

AN IMPLEMENTATION OF PENTAHO IN REPORTING
MANAGEMENT MODULE

ARZTRIE DELAILAH BINTI RUMI

UNIVERSITI TEKNOLOGI MALAYSIA

AN IMPLEMENTATION OF PENTAHO
IN REPORTING MANAGEMENT MODULE

ARZTRIE DELAILAH BINTI RUMI

A project report submitted in partial fulfillment of the requirement for award of
degree of Master of Computer Science in Real Time Software Engineering

Centre for Advanced Software Engineering
Faculty of Computer Science and Information System
Universiti Teknologi Malaysia

OCTOBER 2008

DEDICATION

*To my beloved family my mother, father, sister and brothers
To my beloved love ones.*

ACKNOWLEDGEMENT

Alhamdulillah, all praise is to Allah, the most merciful. My gratitude to ALLAH SWT for allowing me to successfully undertake and complete this Industrial Attachment II thesis report.

First and foremost, I would like to express my deepest appreciation to my academic mentor, Mr Othman Yusop and to Mr Chong Sau Wei for the unconditional support and encouragement, tolerance, patience and guidance during this five months attachment period.

A millions thanks to Scan Associates Berhad, the company which has given me the opportunity to be involved in this project. A thousand appreciations to my colleagues who had shared my anxieties, my ups and down the bitter sweet that I had to go trough during this attachment

Finally to my great family who never stop from encouraging me and lift me up during handling all the challenge. Thank a lot to my mother, father, and brother and especially to my super supportive sister. Not to forget my love ones who never failed to support me and guide me. Without them it was impossible for me to reach the stage where I am now.

ABSTRACT

Reporting Management Module is one of SCAN Associates research and development team projects in order to find the most suitable and effective ways to develop the enhancement of the reporting module previous system. The current system's constraints including limited option in data visualization and also data calculation has effect the productivity of Security Analyst. The research of Pentaho implementation in this Reporting Module should be able to overcome the constraints mentioned. The system is a web based system and develops using JAVA with MySQL as its database. RUP has been chosen as the software development model and UML as the software technique. The analysis and design of this system is using the Scan Associates standard and guideline. The deliverables for this project includes the Software Requirement Specification (SRS) and Software Architecture Design (SAD). The Reporting Management Module is expected to provide a fast and efficient Reporting Management Module in order to improve the productivity of Security Analyst. It is highly hope that this module would be one of the effective ways to improve the company's quality.

ABSTRAK

Perlaksanaan *Pentaho* dalam *Reporting Management* modul merupakan salah satu daripada penyelidikan yang dilakukan oleh bahagian penyelidikan di Scan Associates Berhad untuk menjalankan penyelidikan ke atas perisian-perisian yang sesuai dan terbaik. Perisian-perisian ini digunakan untuk membangunkan sebuah perisian yang lebih baik daripada perisian yang sedia ada. Penyelidikan terhadap *Pentaho* ini sepatutnya mampu membantu *Security Consultant* untuk melakukan tugas dengan lebih cepat and efisien. Perisian ini dibangunkan dengan menggunakan JAVA bersama MySQL sebagai pangkalan data. Fasa analisis dan reka bentuk perisian ini menggunakan Scan Associates panduan dan standard. *Reporting Management* modul ini diharapkan dapat membantu memberikan satu perisian yang berkesan dan cepat bagi meningkatkan daya produktiviti *Security Consultant* dan dalam masa yang sama dapat meningkatkan kualiti syarikat.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	TITLE	i
	DECLARATION	ii
	DEDICATION	iii
	ACKNOWLEDGEMENT	iv
	ABSTRACT	v
	ABSTRAK	vi
	TABLE OF CONTENT	vii
	LIST OF TABLES	xi
	LIST OF FIGURES	xii
	LIST OF ABBREVIATIONS	xv
	LIST OF GLOSSARY	xvi
	LIST OF APPENDICES	xvii
1	INTRODUCTION	1
	1.1 About Scan Associates Berhad	1
	1.2 Background of project involve and important to the company	3
	1.3 Project Objectives	4
	1.4 Project Scope	4
	1.5 Project Deliverables	5
	1.6 Project Plan	5
2	LITERATURE STUDY	6
	2.1 Technology	6
	2.1.1 Tenable Network Security	7
	2.1.2 Business Intelligence	9

	2.1.3	Open Source	10
2.2		Database	11
	2.2.1	Oracle	11
	2.2.2	MySQL	12
	2.2.3	PostgreSQL	12
2.3		UML Tools	13
	2.3.1	Rational Rose	13
	2.3.2	Enterprise Architect	14
2.4		Software Development Model	15
	2.4.1	RUP	15
	2.4.2	Waterfall	17
	2.4.3	Spiral	18
	2.4.4	Prototype	19
	2.4.5	Comparison between RUP, Waterfall and Prototype Model	19
2.5		Application Server	22
	2.5.1	Apache Tomcat	22
	2.5.2	JBoss	23
2.6		Operating System	24
	2.6.1	FreeBSD	24
	2.6.2	Linux	25
	2.6.3	Windows 2000	26
	2.6.4	Comparison between FreeBSD, LINUX and Windows 2000	27
2.7		Reporting	28
	2.7.1	Pentaho Reporting	28
	2.7.2	Crystal Report	29
	2.7.3	Jasper Report	29
3		METHODOLOGY	30
	3.1	Software Development Process	30
	3.2	Software Standard and guideline	34
	3.3	Software Development Tools (Pentaho)	34

	3.3.1	The Pentaho BI Project	35
	3.3.2	Pentaho Projects	36
	3.3.3	The Pentaho BI Project exists to benefit and serve	37
3.4		About Pentaho Reporting	37
	3.4.1	Pentaho Technology Fundamentals	38
	3.4.2	Pentaho BI Platform	38
	3.4.3	Pentaho Reporting	40
	3.4.4	Pentaho Analysis	43
	3.4.5	Pentaho Dashboards	44
	3.4.6	Pentaho Data Integration	46
3.5		Extension to the Project	48
3.6		Software Tools	48
	3.6.1	Microsoft Word 2007	48
	3.6.2	Microsoft Visio 2003	49
	3.6.3	Rational Rose 2000	49
	3.6.4	Pentaho Reporting	50
4		PROJECT DISCUSSION	51
	4.1	Introduction	51
	4.2	Project Implementation Methodology	52
	4.2.1	Project preparation stage	52
	4.2.2	User Requirements	53
	4.2.3	Reporting Management Use Case	54
	4.2.3.1	Upload Report	55
	4.2.3.2	Get Report	56
	4.2.3.3	Use Case of Generate Report	58
	4.2.4	Class Diagram	60
	4.2.5	Sequence Diagram and Collaboration Diagram	61
	4.2.6	System Architecture	62
	4.2.7	Component Diagram	62
	4.2.8	Deployment Diagram	63
5		CONCLUSION	64

5.1	Project Constraints	64
5.1.1	Lack of Knowledge in Pentaho and JAVA	65
5.1.2	Lack of knowledge of information security	65
5.2	Lesson Learnt	66
5.2.1	Exposure to Open Source Environment	66
5.2.2	Other Lesson Learnt	67
5.3	Recommendation	67
5.3.1	Improve the Process flow activity	68
5.3.2	Smaller and Portable Device	68
5.3.3	Improvement on Network Environment	69
5.4	Summary	69
	REFERENCES	70
	APPENDIX A-E	71-90

LIST OF TABLES

TABLE NO.	NAME	PAGE
2.1	Comparison between RUP, Waterfall and Prototype Model	19
2.2	Product Feature Difference	23
2.3	Comparison between FreeBSD, LINUX and Windows 2000	27
4.1	Pentaho BI Platform Advantages	40
4.2	Pentaho Reporting Advantages	41
4.3	Pentaho Analysis Advantages	43
4.4	Pentaho Dashboards Advantages	44
4.5	Pentaho Data Integration Advantages	46

TABLE OF FIGURES

FIGURE NO.	NAME	PAGE
1.1	Company Logo	2
1.2	Modules of NX Project	3
2.1	Waterfall Model	17
2.2	Spiral Model	18
2.3	Main Component of GNU/LINUX Operating System	25
2.4	Windows 2000 Operating System Architecture	26
3.1	RUP overview	31
3.2	Iterative RUP Lifecycle model	32
3.3	Pentaho Open BI Suite	36
3.4	Pentaho Business Intelligence Platform	39
3.5	Pentaho Reporting	42
3.6	Pentaho Dashboard	45
3.7	Pentaho Data Integretion	47
4.1	Use Case Diagram for RM Module	54
4.2	Upload Report Use Case	55
4.3	Interface Upload Report	55
4.4	Get Report Data Use Case	56
4.5	Interface Report of the month	57
4.6	Generate Report Use Case	58
4.7	Interface Top 5 Report	59
4.8	Interface Generate Vulnerability Report	59
4.9	Relationship Among CSCs in RM Module	60
4.10	Class Diagram for RM Module	61
4.11	Component Diagram	62
4.12	Deployment Diagram	63
B.1	Sequence Diagram Upload Report: Basic Flow	72

