

Software Requirements Specification

From TeamF::CGT

Contents

- 1 Abstract
- 2 Introduction
 - 2.1 Purpose of the Document
 - 2.2 Scope of the Document
- 3 Project Information
 - 3.1 Team F
 - 3.2 The Supervisor
 - 3.3 The Client
 - 3.4 Acronyms
 - 3.5 Definitions
 - 3.6 References
- 4 Overview Description
 - 4.1 Project Information
 - 4.2 The Client
- 5 Existing System
 - 5.1 Context
 - 5.2 System Functionalities
 - 5.3 Software Environment
 - 5.4 User Characteristics
 - 5.5 Problems
- 6 Proposed System
 - 6.1 Context
 - 6.1.1 The purpose of Corporate Governance Tool
 - 6.1.2 Corporate Governance
 - 6.2 Project Objective and Scope
 - 6.3 User Characteristics
 - 6.3.1 Consultants and Administrators
 - 6.3.2 Users of the End-System of CGT
 - 6.4 System Characteristics

- 6.5 System Functionalities
 - 6.5.1 High Level Description of a Typical End System of CGT
 - 6.5.2 High Level Description of the Functionalities of CGT
 - 6.5.3 Low-Level Description of the Functionalities of CGT
- 6.6 Software Interface
- 6.7 High Level Use Case Diagrams
- 7 Functional Requirements for First Stage - Setting up System Structure
 - 7.1 Create Database Tables
 - 7.1.1 Proof of Concepts
 - 7.1.2 Dependencies
 - 7.1.3 Related GUI
 - 7.2 Set Icons for Tables and Attributes
 - 7.2.1 Proof of Concepts
 - 7.2.2 Dependencies
 - 7.3 Set Special Attribute Properties
 - 7.3.1 Description
 - 7.3.2 Proof of Concepts
 - 7.3.3 Dependencies
 - 7.4 Support Data Types
 - 7.4.1 Description
 - 7.4.2 Proof of Concepts
 - 7.4.3 Dependencies
 - 7.5 Edit Table Properties
 - 7.5.1 Description
 - 7.5.2 Proof of Concepts
 - 7.5.3 Dependencies
 - 7.6 Delete Tables
 - 7.6.1 Description
 - 7.6.2 Proof of Concepts
 - 7.6.3 Dependencies
 - 7.7 Input Records into Tables
 - 7.7.1 Description
 - 7.7.2 Proof of Concepts
 - 7.7.3 Dependencies
 - 7.8 Modify Records in Tables
 - 7.8.1 Description
 - 7.8.2 Proof of Concepts
 - 7.8.3 Dependencies

- 7.9 Duplicate Records
 - 7.9.1 Description
 - 7.9.2 Proof of Concepts
 - 7.9.3 Dependencies
- 7.10 Define Values in Relation to Other Values in Database
 - 7.10.1 Description
 - 7.10.2 Proof of Concepts
 - 7.10.3 Dependencies
- 7.11 Set Constraints (High Non-Core)
 - 7.11.1 Description
 - 7.11.2 Proof of Concepts
 - 7.11.3 Dependencies
- 7.12 Set Triggers (High Non-Core)
 - 7.12.1 Description
 - 7.12.2 Proof of Concepts
 - 7.12.3 Dependencies
- 7.13 Create Tree Structure
 - 7.13.1 Description
 - 7.13.2 Proof of Concepts
 - 7.13.3 Dependencies
- 7.14 Create Multi-value Attributes
 - 7.14.1 Description
 - 7.14.2 Proof of Concepts
 - 7.14.3 Dependencies
- 8 Functional Requirements for Second Stage - Defining Views
 - 8.1 View Database Trees
 - 8.1.1 Description
 - 8.1.2 Proof of Concepts
 - 8.1.3 Related Requirements
 - 8.2 Set ActiveReports Report as View
 - 8.2.1 Description
 - 8.2.2 Proof of Concepts
 - 8.2.3 Related Requirements
 - 8.3 Set CrystalReports Report as View (Low Non-Core)
 - 8.3.1 Description
 - 8.3.2 Proof of Concepts
 - 8.3.3 Related Requirements
 - 8.4 Set Dundas Chart as View

- 8.4.1 Description
 - 8.4.2 Proof of Concepts
 - 8.4.3 Related Requirements
- 8.5 Set ChartFX Chart as View (Low Non-Core)
 - 8.5.1 Description
 - 8.5.2 Proof of Concepts
 - 8.5.3 Related Requirements
- 8.6 Set Default View
 - 8.6.1 Description
 - 8.6.2 Proof of Concepts
 - 8.6.3 Related Requirements
- 8.7 Plain Text Browser-Based CSS Editor
 - 8.7.1 Description
- 8.8 Load CSS Templates
 - 8.8.1 Description
- 8.9 Support Multi-value Attributes
 - 8.9.1 Description
- 9 Functional Requirements for Third Stage - Using End-System
 - 9.1 View Records by Tree
 - 9.1.1 Description
 - 9.1.2 Proof of Concepts
 - 9.1.3 Related Requirements
 - 9.2 View Records by Spider Web View (Medium Non-Core)
 - 9.2.1 Description
 - 9.2.2 Proof of Concepts
 - 9.3 View Records from Bird's Eye View (Medium Non-Core)
 - 9.3.1 Description
 - 9.3.2 Proof of Concepts
 - 9.4 Filter Records
 - 9.4.1 Description
 - 9.4.2 Proof of Concepts
 - 9.4.3 Related Requirements
 - 9.5 Show Views
 - 9.5.1 Description
 - 9.5.2 Proof of Concepts
 - 9.5.3 Related Requirements
 - 9.6 Context-Specific Buttons
 - 9.6.1 Description

- 9.6.2 Proof of Concepts
 - 9.6.3 Related Requirements
 - 9.7 Select View for Record Type
 - 9.7.1 Description
 - 9.7.2 Proof of Concepts
 - 9.7.3 Related Requirements
 - 9.8 Select Default View
 - 9.8.1 Description
 - 9.8.2 Proof of Concepts
 - 9.8.3 Related Requirements
- 10 Non-Functional Requirements
 - 10.1 Browser Compatability
 - 10.2 Software System Attributes
- 11 User Documentation
 - 11.1 Tool Tips
 - 11.2 User Manual
 - 11.3 Maintenance Documentation
 - 11.4 Online Help
- 12 Acceptance Criteria
 - 12.1 Proof of Concepts: Create Database Tables
 - 12.1.1 Create Table that Represents Risk
 - 12.1.2 Create Table that Represents Location
 - 12.1.3 Create Table that Represents Control
 - 12.1.4 Create Table that Represents Party
 - 12.1.5 Create Table that Represents Cause of Risk
 - 12.1.6 Create Table that Represents Treatment Plan
 - 12.1.7 Create Table that Represents Treatment Plan Task
 - 12.1.8 Create Table that Represents Review
 - 12.1.9 Create Table that Represents Employment Position
 - 12.1.10 Create Table that Represents Contact Details
 - 12.1.11 Create Table that Represents Users
 - 12.1.12 Create Table that Represents Employee
 - 12.1.13 Create Table that Represents Work Group
 - 12.1.14 Create Table that Represents Category
 - 12.1.15 Create Table that Represents Measure Scale
 - 12.1.16 Create Table that Represents an Indicator
 - 12.1.17 Create Table that Represents Email
 - 12.1.18 Create Table that Represents Email Attachments

