a

SOFTWARE REQUIREMENTS SPECIFICATION

District Health Information Software v2 (DHIS2)

This Document explains the Software Requirements Specifications of the District Health Information Software v2 (DHIS2) which is a routine data based health information system which allows for data capture, aggregation, analysis, and reporting of data.

DHIS2 is intended to be used as a Health Management Information System (HMIS) application.

2009

The University of Oslo,

Society for Health Information Systems Programme, India,

HISP Vietnam,

HISP Ethiopia,

06/08/2009

# Contents

[Contents 2](#_Toc237943191)

[Table of Figures 5](#_Toc237943192)

[Introduction 6](#_Toc237943193)

[Purpose 6](#_Toc237943194)

[Aim of the Document 6](#_Toc237943195)

[AIM of the SOFTWARE 6](#_Toc237943196)

[Intended Audience 7](#_Toc237943197)

[Definitions, Acronyms and Abbreviations 7](#_Toc237943198)

[Problem Statement 8](#_Toc237943199)

[Methodology 8](#_Toc237943200)

[Undertaken Tasks 8](#_Toc237943201)

[Overall Description 9](#_Toc237943202)

[Product Perspective 9](#_Toc237943203)

[Product Features 9](#_Toc237943204)

[User Characteristics 10](#_Toc237943205)

[User Requirements 12](#_Toc237943206)

[***1.*** *The user should be able to* **LOGIN TO THE SYSTEM WITH DIFFERENT PRIVILEGES** 12](#_Toc237943207)

[***2.*** *The user should be able to* **LOGOFF FROM THE SYSTEM** 12](#_Toc237943208)

[***3.*** *The user should be able to* **LOGIN AS ADMINISTRATOR & MANAGE USERS** 12](#_Toc237943209)

[***3.1*** *The user should be able to* **MANAGE USER ROLES** 12](#_Toc237943210)

[***3.2*** *The user should be able to* **CHANGE OWN PASSWORD** 12](#_Toc237943211)

[***4.*** *The user should be able to* **CHANGE SYSTEM SETTINGS** 13](#_Toc237943212)

[***5.*** *The user should be able to* **CHANGE USER SETTINGS** 13](#_Toc237943213)

[***6.*** *The user should be able to* **CREATE AND MANAGE ORGANIZATION UNITS** 13](#_Toc237943214)

[***6.1*** *The user should be able to* **MOVE ORGANIZATION UNIT HIERARCHY** 13](#_Toc237943215)

[***7.*** *The user should be able to* **MANAGE DATA ELEMENTS AND INDICATORS** 13](#_Toc237943216)

[***7.1*** *The user should be able to* **CREATE, DELETE OR EDIT DATAELEMENTS** 13](#_Toc237943217)

[***7.2*** *The user should be able to* **TRANSLATE DATAELEMENTS** 14](#_Toc237943218)

[***7.3*** *The user should be able to* **GET AN OVERVIEW OF DATAELEMENTS** 14](#_Toc237943219)

[***7.4*** *The user should be able to* **CREATE, EDIT or DELETE DATA ELEMENT GROUPS** 14](#_Toc237943220)

[***7.5*** *The user should be able to* **TRANSLATE DATA ELEMENT GROUPS** 14](#_Toc237943221)

[***7.6*** *The user should be able to* **GET AN OVERVIEW OF DATAELEMENT GROUP** 14](#_Toc237943222)

[***7.7*** *The user should be able to* **CREATE, DELETE OR EDIT INDICATORS** 14](#_Toc237943223)

[***7.8*** *The user should be able to* **TRANSLATE INDICATORS** 14](#_Toc237943224)

[***7.9*** *The user should be able to* **GET AN OVERVIEW OF INDICATORS** 15](#_Toc237943225)

[***7.10*** *The user should be able to* **CREATE, EDIT or DELETE INDICATOR TYPE** 15](#_Toc237943226)

[***7.11*** *The user should be able to* **TRANSLATE INDICATOR TYPE** 15](#_Toc237943227)

[***7.12*** *The user should be able to* **GET AN OVERVIEW OF INDICATOR TYPE** 15](#_Toc237943228)

[***7.13*** *The user should be able to* **CREATE, EDIT or DELETE INDICATOR GROUPS** 15](#_Toc237943229)

[***7.14*** *The user should be able to* **TRANSLATE INDICATOR GROUP** 15](#_Toc237943230)

[***7.15*** *The user should be able to* **GET AN OVERVIEW OF INDICATOR GROUP** 15](#_Toc237943231)

[***8.*** *The user should be able to* **MANAGE DATASETS** 15](#_Toc237943232)

[***8.1*** *The user should be able to* **CREATE, DELETE OR EDIT DATASETS** 16](#_Toc237943233)

[***8.2*** *The user should be able to* **ASSIGN DATASET TO ORGANIZATION UNITS** 16](#_Toc237943234)

[***8.3*** *The user should be able to* **CREATE DATA ENTRY FORMS FOR DATASETS** 16](#_Toc237943235)

[***8.4*** *The user should be able to* **TRANSLATE NAMES OF DATASETS** 16](#_Toc237943236)

[***8.5*** *The user should be able to* **GET AN OVERVIEW OF ALL DATASETS** 16](#_Toc237943237)

[***9.*** *The user should be able to* **LOCK DATASETS FROM DATAENTRY** 16](#_Toc237943238)

[***10.*** *The user should be able to* **BROWSE DATA ENTRY BY PERIOD** 16](#_Toc237943239)

[***11.*** *The user should be able to* **DO SIMPLE DATA INTEGRITY CHECKS** 17](#_Toc237943240)

[***12.*** *The user should be able to* **SEE DATA STATISTICS** 17](#_Toc237943241)

[***13.*** *The user should be able to* **DO DATA ENTRY** 17](#_Toc237943242)

[***13.1*** *The user should be able to* **ADD MINIMUM & MAXIMUM VALUES to data elements** 17](#_Toc237943243)

[***13.2*** *The user should be able to* **RUN VALIDATION RULES ON DATA ENTRY** 17](#_Toc237943244)

[***13.3*** *The user should be able to* **COMPLETE AND UNDO THE DATASET** 17](#_Toc237943245)

[***14.*** *The user should be able to* **DO DATA QUALITY CHECKS** 18](#_Toc237943246)

[***14.1*** *The user should be able to* **CREATE, EDIT OR DELETE VALIDATION RULES** 18](#_Toc237943247)

[***14.2*** *The user should be able to* **GET OVERVIEW of VALIDATION RULES** 18](#_Toc237943248)

[***14.3*** *The user should be able to* **CREATE, EDIT OR DELETE VALIDATION GROUPS** 18](#_Toc237943249)

[***14.4*** *The user should be able to* **GET OVERVIEW OF VALIDATION GROUPS** 18](#_Toc237943250)

[***14.5*** *The user should be able to* **RUN VALIDATION RULES** 18](#_Toc237943251)