B.2	Sequence Diagram Upload Report: Alternative Flow [A1: Upload]	72
B.3	Sequence Diagram Upload Report: Alternative Flow [A1: Email]	73
B.8	Sequence Diagram Get Report: Basic Flow	73
B.5	Sequence Diagram Get Report: Alternative Flow [A1: Daily]	74
B.6	Sequence Diagram Get Report: Alternative Flow [A1: Weekly]	74
B.7	Sequence Diagram Get Report: Alternative Flow [A1: Monthly]	75
B.8	Sequence Diagram Get Report: Alternative Flow [A1: Yearly]	75
B.9	Sequence Diagram Get Report: Alternative Flow [A1: Date Range]	76
B.10	Sequence Diagram Generate Report: Basic Flow	76
B.11	Sequence Diagram Get Report: Alternative Flow [A1: Daily]	77
B.12	Sequence Diagram Get Report: Alternative Flow [A2: Weekly]	77
B.13	Sequence Diagram Get Report: Alternative Flow [A3: Monthly]	78
B.14	Sequence Diagram Get Report: Alternative Flow [A4: Yearly]	78
B.15	Sequence Diagram Get Report: Alternative Flow [A5: Date Range]	79
B.16	Sequence Diagram Get Report: Alternative Flow [A1: Generate Vulnerability Report]	79
C.1	Basic Flow: Upload Report	80
C.2	Alternative Flow Upload Report: A1 Upload Report	81
C.3	Alternative Flow Upload Report: A1 Email Report	81
C.4	Basic Flow: Get Report	82
C.5	Alternative Flow Get Report: A1 Daily Report	82
C.6	Alternative Flow Get Report: A2 Weekly Report	83
C.7	Alternative Flow Get Report: A3 Monthly Report	83
C.8	Alternative Flow Get Report: A4 Yearly Report	84
C.9	Alternative Flow Get Report: A5 Date Range Report	84
C.10	Basic Flow: Generate Report	85
C.11	Alternative Flow Generate Report: A1 Daily	85
C.12	Alternative Flow Generate Report: A2 Weekly	86
C.13	Alternative Flow Generate Report: A3 Monthly	86

C.14	Alternative Flow Generate Report: A5 Date Rage	87
C.15	Alternative Flow Generate Report: A5 Yearly	87
C.16	Alternative Flow Generate Report: A6 Generate Vulnerability Report	88

LIST OF ABBREVIATIONS

Name		Description
RM	-	Reporting Management
CASE	-	Center for Advanced Software Engineering
SRS	-	Software Requirement Specification
SAD	-	Software Architecture Design
OOA	-	Object Oriented Analysis
UML	-	Unified Modelling Language
DSS	-	Decision Support System
SRS	-	Software Requirement Specification
SAD	-	Software Architecture Design
OEM	-	Oracle Enterprise Manager
JVM	-	Java compiler and Virtual Machine
EA	-	Enterprise Architect
ASF	-	Apache Software Foundation
JSP	-	JavaServer Pages

LIST OF GLOSSARY

Name	Description
Actor	Used to represent something that uses the system.
Sequence Diagram	Used to show interaction between actors and objects and other objects.
Class Diagram	Used to represent the different underlying pieces (Classes), their relationship to each other and to which subsystem they belong to.
Unified Modeling Language	A language for visualizing, specifying, constructing and documenting the artefacts of a software intensive system
Object-oriented Analysis (OOA)	The process that used to attempt to understand a system by using objects and classes for the world relating to the system

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	Project Schedule	81
B	Sequence Diagram	82
C	Collaboration Diagram	93
D	Software Requirement Specification	102
E	System Architecture Design	103

CHAPTER 1

INTRODUCTION

This chapter introduces the research of analysis of the application of Pentaho Business Intelligence in RM Module for NX project. The author will also elaborate the company's background, the project objectives, scope and planning

1.1 About Scan Associates Berhad

SCAN Associates Berhad, incorporated in September 2000, is Malaysia's premier and trusted ICT security solutions provider. It is an MSC-status and an affiliated company of Commerce Asset Ventures Sdn. Bhd., a wholly-owned venture capital arm of Bumiputra-Commerce Holdings Berhad.

With a solid track record of providing ICT security solutions and building its intellectual property, SCAN delivers internationally recognized and enterprise-wide solutions and services.

SCAN is backed by relevant Malaysian Government Agencies, such as the Multimedia Development Corporation and Malaysia Debt Ventures, to spearhead ICT security initiatives in Malaysia as well as overseas. In addition, SCAN currently houses the highest concentration of internationally certified ICT security professionals in Malaysia. SCAN has also made inroads into the international market especially in Indonesia and the Middle East by leveraging on its knowledge capital, to provide leading-edge ICT security solutions.

SCAN upholds Trust as its fundamental value proposition in providing customer-centric solutions and services to the growing regional and global markets, where it aims to be one of the leading global ICT Security Solutions providers. The company logo is depicted in Figure 1.1.



Figure 1.1 : Company Logo

1.2 Background of project involve and important to the company

The primary goal of NX project is to enhance the features of previous version network security solution and it intends to become a commercial product to compete with the worldwide Managed Security Service (MSS) providers. NX provides a suite of solutions in order to prevent, detect and respond to the vulnerabilities, threats and events in real time as well as to support future MSS Security Operation Center (SOC) services. The security analysts, pen testers and the managers are able to monitor, assess and generate reports more rapidly through a dynamic and interactive Rich Internet Application (RIA) user interface.

Figure 1.2 illustrates the modules of NX project.

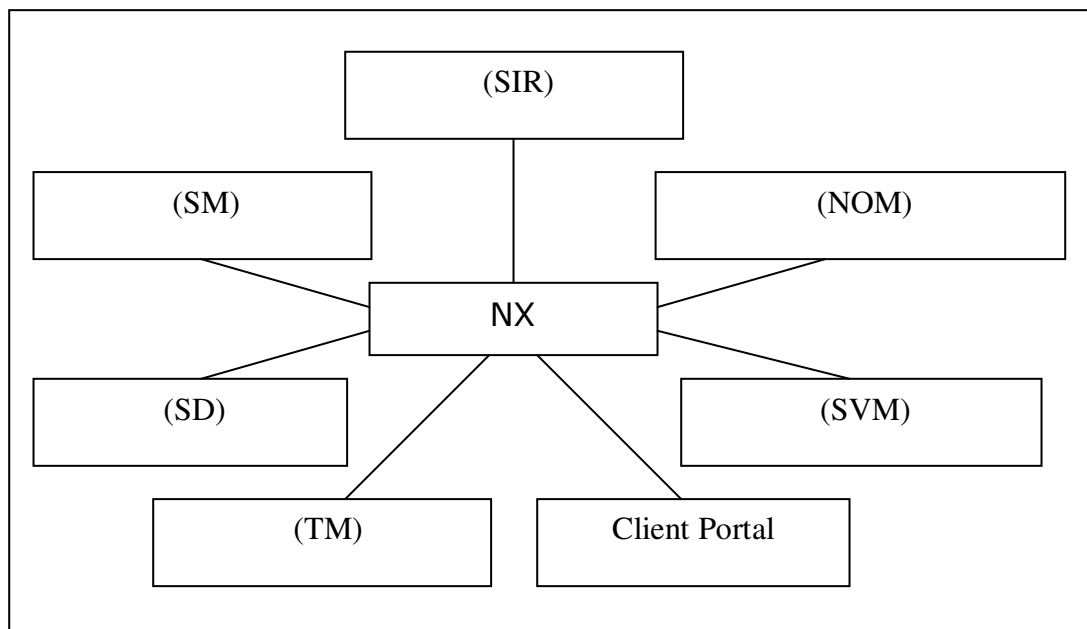


Figure 1.2 : Modules of NX Project

1.3 Project Objectives

The main objective of this project is to do Research and Development (R&D) on the suitability of Pentaho for this module. Below are the objectives of the project:

- i) To enable the system to provide the capability to display data in a chart or graph and make data interpretation much easier for Security Analyst. And
- ii) To improve the reporting tools and overcome the previous system's limitation for reporting module.

1.4 Project Scope

The scopes of the project are:

- i) Explore and study the new technology, Pentaho and the latest Security network monitoring software.
- ii) The core functionality is to display an interactive user interface for the reporting module and provide the flexibility for adjustment for various clients.
- iii) All the data are obtain from MySQL database.

1.5 Project Deliverables

The project deliverables are:

- i) Software Requirement Specification (SRS)
- ii) Software Architecture Design (SAD)

1.6 Project Plan

All work had been done accordingly the project schedule. Please refer to Appendix A for the project schedule. RM module was an internal project which did not involve client during its development stages. The development was based on the previous system which was done by the company development team. The author responsibilities are to do a research on Pentaho and define the suitability of Pentaho for the Reporting Module and also to gather requirements for the project.

CHAPTER 2

LITERATURE STUDY

This chapter describes the literature study that has been made to find the best solution to develop the RM Module. In order to make sure the capability is same as the previous system

2.1 Technology

The technology that has been used in this project will be described in detail in this chapter.

2.1.1 Tenable Network Security

The current system that being use by the company is Tenable Network Security. The author has been given the opportunity to explore and observe Tenable Network Security performance, capability and to identify the limitation for Security Analyst.

Tenable Network Security is the world leader in Unified Security Monitoring. Tenable provide agent less solutions for continuous monitoring of vulnerabilities, configurations, data leakage, and log analysis and compromise detection.

Tenable is the sole sponsor of the Nessus vulnerability scanner which provides to the internet community a free, powerful, up-to-date, and easy to use remote security scanner. Nessus is currently rated among the top products of its type throughout the security industry and is endorsed by professional information security organizations such as the SANS Institute. It is estimated that the Nessus scanner is used by more than 75,000 organizations world-wide.

Tenable Network Security, Inc. (Tenable) is a US-based Delaware C Corporation. Tenable's corporate offices are located in Columbia, Maryland with additional offices in New England, California, Virginia, Pennsylvania, and Georgia. Tenable was founded in September of 2002 and is privately owned. Tenable has hundreds of Global 2000 customers in the US, Canada, Asia Pacific, and Europe.

