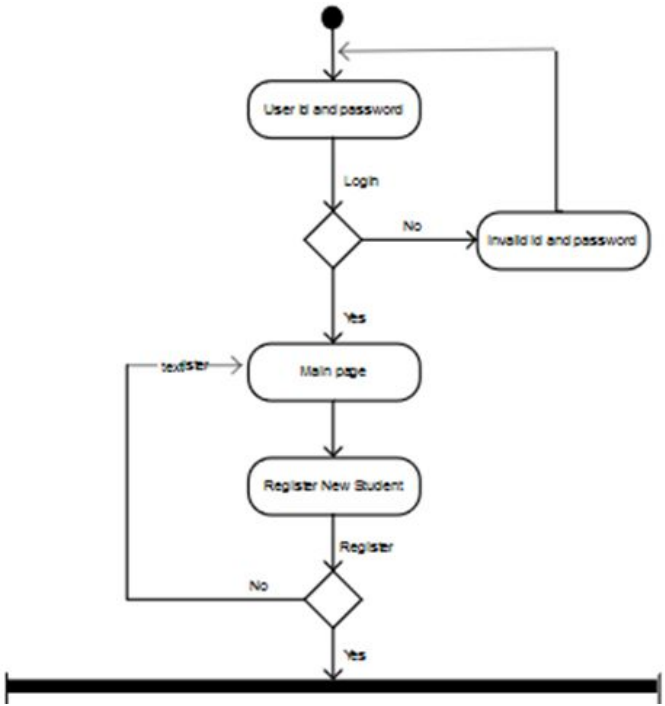
2.3 User Classes and Characteristics

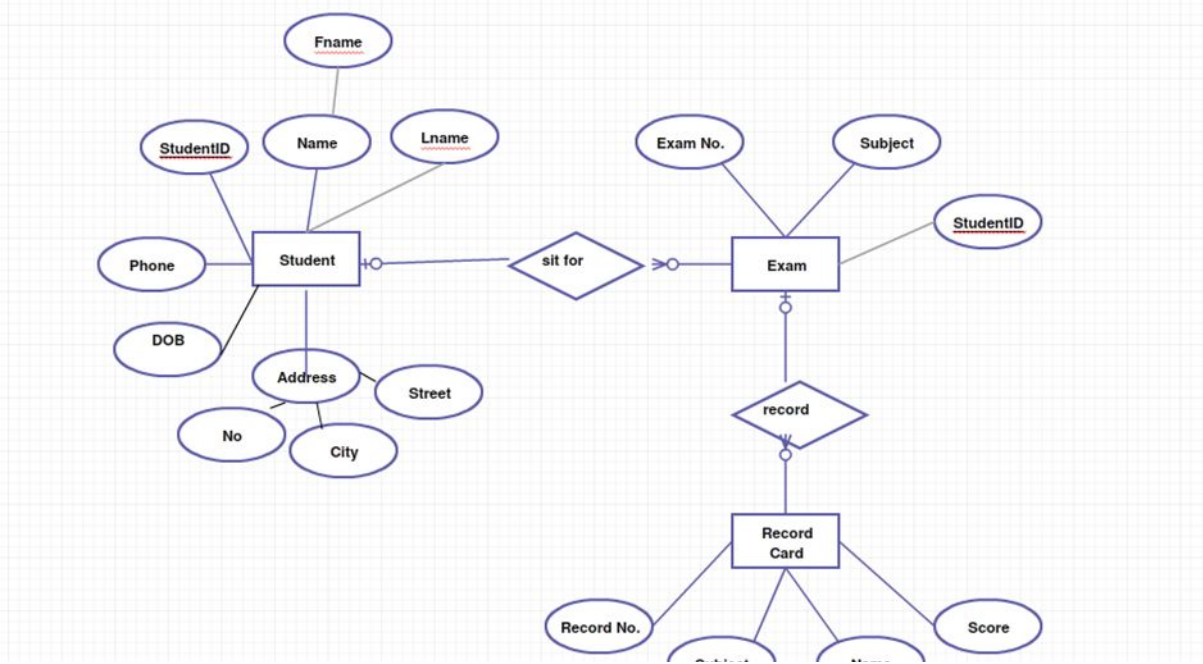


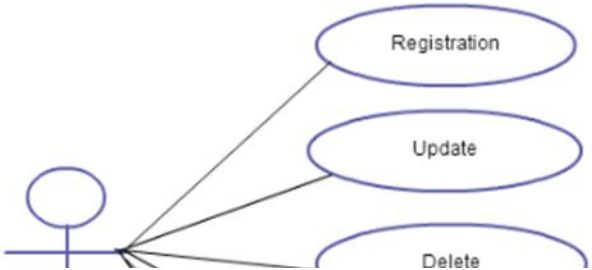
Class Diagram


User
User Terminal
Database
Login
confirm user
register student
register successful
view information
send request
view student information
update student information
update student information
display updated successful
delete student information
send request for deleting
deleted successful
logout
validate
logout successful