[***15.*** *The user should be able to* **DO GRAPHICAL ANALYSIS OF DATA** 19](#_Toc237943252)

[***15.1*** *The user should be able to* **DO GRAPHICAL ANALYSIS FOR ANNUAL DATA** 19](#_Toc237943253)

[***16.*** *The user should be able to* **DO TABULAR ANALYSIS OF DATA** 19](#_Toc237943254)

[***17.*** *The user should be able to* **VIEW THE DATA STATUS** 19](#_Toc237943255)

[***18.*** *The user should be able to* **TAKE BACKUP OF DATA** 19](#_Toc237943256)

[***19.*** *The user should be able to* **VIEW DATA ON A MAP AS A GEOGRAPHIC INFORMATION SYSTEM** 19](#_Toc237943257)

[***20.*** *The user should be able to* **VIEW HELP ON THE APPLICATION** 20](#_Toc237943258)

[***21.*** *The user should be able to* **SEE APPLICATION INFORMATION** 20](#_Toc237943259)

[System Requirements 36](#_Toc237943260)

[System Functional Requiremnets 36](#_Toc237943261)

[1. ***The system shall*** BE BUILT AS A COMBINATION OF MODULES 36](#_Toc237943262)

[2. ***The system shall*** NOT DISPLAY PAGES WITHOUT VALID USER CREDENTIALS 36](#_Toc237943263)

[3. ***The system shall*** CREATE/EDIT USER WITH VALID DETAILS 37](#_Toc237943264)

[Nonfunctional Requirements 38](#_Toc237943265)

[1. Product Requirements 38](#_Toc237943266)

[2. Organization Requirements 40](#_Toc237943267)

[External Interface Requirements 40](#_Toc237943268)

[User Interfaces 40](#_Toc237943269)

[Hardware Interfaces 40](#_Toc237943270)

[Software Interfaces 40](#_Toc237943271)

[System Architecture 41](#_Toc237943272)

[References 42](#_Toc237943273)

# Table of Figures

[Figure 1- PRIMARY USE-CASE OF DHIS2 MODULES 22](#_Toc237943274)

[Figure 2- USE-CASE of USER MANAGEMENT MODULE 23](#_Toc237943275)

[Figure 3- USE-CASE of ROLE MANAGEMENT MODULE 23](#_Toc237943276)

[Figure 4- USE-CASE of SETTINGS MODULE 24](#_Toc237943277)

[Figure 5- USE-CASE of ORGANIZATION UNIT MANAGEMENT MODULE 25](#_Toc237943278)

[Figure 6- USE-CASE of ORGANIZATION UNIT GROUP MANAGEMENT MODULE 25](#_Toc237943279)

[Figure 7- USE-CASE ORGANIZATION UNIT GROUPSET MANAGEMENT MODULE 26](#_Toc237943280)

[Figure 8- USE-CASE of ORGANIZATION UNIT HIERARCHY MANAGEMENT MODULE 26](#_Toc237943281)

[Figure 9- USE-CASE of DATASET MANAGEMENT MODULE 27](#_Toc237943282)

[Figure 10- USE-CASE of DATA ELEMENT MANAGEMENT MODULE 28](#_Toc237943283)

[Figure 11- USE-CASE of DATA ELEMENT GROUP MANAGEMENT MODULE 28](#_Toc237943284)

[Figure 12- USE-CASE of INDICATOR MANAGEMENT MODULE 29](#_Toc237943285)

[Figure 13- USE-CASE of INDICATOR TYPE MANAGEMENT MODULE 30](#_Toc237943286)

[Figure 14- USE-CASE of INDICATOR GROUP MANAGEMENT 31](#_Toc237943287)

[Figure 15- USE-CASE of DATA ADMINISTRATION MODULE 31](#_Toc237943288)

[Figure 16- USE-CASE of DATA ENTRY MODULE 32](#_Toc237943289)

[Figure 17- USE-CASE of DATABASE MANAGEMENT MODULE 32](#_Toc237943290)

[Figure 18- USE-CASE OF VALIDATION RULE MANAGEMENT MODULE 33](#_Toc237943291)

[Figure 19- USE-CASE of VALIDATION RULE GROUP MANAGEMENT MODULE 33](#_Toc237943292)

[Figure 20- USE-CASE of RUNNING VALIDATION RULES 34](#_Toc237943293)

[Figure 21- USE-CASE of DATA ANALYSIS 34](#_Toc237943294)

[Figure 22- USE-CASE of IMPORT/EXPORT MODULE 35](#_Toc237943295)

[Figure 23- System Architecture 41](#_Toc237943296)

# Introduction

## Purpose

### Aim of the Document

This Software Requirements Specification (SRS) for DHIS2 is a document that aims and focuses to describe the main structure of the system, how its components interact with one another, and the list of requirements met by the system on a simple and thorough scale. Relationships between entities of the system and functional use-cases are often modeled or illustrated in various parts of the document. In other words, the SRS should completely describe the external behavior of the application. It also explains and describes nonfunctional requirements, design constraints and other factors necessary to provide a complete, broad and comprehensive description of the requirements for DHIS2.

### AIM of the SOFTWARE

DHIS2 is targeted at the distributed collection and analysis of routine data, specifically for primary health care, but potentially any kind of data. Henceforth, the aim of the software is easy data collection, data mart creation, data warehouse management, analysis of data and reporting information for action. Public health researchers, health ministries and the health staff are in need of easy to use Health Management Information System (HMIS) software. DHIS2 tries to meet these needs and the SRS explains different use-cases and processes through which it meets the requirements.

DHIS2 aims to be a tool for collection, validation, analysis, and presentation of aggregate statistical data, tailored (but not limited) to integrated health information management activities. It is a generic tool rather than a pre-configured database application, with an open meta-data model and a flexible user interface that allows the user to design the contents of a specific information system without the need for programming

## Intended Audience

The SRS document is intended for anyone who wants to have a holistic understanding of the system and tries to give a clear idea to decision makers on what requirements the software tries to meet. The SRS is also useful for design and development of the application and is considered as a constitution document that has to be followed for any further development of the software. Ministries of health can understand use of the application by reading this document. External testers can perform functional testing by understanding expected results of the application.

## Definitions, Acronyms and Abbreviations

* HMIS : Health Management Information System
* DHIS : District Health Information Software
* FOSS : Free and Open-Source Software
* GUI : Graphical User Interface
* ORG UNIT: Organization Unit of Data Capture
* VPN : Virtual Private Network

# Problem Statement

To design and develop a web-based, free and open-source system for data collection, validation, analysis, and presentation of aggregate statistical data. This system should allow distributed data collection and distributed dissemination of data, meeting the requirements of a health management information system.

# Methodology

The system is developed on the principles of Xtreme Programming (XP) which includes simplicity, communication, feedback and courage. Simple design and continuous refactoring through feedback is the process followed for development and implementation of the system. The “bazaar” model of distributed development has been adopted with developers spread across countries like Norway, India, Ethiopia and Vietnam.