The technology founders include the creator of the award-winning Dragon intrusion detection system, Ron Gula, and the creator of the award-winning "open-source" Nessus Security Scanner, Renaud Deraison.

Tenable Network Security provides a suite of solutions which unify real-time vulnerability, event and compliance monitoring into a single, role-based, interface for administrators, auditors and risk managers to evaluate, communicate and report needed information for effective decision making and systems management.

Solutions:

- i) Compliance Reporting: Instantly report to management on the level of your network's policy compliance
- ii) Security Event Management: Correlate and analyze log events from hundreds of devices in one unified view.
- iii) Security Auditing: Audit the configuration of the hosts on your network and make sure all their patches are up-to-date

Products:

- i) Nessus: The network vulnerability scanner
- ii) Security Center: Unify your Security Monitoring
- iii) Log Correlation Engine: Aggregate and correlate your logs
- iv) Direct Feed: Check the policy compliance of your scanned hosts
- v) Passive Vulnerability Scanner: Listen to your vulnerabilities

2.1.2 Business Intelligence

Business intelligence (BI) refers to technologies, applications and practices for the collection, integration, analysis, and presentation of business information and sometimes to the information itself. The purpose of business intelligence term that dates at least to 1958- is to support better business decision making. Thus, BI is also described as a decision support system (DSS).

BI is sometimes used interchangeably with briefing books, report and query tools and executive information systems. In general, business intelligence systems are data-driven DSS.

BI systems provide historical, current, and predictive views of business operations, most often using data that has been gathered into a data warehouse or a data mart and occasionally working from operational data. Software elements support the use of this information by assisting in the extraction, analysis, and reporting of information. Applications tackle sales, production, financial, and many other sources of business data for purposes that include, notably, business performance management. Information may be gathered on comparable.

Below are the examples of Business intelligence:

- i) Operational reporting - Typically static, based on a single operational system.
Examples: current inventory report, customer invoice
- ii) Standardized reporting - Typically static, IT-driven based on a single system, on a recurring basis. Examples: Monthly sales report, Quarterly customer satisfaction report Ad hoc reporting frequently dynamic, end user driven, on an ad hoc basis. Example: Business user wants to see what products a certain customer has purchased over the last 6 months
- iii) Embedded reporting - Typically blends report content directly into packaged or custom operational applications

- iv) OLAP reporting (Online Analytical Processing) - Typically interactive, exploratory, dimensional (sales by product by time period by geography by channel)
- v) Dashboards - Typically a single, simplified view of Key Performance Indicators.
- vi) Data mining - Involves automated analysis of large volumes of data, usually customer or consumer data. Tries to uncover hidden patterns and correlations in the data
- vii) Data warehousing - Integrating information from multiple different systems into a centralized data store. Example: integrating customer order, customer service, and customer marketing data into a centralized database.

2.1.3 Open Source

Open source can be defined as computer software for which the human-readable source code is made available under a copyright license. Open source is considered as one of various possible design approaches, while others consider it a critical strategic element of their operations. Before open source became widely adopted, developers and producers used a variety of phrases to describe the concept; the term open source gained popularity with the rise of the Internet, which provided access to diverse production models, communication paths, and interactive communities.

The open source model of operation and decision making allows concurrent input of different agendas, approaches and priorities, and differs from the more closed, centralized models of development. The principles and practices are commonly applied to the development of source code for software that is made available for public collaboration, and it is usually released as open-source software.

2.2 Database

A few databases have been identified as the suitable for this project. The database that has been considered will be explained in detail in this chapter.

2.2.1 Oracle

Oracle Database is available in Personal, Standard, and Enterprise Editions. The Oracle Universal Installer installs the Oracle Database software, and calls configuration assistants to setup the networking components and create databases. Oracle Enterprise Manager (OEM) is a graphical application included with all editions that administrators use to manage the Oracle instances.

All editions include the PL/SQL engine to develop stored procedures, triggers, and functions, a Java compiler and Virtual Machine (JVM) to develop Java stored procedures and triggers, XML support an Apache Web server and object-relational capabilities.

2.2.2 MySQL

Developers can embed MySQL as a library within an application, or use MySQL as a standalone database engine. MySQL is available as binaries, or because it is Open Source, developers can download, modify, and compile the source code on the target server.

MySQL supports transactional and non-transactional tables, and one-way replication. The MySQL developers built the database for high performance, especially with the use of non-transactional tables. In addition, MySQL takes advantage of multiple processors, as it is fully multi-threaded using kernel threads.

2.2.3 PostgreSQL

PostgreSQL is an object-relational database management system. It is released under a BSD-style license and is thus free software. As with many other open-source programs, PostgreSQL is not controlled by any single company, but relies on a global community of developers and companies to develop it.

2.3 UML Tools

There are two UML tools that have been considered in this project including Rational Rose and also Enterprise Architect.

2.3.1 Rational Rose

Rational Rose Enterprise provides a common modeling language for enabling faster creation of quality software. Includes Unified Modeling Language support and is one of the most comprehensive products in the Rational Rose family. Rational Rose offers code quality analysis abilities and code generation, with configurable model-to-code synchronization capabilities, as well as more granular management and use of models with the separately controllable model components feature. Integrates with other IBM Rational lifecycle development tools and also provides UML modeling for database designs, with the ability to represent the integration of data and application requirements through logical and physical designs. Creates XML document type definitions for use in your application.

2.3.2 Enterprise Architect

Enterprise Architect (EA) is a Unified Modeling Language modeling tool produced by Sparx Systems. Enterprise Architect combines the power of the latest UML specification with a high performance, intuitive interface, to bring advanced modeling to the desktop, and to the complete development and implementation team. With a great feature set and unsurpassed value for money, EA can outfit the whole team, including analysts, testers, project managers, quality control staff, deployment team and more, for a fraction of the cost of some competing products.

EA's feature set supports the following capabilities:

- i) UML 2.1 support
- ii) XMI 2.1 support
- iii) SysML 1.0 support (via a plug-in)
- iv) BPMN 1.0 support (via a plug-in)
- v) Requirements management
- vi) Automated document generation
- vii) Debug of running code
- viii) Model-to-model transformations

Among the software techniques, Unified Modeling Language (UML) is the simplest way and the best engineering practices to model the software processes. It is important and useful for the software analysis and design phase and enables the user get better understanding of the software workflows. Therefore UML has been chosen along the software development.

2.4 Software Development Model

RUP has been chosen as the software development tools in this project. A few other software development models have been compares in detail in this chapter including waterfall model, spiral model and also prototype model.

2.4.1 RUP

The Rational Unified Process (RUP) is an iterative software development process framework created by the Rational Software Corporation, a division of IBM since 2003. RUP is not a single concrete prescriptive process, but rather an adaptable process framework, intended to be tailored by the development organizations and software project teams that will select the elements of the process that are appropriate for their needs

The RUP has determined a project lifecycle consistent of four phases. The first is the inception phase, which should establish baseline by which to compare actual expenditures versus planned expenditures. If the project does not pass this milestone, called the "Lifecycle Objective Milestone", it can either be canceled outright or it can repeat this phase after being redesigned to better meet the criteria.

The second elaboration phase is where the project starts to take shape. In this phase the problem domain analysis is made and the architecture of the project gets its basic form. This phase must pass the "Lifecycle Architecture Milestone". If the project cannot pass this milestone, there is still time for it to be canceled or redesigned. After

leaving this phase, the project transitions into a high-risk operation where changes are much more difficult and detrimental when made. The key domain analysis for the elaboration is system architecture.

The third phase is the construction phase , which main focus goes to the development of components and other features of the system being designed. This is the phase when the bulk of the coding takes place. In larger projects, several construction iterations may be developed in an effort to divide the use cases into manageable segments that produce demonstrable prototypes. This phase produces the first external release of the software. Its conclusion is marked by the Initial Operational Capability Milestone.

The forth and last is the transition phase. In the transition phase, the product has moved from the development organization to the end user. The activities of this phase include training of the end users and maintainers and beta testing of the system to validate it against the end users' expectations. The product is also checked against the quality level set in the Inception phase. If it does not meet this level, or the standards of the end users, another iteration of the phase begins. If all objectives are met, the "Product Release Milestone" is reached and the development cycle ends.

2.4.2 Waterfall

The waterfall model is a sequential software development process (a process for the creation of software) in which development is seen as flowing steadily downwards just like a waterfall through the phases of requirements analysis, design, implementation, testing , validation, integration, and maintenance.

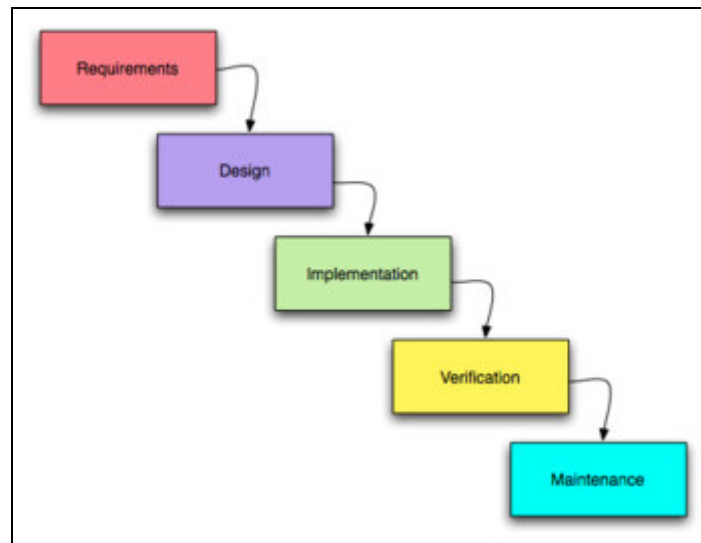


Figure 2. 1 : Waterfall Model

2.4.3 Spiral

The spiral model is a software development process combining elements of both design and prototyping-in-stages, in an effort to combine advantages of top-down and bottom-up concepts. Also known as the spiral lifecycle model, it is a systems development method used in information technology . This model of development combines the features of the prototyping model and the waterfall model. The spiral model is intended for large, expensive and complicated projects.

