Software Requirements Specification

for

Student Information System

Prepared by

AMIT(16IT106)

ARUNABHA(16IT109)

HIMABINDU(16IT117)

Table of Contents

Table of Contents ii
Revision History iii
1. Introduction 4
1.1 Purpose 4
1.2 Document Conventions 4
1.3 Intended Audience and Reading Suggestions 4
1.4 Product Scope 4
1.5 References 4
2. Overall Description 5
2.1 Product Perspective 5
2.2 Product Functions 5
2.3 User Classes and Characteristics 7
2.4 Operating Environment 12
2.5 Design and Implementation Constraints 12
2.6 User Documentation 12
2.7 Assumptions and Dependencies 12
3. External Interface Requirements 13
3.1 User Interfaces 13
3.2 Hardware Interfaces 13
3.3 Software Interfaces 13
3.4 Communications Interfaces 13
4. System Features 13
4.1 System Feature 1 13
4.2 System Feature 2 (and so on) 15
5. Other Nonfunctional Requirements 17
5.1 Performance Requirements 17
5.2 Safety Requirements 17
5.3 Security Requirements 17
5.4 Software Quality Attributes 18
5.5 Business Rules 18
6. Other Requirements 18
Appendix A: Glossary 18
Appendix B: Analysis Models 18
Appendix C: To Be Determined List 18

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.

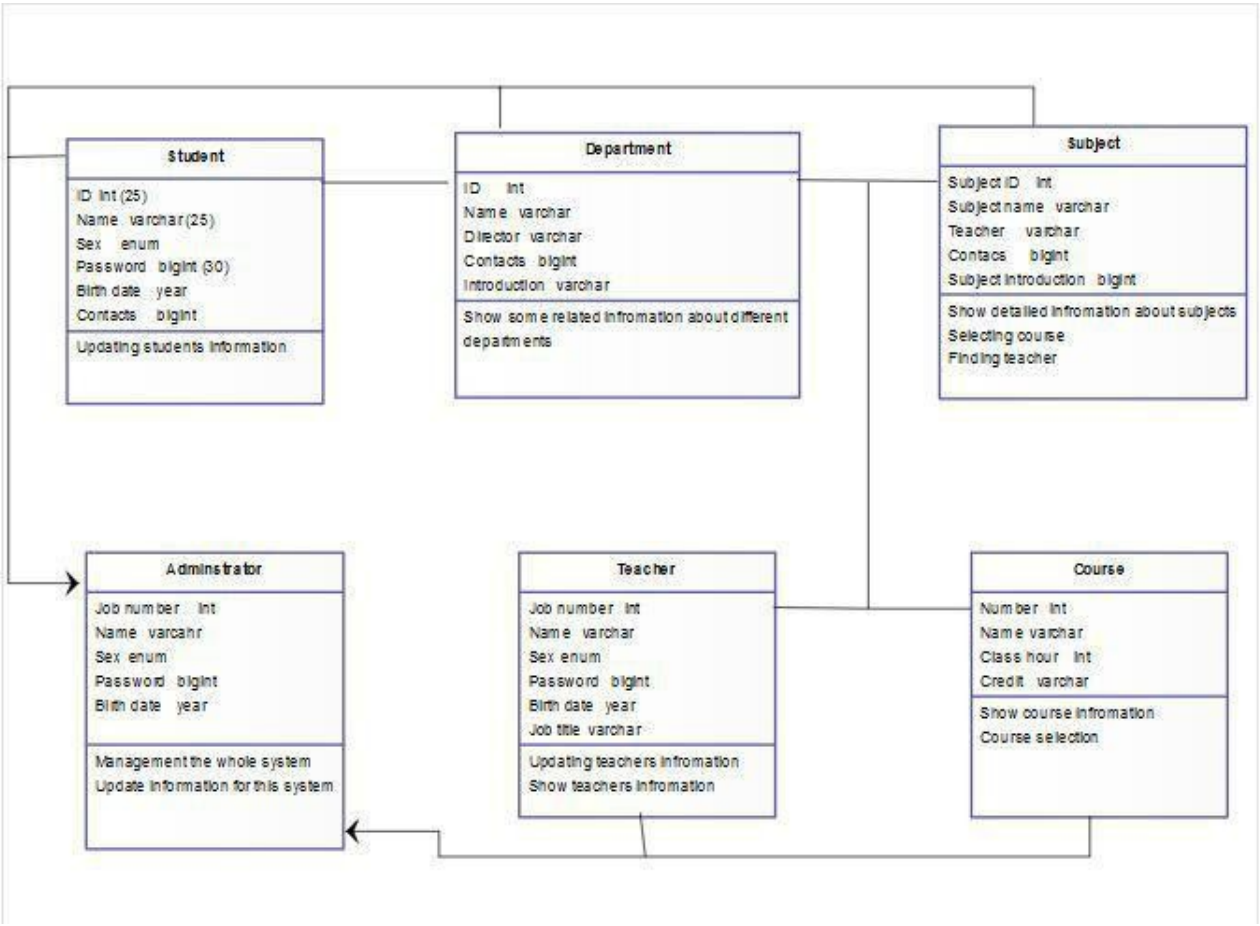
Functional component 6:Attendance details