## Undertaken Tasks

* **Gathering information**
* **Identify and classify the system and user requirements**
* **SRS creation**
  + Successfully identifying system requirements and documenting them in this SRS
* **System Design**
  + Decide upon the FOSS frameworks to be used for the system
  + Design a flexible data model that can be customized to implementation needs
  + Design a common portal framework to be used by all modules
* **System Implementation**
  + Implementing the different components and subsystems of the project
* **Unit Testing**
  + Testing each subsystem independently
* **System Integration**
  + Integration of different subsystems to form the whole system.
* **System Testing**
  + Testing the system as a whole
* **Test Specification**
  + Submission a full report of the known bugs and errata of the system.

# Overall Description

## Product Perspective

DHIS2 is the next version to the popular District Health Information System 1.3. DHIS2 is web-based and uses open-source frameworks. The system runs on a Java Servlet Container and is accessed through a web browser. It is platform independent (runs on Windows, Linux, OSX etc.) and browser independent (runs on Mozilla Firefox, Internet Explorer, Opera etc.). DHIS2 can use any database management system (runs on MySQL, PostgreSQL, Oracle etc.) and can run on a number of Java Servers (runs on Apache Tomcat, Jetty, Glassfish etc.)

Using DHIS2 does not imply any licensing fees. DHIS2 is released under the BSD license. Though some parts of DHIS2 are also under the GNU Public License v3

## Product Features

Based on the requirements DHIS2 intends of provide the following features:

* Easy and distributed data collection
* Easy management of collected data
* Flexible data model for collecting all kinds of data
* Scalable access to data by creating a data warehouse
* Abstraction of data and datasets based on hierarchy of organization units
* Powerful user management for system security and data management
* Intuitive reporting for managers and decision makers using graphs, charts and maps
* Generates reports in different formats that can be exported to other applications
* Validation rules for improving data quality
* Import / Export data to other applications

## User Characteristics

Since the system is to be used for data collection as well as decision making, it should cater to the needs of a wide audience of users. The users range from grass-root health workers, data entry operators, health officers, state health ministries, public health researchers and national policy formulators. Thus, DHIS2 needs to be powerful, yet simple to use as a HMIS.

# User Requirements

## *The user should be able to* **LOGIN TO THE SYSTEM WITH DIFFERENT PRIVILEGES**

***Rationale:*** Since data entry, validation, analysis and reporting are all specific to the user and their privileges; each user should have their own login credentials which allow access to only certain parts of the system. Each user should be able to change their login credentials.

## *The user should be able to* **LOGOFF FROM THE SYSTEM**

***Rationale:*** Since all activities of the system are associated to a user, after the user has finished using the system, they should be allowed to logoff from the system. To again use the system, the user has to login.

## *The user should be able to* **LOGIN AS ADMINISTRATOR & MANAGE USERS**

***Rationale:*** The administrator user of the system should be able to set-up the system by creating and managing users of the system. The admin user should be able to search existing users and change their details like username, password, first name, last name, phone and email.

## *The user should be able to* **MANAGE USER ROLES**

***Rationale:*** The admin user should also be able to create new user roles and assign these roles to different users. The user roles can be used to allow viewing of datasets, reports and different authorities to use the system.

## *The user should be able to* **CHANGE OWN PASSWORD**

***Rationale:*** Each user of the system should be able to change their own password. Admin users should not be able to look-up this password, but still should be allowed to change the password for any user of the system.

## *The user should be able to* **CHANGE SYSTEM SETTINGS**

***Rationale:*** Users with roles that can setup the system should be able to change the settings of the system. The user should also be able to change important systems settings like Application Title, Logo and Start page.

## *The user should be able to* **CHANGE USER SETTINGS**

***Rationale:*** User should be able to change the GUI Language, Database Language and change the style (look and feel) of the system.

## *The user should be able to* **CREATE AND MANAGE ORGANIZATION UNITS**

***Rationale:*** Every data value should be captured or accessed only through an organization unit. The organization unit is representative of the location, area or unit of data capture. The user should be able to create new organization units, associate them as a hierarchy and identify different levels of organization units. The user should also be able to move organization units as and when required.

## *The user should be able to* **MOVE ORGANIZATION UNIT HIERARCHY**

***Rationale:*** The user should also be able to move organization units as and when required and along with the data should also be moved to the new organization unit hierarchy.

## *The user should be able to* **MANAGE DATA ELEMENTS AND INDICATORS**

***Rationale:*** Data elements and indicators are the basic models of data collection and analysis in the system. Data elements act as data capturing units while indicators act as analysis units.

## *The user should be able to* **CREATE, DELETE OR EDIT DATAELEMENTS**

***Rationale:*** The user should be able to create or edit data elements by giving name, codes, data type and aggregation operator. The data type should be Number, Text or YES/NO while the aggregation operator should be SUM or AVERAGE. The data type is representative of the type of data being stored and will be used to show the user-interface to the user. The aggregation operator will be useful to aggregate values over a given period or across different org units.

## *The user should be able to* **TRANSLATE DATAELEMENTS**

***Rationale:*** The user should be able to create new locales and give names to data elements.

## *The user should be able to* **GET AN OVERVIEW OF DATAELEMENTS**

***Rationale:*** The user should be able to see name, description, type and aggregation operator.

## *The user should be able to* **CREATE, EDIT or DELETE DATA ELEMENT GROUPS**

***Rationale:*** Data Element Groups are different from datasets. Data elements can be put together in a group and these may be used in different datasets. Data element groups can also be used to segregate data elements that are used for a common purpose. Thus, the user should be able to create, edit or delete data element groups by adding data elements to the groups.

## *The user should be able to* **TRANSLATE DATA ELEMENT GROUPS**

***Rationale:*** The user should be able to create new locales and give names to data element groups based on their locale.

## *The user should be able to* **GET AN OVERVIEW OF DATAELEMENT GROUP**

***Rationale:*** The user should be able to see name and number of data elements in a data element group.

## *The user should be able to* **CREATE, DELETE OR EDIT INDICATORS**

***Rationale:*** The user should be able to create or edit indicators by giving name, codes, whether annualized and type of indicator. As indicators are generally formulas, the user should be able to edit numerator and denominator of the indicators and while setting these should be able to define formulas from data elements.

## *The user should be able to* **TRANSLATE INDICATORS**

***Rationale:*** The user should be able to create new locales and give names to indicators based on their locale.

## *The user should be able to* **GET AN OVERVIEW OF INDICATORS**

***Rationale:*** The user should be able to see name, description, whether annualized, indicator type, numerator description and denominator description for each indicator as an overview.

## *The user should be able to* **CREATE, EDIT or DELETE INDICATOR TYPE**

***Rationale:*** Indicator types are useful for creating indicators and how they are calculated. The user should be able to give name and factor while creating Indicator Type.