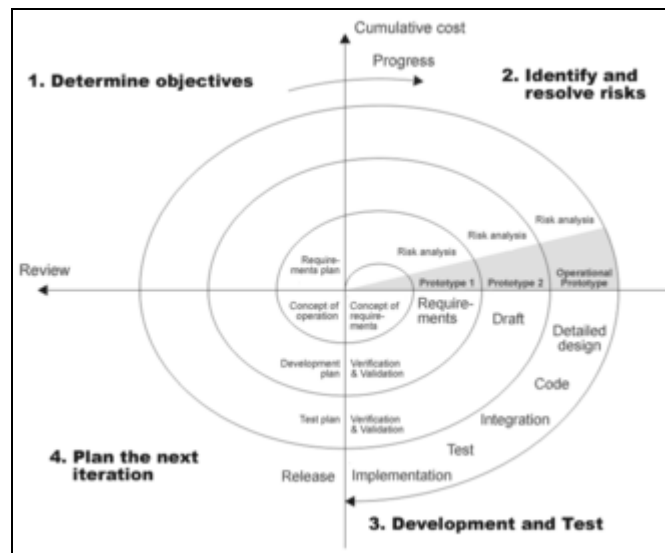


Figure 2. 2 : Spiral Model

2.4.4 Prototype

Prototype model is a static system that replicates the system that wants to be developed. It demonstrates the best practices of a user-centered design process and the key features that are valued by the user community and also stakeholders. The prototype is to be used as a training tool and design model for developers and designers who are working towards completing of similar system.

2.4.5 Comparison between RUP, Waterfall and Prototype Model

Table below shown the comparison between RUP, Waterfall and Prototype Model.

Table 2.1 : Comparison between RUP, Waterfall and Prototype Model

Model	RUP	WATERFALL	SPIRAL	PROTOTYPE
Advantages	1. Well documented and complete methodology 2. Open and Public 3. Training readily available 4. Reduce	1. Clear project objectives. 2. Stable project requirements. 3. Progress of system is measurable. 4. Strict sign-off requirements.	1. Avoidance of Risk is enhanced. 2. Strong approval and documentation control. 3. Implementation has	1. Strong Dialogue between users and developers 2. Missing functionality can be identified easily 3. Confusing or difficult

Model	RUP	WATERFALL	SPIRAL	PROTOTYPE
	<p>integration time and effort.</p> <p>5. Higher level of reuse</p>		<p>priority over functionality.</p> <p>4. Additional Functionality can be added at a later date.</p>	<p>functions can be identified</p> <p>4. Requirements validation, Quick implementation of, incomplete, but functional, application</p> <p>5. May generate specifications for a production application</p> <p>6. Environment to resolve unclear objectives</p> <p>7. Encourages innovation and flexible designs</p>
Disadvantages	<p>1.The process is too complex</p> <p>2.Sociological aspects</p> <p>3. Disorganized development.</p>	<p>1. Time consuming</p> <p>2. Never backward (Traditional)</p> <p>3. Little room for iteration</p> <p>4. Difficulty responding to changes</p>	<p>1. Highly customized limiting re-usability</p> <p>2. Applied differently for each application</p> <p>3. Risk of not meeting budget or</p>	<p>1. Contract may be awarded without rigorous evaluation of Prototype</p> <p>2. Identifying non-functional elements difficult to document</p> <p>3. Incomplete</p>

Model	RUP	WATERFALL	SPIRAL	PROTOTYPE
			schedule 4. Possibility to end up implemented as the Waterfall framework	application may cause application not to be used as the full system was designed 4. Incomplete or inadequate problem analysis 5. Client may be unknowledgeabl e 6. Approval process and requirement is not strict 7. Requirements may frequently change significantly

RUP model has being selected as the development model for this project. The company has long use RUP model referring to its advantages and best practices.

2.5 Application Server

Earlier during the requirement phase, the project management has assigned the author to use Apache Tomcat as the servlet container. But later on has been changed to JBoss. The reason JBoss is chosen to be the servlet container is for easy integration since the other application is using JBoss.

2.5.1 Apache Tomcat

Apache Tomcat is a Servlet container developed by the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, and provides a HTTP web server environment for Java code to run.

Tomcat should not be confused with the Apache web server, which is a C implementation of an HTTP web server; these two HTTP web servers are not bundled together. Apache Tomcat includes tools for configuration and management, but can also be configured by editing configuration files that are normally XML-formatted.

2.5.2 JBoss

JBoss Application Server is a free software / open source Java EE-based application server. Because it is Java-based, the JBoss application server is cross-platform, usable on any operating system that Java supports

Table 2. 2 : Product Feature Difference

	JBoss	Apache Tomcat
Product Feature	<ol style="list-style-type: none"> 1. Clustering 2. Failover (including sessions) 3. Load balancing 4. Distributed caching (using JBoss Cache, a standalone product) 5. Distributed deployment (farming) 6. Enterprise Java Beans version 3 7. Aspect-Oriented Programming(AOP)-support 8. Hibernate-integration (for persistence programming;JPA) 9. Support for J2EE-Web Services like JAX-RPC (Java API for XML for Remote Procedure Call) 	<ol style="list-style-type: none"> 1. Implements the Servlet 2.3 and JSP 1.2 specifications 2. Servlet container redesigned as Catalina 3. JSP engine redesigned as Jasper 4. Coyote HTTP connector 5. Java Management Extensions (JMX), JSP and Struts-based administration 6. Reduced garbage collection, improved performance and scalability 7. Native Windows and Unix wrappers for platform integration

Apache Tomcat has been chosen by the other as the servlet container because for Pentaho itself unable to integrate with JBoss. After a few attempt and changes being made, the performance of Pentaho with Apache Tomcat is easier and more reliable

2.6 Operating System

There were a few Operating System that has been put into research as a way to find the most suitable Operating System for this project because of its robustness and also because it is an open source software.

2.6.1 FreeBSD

FreeBSD is a Unix-like free operating system descended from AT&T Support for other architectures is in varying stages of development. FreeBSD has been characterized as the unknown giant among free operating systems. It is not a clone of UNIX, but works like UNIX, with UNIX-compliant internals and system APIs. FreeBSD is generally regarded as reliable and robust. Among all operating systems which can accurately report uptime remotely, FreeBSD is the free operating system. A long uptime also indicates no crashes have occurred and no kernel updates have been deemed needed, since installing a new kernel requires a reboot, resetting the uptime counter of the system.

FreeBSD is developed as a complete operating system. The kernel, device drivers and all of the user land utilities, such as the shell, are held in the same source code revision tracking tree, whereas with Linux distributions, the kernel, user land utilities and applications are developed separately, then packaged together in various ways by others.

2.6.2 Linux

Linux is a Unix-like computer operating system family which uses the Linux kernel. Linux is one of the most prominent examples of free software and open source development; typically all the underlying source code can be freely modified, used, and redistributed by anyone.

Linux is well known for its use in servers, it is installed on a wide variety of computer hardware, ranging from embedded devices and mobile phones to supercomputers.

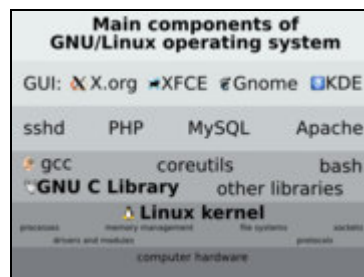


Figure 2. 3 : Main Component of GNU/LINUX Operating System

2.6.3 Windows 2000

Windows 2000 also known as Win2K is a preemptive, interruptible, graphical and business-oriented operating system designed to work with either uniprocessor or symmetric multi-processor computers. It is part of the Microsoft Windows NT line of operating systems. It is a hybrid kernel operating system.

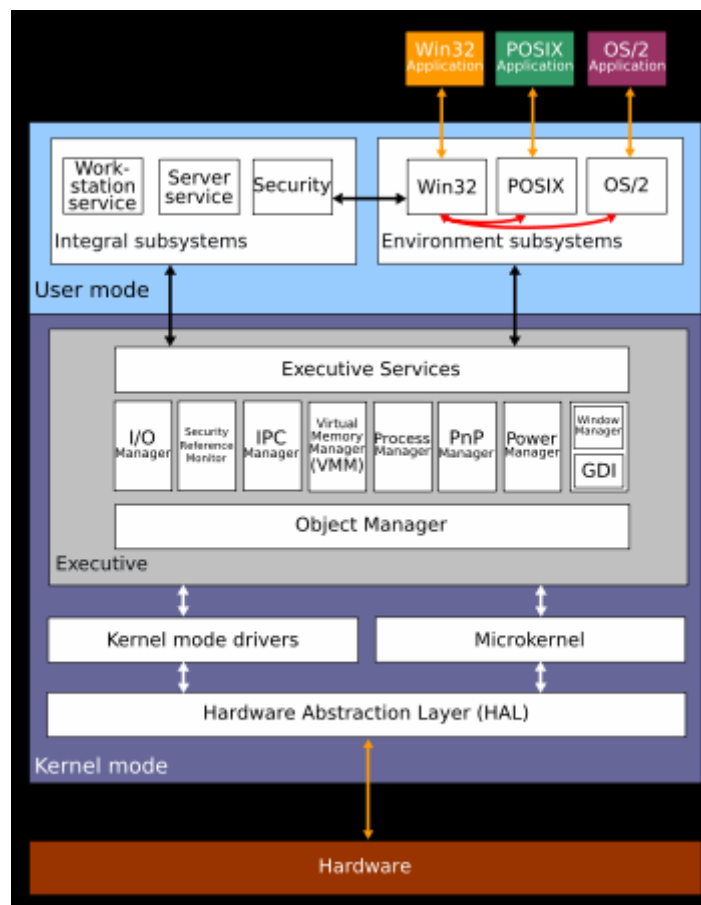


Figure 2. 4 : Windows 2000 Operating System Architecture

2.6.4 Comparison between FreeBSD, LINUX and Windows 2000.

Table below shows the reliability comparison between FreeBSD, LINUX and Windows 2000.