- 12.1.19 Create Table that Represents a Survey Template
- 12.1.20 Create Table that Represents a Survey
- 12.1.21 Create Table that Represents a Question
- 12.1.22 Create Table that Represents a Result
- 12.1.23 Create Table that Represents a Question Pool
- 12.1.24 Create Table that Represents an Audit
- 12.2 Proof of Concepts: Set Special Attribute Properties
 - 12.2.1 Log Attributes (Non-Core)
 - 12.2.2 Set Visibility of Attributes
 - 12.2.3 Report on Attributes
- 12.3 Proof of Concepts: Input Records into Tables
 - 12.3.1 Preset Data in Control Table
- 12.4 Proof of Concepts: Duplicate Records
 - 12.4.1 Duplicate Risk Record
- 12.5 Proof of Concepts: Define Values in Relation to Other Values
 - 12.5.1 Calculation of Risk Attribute
 - 12.5.2 Calculation of Risk Average Likelihood
 - 12.5.3 Calculation of Risk Severity Attribute
 - 12.5.4 Calculation of Treatment Plan's Attribute
 - 12.5.5 Calculation of Risk's Total Control Cost
 - 12.5.6 Calculation of Tolerance Attribute
 - 12.5.7 Calculation of Risk Attribute in Relation to Child
- 12.6 Proof of Concepts: Create Tree Structure
 - 12.6.1 Hierarchical Tree Structure
 - 12.6.2 Employment Title Hierarchies
 - 12.6.3 Risk Hierarchies
 - 12.6.4 Compliance Hierarchies
 - 12.6.5 Categorical Hierarchies
- 12.7 Proof of Concepts: Set Constraints (High Non-Core)
 - 12.7.1 Constraints on Treatment Plans
 - 12.7.2 Constraints on Viewing Risks
 - 12.7.3 Constraints on Controls
 - 12.7.4 Read-Only Option
 - 12.7.5 Related Requirements
- 12.8 Proof of Concepts: Set Triggers (High Non-Core)
 - 12.8.1 Record Trigger (high non-core)
 - 12.8.2 Cost/Effectiveness Trigger (high non-core)
 - 12.8.3 Overdue Trigger (high non-core)

- 12.8.4 Changed Category Trigger (high non-core)
- 12.8.5 Out-of-Tolerance Trigger (high non-core)
- 12.8.6 Survey Trigger (high non-core)
- 12.8.7 Snapshot Trigger (high non-core)
- 12.8.8 Duplicate Trigger (high non-core)
- 12.8.9 Create Log (high non-core)
- 12.9 Proof of Concepts: Set ActiveReports Report as View
 - 12.9.1 Create ActiveReport Report from Risk Data
- 12.10 Proof of Concepts: Set Dundas Chart as View
 - 12.10.1 View Risk Data in a Matrix
 - 12.10.2 Active cells in Matrix (non-core)
 - 12.10.3 Drill Down (non-core)
 - 12.10.4 Combined or Overlaid Matrices (non-core)
 - 12.10.5 Lines Between Items (non-core)
 - 12.10.6 Legends for Matrices
 - 12.10.7 Interchangeable Legends
 - 12.10.8 Searchable Matrix (non-core)
- 13 Client Sign-off
- 14 Notes

1 Abstract

This document formally specifies the requirements of the Corporate Governance Toolkit. The purpose of this document is to facilitate an agreement between the client and Team F as to the software product to be created. It will also be used in design, testing, and validation and verification in the later stages of this project.

2 Introduction

2.1 Purpose of the Document

The purpose of this document is to formally outline all the software requirements for the Corporate Governance Toolkit and to ensure the software produced satisfies the client's expectations.

Also, this document serves as:

- a technical contract that binds the client and team to an agreement. The client agrees that the requirements specified are accurate while the team consents to producing the software according to the requirements specified;
- a reference for the stake holders involved in the development of the software. The document serves to present the requirements to the technical and non-technical stake holders in a concise manner.

The intended audience of this document are the team members and the client.

2.2 Scope of the Document

This SRS will provide details about the general and specific requirements that are associated with the software. This SRS will propose a proof of concept with each functional requirement stated. The reasoning behind for the proof of concept is that the software is abstract and flexible.

3 Project Information

3.1 Team F

The team that will be responsible for creating the software will be 433-440^[1] Team F, also known as Team F. The members are:

Team F members' details and roles

Name	Team Role	Email(@students.csse.unimelb.edu.au)
Stephan Taitz	Project Manager	sptaitz
Tim Fang	Project Manager	tfang
Derek Tong	Client Liaison	tong
Amiel Zwier	Quality Assurance	aczwier

Fan Zhang	Software Admin	fzhan
Ivan Seow	Research Manager	iwhs
Jack Gao	Design Manager	jbgao
Luke Jones	Risk Manager	ljjones
Shankar Sribalachandran	Requirements	srib
Sie Ming Huang	Requirements	smhuang
Yan Ming (Eric) Huang	Implementation Manager	ymh
Zhao Hui (Gary) Liang	Test Manager	zhl

3.2 The Supervisor

The project supervisor is Steele Griffiths.

3.3 The Client

The Client is Risk Wizard. Risk Wizard provides corporate governance software and services for risk management, performance, voting systems and whistle-blowing. Risk Wizard has clients based in North America Europe, Middle East and Australasia. Risk Wizard's Risk Management Software is a user-friendly system for managers to analyze risk and adopt a holistic approach regardless of size, industry, location or complexity. For more information regarding the current system, please refer to Existing System.

3.4 Acronyms

Abbreviations of terms

Term	Definition
CGT	Corporate Governance Toolkit
Client	Risk Wizard
ChartFX	A Graphical Program
Dundas	Another Graphical Program
End-User	The users of the End-System
End-System	A system created with the CGT
GUI	Graphical User Interface
PDF	Portable Document Format

Stakeholders	People who have a share in the project
SRS	Software Requirements Specification
RW	Risk Wizard
W3C	World Wide Web Consortium

3.5 Definitions

Core: In the context of the Corporate Governance Toolkit project, the term 'core' is used throughout this document to describe requirements that are essential functionalities of CGT. All core requirements must be implemented satisfactorily for the software to be accepted by the client.

High Non-Core: The term "High Non-Core" used to describe requirements that are not essential to the functionalities of CGT. But they must be implemented satisfactorily for the software to be accepted by the client.

Medium Non-Core: The term "Medium Non-Core" used to describe requirements that give desirable functionalities to CGT. While these requirements are expected to be implemented, but they are not included as part of the requirements that must be implemented for the software to be accepted by the client.

Low Non-Core: In the context that the term "Low Non-Core" is being used throughout this document, it refers to requirements which have the lowest priority in being implemented. These additional requirements are to be implemented only if the time permits.

3.6 References

Risk Wizard Website - <http://www.riskwizard.com>

4 Overview Description

4.1 Project Information

The Corporate Governance Toolkit (CGT) is a toolkit that can be used to create customised corporate governance applications. These customised applications may include:

- Risk Management Systems;

- Compliance Systems;
- Performance Management Systems; and
- other Corporate Governance Systems.

4.2 The Client

The Client will use the Corporate Governance Toolkit in multiple ways. The Client may:

- Distribute the Corporate Governance Toolkit as is, with the intention that the end-user will use the Corporate Governance Toolkit in order to create their own corporate governance software; or
- Consult with an end-user, create customised corporate governance software for the end-user, then distribute it to the end-user.

5 Existing System

5.1 Context

The existing system, Risk System is a tool that assists an organisation identify, analyse, measure and report the risks it faces. The Risk system essentially provides an organisation with a central database that holds information about risks from all the different division and produce a report that calculates the risk profiles and the impact on each division and on the whole organisation.

Risk System was developed our client, Risk Wizard to manage the risks as recommended by Australian Risk Management Standards. It has been sold to their clients as packaged software.

5.2 System Functionalities

Risk System software breaks the risk management into Identification, Analysis, Evaluation, Treatment, Review and Reporting phases. It allows the users to enter information about a risk after a user has completed the recommended risk management procedures in these phases. Risk System also produces necessary reports to support the analysis of risks.