## *The user should be able to* **TRANSLATE INDICATOR TYPE**

***Rationale:*** The user should be able to create new locales and give names to indicator type for their locale.

## *The user should be able to* **GET AN OVERVIEW OF INDICATOR TYPE**

***Rationale:*** The user should be able to see name and factor for any indicator type as an overview.

## *The user should be able to* **CREATE, EDIT or DELETE INDICATOR GROUPS**

***Rationale:*** Indicators used for common analysis can be put into a group for easy management. User should be able to create, edit or delete indicator groups by adding or removing indicators from a group.

## *The user should be able to* **TRANSLATE INDICATOR GROUP**

***Rationale:*** The user should be able to create new locales and translate indicator groups for their locale.

## *The user should be able to* **GET AN OVERVIEW OF INDICATOR GROUP**

***Rationale:*** The user should be able to see name and number of indicators for any indicator group.

## *The user should be able to* **MANAGE DATASETS**

***Rationale:*** Data elements form the basic unit of data. Groups of data elements are known as datasets. Users should be able to manage datasets.

## *The user should be able to* **CREATE, DELETE OR EDIT DATASETS**

***Rationale:*** Users should be able to create, delete or edit datasets at any time. Users should be able to give names to datasets, assign codes to datasets, add/remove necessary data elements and store the capture frequency of the dataset.

## *The user should be able to* **ASSIGN DATASET TO ORGANIZATION UNITS**

***Rationale:*** Users should be able to assign created datasets to organization units so that only assigned data can be captured. Not all data is captured from each organization unit and helps in easy data management.

## *The user should be able to* **CREATE DATA ENTRY FORMS FOR DATASETS**

***Rationale:*** Datasets are representative of forms in the system. The user should be able to create the data entry screens for capturing data in the datasets.

## *The user should be able to* **TRANSLATE NAMES OF DATASETS**

***Rationale:*** The user should be able to create new locales and give names to datasets based on their locale.

## *The user should be able to* **GET AN OVERVIEW OF ALL DATASETS**

***Rationale:*** The user should be able to see name, number of elements and frequency of capturing of datasets

## *The user should be able to* **LOCK DATASETS FROM DATAENTRY**

***Rationale:*** The user should be able to lock datasets so that further data entry is not possible. Data locking is useful for finalizing data by making the data values from a dataset for a given period (monthly, weekly etc.) uneditable.

## *The user should be able to* **BROWSE DATA ENTRY BY PERIOD**

***Rationale:*** The user should be able to see how much data was entered in a given period for a selected dataset or organization unit. The user should be able to select Period Type, the start date and end date and the dataset or organization and browse the data.

## *The user should be able to* **DO SIMPLE DATA INTEGRITY CHECKS**

***Rationale:*** The user should be able to do simple data integrity checks and look through which of them are causing violations

## *The user should be able to* **SEE DATA STATISTICS**

***Rationale:*** The user should be able to see number of data elements, data element groups, indicator types, indicators, indicator groups, datasets, data dictionary, organization units, validation rules, periods and data values.

## *The user should be able to* **DO DATA ENTRY**

***Rationale:*** The user should be able to select organization unit, dataset and the period which allows to data entry on the dataset. There should be on-the-fly saving of data values so that data values are not lost due to the intermittent loss of power and internet infrastructure. Whatever data is entered should be saved and retrievable next time.

## *The user should be able to* **ADD MINIMUM & MAXIMUM VALUES to data elements**

***Rationale:*** The user should be able to set minimum and maximum values to a data element and during data entry the user should conform to these rules of minimum and maximum. The user should also be able to see the history of data values that have been entered for the data element.

## *The user should be able to* **RUN VALIDATION RULES ON DATA ENTRY**

***Rationale:*** While the user is doing data entry, the user should be able to run the already existing validation rules on the dataset.

## *The user should be able to* **COMPLETE AND UNDO THE DATASET**

***Rationale:*** The user should be able to specify that they have finished data entry on a given dataset. The user should also be able to revert back to say that the data entry is not complete.

## *The user should be able to* **DO DATA QUALITY CHECKS**

***Rationale:*** The user should be able to do quality checks on correctness, completeness, consistency and timeliness. Other than running validation rules during data entry, user should be able to define and run validation rules on all data that has been collected.

## *The user should be able to* **CREATE, EDIT OR DELETE VALIDATION RULES**

***Rationale:*** The user should be able create or edit validation rules by giving name of validation rules, description and operator for validation rules. The user should be able to define the left-hand side and right-hand side of the formula for validation rules.

## *The user should be able to* **GET OVERVIEW of VALIDATION RULES**

***Rationale:*** The user should be able to see name, description, left-side description, right-side description and operator for each validation rule.

## *The user should be able to* **CREATE, EDIT OR DELETE VALIDATION GROUPS**

***Rationale:*** The user should be able to see add or remove validation rules and create a group of validation rules

## *The user should be able to* **GET OVERVIEW OF VALIDATION GROUPS**

***Rationale:*** The user should be able to see name, description and number of elements in validation group

## *The user should be able to* **RUN VALIDATION RULES**

***Rationale:*** The user should be able to run validation rules by selecting the start date and end date, validation rule group and the organization unit. The user should be able to see the results of the validation check and be able to identify which data elements are failing the validations.

## *The user should be able to* **DO GRAPHICAL ANALYSIS OF DATA**

***Rationale:*** The user should be able to view graphical representation of the data collected. The user should be able to analyze data on indicator and data elements and view it for organization units over a selected period. The user should be able to see charts or summary of the results. The user should also be able to export the data into excel sheets or print out the graphs and charts.

## *The user should be able to* **DO GRAPHICAL ANALYSIS FOR ANNUAL DATA**

***Rationale:*** The user should be able to create graphs and charts by selecting indicators or data elements over a period of year/years for a selected organization unit.

## *The user should be able to* **DO TABULAR ANALYSIS OF DATA**

***Rationale:*** The user should be able to view at data in the form of tables for any selected indicator or data element for a given period or organization unit

## *The user should be able to* **VIEW THE DATA STATUS**

***Rationale:*** The user should be able to look at the number of data elements that are being collected from an organization unit for a given period. Data status should be seen as the percentage of data elements that have been filled to the total number of elements in the dataset.

## *The user should be able to* **TAKE BACKUP OF DATA**

***Rationale:*** The user should be able to take backup of the data from the user-interface. The user should be able to click on a button and download the backup of data from the server. ***The user should be able to*** view the files in a file explorer

## *The user should be able to* **VIEW DATA ON A MAP AS A GEOGRAPHIC INFORMATION SYSTEM**

***Rationale:*** The user should be able to view collected data represented on a map. Each organization unit can be associated as an area on the map and the user should be able to view data values/ indicators on the map.

