

SOFTWARE REQUIREMENTS SPECIFICATION

FOR

COLUMN DATASTORE FOR DATA WAREHOUSE

Project Guide

Prof Ramanathan Chandrashekar

Version No. 1.0

Submitted By

K.Sravya(MT2012070)

Nagashree D A(MT2012084)

TABLE OF CONTENTS

Contents	Page
1 Introduction	3
1.1 Purpose	3
1.2 Scope	3
1.3 Definitions	3
1.4 Abbreviations and Acronyms	3
2 Overall Description	4
2.1 System Perspective	4
2.1.1 User Interface	4
2.2 System Functions	4
2.3 User Characteristics	5
2.4 Constraints	5
2.4.1 Design Constraints	5
2.4.2 Reliability Requirements	6
3 Requirements	6
3.1 Functional Requirements	6
3.2 Non-Functional Requirements	7
3.2.1 Reliability	7
3.2.2 Usability	7
3.2.3 Maintainability	8
3.2.4 Performance	8
3.3 Software Requirements	8
4 Inputs and Outputs	8

1. INTRODUCTION

1.1 PURPOSE

This project aims at implementing a column based data store which facilitates a user to insert, retrieve, update, delete data efficiently.

1.2 SCOPE

This system developed is capable of :

- Accepting schema from the user in the form of XML file.
- Constructing a Column Data Store based on the schema input from the user.
- Allowing the user to perform CRUD operations on the data in the data store.
- Enabling the user to view the retrieved data in the form of a tuple irrespective of internal organization of data.

1.3 DEFINITIONS

Data Store : A data store is a data repository of a set of integrated objects. These objects are modelled using classes defined in database schemas.

Database Schema : The description of a database is called the database schema which is specified during database design and is not expected to change frequently.

Column Data Store : A Column data store is a system that stores data tables as sections of columns of data rather than as rows of data.

Data Warehouse : Data Warehouse is a database used for reporting and data analysis.

Database Index : A Database Index is a data structure that improves the speed of data retrieval operations on a database table at the cost of slower writes and less storage space.

1.4 ABBREVIATIONS and ACRONYMS

CRUD : Create, Read, Update, Delete

CDS : Column Data Store

XML : Extensible Markup Language

UML : Unified Modelling Language

BST : Binary Search Tree

2 OVERALL DESCRIPTION

2.1 SYSTEM PERSPECTIVE

Column Data Store aims at providing user an efficient way of retrieving data from a large database.

Traditionally, a Relational Database will store data in the form of tuples which hinders data retrieval performance during non-index access to the tuples.

CDS will overcome this drawback by storing each attribute of an entity in a separate data file and using indexing structures to access data from different data files.

2.1.1 USER INTERFACES

- User gives the schema file as a command line input from which the CDS is constructed.
- User can perform CRUD operations by specifying queries in a file which is again given as a command line input .
- The results of the CRUD operations will be displayed to the user on the standard output. For example, Create, Update and Delete show the output as "Successful" or "Not Successful" and Read will display the desired tuples or "Record not found", if not found.

2.2 SYSTEM FUNCTIONS

The main functions of the system are:

- Accepts XML Schema File from the user as a command line argument
- Parses the schema file and creates different data files based on the tags in the XML schema file.
- Allows user to perform CRUD operations on the data
- Enables an efficient retrieval of data by constructing indexing structures such as BST and Hash table.

- Displays the results of CRUD operations to the user.