Table 2.3 : Comparison between FreeBSD, LINUX and Windows 2000

	FreeBSD	LINUX	Windows 2000
Reliability	<ol style="list-style-type: none"> 1. Extremely robust. uptimes measured in years 2. New Soft Updates file system 3. Optimizes disk I/O 4. High performance ensures reliability for transaction based applications. 	<ol style="list-style-type: none"> 1. Servers often stay up for years. disk I/O is non-synchronous by default 2. Less reliable for transaction based operations 3. Produce a corrupted file system after a system crash or power failure 4. Very dependable OS. 	<ol style="list-style-type: none"> 1. Poor reliability 2. Code bloat has introduced many more reliability problems. 1. Uses a lot of system resources and it is 2. Very difficult to keep the system up for more than a couple of months

FreeBSD has been chosen as the operating system after several comparisons has been made by the author and the project's team. From the comparison that has been made. It is proved that it has a better reliability.

2.7 Reporting

Three reporting software has been considered for this project including Pentaho Reporting, Crystal Report and also Jasper Report. Pentaho Reporting is chosen by the management for this project as an effort to explore the Pentho itself since the company has used Crystal Report and Jasper report in different projects.

2.7.1 Pentaho Reporting

Pentaho Reporting is a class library for generating reports. XML-based templates provide flexible reporting and printing functionality using data from multiple sources. It supports output to display devices, printers, PDF, Excel, HTML, XHTML, PlainText, XML and CSV files.

Pentaho Reporting was formerly known as JFreeReport, but has been renamed to 'Pentaho Reporting' to avoid confusion with the other JFree.org projects.

2.7.2 Crystal Report

Crystal Reports is a business intelligence application used to design and generate reports from a wide range of data sources. Several other applications, such as Microsoft Visual Studio, bundle an OEM version of Crystal Reports as a general purpose reporting tool. Crystal Reports became the de facto report writer when Microsoft released it with Visual Basic.

2.7.3 Jasper Report

JasperReports is an open source Java reporting tool that can write to screen, to a printer or into PDF, HTML, Microsoft Excel, RTF, ODT, Comma-separated values and XML files.

It can be used in Java-enabled applications, including J2EE or Web applications, to generate dynamic content. It reads its instructions from an XML or .jasper file.

CHAPTER 3

METHODOLOGY

This chapter describes the literature study that has been made to find the best solution to develop the RM Module. In order to make sure the capability is same as the previous system

3.1 Software Development Process

Scan Associates Bhd. practices the Iterative Rational Unified Process (RUP) as their software process with few trade-off due to industrial needs and to suit the nature of the various projects.

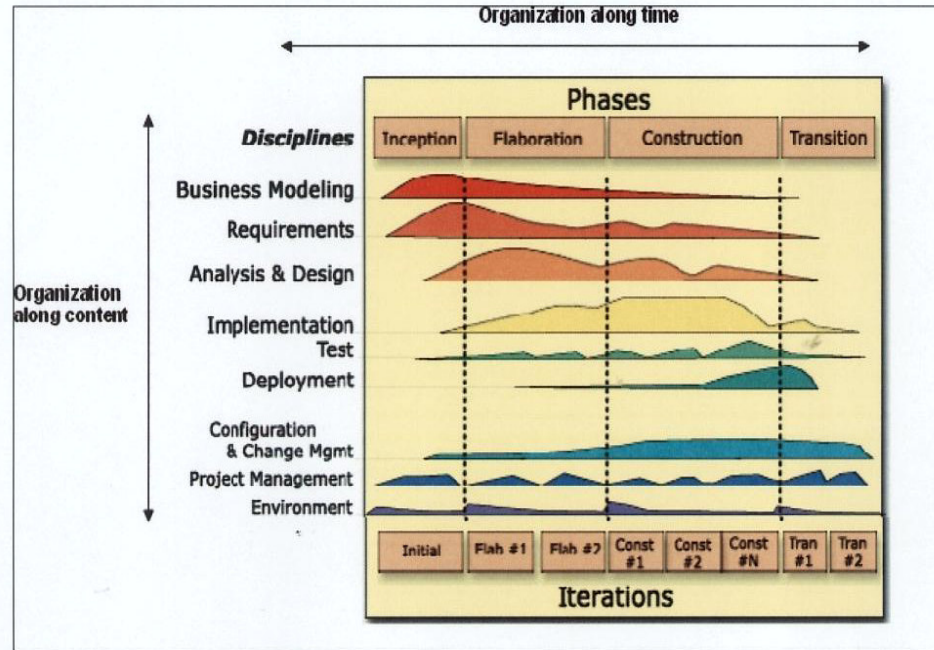


Figure 3.1 : RUP overview

The RUP is organized in both the time (the life cycle model, phases and iterations) and content (the disciplines to be used) as shown by the “iteration cycle graph” above.

Iteration consists of planning, requirements, analysis and design, implementation and testing in various proportions depending on where the iteration is in the software lifecycle. Early iterations focus on requirements analysis and architectural design, whereas late iterations focus more on design, implementation, and testing. The “iteration cycle graph above illustrates typical proportions. Thus, part of the iteration plan is to decide how the various disciplines are to be exercised for every iteration.

The process architecture adopted is that of the RUP. In this architecture a clear separation between the time dimension of a project (represented by phases and milestones of the process lifecycle model), and the process components (the disciplines, workflow details, roles, activities, artifacts, templates and guidelines that define the static elements of the process).

The lifecycle adopted is based on the RUP. It is tailored for any SCAN product development project as follows:

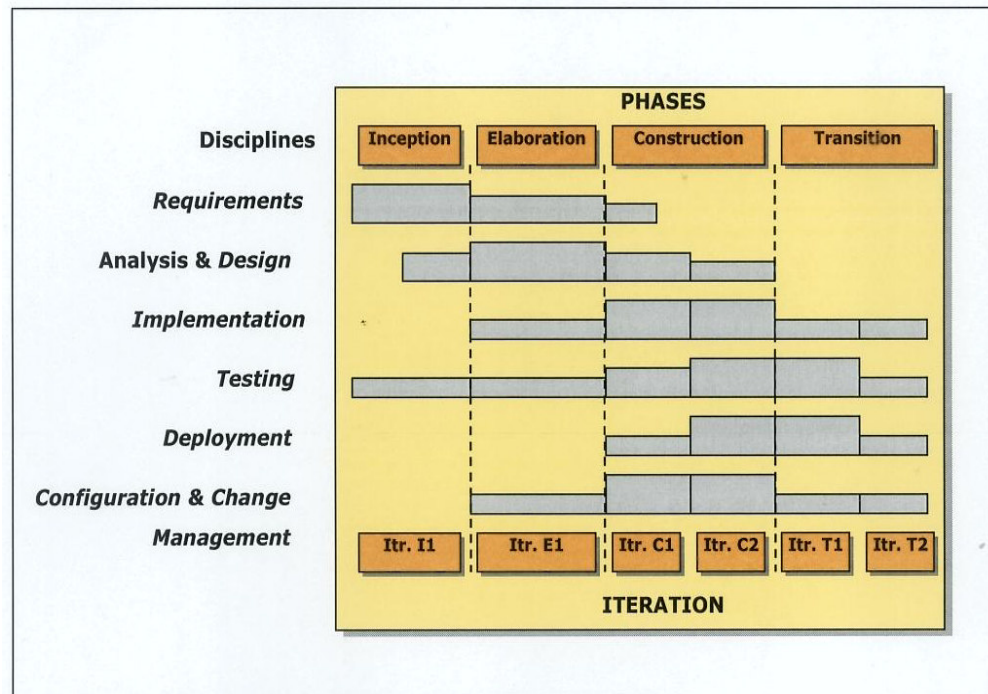


Figure 3.2 : Iterative RUP Lifecycle model

In this tailored software lifecycle, it is advisable to iterate the RUP standard iteration. To adapt to the adaptive software development, the software is divided into Release/Version, cycle and build. One particular release consists of one or multiple cycles and one particular cycle consists of one or multiple builds.

One particular release is considered as either a system that has implemented all the required functionality or a system in a certain period of development stage or a system that has reached the end of the project, whichever comes first. Client testing is not compulsory in each release but advisable, as each release is actually n(th) number of the cycle, and user testing is compulsory to be conducted.

One particular cycle considered as a minor will involved and user testing. Each cycle have range from two to eight weeks depending on the uncertainly of the requirements in hand. For requirements that are ambiguous and volatile, the cycle duration could be as short as two weeks and for stable and clear requirements, the cycle duration could be as long as eight weeks. For the first two cycles, it is mainly used to confirm the core requirements. At the end of each cycle, at a particular phase of the project, the product will be tested by internal tester and that testing could be considered as acceptance test.

The size and complexity of a product development does not affect much of the life cycle. The only difference between big and complicated product development and small and simple product development only lies on the number of cycles and builds. For a small and simple product, the cycle could be less than five and build could be less than ten for example. It is all depends on the judgment of the product manager to plan and the involvement of the client at the beginning of the product development.