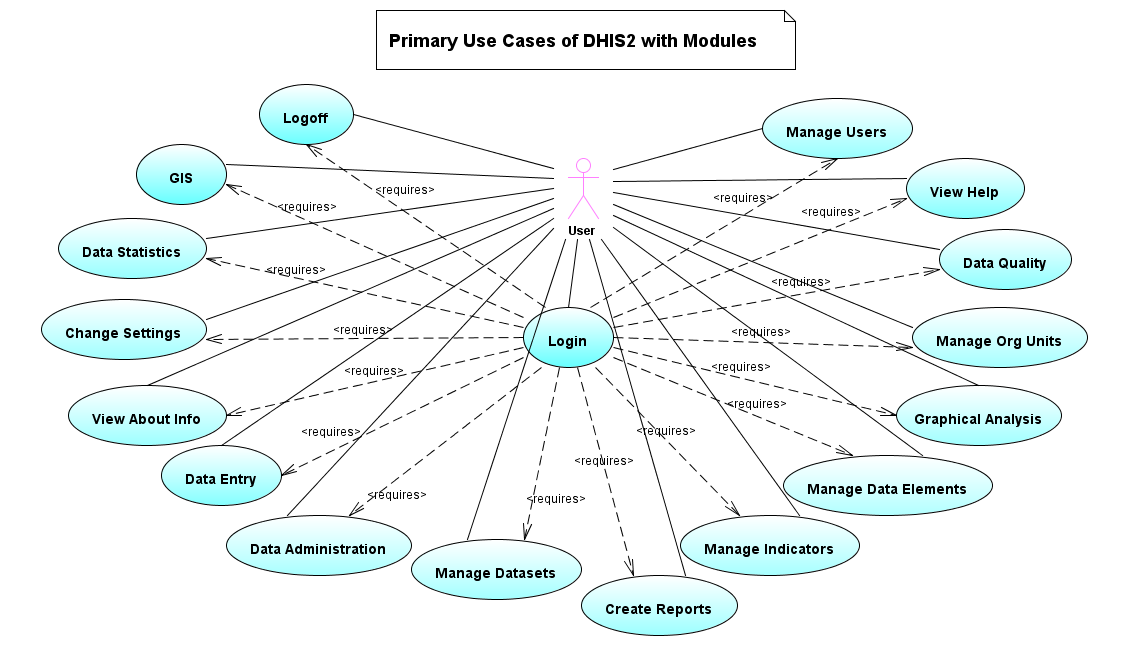
## *The user should be able to* **VIEW HELP ON THE APPLICATION**

***Rationale:*** The user should be able to receive online help in the application

## *The user should be able to* **SEE APPLICATION INFORMATION**

***Rationale:*** The user should be able to look at the application information like current user, version, browser agent, configuration directory, environment variables, database names and database user from the about screen of the system

Use-Case Diagram

Figure - PRIMARY USE-CASE OF DHIS2 MODULES

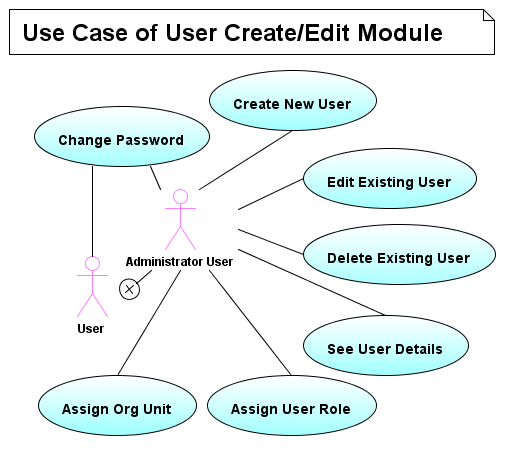


Figure - USE-CASE of USER MANAGEMENT MODULE

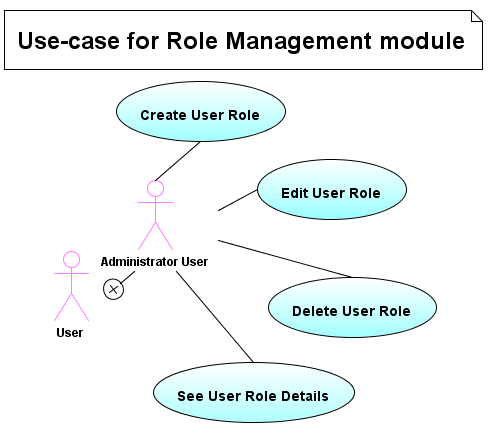
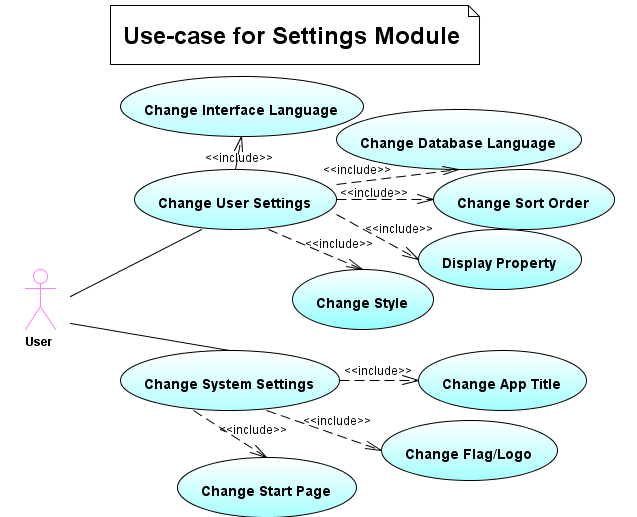