The functions of the system are described below during each phase of risk management:

- Identification: Allows the users to create new risk by entering information about its characteristics.
- Analysis: Allows the users to enter information from the analysis of a risk such as the impact of the risk and parties affected
- Evaluation: Risk System calculates the risk rating for the entity for a particular risk before and after a preventive action is taken.
- Treatment: Allows the users to enter information about how to tackle a risk.
- Review: User enters the information from the outcome of the review of risk.
- Reporting: Produces the a comprehensive report that details attributes of the risk such as source of risk and parties affected and calculates the impact of risk for the work group that risk belongs to and for the whole organization.

5.3 Software Environment

The Risk System is currently available as both a web-based system and stand-alone to be used on personal computers. Both web-based and stand-alone offers the same functionalities and similar user interfaces.

5.4 User Characteristics

The Risk System is currently used in medium businesses to large multi-antintional companies and government departments. The primary users of the current software are professionals working in an organization.

5.5 Problems

The major limitation of the current software is the inability to easily customize the software to match the users' needs. Risk System is based on the recommended Australian Risk Management Standards however the risk management policies tend to vary across the organizations, as the result, the existing system can not support the organizations' needs. In particular, the fields of data become irrelevant and certain phases are not applicable to certain organization. This makes it cumbersome for the organizations to implement their risk management policy using Risk System. The lack of flexibility reduced the opportunity for Risk Wizard to sell the product.

There are also some technical deficiencies with the current system. Risk System acts more of a data collection and presentation tool with capacity to produce only one non-customizable report.

Although the current system is unable to meet different organisation needs, it provides a good platform to build more powerful and sophisticated software.

6 Proposed System

6.1 Context

6.1.1 The purpose of Corporate Governance Tool

The aim of the proposed system, the Corporate Governance Toolkit (CGT) is to provide organizations with a single system to manage all their corporate governance needs. CGT must not only overcome the lack of customization of the existing system, but also provide multiple integrated tools as part of a single system to record, analyze and report on different corporate governance issues.

In order to better understand the purpose of the CGT, the nature of corporate governance must be understood.

6.1.2 Corporate Governance

Corporate governance is a term used to describe the organizational structure, business policy, guidelines, internal and external regulation and performance monitoring mechanisms that ensure that the corporate affairs are conducted in a transparent and accountable manner. The objectives of the corporate governance policies are to promote efficient use of resources and give a sense of direction and control to an organization.

While the corporate governance policies of different organization may share some common principles, they are often different and are tailored to suit each organization's unique culture, their set of goals and the industry they operate in. For example, organizations, operating in different industries need to comply with different government, accounting, health and environmental regulations. These organizations need tools to assist them with compliance and control. An Organization that operates in a dynamic market such financial trading will need regular risk assessments of its operations and will require tools that assist in risk management. All organizations typically measure the performance of their various business units and staffs to formulate their business strategy and reward their staff.

Given the nature of corporate governance, a centralized software that contain integrated multiple tools that are tailored to manage different aspects of an organization's corporate governance would provide an organization with a powerful and sophisticated business tool.

6.2 Project Objective and Scope

CGT aims to provide an organisation with a single centralised system that consists of all the different tools needed to implement an organisation's corporate governance policies. Given that each organization's corporate policy is quite different, any tool built must be tailored to an organization's needs.

CGT addresses these two identified needs for the corporate governance software by providing web-based software that gives the users the capacity to build their own corporate governance tool kit.

In order to achieve this goal, CGT must be able to dynamically build any tools needed to address an area of corporate governance and integrate these different tools into a single functioning system.

CGT endeavors to be able to create applications to support Risk Management, Control and Compliance Management, Privacy Management, Dashboards, Activity Alert, Performance Management, Incident management and Issue Registers. However, given the limited resources and time constraints of the 440 team, the CGT software will only be able to dynamically create applications or end-system that support:

- Risk Management
- Compliance Management
- Performance Management

CGT will be expanded in the future by Risk Wizard to support the creation of other types of tools.

6.3 User Characteristics

The CGT software and the End-System will be used within medium to large multinational organizations. The End-System will be accessible to the users of End-System on the organization's intranet, internet or via the LAN/WAN.

The target users of CGT belong to three categories:

1. Consultants
2. Administrators of the end-system of CGT
3. Users of the end-system of CGT

6.3.1 Consultants and Administrators

The consultants and the administrators are the users of CGT who create an end-system using CGT. The consultants are the employees of our client, Risk Wizard who may be asked by an organisation to build them a particular corporate governance tool.

Administrators refer to employees of the clients of Risk Wizard. Administrators will also build or modify the end-product.

The consultants and the administrators will be expected to have comprehensive knowledge of the corporate governance policy of an organization. They are also expected to be familiar with the basics of database structure such as different data-types and the constraints placed on them, in order to create an end-system in CGT.

6.3.2 Users of the End-System of CGT

The users of the end-system developed using CGT will typically be employees of an organization. These users are expected to possess good computer skills and are not expected to require any training to familiarize them with the interface of the end-system of CGT.

6.4 System Characteristics

The CGT should be robust in order to allow rapid development of applications that facilitate implementation of a corporate governance policy. While the software is geared towards dynamically building corporate governance tools, it is essential that software remains as much flexible as possible in order for it to be extended. The key success factors of CGT software are as follows (not in any order of importance):

- Flexibility to cater for organization differences.
- Provide facilities to easily customize the end-system.
- Graphic aesthetics of the end-system
- Support for languages other than English.
- Build applications that interact and inter-link and remain integrated within a single system
- Efficient information sharing, filtering and retrieval in the end-system
- Extendibility to facilitate future development
- Process automation
- Ease of maintenance

6.5 System Functionalities

Since CGT is used to dynamically build an application, the functionalities of the end-system of CGT must be understood so that CGT can provide the means to build the end-system with the required functionalities.

6.5.1 High Level Description of a Typical End System of CGT

While not all the corporate governance tools are same, there are fundamental principles or similarities across all corporate governance tools.

A typical corporate governance tool, therefore the end system of CGT must have the following functionalities:

- **Information Collection and Storage:** The end-system must have means of collecting information from users and store them in a data repository for future retrieval. Since collection of the information about various corporate aspects is fundamental to allow analysis, the information must be validated to ensure the integrity of data and shared in an efficient manner. The user must be able to view the information and update the information in the database. For example, In an end-system for risk management, the users of that end system will need to define risk and enter information about the risks.
- **Analysis of Information:** The end-system must be able to facilitate analysis of information, stored in the database. The end-system must perform calculations and interpret the information to produce meaningful reports. These reports must be displayed to the users. For example, an end-system for risk management must be able to retrieve information about risks of a particular department and produce risk profile for that department based on the specified risk-rating calculations.

- **Generation of Reports and Graphs:** The end-system will need to produce various reports and graphs based on the information. For example, the end-system for performance management may need to generate a bar graph that shows the sales of different business units.
- **User Communication:** The end-system must be able to automatically send alerts by email to the users of an impending event. For example, an end-system for risk management may send a warning message by email to the users from a department when the risk rating for a particular risk increases above a certain level.
- **Privacy and User permissions:** The end-system may contain sensitive information the operation of an organization. Therefore the access to such information must be restricted to only few users. The end-system must be able to define the access of each users based on their level of authority in the organization and permission.

6.5.2 High Level Description of the Functionalities of CGT

CGT will provide the functionalities to build a typical end-system (see).

The primary functionalities of CGT are as follows:

- Set up structure of the end-system

CGT will provide the means to set up the structure of database to collect and store information and organise the information in a meaningful form.

CGT must also provide means to design the reports needed for the end-system and the facilities to view those reports in the end-system. This will require the CGT to provide functionalities that allow the mathematical calculations to be performed on the relevant information.