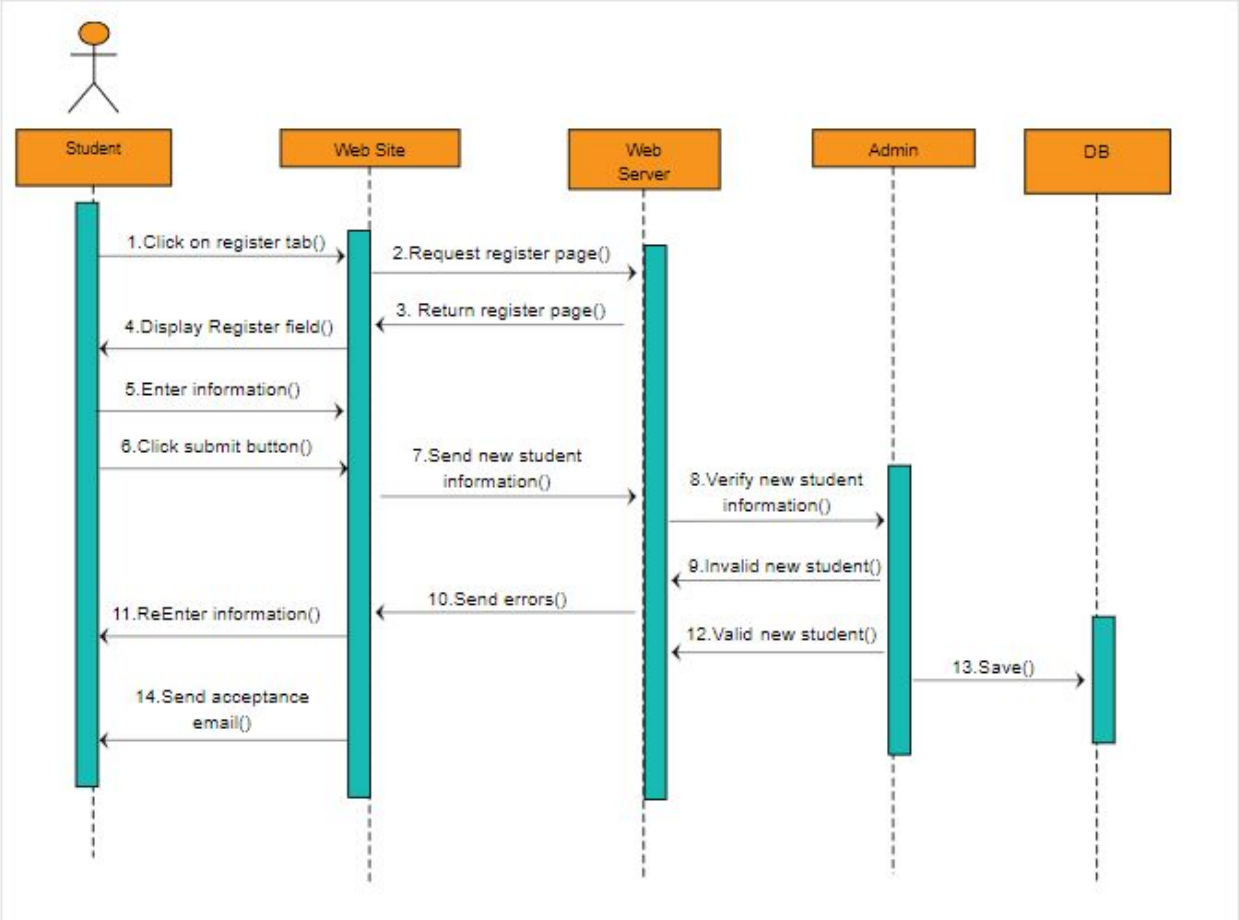
Sequence Diagram




Fname
StudentID
Name
Lname
Exam No.
Subject
StudentID
Phone
Student
sit for
Exam
DOB
Address
Street
record
No
City
Record Card
Record No.
Score


Registration
Update
Delete

Use Case Diagram



2.4 Operating Environment

Hardware Requirements

- *Computer Machine*
- *External Disk*
- *Printer*
- *Printing Material*

Software Requirements

- *Xampp (My-SQL)*
- *Netbeans*
- *Operating system : Windows XP or above*
- *Languages used : Java*

2.5 Design and Implementation Constraints

<Describe any items or issues that wUser 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.>

2.6 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

- *Login*
- *Details of new students*

- *Student Details*

3.2 Hardware Interfaces

The system shall run on Microsoft Windows based system.

3.3 Software Interfaces

The system shall interface with Access database.

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.>