Figure - USE-CASE of ROLE MANAGEMENT MODULE

Figure - USE-CASE of SETTINGS MODULE



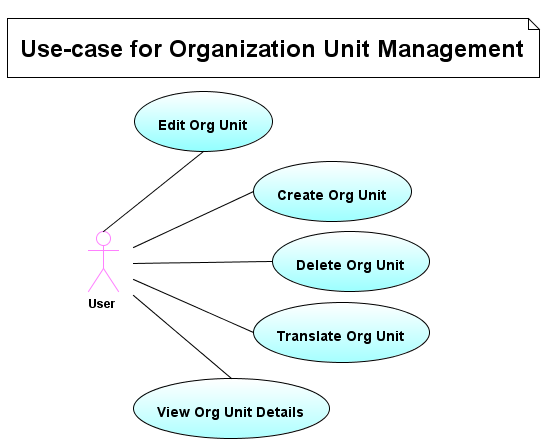


Figure - USE-CASE of ORGANIZATION UNIT MANAGEMENT MODULE

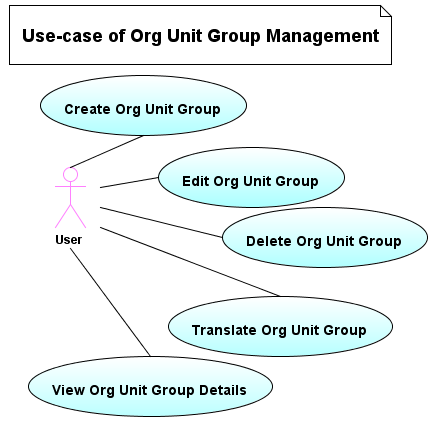


Figure - USE-CASE of ORGANIZATION UNIT GROUP MANAGEMENT MODULE

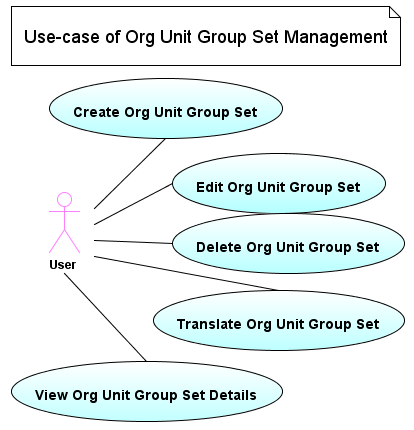


Figure - USE-CASE ORGANIZATION UNIT GROUPSET MANAGEMENT MODULE

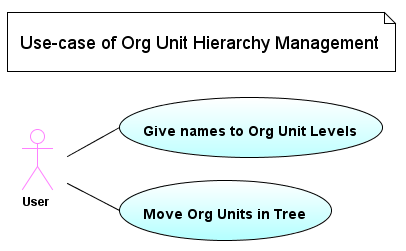
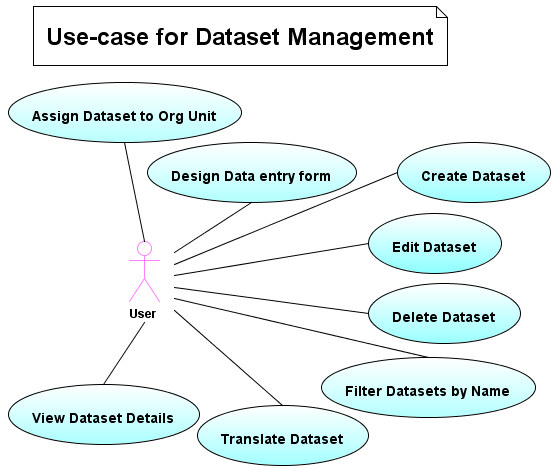
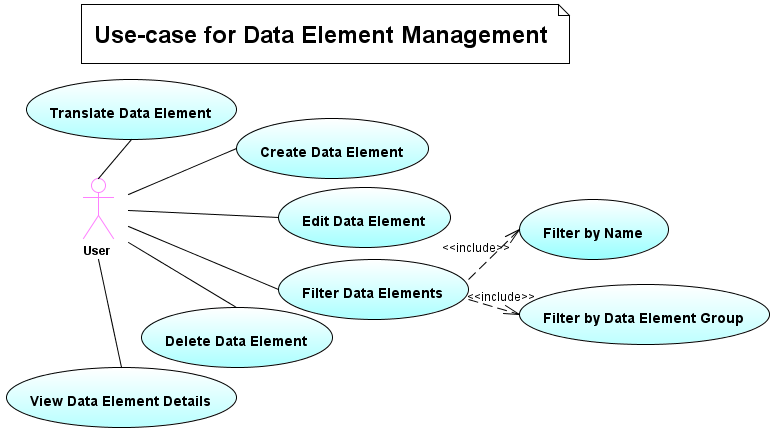


Figure - USE-CASE of ORGANIZATION UNIT HIERARCHY MANAGEMENT MODULE

Figure - USE-CASE of DATASET MANAGEMENT MODULE



Figure - USE-CASE of DATA ELEMENT MANAGEMENT MODULE

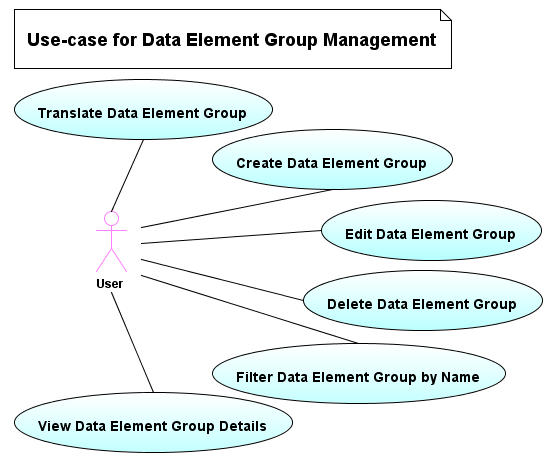


Figure - USE-CASE of DATA ELEMENT GROUP MANAGEMENT MODULE

Figure - USE-CASE of INDICATOR MANAGEMENT MODULE

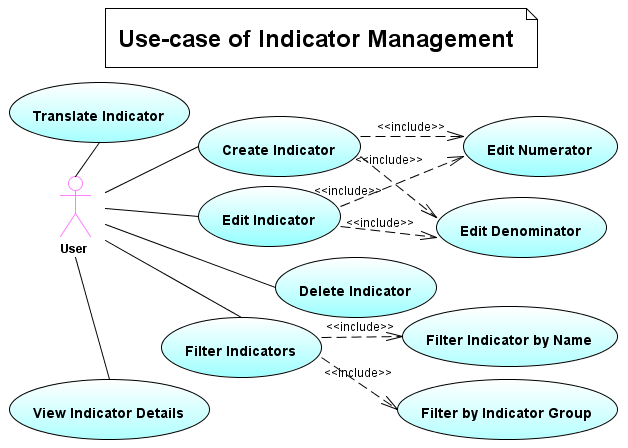
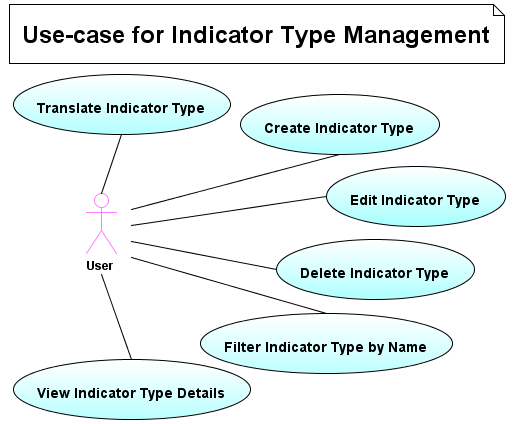