CGT must also provide ways to build an email communication in the end-system and the automatically send emails to users if a specified condition is met.

- Set up the interface for the end-system

CGT must provide means to create Graphical User Interface (GUI) for the end-system. The interface must be easily linked to the information stored in the database of the end-system and the defined reports to facilitate the collection of information, viewing of information reports and graphs in the end-system. CGT must allow the interface of the end-system to be easily customised.

- Compile and generate the end-system

CGT must be able to automatically compile the end-system once it has been built and allow the end-system to be deployed and used by the users of the end-system.

6.5.3 Low-Level Description of the Functionalities of CGT

The functionalities of CGT can be categorized into three broad groups:

1. Setting up the Structure of End-System
2. Designing the Interface for the End-System

Setting up the Structure of End-System

CGT must provide functionalities to set up the structure of the end-system by allowing the users to create the data structure and design reports for the end-system. The user of CGT will be expected to have a comprehensive understanding the corporate governance policy to be implemented and the knowledge of basic database design.

CGT provide the functionality to create a database as part of the end-system. The database table will be used as the fundamental unit of data structure to represent an object. CGT will create database table by providing the means to specify the name, its attributes and any constraints that apply on the table. For example, in a risk management end-system, a "risk" will be represented by a database table with all its attributes as the fields of the table. CGT will also have the functionalities to add, delete or modify any database tables. The CGT will allow the users to set the dependencies between the database tables.

In order to organise the information contained in the database in a meaningful form, CGT will allow the users to create a hierarchical tree structure. The tree structure will show the relationships between each database table.

CGT will have the functionalities to create reports. CGT will allow the users to select the database tables and attributes to be used to generate the reports and support mathematical functions used in report generation. CGT will produce reports in a format that can be read by Dundas or Chartfx reporting tools.

CGT will provide standard database structures and process automation to speed up the process of creating the end-system.

Designing the interface for the End-System

Once the structure of the end-system has been set up, CGT must provide the functionalities to set up the interface of the end-system so that the users of the end-system can enter information and view reports.

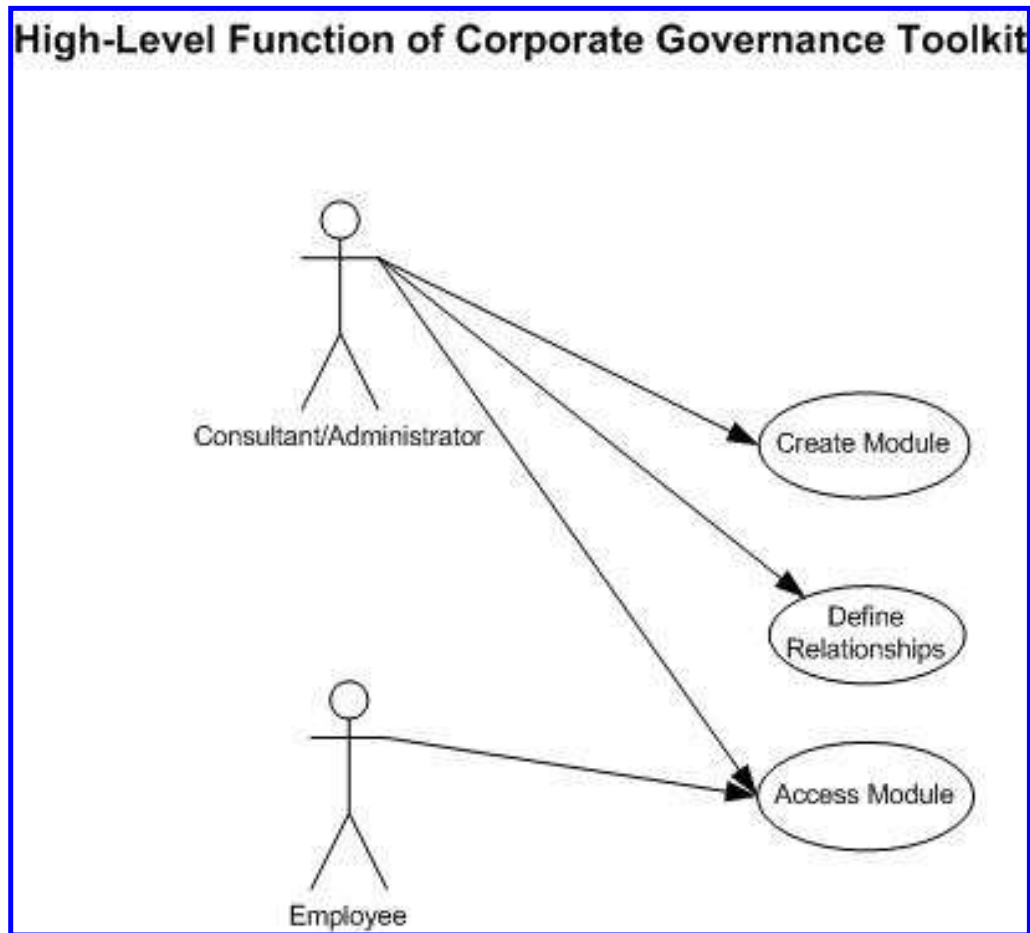
CGT will allow the users to define views for the objects in the database tables. CGT will provide a CSS editor to design the look and feel of the interface. CGT will also contain ready-made templates for the interface that users can import for the interface.

CGT will support the report design by using ActiveReports, Dundas Chart and ChartFX.

6.6 Software Interface

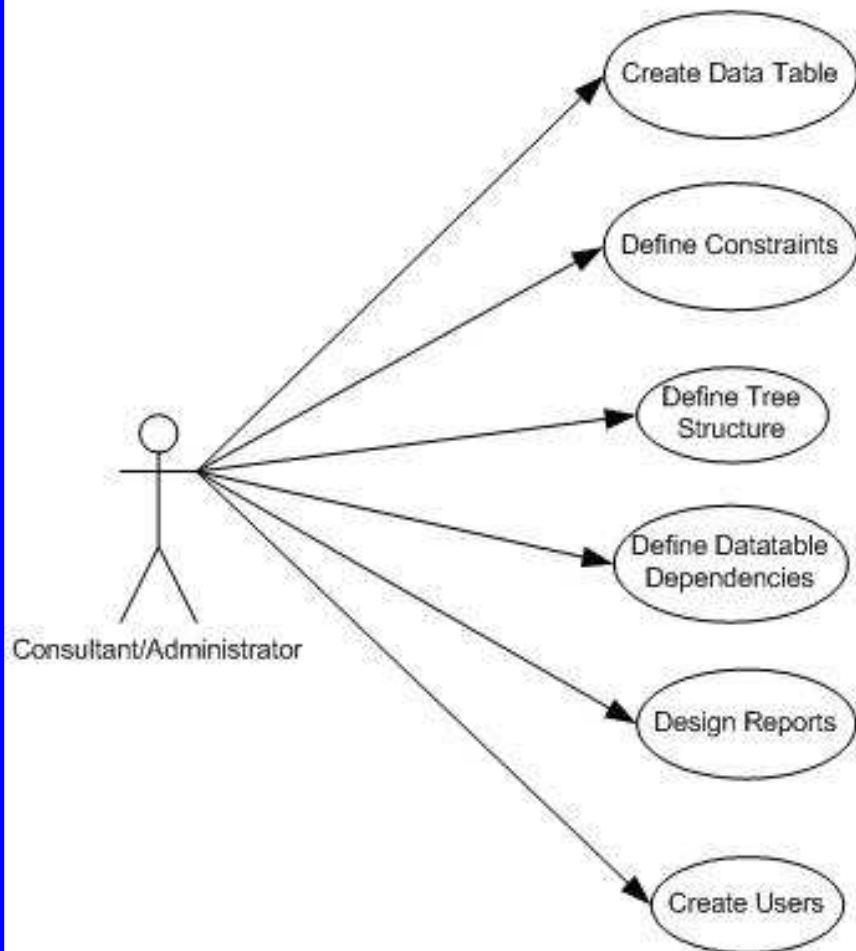
CGT and the end-system will be interfaced through a web browser that supports .NET UI control rendering, mainly Internet Explorer and FireFox. The interface of the software should be intuitive enough to allow rapid development of end-system. The interface of the end-system is extra important as it is one of the key success factors of the software. The software must provide an intuitive yet very eye-catching and customisable interface. The interface must facilitate very pleasant interaction with the users.