The RUP model is used in this project by using the basic phase the Requirements, Analysis and Design, Implementations, Testing and Deployment. Since the project is only up to Requirements and Analysis and Design phase. The author has been complying with the inception, elaboration, construction and transition iteratively.

3.2 Software Standard and guideline

During the industry training, DoD Standard 1267A is preferred for the purpose of documentation. There are three documents will be produced based on this standard guideline, which are:

- i) Software Requirement Specification (SRS)
- ii) Software Design Document (SDD)
- iii) For the final report, the document will follow the UTM Thesis Manual guideline

But as a request from to the SCAN Associates Bhd to follow its own standard and guidelines:

- i) SAD (Software Architecture Design)
- ii) SRS (Software Requirement specification)

3.3 Software Development Tools (Pentaho)

Pentaho is the commercial open source alternative for business intelligence (BI). Pentaho Open BI Suite provides comprehensive reporting, OLAP analysis, dashboards, data integration, data mining and a BI platform that have made it the world's leading and most widely deployed open source BI suite. Pentaho's commercial open source business model eliminates software license fees, providing support, services, and product enhancements via an annual subscription. In the years since Pentaho's inception as the pioneer in commercial open source BI, Pentaho's products have been downloaded more than three million times, with production deployments at companies ranging from small organizations to The Global 2000.

3.3.1 The Pentaho BI Project

Pentaho Corporation is the primary sponsor and leader of The Pentaho BI Project. The Pentaho BI Project is an ongoing initiative by the open source community that provides organizations with best-in-class solutions for their enterprise Business Intelligence (BI) needs. By leveraging the wealth of open source technologies and the contributions of the open source development community we are able to innovate much faster than commercial vendors.

As a result, it provides an open source alternative that surpasses the leading proprietary BI suites in many areas of architecture, standards support, functionality, and ease of deployment. The Pentaho BI Project encompasses the following major application areas:

- i) Reporting
- ii) Analysis
- iii) Dashboards
- iv) Data Mining
- v) Workflow
- vi) BI Platform

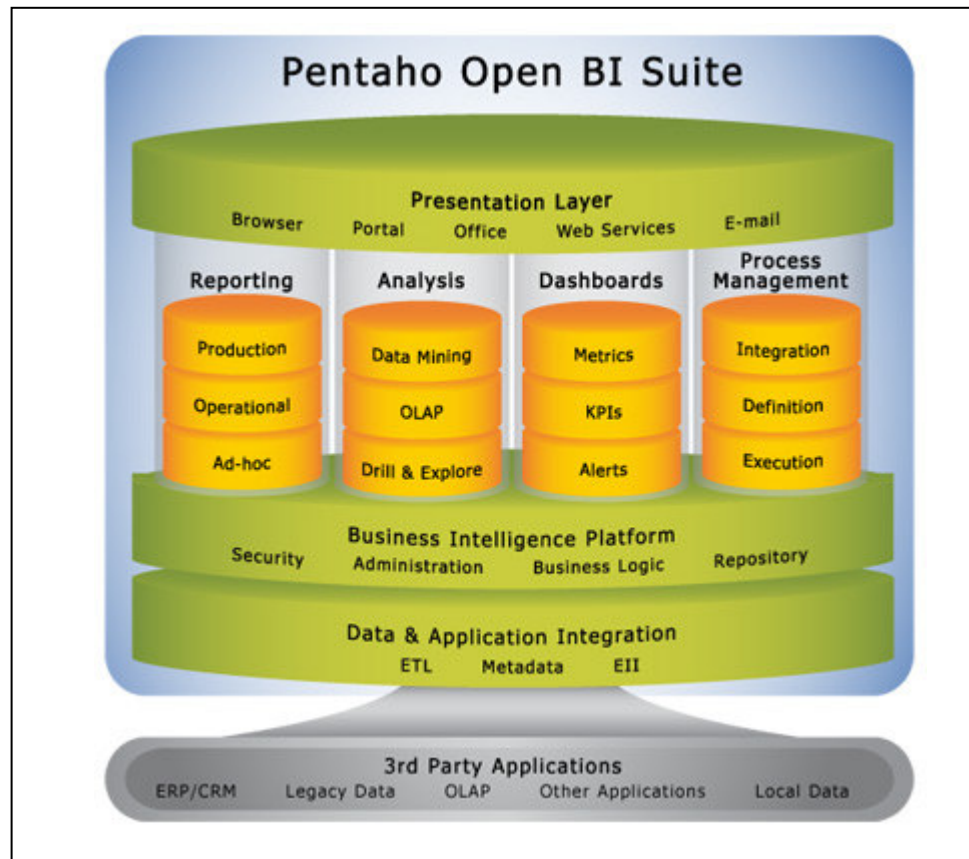


Figure 3.3 : Pentaho Open BI Suite

3.3.2 Pentaho Projects

Pentaho is the sponsor and owner of other open source projects. These projects provide components and functionality integrated with the Pentaho BI Platform.

- i) Mondrian - Open Source OLAP Server
- ii) Pentaho Reporting Engine - Open Source Reporting
- iii) Kettle - Open Source Data Integration (ETL)
- iv) Pentaho - Comprehensive Open Source BI Suite
- v) Weka - Open Source Data Mining

3.3.3 The Pentaho BI Project exists to benefit and serve

Below are the list of BI Project exists to benefit and serve:

- i) Java developers who can use project components to rapidly assemble custom BI solutions.
- ii) ISVs that can enhance the value and capability of their solutions by embedding BI functionality.
- iii) End-Users who can rapidly deploy packaged BI solutions which are either competitive or superior to traditional commercial offerings at a dramatically lower cost.
- iv) The Pentaho BI Project teams who are free to pursue their passion for BI while being a part of the Open Source movement which is revolutionizing the software industry to better serve users.

3.4 About Pentaho Reporting

All organizations use reporting in one form or another. As a result, reporting is considered a core Business Intelligence (BI) need and is frequently the first BI application deployed. Pentaho Reporting allows organizations to easily access, format, and distribute information to employees, customers, and partners.

- i) Flexible deployment from standalone desktop reporting, to interactive web-based reporting to enterprise business intelligence – with a free open source alternative at every step
- ii) Broad data source support including relational, OLAP, or XML-based data sources
- iii) Flexible output options including Adobe PDF, HTML, Microsoft Excel, Rich Text Format, or plain text
- iv) Wizard-driven report design for fast, easy report creation

- v) Professional Edition available with additional deployment capabilities including clustering, subscriptions, directory integration, versioning, auditing, and more

3.4.1 Pentaho Technology Fundamentals

Pentaho is componentized and modular and also perform Service-implemented architecture. It is built from the ground up as a set of services and exposed via AJAX. Pentaho is Web Services 100%. It also has the Java EE server side scalable, standards-based Web-based, thin-client end user interfaces, eclipse-based design and interfaces Embedded workflow / process engine.

3.4.2 Pentaho BI Platform

Pentaho BI Platform provides critical services for end users and also provide scheduling Subscriptions Bursting/Distribution Alerting, notification Integration of end user tools. It also provides critical services for administrators Integration with external business logic definition and execution of business rules. This BI platform content presentation, integration points to applications, portals process management and integration centralized security.

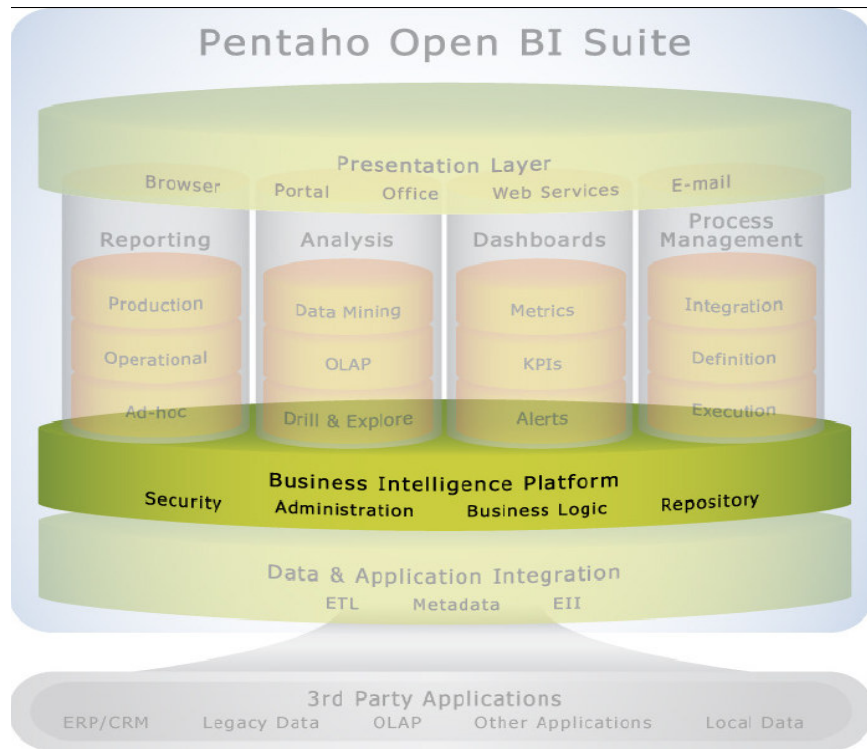


Figure 3.4 : Pentaho Business Intelligence Platform

Table 4.1 : Pentaho BI Platform Advantages

	Pentaho BI Platform
Advantages	<ol style="list-style-type: none"> 1. Embeddable and componentized 2. Not a “BI Monolith” 3. Delivered as libraries 4. Based on open technologies and standards (J2EE, JDBC, JNDI, JSR-168, JSP, HTML, XSL, and XML) 5. 100% Java Portability, scalability, integration 6. Operational BI tightly integrates business processes, business rules and components in BI Suite.