Figure - USE-CASE of INDICATOR TYPE MANAGEMENT MODULE



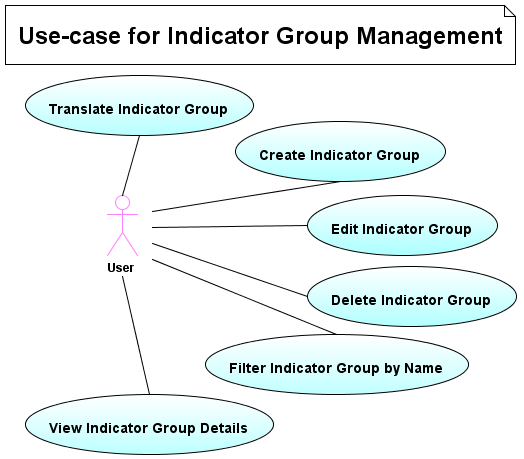


Figure - USE-CASE of INDICATOR GROUP MANAGEMENT

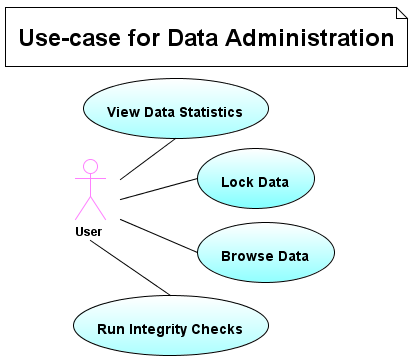


Figure - USE-CASE of DATA ADMINISTRATION MODULE

Figure - USE-CASE of DATA ENTRY MODULE

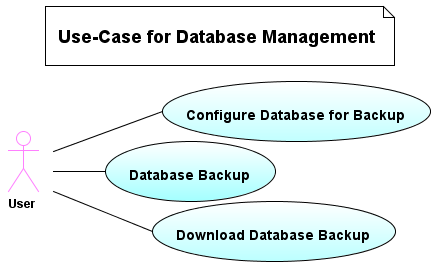


Figure - USE-CASE of DATABASE MANAGEMENT MODULE

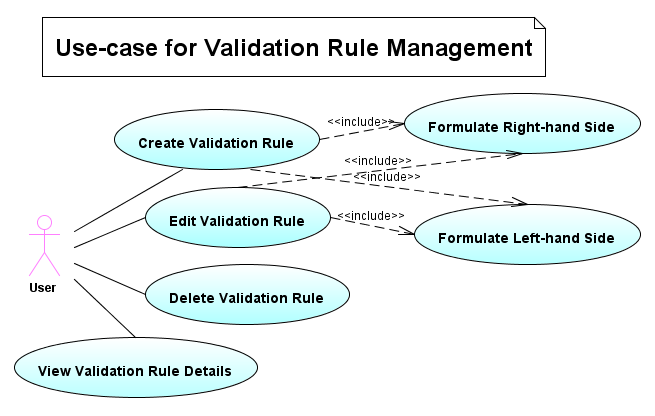
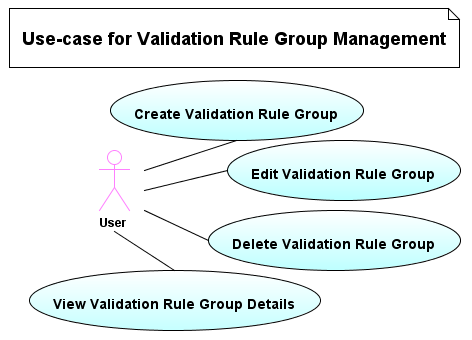
Figure - USE-CASE OF VALIDATION RULE MANAGEMENT MODULE

Figure - USE-CASE of VALIDATION RULE GROUP MANAGEMENT MODULE



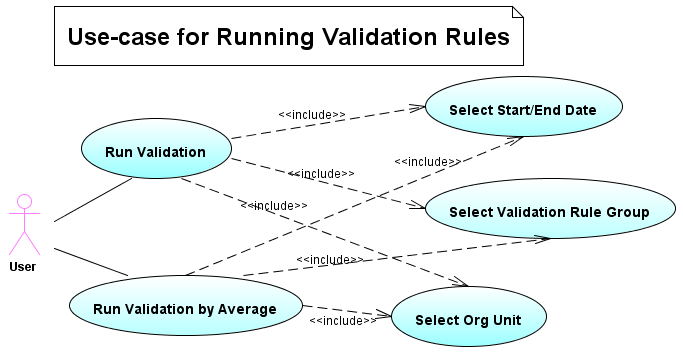
Figure - USE-CASE of RUNNING VALIDATION RULES

Figure - USE-CASE of DATA ANALYSIS

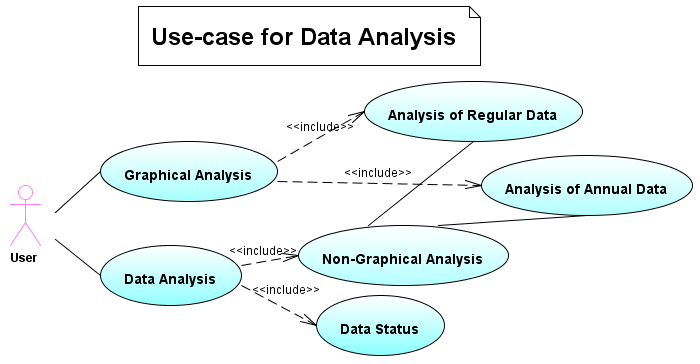
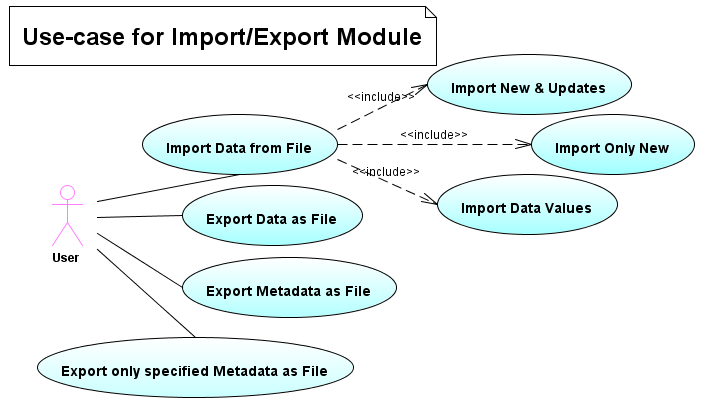


Figure - USE-CASE of IMPORT/EXPORT MODULE



# System Requirements

## System Functional Requiremnets

### ***The system shall*** BE BUILT AS A COMBINATION OF MODULES

**Description:** The system shall be built by using a combination of modules that can be assembled to create a common portal based on user’s functional requirements

**Inputs:** The implementer chooses the modules that are to be part of the DHIS2 Portal.

**Source:**  Maven.

**Outputs:**  A web application that can deployed into the Servlet Container.

**Destination:** N/A

**Action:** Run ‘mvn clean install’ on the source root folder

**Requires:** The implementer modifies the pom.xml configuration file.

**Pre-Condition:** The source code is present for required modules. Maven and JDK are correctly configured. All the necessary libraries are present or internet connection is present to automatically download the libraries from central maven repositories.

**Post-Condition:** The implementer will find the dhis2.war file in the target folder, as well as built jar files for all the compiled modules.

**Side Effects:** N/A.

### ***The system shall*** NOT DISPLAY PAGES WITHOUT VALID USER CREDENTIALS

**Description:** The system will not display pages only without valid username and password for the system.

**Inputs:** Username and password entered into login screen for an existing user

**Source:**  The User

**Outputs:**  Displays the page

**Destination:** The start page configured in the system settings.

**Action:** The system will always load the login page, if the user opens a page without logging into the system. The web page location is filtered to check valid login from the user before showing the page.