6.7 High Level Use Case Diagrams



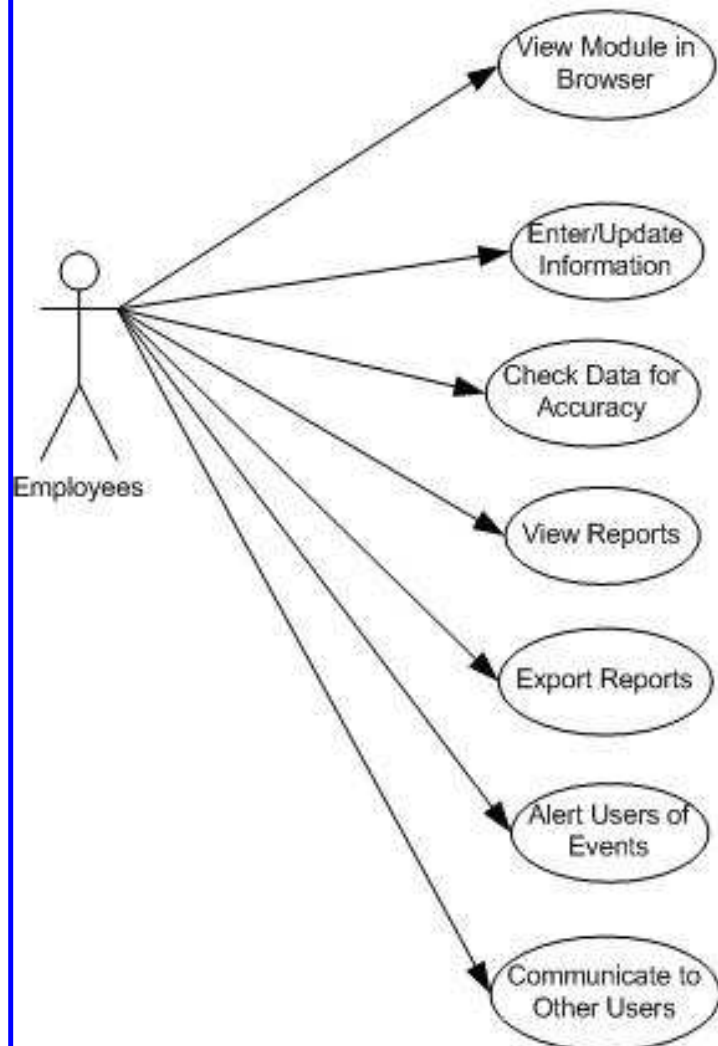
High Level Functions of CGT

High-Level Functions In Creating a Module in Corporate Governance Toolkit



Creating a Module in CGT

High-Level Functions of End-Product Created Using Corporate Governance Toolkit



Using a Module in CGT

7 Functional Requirements for First Stage - Setting up System Structure

This section outlines the functional requirements of the first stage of using the CGT system. In this stage a Consultant or Advanced User creates and configures the database structure of an End-System for use by an End-User. A sample GUI of how this stage may look is provided below for guidance only.

CGT

NEXT >

VIEW TABLE

INSERT RECORD
INTO SELECTED
TABLE

EDIT SELECTED
TABLE

DELETE SELECTED
TABLE

INSERT ICON FOR
SELECTED TABLE

CREATE TABLE

CREATE CONSTRAINTS

CREATE TRIGGER

CREATE TREE

LOGOUT

TABLES

RISK

CONTROL_

USER

CGT First Stage



7.1 Create Database Tables

A user shall be able to create a database table through a GUI by specifying:

1. a name;
2. the number of attributes;
3. the name of each attribute;
4. the primary key of the table (if applicable);
5. the foreign keys of the table (if applicable);
6. the data type of each attribute (see Support Data Types);
7. the constraints of each attribute.

7.1.1 Proof of Concepts

Please refer to Proof of Concepts: Create Database Tables.

7.1.2 Dependencies

Selectable Data Types (see Support Data Types).

7.1.3 Related GUI

7.2 Set Icons for Tables and Attributes

A user shall be able to set icons to represent each table and each attribute in a table.

7.2.1 Proof of Concepts

7.2.2 Dependencies

Create Database Tables (see Create Database Tables).

7.3 Set Special Attribute Properties

7.3.1 Description

A user shall be able to set states for each individual attribute of each table created above through a GUI.

7.3.2 Proof of Concepts

Please refer to Proof of Concepts: Set Special Attribute Properties.

7.3.3 Dependencies

Create Database Tables (see Create Database Tables).

7.4 Support Data Types

7.4.1 Description

For each attribute of each database table created, a user shall be able to use a GUI to specify the type of the attribute as one of:

- bitint
- int
- tinyint
- nchar
- nvarchar
- ntext
- bit
- datetime
- images
- money
- xml
- double
- float
- varchar
- blob

7.4.2 Proof of Concepts

Please refer to Proof of Concepts: Create Database Tables.

7.4.3 Dependencies

Create Database Tables (see Create Database Tables).

7.5 Edit Table Properties

7.5.1 Description

A user shall be able to edit table properties through a GUI, including:

1. the table name;
2. the number of attributes;
3. the name of each attribute;
4. the primary key of the table (if applicable)
5. the foreign keys of the table (if applicable);
6. the data type of each attribute (see Support Data Types);
7. the constraints of each attribute.

7.5.2 Proof of Concepts

7.5.3 Dependencies

Create Database Tables (see Create Database Tables).

7.6 Delete Tables

7.6.1 Description

A user shall be able to delete a selected table through a GUI.

7.6.2 Proof of Concepts

7.6.3 Dependencies

Create Database Tables (see Create Database Tables).

7.7 Input Records into Tables

7.7.1 Description

A user shall be able to input records into tables through a GUI. This will allow tables to already have preset data when an End-User starts using the created module. Note that the End-User will also be able to add records to these tables during the Using the Module stage.

7.7.2 Proof of Concepts

Please refer to Proof of Concepts: Input Records into Tables.

7.7.3 Dependencies

Create Database Tables (see Create Database Tables).

7.8 Modify Records in Tables

7.8.1 Description

A user shall be able to modify an existing record in a database table through a GUI.

7.8.2 Proof of Concepts

7.8.3 Dependencies

1. Create Database Tables (see Create Database Tables).
2. Input Records into Tables (see Input Records into Tables).

7.9 Duplicate Records

7.9.1 Description

A user shall be able to duplicate an existing record in a database table through a GUI without having to create a new record and manually fill in identical data.

7.9.2 Proof of Concepts

Please refer to Proof of Concepts: Duplicate Records.

7.9.3 Dependencies

1. Create Database Tables (see Create Database Tables).
2. Input Records into Tables (see Input Records into Tables).

7.10 Define Values in Relation to Other Values in Database

7.10.1 Description

When inputting data into a table, a user shall be able to define a values in relation to values in other tables through a GUI. The user shall be able to do this by:

1. concatenating strings; or
2. arithmetic data manipulation; or
3. arithmetic number manipulation.

7.10.2 Proof of Concepts

Please refer to Proof of Concepts: Define Values in Relation to Other Values.