3.4.3 Pentaho Reporting

Pentaho Reporting allow user to access and format data from disparate sources RDBMS, XML, and OLAP. Pentaho Reporting produce output in popular formats and in multiple report types. It performs Operational Analytical Financial Directly against sources or using centralized metadata layer pixel perfect report designer Web-based business user ad hoc.

Table 4.2 : Pentaho Reporting Advantages

	Pentaho Reporting
Advantages	<ol style="list-style-type: none">1. Flexible Report2. Design Graphical design environment3. Externally reference commonly used Report Sections4. Conditional hiding of report objects5. Access relational, OLAP, or XML data6. Embeddable Lightweight Easy to extend7. 100% Java Portability, scalability, integration8. Cross Platform (client and server) Mac, Linux/Unix, Windows

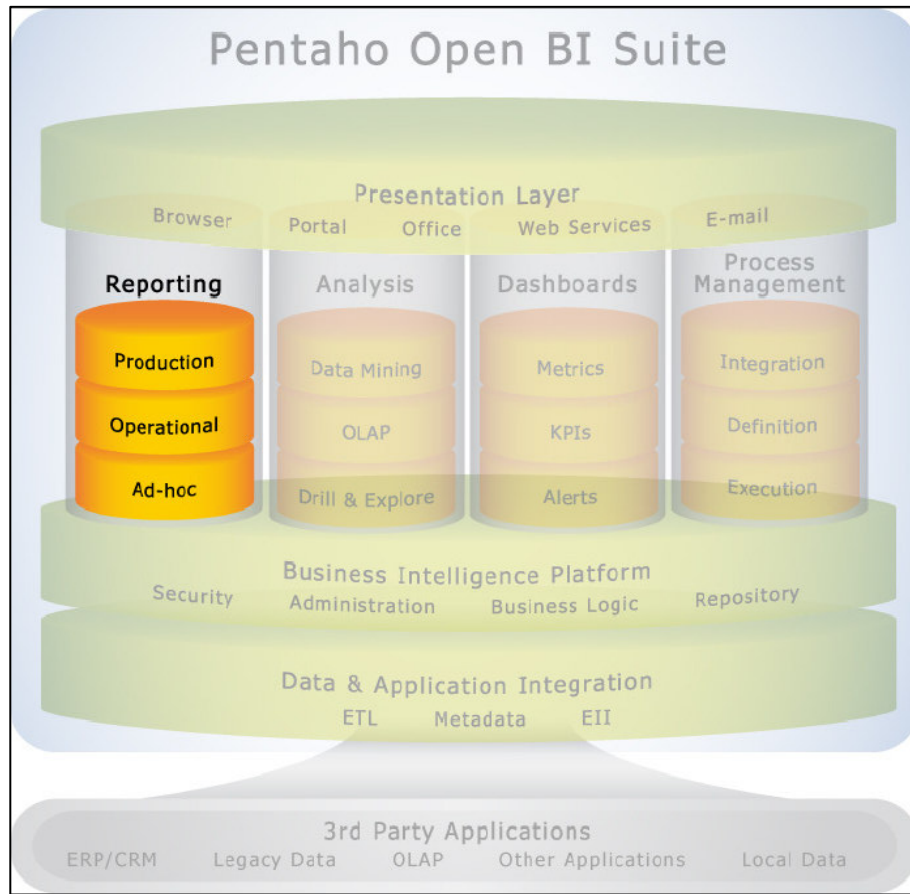


Figure 3.5 : Pentaho Reporting

3.4.4 Pentaho Analysis

Pentaho analysis is also known as ROLAP architecture. It works with all popular open source and proprietary database. Pentaho analysis allow user to view data dimensionally and also navigate and explore ad hoc analysis drill-down from year to quarter Pivot. It also allow user to select specific members for analysis and web-based or Excel front ends.

Table 4.3 : Pentaho Analysis Advantages

	Pentaho Analysis
Advantages	<ol style="list-style-type: none"> 1. Standards-based architecture 2. J2EE architecture 3. JDBC and JNDI connectivity 4. SQL-based data retrieval XML/A and MDX front-end support 5. Embed ability and extensibility Rich, interactive analysis 6. Web- or Excel-based access Performance and scalability 7. Optimized SQL Aggregate table support 8. Aggregation Designer Pentaho Open BI Suite Integration 9. Comprehensive auditing of user activity, performance and data access 10. Integrated security, scheduling, alerting, portal integration, and metadata

3.4.5 Pentaho Dashboards

Pentaho dashboard is tight business process integration. It contain embedded workflow and can receive events from or trigger events in external systems. Pentaho Open BI Suite Integration Comprehensive auditing of user activity, performance and data access. The best feature of pentaho dashboard is the context-sensitive drilling to reports or analysis. It has the integrated security, scheduling, alerting, portal integration.

Table 4.4 : Pentaho Dashboards Advantages

	Pentaho Dashboard
Advantages	<ol style="list-style-type: none"> 1. Flexible deployment options 2. Out-of-the box Fully integrated with entire Pentaho Open BI Suite 3. AJAX components provide high performance, customizable extensions 4. Embeddable into existing solutions 5. Portal integration Rich, dynamic information delivery 6. Wide variety of visual components 7. Role or subject-based filtering 8. Drill to detailed reports, analysis views or other Dashboards with context 9. Integration with Google Maps for location intelligence

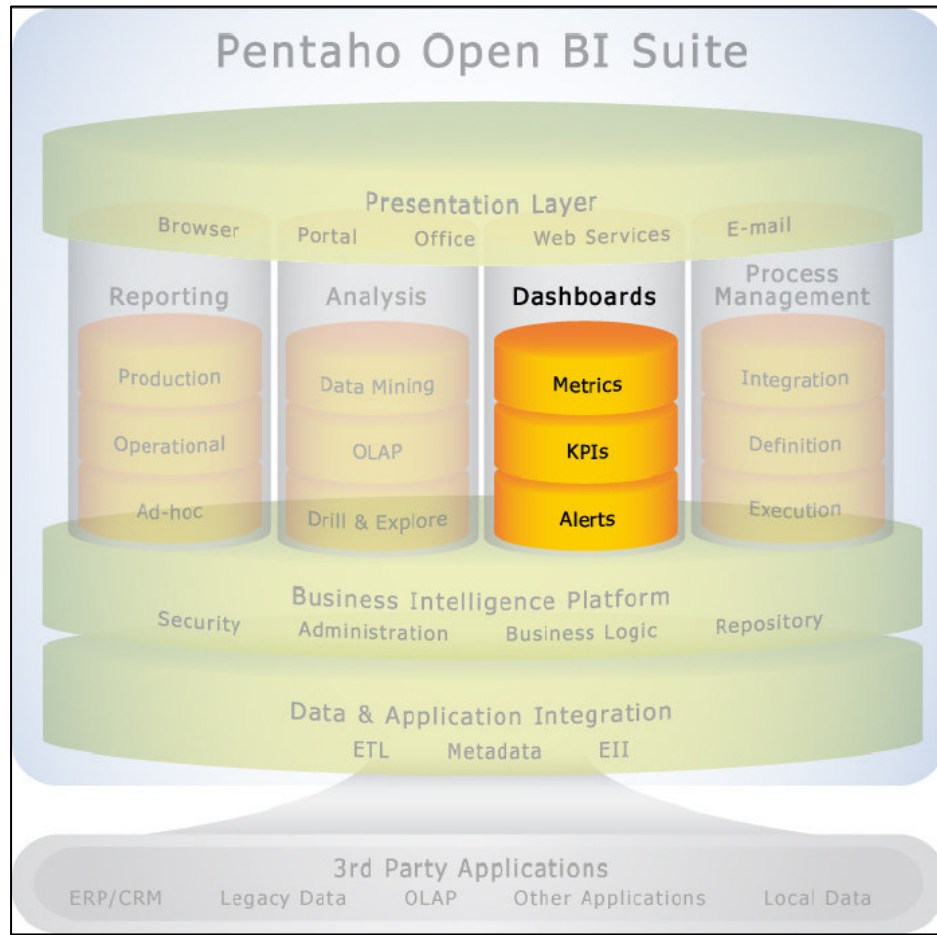


Figure 3.6 : Pentaho Dashboard

3.4.6 Pentaho Data Integration

Pentaho Data Integration consist Rich Feature Set Enterprise-class with high performance and scalability. It is 100% Broad Database Support, Meta-data Driven Graphical and model-driven design Mature.

Table 4.5 : Pentaho Data Integration Advantages

	Pentaho Data Integration
Advantages	<ol style="list-style-type: none"> 1. Meta-data, model-driven approach 2. Tell it WHAT to do and not HOW to do it 3. Complex transformations with zero code 4. Graphically design data transformations and jobs 5. 100% Java, cross-platform support 6. Extensible architecture Easy to develop and plug in custom connectors 7. SAP connector available 8. Repository-based 9. Structured management of models, connections, logs and more 10. Easy re-use of queries and transformation components.