Input: course, semester, subject, total classes.

Output: All the entered data will be stored in respective database and will be displayed in the grid.

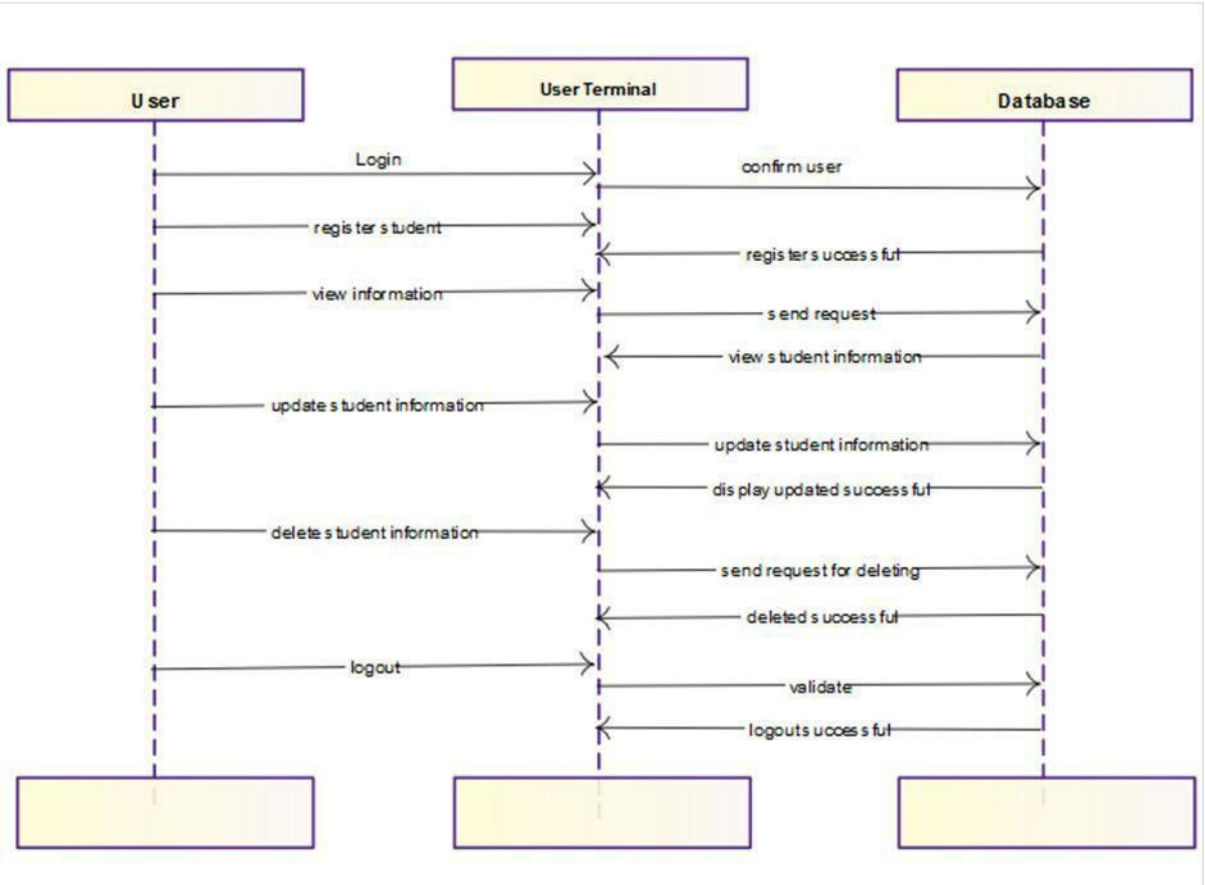
File I/O interface: This module is not related to other functional components.