7.10.3 Dependencies

1. Create Database Tables (see Create Database Tables).
2. Input Records into Tables (see Input Records into Tables).

7.11 Set Constraints (High Non-Core)

7.11.1 Description

A user shall be able to set the behaviour of an object to be dependent on certain values in the database using a GUI.

7.11.2 Proof of Concepts

Please refer to Proof of Concepts: Set Constraints.

7.11.3 Dependencies

1. Create Database Tables (see Create Database Tables).
2. Input Records into Tables (see Input Records into Tables).

7.12 Set Triggers (High Non-Core)

7.12.1 Description

A user shall be able to use a wizard system in order to define an event that if occurs, will make the system perform one of the following:

1. Send an email to an address either manually entered from stored as a value in any table in the database.
2. Create a log of attributes that are set to be logged.
3. Take a snapshot of the database.
4. Create a new record.
5. Duplicate an existing record.

7.12.2 Proof of Concepts

Please refer to Proof of Concepts: Set Triggers.

7.12.3 Dependencies

1. Create Database Tables (see Create Database Tables).
2. Input Records into Tables (see Input Records into Tables).

7.13 Create Tree Structure

7.13.1 Description

A user shall be able to create tree structures using tables. The user shall be able to specify the:

1. name of the tree;
2. value of each node in the tree;
3. type of each node in the tree;
4. parent of each node in the tree (if any);
5. child(ren) of each node in the tree (if any).

7.13.2 Proof of Concepts

Please refer to Proof of Concepts: Create Tree Structure.

7.13.3 Dependencies

1. Create Database Tables (see Create Database Tables).
2. Input Records into Tables (see Input Records into Tables).

7.14 Create Multi-value Attributes

7.14.1 Description

A user shall be able to define an attribute as having multiple values.

7.14.2 Proof of Concepts

None.

7.14.3 Dependencies

None.

8 Functional Requirements for Second Stage - Defining Views

This section contains functional requirements for the second stage of the system. During this stage, a Consultant or Advanced User would customise the visual appearance of the End-System by:

- defining views for objects created in the First Stage; and
- styling the End-System through CSS.

A sample GUI of how this stage of the system may look is provided below for reference only.

CGT

NEXT >

RISK TREE

RISK

+ CONTROL

- PARTY IMPACTED

→ ATTRIBUTE 1

→ ATTRIBUTE 2

CREATE DUNDAS CHART WIZARD

NEW CHART

NEW REPORT

CSS EDITOR

FINISHED

CGT Second Stage



8.1 View Database Trees

8.1.1 Description

The user shall be able to view the Trees of an End-System created in the First Stage in order to define the views for each object in the Tree. When a Tree is selected, all attributes of each object in the Tree shall be displayed.

8.1.2 Proof of Concepts

8.1.3 Related Requirements

8.2 Set ActiveReports Report as View

8.2.1 Description

The user shall be able to define an ActiveReport as a view for a particular object in the End-System.

8.2.2 Proof of Concepts

Please refer to Proof of Concepts: Set ActiveReports Report as View.

8.2.3 Related Requirements

8.3 Set CrystalReports Report as View (Low Non-Core)

8.3.1 Description

The user shall be able to define an Crystal Reports report as a view for a particular object in the End-System.

8.3.2 Proof of Concepts

8.3.3 Related Requirements

8.4 Set Dundas Chart as View

8.4.1 Description

The user shall be able to define a Dundas chart as a view for a particular object in the End-System.

8.4.2 Proof of Concepts

Please refer to Proof of Concepts: Set Dundas Chart as View.

8.4.3 Related Requirements

8.5 Set ChartFX Chart as View (Low Non-Core)

8.5.1 Description

The user shall be able to define a ChartFX chart as a view for a particular object in the End-System.

8.5.2 Proof of Concepts

8.5.3 Related Requirements

8.6 Set Default View

8.6.1 Description

When a record is selected, the user shall be select which view to be the default view for either:

1. all records of that type; or
2. the selected record only.

8.6.2 Proof of Concepts

8.6.3 Related Requirements

8.7 Plain Text Browser-Based CSS Editor

8.7.1 Description

The user shall be able to use a hierarchical plain-text web-based CSS editor to edit the appearance of the End-System.

8.8 Load CSS Templates

8.8.1 Description

The user shall be able to change the appearance of the End-System by loading an existing .css file into the plain-text web-based CSS editor.

8.9 Support Multi-value Attributes

8.9.1 Description

If a user has created tables with multi-value attributes using the Create Multi-value Attributes feature in the First stage, the End-System shall have an option whereby different values can be shown.

9 Functional Requirements for Third Stage - Using End-System

This section outlines the requirements for a created End-System. Although the End-System is in fact generated at the end of the Second Stage, and therefore this generation is technically and requirement of the Second Stage, the criteria are expressed separately to give a better perspective on the features of the generated End-System. A sample GUI of a possible End-System is provided below for reference only.

COMPANY LOGO GOES HERE

LOGOUT

RISK TREE

RISK

+ CONTROL

- PARTY IMPACTED

→ ATTRIBUTE 1

→ ATTRIBUTE 2

FILTERS

VIEW

DUNDAS CHART

CREATE NEW RISK

EDIT RISK

DELETE RISK

DUPLICATE RISK

VIEWS

DUNDAS CHART ▾

CGT Third Stage



9.1 View Records by Tree

9.1.1 Description

A created End-System shall allow a user to view all Trees created in the First stage. When a Tree is selected, all records in that Tree shall be displayed hierarchically according to their level in the Tree.

9.1.2 Proof of Concepts

9.1.3 Related Requirements

9.2 View Records by Spider Web View (Medium Non-Core)

9.2.1 Description

The system shall be able to view records in a spider web.

9.2.2 Proof of Concepts

Please refer to Proof of Concepts: View Data in a Spider Web View (Medium Non-Core).

9.3 View Records from Bird's Eye View (Medium Non-Core)

9.3.1 Description

The system shall be able to view records as a bird's eye view. The data shall be displayed in terms of rings. The size of each ring is dependent on the total number of records that fall within the band. Each ring can then be segmented according to the number of members that they have.

9.3.2 Proof of Concepts

Please refer to Proof of Concepts: View Data from Bird's Eye View (Medium Non-Core).

9.4 Filter Records

9.4.1 Description

There will be a filter function for viewing records that allows only certain records in the selected Tree to be seen.

9.4.2 Proof of Concepts

9.4.3 Related Requirements

9.5 Show Views

9.5.1 Description

When a record within the Tree is selected, the default view for that record will be displayed.

9.5.2 Proof of Concepts

9.5.3 Related Requirements

View Records by Tree

9.6 Context-Specific Buttons

9.6.1 Description

The End-System shall show buttons relevant to the current record selected. There will be buttons to perform each of the following functions:

1. Add new record of type currently selected
2. Edit selected record
3. Delete selected record
4. Duplicate selected record

However, some of these buttons may be unavailable if the modification or view of that record is not permitted because of a constraint.

9.6.2 Proof of Concepts

9.6.3 Related Requirements

Display Trees/Tables

9.7 Select View for Record Type

9.7.1 Description

Once a record has been selected and the default view has been displayed, there shall be a drop-down box in a panel within the web browser window that allows a user to change the view for that record. The available views shall be those created for that record type in Functional Requirements for Second Stage - Defining Views.

9.7.2 Proof of Concepts

Display Trees/Tables

9.7.3 Related Requirements

Display Trees/Tables

9.8 Select Default View

9.8.1 Description

Once a view has been selected, the user will be able to set that view as the default view so that either:

1. the next time a record of that type is displayed, the current view will be displayed as the default; or
2. the next time that specific record is displayed, the current view will be displayed as the default.

