System Requirements Specification for: DBIS Course Project SRS

S. Sudarshan

September 2, 2000

`The format of this SRS is based on an IEEE standard plus modifications of my own. Not all sections are required for all projects. If not applicable just say not applicable or delete.`

1 Introduction

1.1 Overall Description

Clearly state purpose of the project. By whom and how it will be used

Eg. This project aims to computerize the operations of the library. There are several aspects to this, including recording of books/journals, issue and return of books, handling late returns and fines, etc. The users of the system are the library staff and institute staff/students.

1.2 Definitions

Acronyms and abbreviations as applicable

1.3 References

References to other documents

1.4 Overview of Developer's Responsibilities

Responsibilities in terms of development, installation, training, maintenance etc.

Eg. The developer is responsible for development, installation, loading of all the data relating to books in the library, and training of employees.

2 General Description

2.1 Product Perspective

Relationship with other products and principal interfaces

Eg. This product will interface with the existing database of staff/student information. There are four main interfaces. One for book issual/return and fines, one for querying of books, one for adding/deleting books, and one for administration of users.

2.2 Product Functions Overview

A library example.

1. Book issue and return

2. Fine calculation and collection
3. Adding and deleting books
4. Querying to find books
5. Report generation
6. Automatic late book reminder

2.3 User Characteristics

Who the users are, and what training they may need

2.4 General Constraints

About schedule, resources, cost etc.

Eg. This project is to be completed, including coding, testing and loading of the database, by Nov 15, 2000. A prototype demo is due on Oct 22, 2000.

3 Information Description

3.1 Entities and Relationships

Give a list of entities/relationships that are needed, and ER diagrams.

3.2 Data Dictionary

Give the relations, their attributes, and for each attribute the type and a description of the attribute. Here's an example.

1. customers

account_no	varchar(8)	primary key	account number
name	varchar(20)	not null	Name
profession	varchar(10)	not null	Profession
address	varchar(40)	not null	Address
email	varchar(40)	-	Email Address

3.3 Data Flow

Give data flow between major units of your software. E.g., Most useful in case a task has multiple steps requiring interaction with other software or other humans. E.g. To purchase a book, user can enter request, department head can approve, then librarian approves, then order is made.

4 Functional Requirements

Give a description of each process/function. Start from what happens when the user connects to the system. E.g. The main screen and alternatives available to choose from would be a functional requirement. Authentication when entering the system would be another. An interface to issue a book would be a third.

Have one subsection for each functional requirement.

4.1 Functional Requirement 1

Note what a particular functionality does. Make sure to also note what happens in special cases such as out of stock.

4.2 Functional Requirement 2

...

5 External Interface Requirements

5.1 User Interfaces

This used to be important, and required a user manual giving commands screen formats, outputs, error messages, etc. In your case just say the web is used as interface.

5.2 Hardware Interfaces

Interfaces with existing as well as new or special purpose hardware. E.g. smart card reader for authentication. In your case say no special hardware.

5.3 Software Interfaces

List software platforms that are to be supported.

List interfaces with other software packages etc. Make sure to list all software that you interface with.

6 Performance Requirements

Capacity requirements (estimates of no of users, database size, ..) Response time requirements (for different tasks) and throughput (estimates of avg/peak number of transactions per minute, etc). Don't worry about exact numbers but give ballpark figures.

7 Design Constraints

7.1 Standards Compliance

Software development standards asn well as organizational standards (eg reports, auditing).

In your case, say not applicable.

7.2 Hardware Limitations

Requirements on machines, operating systems, storage capacities etc. This used to be very important, but these days this matters only for very fast/highthroughput systems, not so much for the things you are doing.

Fault tolerance and reliability requirements should also be listed here.

In your case this is not considered

8 Validation Criteria

Validation criteria for demonstrating successful implementation and user acceptance. In general includes classes of tests and expected software response.

E.g. I will load the database with X amount of data for testing, and an overview of tests you will run.

9 Other Requirements