2.3 User Classes and Characteristics

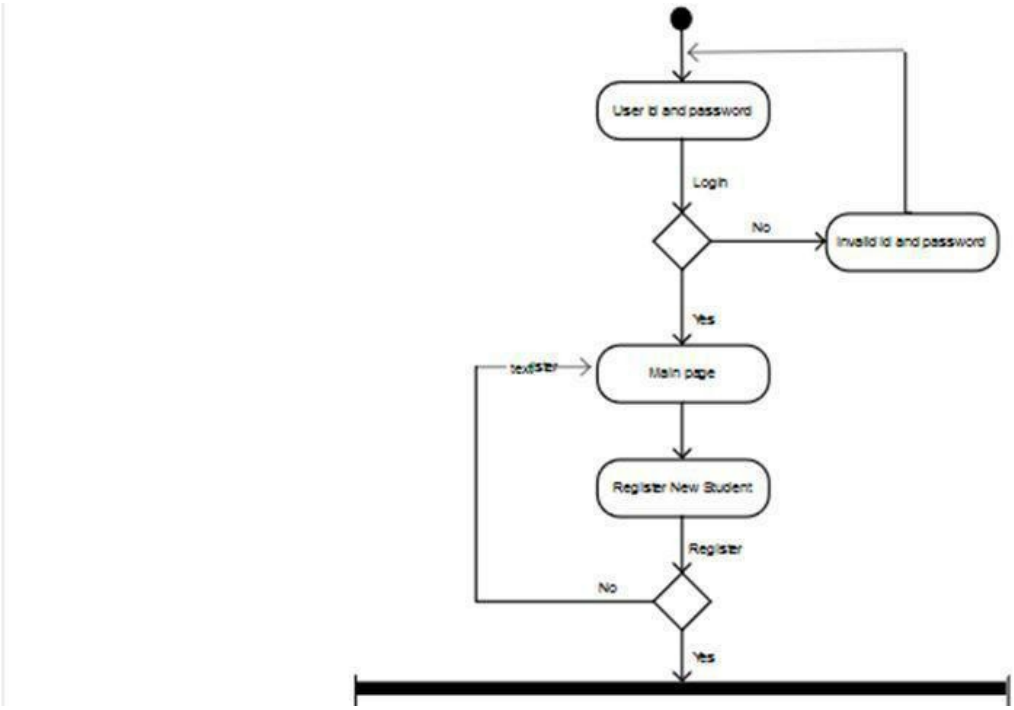


Class Diagram

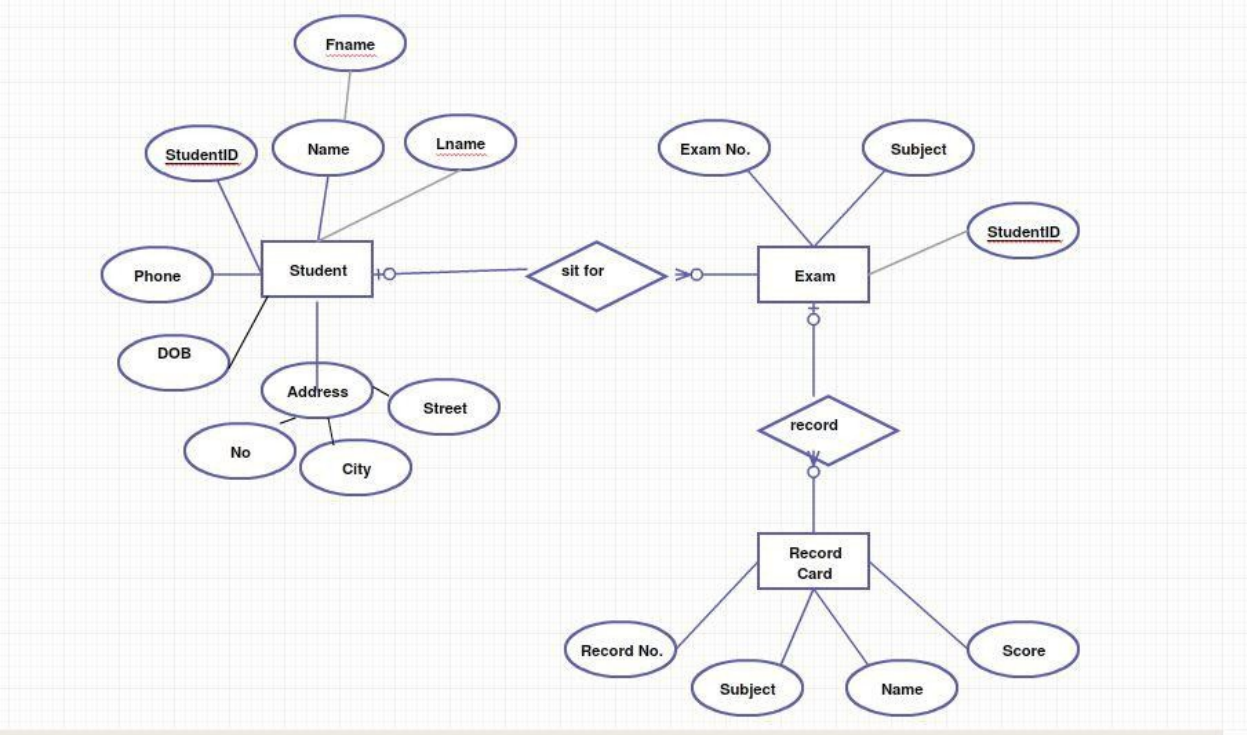
Sequence Diagram



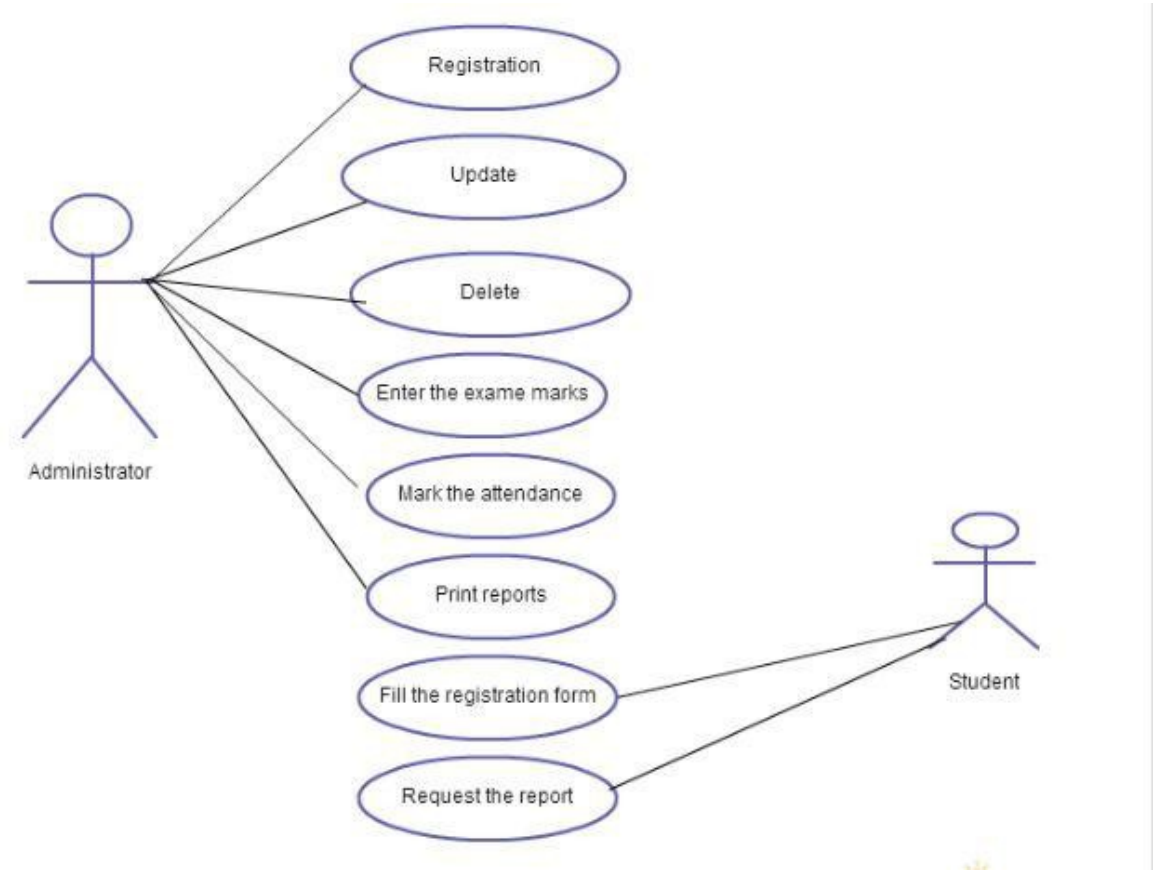
Activity Diagram



ER Diagram


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

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

There will be only one admin.
The deletion features will be available only for admin.

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

4.1 Admin Login

4.1.1 Description and priority

Login allows admin of this software to login with his username and password.Only admin can login.

4.1.2 Stimulus/Response Sequences

Upon login administrator is directed to the home page.The admin has to enter his credentials before logging.

4.1.3 Functional Requirements

REQ-1: The admin has to provide his/her username and password.

REQ-2: The system has to check the entered details and allow login only when entered details are correct.

REQ-3: When the user name and password is correct the admin should be successfully logged into his/her account.

REQ-4: Logout functionality has to provided as well.

4.2 Student details

4.1.1 Description and priority

Admin should be able to add details of new students as well as manage details of existing students.

4.1.2 Stimulus/Response Sequences

When the admin logins the students module should be displayed where admin can add and manage students details.

4.1.3 Functional Requirements

REQ-1: The admin should be able to add details of new students like reg.no, first name, last name, dob, father's name, sex, address, contact no,course, semester.