9.8.2 Proof of Concepts

9.8.3 Related Requirements

1. Display Trees/Tables
2. Select View for Record

10 Non-Functional Requirements

This subsection outlines the non-functional requirements and constraints of each of the three systems that form the CGT.

10.1 Browser Compatability

The system shall be able to run on IE 6.0+ and Firefox 1.5+.

10.2 Software System Attributes

There are three quality goals that the software system has to achieve:

1. Correctness;
2. Completeness;
3. Consistency.

11 User Documentation

User documentation will include a User Manual, Maintenance Document and Online Help.

11.1 Tool Tips

The system shall be able to show tool tips on all the relevant entities or attributes of the system. The tool tips shall be information on the entity or attributes itself. The tool tips for each of the entities are provided to help inexperienced users of the system.

11.2 User Manual

A user manual will be supplied with the software, incorporating the installation documentation. The user manual will also include a brief description of all the functional requirements and a simple tutorial aimed at the inexperienced users.

The user manual will come in the form of a PDF and also an HTML file.

11.3 Maintenance Documentation

All of the code will be provided to the client. The code will be documented so that the code can be maintained.

11.4 Online Help

There will be an online tutorial incorporated in the documentation.

12 Acceptance Criteria

The Corporate Governance Toolkit will be deemed acceptable when the following Proof of Concepts have been met (excluding non-core Proof of Concepts).

12.1 Proof of Concepts: Create Database Tables

For the following proof of concepts, a "*" after an attribute shall indicate that it is a multi-value attribute. Multi-value attributes may require extra resolving tables in order to be implemented. Unless indicated otherwise, all Proof of Concepts are core.

Please refer to Create Database Tables.

12.1.1 Create Table that Represents Risk

The system must be able to create a table that represents a risk, including the following attributes:

1. Name;
2. Description;
3. Categories;
4. Owner;

5. Impact;
6. Party Impacted (see Create Table that Represents Party);
7. Average Likelihood;
8. Severity;
9. Cause (see Create Table that Represents Cause of Risk);
10. Total Control Cost;
11. State (Treatment Required, etc).

12.1.2 Create Table that Represents Location

The system must be able to create a table that represents a location, including the following attributes:

1. Long title;
2. Short title.

12.1.3 Create Table that Represents Control

The system must be able to create a table that represents a control, including the following attributes:

1. Associated Action;
2. Way to reduce risk;
3. Cost/Effectiveness rating;
4. Last date reviewed;
5. State (New/Existing).

12.1.4 Create Table that Represents Party

The system must be able to create a table that represents a party impacted by a risk, including the following attributes:

1. Party Name;
2. Impact;
3. Control (see Create Table that Represents Control).

12.1.5 Create Table that Represents Cause of Risk

The system must be able to create a table that represents a cause of a risk, including the following attributes:

1. Description of Cause;
2. Likelihood;
3. Control (see Create Table that Represents Control).

12.1.6 Create Table that Represents Treatment Plan

The system must be able to create a table that represents a treatment plan, including the following attributes:

1. Treatment Plan Task (see Create Table that Represents Treatment Plan Task);
2. Percentage complete;
3. Sign-off/Completed boolean.

12.1.7 Create Table that Represents Treatment Plan Task

The system must be able to create a table that represents a treatment plan task, including the following attributes:

1. Name;
2. Control (see Create Table that Represents Control);
3. Percentage complete.

12.1.8 Create Table that Represents Review

The system must be able to create a table that represents a review, including the following attributes:

1. Risk to be reviewed (see Create Table that Represents Risk).

12.1.9 Create Table that Represents Employment Position

The system must be able to create a table that represents an employment position, including the following attributes:

1. Title;
2. Description.

12.1.10 Create Table that Represents Contact Details

The system must be able to create a table that represents contact details, including the following attributes:

1. Mobile number;
2. Email;
3. Address;
4. Pager number.

12.1.11 Create Table that Represents Users

The system must be able to create a table that represents a user including the following attributes:

1. User ID.

12.1.12 Create Table that Represents Employee

The system must be able to create a table that represents an employee, including the following attributes:

1. User (see Create Table that Represents Users);
2. Contact details (see Create Table that Represents Contact Details);
3. Employment title (see Create Table that Represents Employment Position);
4. Work Group (see Create Table that Represents Work Group).

12.1.13 Create Table that Represents Work Group

The system must be able to create a table that represents a work group, including the following attributes:

1. Title;
2. Description.

12.1.14 Create Table that Represents Category

The system must be able to create a table that represents a category, including the following attributes:

1. Name;
2. Description.

12.1.15 Create Table that Represents Measure Scale

The system must be able to create a table that represents a measure scale, including the following attributes:

1. Upper bound for scale;
2. Lower bound for scale;
3. Upper bounds for band (if any);
4. Lower bounds for band (if any);
5. Band name (if applicable);
6. Band colour (if applicable);
7. Band description (if applicable).

12.1.16 Create Table that Represents an Indicator

The system must be able to create a table that represents an indicator, including the following attributes:

1. Name;
2. Description;
3. Type;
4. Value;
5. Frequency to be recorded;
6. Measure Scale (see Create Table that Represents Measure Scale);
7. Above Tolerance Boolean.

12.1.17 Create Table that Represents Email

The system must be able to create a table that represents a Email message, including the following attributes:

1. To field;
2. From field;
3. Subject field;
4. Body field;
5. Attachment (see Create Table that Represents Email Attachments).

12.1.18 Create Table that Represents Email Attachments

The system must be able to create a table that represents an Attachment, including the following attributes:

1. File or URL.

12.1.19 Create Table that Represents a Survey Template

The system must be able to create a table that represents a Survey Template, including the following attributes:

1. Owner (see Create Table that Represents Users);
2. Associated Work Group (see Create Table that Represents Work Group);
3. Frequency;
4. Duration;
5. Start Date;
6. End Date;
7. Governing Document;
8. Question (see Create Table that Represents a Question);
9. State (Deleted, inactive, active);
10. Associated Risk (see Create Table that Represents Risk);
11. Associated User Guide for Template.

12.1.20 Create Table that Represents a Survey

The system must be able to create a table that represents a survey, including the following attributes:

1. Owner (see Create Table that Represents Users);
2. Associated Work Group (see Create Table that Represents Work Group);
3. Frequency;
4. Duration;
5. Start Date;
6. End Date;
7. Question (see Create Table that Represents a Question);
8. Associated Risk (see Create Table that Represents Risk).

12.1.21 Create Table that Represents a Question

The system must be able to create a table that represents a question, including the following attributes:

1. QuestionID;
2. Question;
3. Required Answer format.

12.1.22 Create Table that Represents a Result

The system must be able to create a table that represents a result, including the following attributes:

1. Question ID;
2. Answer.

12.1.23 Create Table that Represents a Question Pool

The system must be able to create a table that represents a question pool, including the following attributes:

1. Question pool ID;
2. Question (see Create Table that Represents a Question).

12.1.24 Create Table that Represents an Audit

The system must be able to create a table that represents an audit, including the following attributes:

1. Question (see Create Table that Represents a Question);
2. Performance rating.

12.2 Proof of Concepts: Set Special Attribute Properties

Please refer to Set Special Attribute Properties.

12.2.1 Log Attributes (Non-Core)

For each attribute in each table above under Proof of Concepts: Create Database Tables, a boolean logging option shall be able to be set to specify whether changes to that property's value should be recorded in a log file that can be used as input for charts and matrices.

12.2.2 Set Visibility of Attributes

For each attribute in each table created under Proof of Concepts: Create Database Tables, a boolean visibility option shall be able to be set to specify whether that property shall be visible when records are displayed at the front-end.

12.2.3 Report on Attributes

For each attribute in each table created under Proof of Concepts: Create Database Tables, a boolean reporting option shall be able to be set to specify whether that property is available as an option in a report or chart designer.