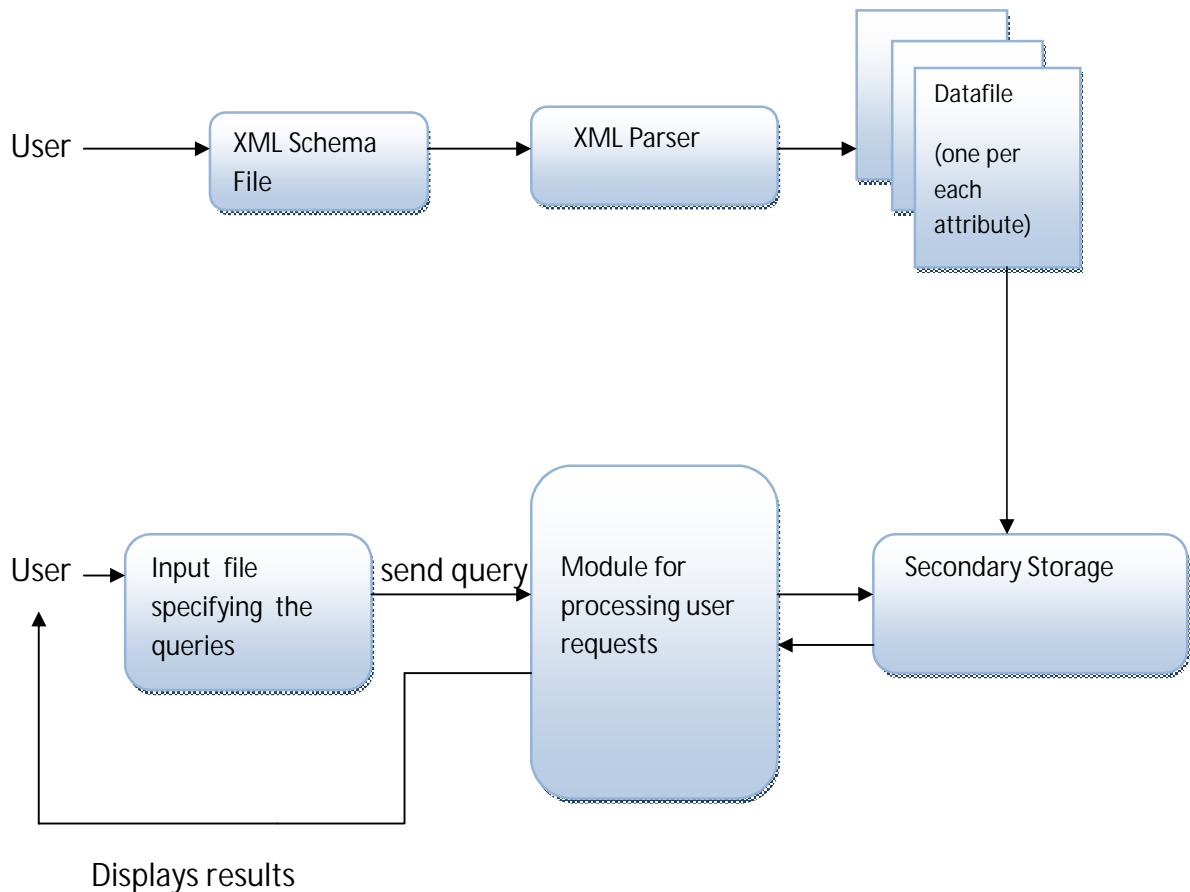


Figure 1: System Architecture

2.3 USER CHARACTERISTICS

- User must be able to specify his schema in XML .
- User should be able to specify queries in a file and give it as command-line input.
- User must be familiar with schema and schema constructs.

2.4 CONSTRAINTS

2.4.1 DESIGN CONSTRAINTS

User has to provide schema only in XML since the data files are obtained as an output of the parser. If any other model is used for specifying the schema then it has to be mapped to XML.

2.4.2 RELIABILITY REQUIREMENTS

The system should be able to link all the data files accurately to form desired tuple to be presented to the user.

3. REQUIREMENTS

3.1 FUNCTIONAL REQUIREMENT: Functional Requirements of the system are as follows:

Actor	Use Case	Description
User	Gives schema as input	Schema is given in an XML file, where each tag represents a column (attribute).
	Inserts data	User will provide data in the form of a query which is then processed and stored in respective files
	Read data	User will provide the query for retrieval of data. This query is processed and result will be presented in the form of tuple(s).
	Update data	User will provide the query. This query will be processed and old value will be replaced with the new value.
	Delete data	User will provide the query for deletion. This query is processed and the specified data is deleted.
	View Result	User will be able to view the result of the queries after they are processed.

Table 1 : Use Cases

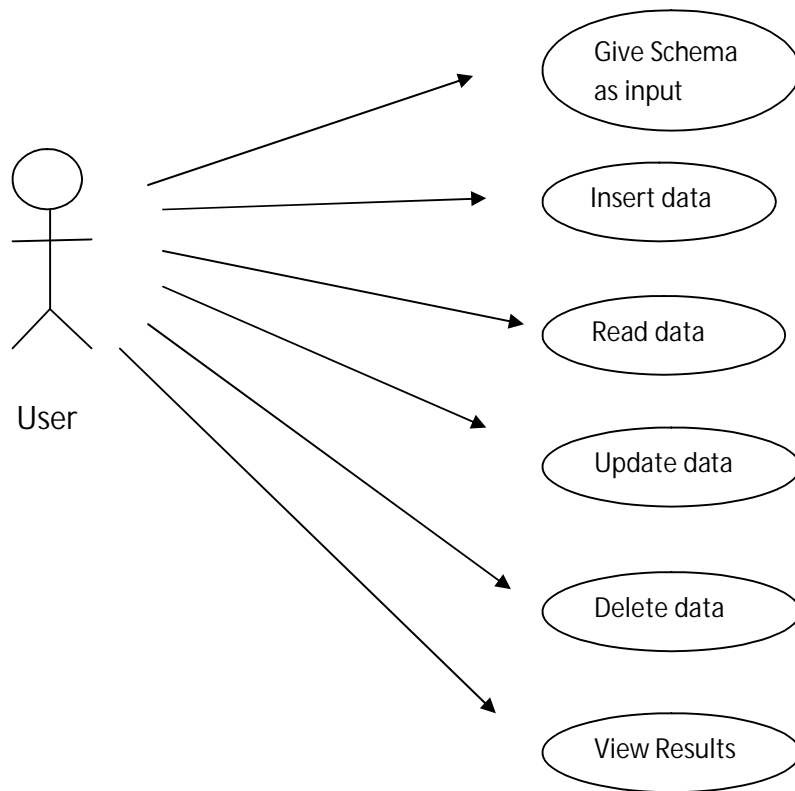


Figure 2: Use case diagram

3.2 NON-FUNCTIONAL REQUIREMENTS

3.2.1 RELIABILITY

The system is expected to provide accurate results at any instant of time. The system is not expected to go to an inconsistent state, eventually.

3.2.2 USABILITY

It is sufficient for the user to provide the query and expect desired results without bothering about the internal organisation of data and processing of the query.

3.2.3 MAINTAINABILITY

The modules are loosely coupled. Changes can be made in each module without affecting the other modules, according to changing requirements.

3.2.4 PERFORMANCE

Retrieval of data is faster in CDS than that of the RDBMS

3.3 SOFTWARE REQUIREMENTS

- Programming Language : C
- Any 'C' Compiler

4 INPUTS AND OUTPUTS

- a) Input : Schema in the form of an XML file

Output : Creation of the datafiles

- b) Input : Query for insertion

Output : "Insertion Successful" or "Insertion Not Successful"

- c) Input : Query for retrieval of data

Output : Displays the data records if successful or displays "No records Found" on failure

- d) Input : Query for update

Output : "Update Successful" or "Update Not Successful"

- e) Input : Query for deletion

Output : "Delete Successful" or "Delete Not Successful"