REQ-2: The admin should be able to manage details of existing students.

REQ-3: The details of each and every student should be displayed in managing module.

4.3 Course details

4.1.1 Description and priority

Admin should be able to add new course details and manage existing course details.

4.1.2 Stimulus/Response Sequences

When the admin logins the courses module should be displayed where admin can add and manage course details.

4.1.3 Functional Requirements

REQ-1: The admin should be able to add details of new courses like course id, course name, comment, course key.

REQ-2: The admin should be able to manage details of existing courses.

REQ-3: All the entered data will be stored in respective database and will be displayed in the grid.

4.4 Subject details

4.1.1 Description and priority

Admin should be able to add new course details and manage existing subject details.

4.1.2 Stimulus/Response Sequences

When the admin logins the subjects module should be displayed where admin can add and manage subject details.

4.1.3 Functional Requirements

REQ-1: The admin should be able to add details of new subjects like subject id, subject name, comment, course, subject type, semester.

REQ-2: The admin should be able to manage details of existing subjects.

REQ-3: All the entered data will be stored in respective database and will be displayed in the grid.

4.5 Examination details

4.1.1 Description and priority

Admin should be able to enter details of examination and get marks of student when details are entered.

4.1.2 Stimulus/Response Sequences

When the admin logins the examinations module should be be displayed.

4.1.3 Functional Requirements

REQ-1: The admin should be able to add details of examination like course, semester, internal type,marks

REQ-2: Marks of the selected student from database should be displayed

4.6 Attendance details

4.1.1 Description and priority

Admin should be able to enter and retrieve attendance of students.

4.1.2 Stimulus/Response Sequences

When the admin logins the attendance module should be be displayed.

4.1.3 Functional Requirements

REQ-1: Each student attendance in a course should be entered

REQ-2: Each student attendance should be displayed when details are entered

5.Other Nonfunctional Requirements

5.1 Performance Requirements

The product should perform well and fast and satisfy the intended purposes.

5.2 Safety Requirements

·The login Id and password must be created by system administrator and communicated to the concerned user confidentially to avoid unauthorized access to the system.
·It is assumed that a student registering for a course has paid desired university fee.

5.3 Security Requirements

·Restricted access to functionalities based on role.

·The delete operation is available only to the administrator. To reduce the complexity of the system, there is no check on delete operation. Hence, administrator should be very careful before deletion of any record and he/she will be responsible for data consistency.

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.>