12.3 Proof of Concepts: Input Records into Tables

12.3.1 Preset Data in Control Table

The system shall allow to have preset data in the Control table created under Create Table that Represents Control that corresponds to the following generic business risk controls:

1. coso;
2. sox.

12.4 Proof of Concepts: Duplicate Records

Please refer to Duplicate Records.

12.4.1 Duplicate Risk Record

The system shall allow a risk record to be duplicated. If the risk record has any children nodes as part of a tree structure, the user will be given option as to whether to duplicate all those children nodes as well, or whether to just duplicate the risk record.

12.5 Proof of Concepts: Define Values in Relation to Other Values

Please refer to Define Values in Relation to Other Values in Database.

12.5.1 Calculation of Risk Attribute

The system shall allow a Risk's Impact attribute (see Create Table that Represents Risk) to be calculated as the sum of each impact for each associated Party Impacted by the Risk (see Create Table that Represents Party).

12.5.2 Calculation of Risk Average Likelihood

The system shall allow a Risk's Average Likelihood (see Create Table that Represents Risk) attribute to be calculated as the highest likelihood for all associated Cause's of the risk (see Create Table that Represents Cause of Risk).

12.5.3 Calculation of Risk Severity Attribute

The system shall allow a Risk's Severity attribute (see Create Table that Represents Risk) to be calculated as the product of its average impact and average likelihood.

12.5.4 Calculation of Treatment Plan's Attribute

The system shall allow a Treatment Plan's Percentage Complete attribute to be defined as the average of percentage complete attributes of each of its treatment plan tasks.

12.5.5 Calculation of Risk's Total Control Cost

The system shall allow a risk's Total Control cost to be the sum of all costs of each of the controlf for each of its particles impacted and causes.

12.5.6 Calculation of Tolerance Attribute

The system shall allow an indicator's above tolerance attribute to be set to true of the value entered is within the tolerance thresholds for that indicator.

12.5.7 Calculation of Risk Attribute in Relation to Child

The system shall be able to define a attribute in the Risk table in relation to attributes of its child nodes as defined in Proof of Concepts: Create Tree Structure.

12.6 Proof of Concepts: Create Tree Structure

Please refer to Create Tree Structure.

12.6.1 Hierarchical Tree Structure

A tree structure must be created that represents a hierarchical organisation structure, i.e.:

(Work Group (Work Group*))

12.6.2 Employment Title Hierarchies

A tree structure must be created that represents employment title hierarchies, i.e.:

(Employment Title (Employment Title*))

12.6.3 Risk Hierarchies

A tree structure must be created that represents a risk hierarchy, i.e.:

(Risk (Cause* (Control*), PartyImpacted* (Control*)))

12.6.4 Compliance Hierarchies

A tree structure must be created that represents a compliance tree, i.e.:

(Template (Question*))

12.6.5 Categorical Hierarchies

A tree structure must be created that represents a hierarchical category structure, i.e.:

(Category (Category*))

12.7 Proof of Concepts: Set Constraints (High Non-Core)

Please refer to Set Constraints.

12.7.1 Constraints on Treatment Plans

The system shall be able to prevent treatment plans from being signed off unless all of its treatment plan tasks are completed.

12.7.2 Constraints on Viewing Risks

The system shall be able to restrict the viewing of risks, to be dependent on any of the following:

1. Certain level or higher in Employment Title hierarchy (See Section 10.1.9);
2. Particular Category;
3. Particular Work Group.

12.7.3 Constraints on Controls

The system shall be able to restrict the changing of the state of a control from "new" to "existing" once the associated risk's state becomes "treatment required".

12.7.4 Read-Only Option

The system should allow a record to be set as read-only.

12.7.5 Related Requirements

12.8 Proof of Concepts: Set Triggers (High Non-Core)

Please refer to Set Triggers.

12.8.1 Record Trigger (high non-core)

The system shall be able to send an email or report when a record is:

1. Created;
2. Deleted.

12.8.2 Cost/Effectiveness Trigger (high non-core)

The system shall be able to send an email or report when a cost/effectiveness rating exceeds a given value.

12.8.3 Overdue Trigger (high non-core)

The system shall be able to send an email or report when an action is overdue.

12.8.4 Changed Category Trigger (high non-core)

The system shall be able to send an email or report when a category is changed.

12.8.5 Out-of-Tolerance Trigger (high non-core)

The system shall be able to send an email or report when an indicator becomes out of tolerance.

12.8.6 Survey Trigger (high non-core)

The system shall be able to generate a new survey from a template when the period for the survey recurrence has come, and send an email or report.

12.8.7 Snapshot Trigger (high non-core)

The system shall be able to take a snapshot of the entire database in response to a particular event.

12.8.8 Duplicate Trigger (high non-core)

The system shall be able to duplicate a record in the database in response to a particular event.

12.8.9 Create Log (high non-core)

The system shall be able to create a log whenever a property set to be logged is changed (see Set Triggers).

12.9 Proof of Concepts: Set ActiveReports Report as View

Please refer to Set ActiveReports Report as View.

12.9.1 Create ActiveReport Report from Risk Data

The system shall allow export of Risk attributes to be used as input to ActiveReports.

12.10 Proof of Concepts: Set Dundas Chart as View

Please refer to Set Dundas Chart as View.

12.10.1 View Risk Data in a Matrix

The system shall be able to create a Dundas Chart to view data in a matrix form, by allow a user to specify:

1. the data for the x, y and z axes;
2. the cell structure and number of cells;
3. the color, borders, shades, text content and icons of the cell;
4. the scale of the axes;
5. a legend for the matrix;
6. comments for the matrix;
7. tooltips for cells.

12.10.2 Active cells in Matrix (non-core)

The cells in View Risk Data in a Matrix shall be active and may contain relative data.

12.10.3 Drill Down (non-core)

The cells in View Risk Data in a Matrix shall be able to have a "drill down" capacity where an open cell is able to show a line of information in a grid or to display the nodes within a related tree view.

12.10.4 Combined or Overlaid Matrices (non-core)

A matrix created in View Risk Data in a Matrix will be able to be combined with other matrices. For example, matrix with current impact and current probability as the y and x axes respectively shall be able to be combined with another matrix with different x and y axes like target impact and target probability.

12.10.5 Lines Between Items (non-core)

A matrix created in View Risk Data in a Matrix shall be able to have tie lines in-between like items.

12.10.6 Legends for Matrices

A matrix created in View Risk Data in a Matrix will be able to have a legend that possesses a value of the following types:

1. Currency;
2. Percentages;
3. Number;
4. Fraction;
5. Text Description.

12.10.7 Interchangeable Legends

A legend created in Legends for Matrices The system shall be able to interchange legends and axes.

12.10.8 Searchable Matrix (non-core)

A matrix created in View Risk Data in a Matrix will be able to have a search filter applied to it.

13 Client Sign-off

This SRS will act as a formal agreement between the client and Team F as to the specifications of the system to be developed. Any changes to the SRS after the sign off will need to follow the change control procedures outlined in the SPMP.

I (the client) hereby assent that the requirements specified in this document agree with the my requirements and expectations from the product.

Signatures

.....(Client Signature)

.....(Client Name)

.....(Team F Representative Signature)

.....(Team F Representative Name)

.....(Witness Signature)

.....(Witness Name)

.....(Date)

14 Notes

1. ↑ University of Melbourne, 433-440 Advanced Software Engineering Project, 2006

Retrieved from "http://www.eirikr.net/projects/cgt/wiki/index.php?title=Software_Requirements_Specification"

This page has been accessed 500 times.

This page was last modified 2006-04-27 17:22:44.

Content is available under Attribution-NonCommercial-ShareAlike 2.5.