**Requires:** Valid username and password.

**Pre-Condition:** The user should exist in the system

**Post-Condition:** N/A

**Side Effects:** The startup page is always loaded after login even if the user is shown the login screen from a different page

### ***The system shall*** CREATE/EDIT USER WITH VALID DETAILS

**Description:** The system requires some basic details like username, password, surname, first name to create/edit a user.

**Inputs:** Username, 2 Matching passwords, Surname and First name

**Source:**  The User

**Outputs:**  A new user is created

**Destination:** A list of users is shown

**Action:** The system shall create the new user and show the list of users after the user has entered the correct details and saved

**Requires:** N/A

**Pre-Condition:** N/A.

**Post-Condition:** N/A.

**Side Effects:** The project GUI is updated and the files tree view is displayed in the files explorer.

## Nonfunctional Requirements

### Product Requirements

#### 1.1 Performance Requirements

* **Speed**

The system will be accessed from the lowest levels of health hierarchy to the highest levels. Thus, the system is designed to be fast and scalable across different types of internet connections. Large-sized graphics are avoided, so that the system can work fast in low bandwidth conditions. The system is designed to be scalable and allows data entry and data analysis from multiple levels of health system hierarchy.

* **Space**

The system should allocate memory efficiently, so as not to cause degradation in the performance and speed. The system can be hosted on a single server or across 2 servers i.e. application server and database server to manage load and share space between the servers.

The product should consume the minimal amount of memory from the user’s computer. Since only the browser is used on the client computer, the space and processor requirements are bare minimum.

#### 1.2 Reliability Requirements

According to the IEEE, reliability is defined as “the probability that software will not cause the failure of a system for a specified time under specified conditions”. The system is designed to work reliably under full load. The system can reliably generate reports, graphs, charts, maps when under full load of data entry.

The system should work at all times since data entry may be done from anywhere at any time. The system should deal with denial of service (DOS) and distributed denial of service (DDOS) attacks which are common for large systems on the internet. The system should also work reliably for clients within VPN and wireless networks.

#### 1.3 Usability Requirements

The application should be user-friendly and easy to use. Like explained in the User Characteristics, the software users come from different backgrounds and needs to be easy for everyone to use. Both data entry and data analysis should be easy to use. The system should have good readability and accessibility controls. The system should be available in local language and all components of the system should be internationalized. The system controls should be intuitive and user need not go through a lot of training to use the application.

The system should be customizable to meet the user needs. Thus, it can bring in the current formats and reports into the system, making it easier for users to understand.

#### 1.4 INTER-OPERABILITY

DHIS2 should be inter-operable with other types of health information systems. Being open-source, there are more chances of the system being inter-operable with other products. The system should support a wide variety of information exchange formats like DXF, IXF, SDMX etc.

#### 1.5 Correctness

The system should behave as expected. User and functional requirements should be met. The system should support synchronization between offline and online versions of the application and should result in correct update of data. Reports that are to be sent to other application should be correctly generated and state of data stored in the system should not automatically change over time.

### Organization Requirements

#### 2.1 Implementation Requirements

The implementation requirements come from a variety of sources - from grass-root users to ministries of health. Because of the lack of internet availability at all health facilities, the system should also be able to work in offline mode and the system should be able to synchronize data between such offline and online systems. Data Elements, Datasets, Organizational hierarchies, Indicators, Validation rules and what modules to use are all customizable to the needs of the implementation.

#### 2.2 Delivery Requirements

The system is incrementally developed through the use of modules. Each module is to be delivered at a different time and depends on the implementation need. But the timeframe of delivery of a module may range from 1-6 months. The initial portal architecture will be developed in a period of 6-12 months.

# External Interface Requirements

## User Interfaces

Since DHIS2 is a web application, it can be viewed through a web browser. Mozilla Firefox is the recommended browser for using DHIS2. JavaScript and Cookies must be enabled in the browser for the system to work properly. Only CSS2 complaint browsers would be able to display the system correctly. The user is advised to use a minimum screen resolution of 1024x768 for correct view of the user interface.

For GIS functionality, the user interface requires a SVG renderer. This can be browser native like in Firefox 3 or through a plug-in like Adobe SVG Viewer for Internet Explorer.

## Hardware Interfaces

The system is hardware independent and does not require any special hardware.

## Software Interfaces

The system makes use of Java and Java-based web frameworks like Struts, Spring, Hibernate, Webwork and their APIs. The system also uses JavaScript, Velocity, SVG for the user interfaces.

# System Architecture

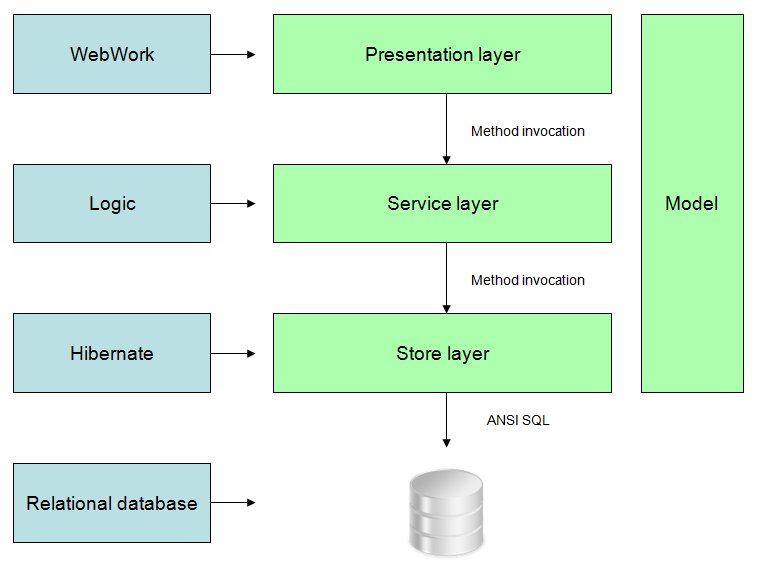


Figure - System Architecture

A detailed system architecture description can be found in the Technical Architecture document that is part of the source code.

**Available Modules**

DHIS2 plans to have the following web modules available and new modules are developed on need:

1. Dashboard-Integration Module
2. Linelisting Module
3. Data entry Module
4. Datamart Module
5. GIS Module
6. Import/Export Module
7. Indian Dashboard Module
8. JForum Integration Module
9. Data Admin Maintenance Module
10. Data Dictionary Maintenance Module
11. Dataset Maintenance Module
12. Organization Unit Maintenance Module
13. Settings Maintenance Module
14. User Maintenance Module
15. NRHM Reports Module
16. OpenHealth Integration Module
17. Reporting Module
18. Validation Rule Module

# References

1. The Struggle for District-Based Health Information Systems in South Africa, Braa & Hedberg
2. Modularization and Demodularization: Levels of a Java Web Application for Open Health, Torgeir Lorange Østby
3. Technical Architecture, DHIS2
4. DHIS2 Wiki - <http://208.76.222.114/confluence/display/DHIS2/Home>
5. DHIS2 User Manual