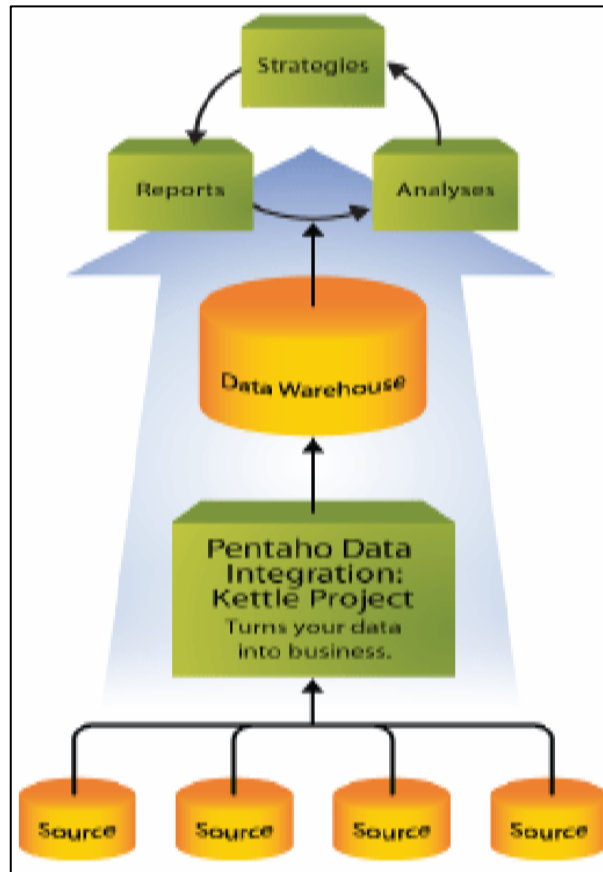


Figure 3.7 : Pentaho Data Integration

3.5 Extension to the Project

For the project Research and Development, the commercial open source alternative for Business Intelligence, Pentaho is the main research. But the focus is on Pentaho Reporting which is one of Pentaho product. Embedded software reporting tools has been used majorly in this research. The performance and suitability of this software is being studied by the author for suitability and easy integration purposes.

3.6 Software Tools

For this project, the software tools that will be used are as following:

3.6.1 Microsoft Word 2007

Microsoft Office 2007 or in other name officially called 2007 Microsoft Office system is the most recent Windows version of the Microsoft Office system, Microsoft's productivity suite. Formerly known as Office 12 in the initial stages of its beta .These are, respectively, the same dates Windows Vista was released to volume licensing and retail customers. Office 2007 contains a number of new features, the most notable of which is the entirely new graphical user interface. It was purposely replacing the menus and toolbars that have been the cornerstone of Office since its inception with a tabbed toolbar, known as the Ribbon.

3.6.2 Microsoft Visio 2003

Microsoft Visio is diagramming software for Microsoft Windows. It uses vector graphics to create UML diagrams. It is currently available in two editions, Standard and Professional.

The Standard and Professional editions both share the same interface, but the latter has additional templates for more advanced diagrams and layouts as well as unique functionality that makes it easy for users to connect their diagrams to a number of data sources and display the information graphically.

3.6.3 Rational Rose 2000

Rational Rose is an object-oriented Unified Modeling Language (UML) software design tool intended for visual modeling and component construction of enterprise-level software applications. In much the same way a theatrical director blocks out a play, a software designer uses Rational Rose to visually create (model) the framework for an application by blocking out classes with actors (stick figures), use case elements (ovals), objects (rectangles) and messages/relationships (arrows) in a sequence diagram using drag-and-drop symbols. Rational Rose documents the diagram as it is being constructed and then generates code in the designer's choice of C++, Visual Basic, Java, Oracle8, CORBA or Data Definition Language.

3.6.4 Pentaho Reporting

Pentaho Reporting allows organizations to easily access, format, and distribute information to employees, customers, and partners.

- i) Flexible deployment from standalone desktop reporting, to interactive web-based reporting to enterprise business intelligence
- ii) Broad data source support including relational, OLAP, or XML-based data sources
- iii) Flexible output options including Adobe PDF, HTML, Microsoft Excel, Rich Text Format, or plain text
- iv) Wizard-driven report design for fast, easy report creation
- v) Web-based ad hoc query and reporting for business users
- vi) Full-featured graphical Report Designer for report developers
- vii) Enterprise Edition provides enhanced software functionality, comprehensive professional technical support, product expertise, certified software and software maintenance, and more

CHAPTER 4

PROJECT DISCUSSION

This chapter describes the overall project discussion including the analysis and design. The project discussion will describe detail will covers the project management, project implementation methodology

4.1 Introduction

The project implementation approaches are using methodology to provide a comprehensive framework to both project management and the security analyst.

4.2 Project Implementation Methodology

The RM Module was implemented by following the process below:

- i) Project preparation
- ii) Requirement Study/Business Modelling
- iii) System Implementations
- iv) Deployment

4.2.1 Project preparation stage

Requirement study stage is one of the earliest phases in software development. The author first task is to finalize the overall project scope and objective. Table below shows the task and deliverable needed during project development.

Table 4.1 : Task and Deliverable

Stage	Teams Task	User Task	Deliverables
Project Preparation	<ul style="list-style-type: none"> i) Finalize overall project scope with team members. ii) Determine the project standard and guideline for documentation. iii) Prepare the project schedule 	Assemble all the security consultant team	<ul style="list-style-type: none"> i) Initial work plan ii) Meeting iii) Technology and platform identified

4.2.2 User Requirements

The requirement study phase consist the Use Case, Sequence Diagram, Collaboration Diagram, Class Diagram, Component Diagram and also the Deployment Diagram that has been used by the author to interpret the requirement into business modelling.

4.2.3 Reporting Management Use Case

This section describes the requirements for Reporting Management (RM) module. There are three use cases in RM which are Upload Report, Get Report and Generate Report. Figure 4.1 illustrates Use Case Diagram for RM Module.

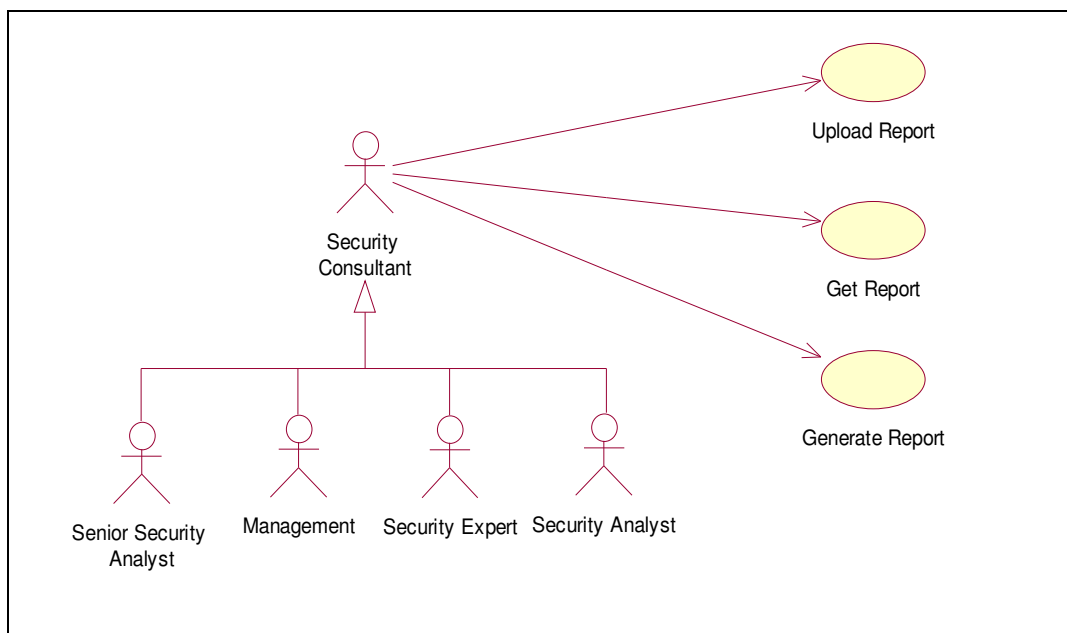


Figure 4.1 : Use Case Diagram for RM Module

4.2.3.1 Upload Report

This use case is initiated by the Security Consultant. It provides the capability for Security Consultant to upload report to customer. Following figure shows the Upload Report Use Case.

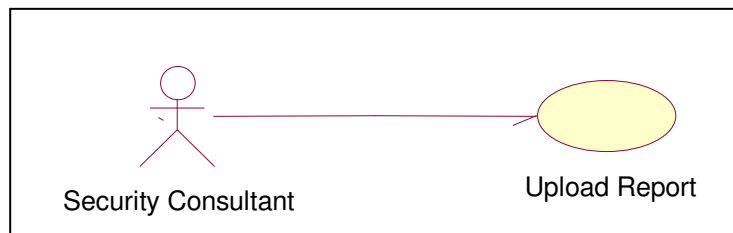


Figure 4.2 : Upload Report Use Case

Following figure shows the Interface Design Specification for Upload Report.

The interface design specification is contained within a rectangular border. It features the following elements:

- Upload /Email to:** A dropdown menu with a blue background and the text "Select Client".
- Reportname:** A text input field.
- Description:** A text input field.
- Upload /File:** A text input field with a "Browse..." button to its right.
- Buttons:** Two buttons labeled "Upload" and "Email" are positioned at the bottom center of the form.

Figure 4.3 : Interface Upload Report

4.2.3.2 Get Report

This use case is initiated by the Security Consultant. It provides the capability for Security Consultant to get report. Following figure shows the Get Report Use Case.

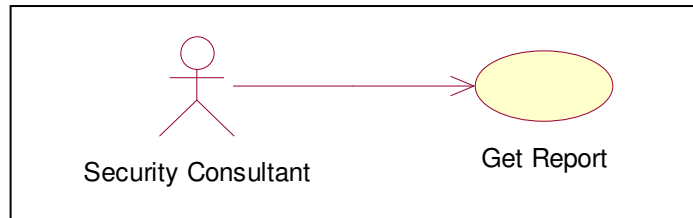


Figure 4.4 : Get Report Data Use Case

Following figure shows the Interface Design Specification for Get Report.

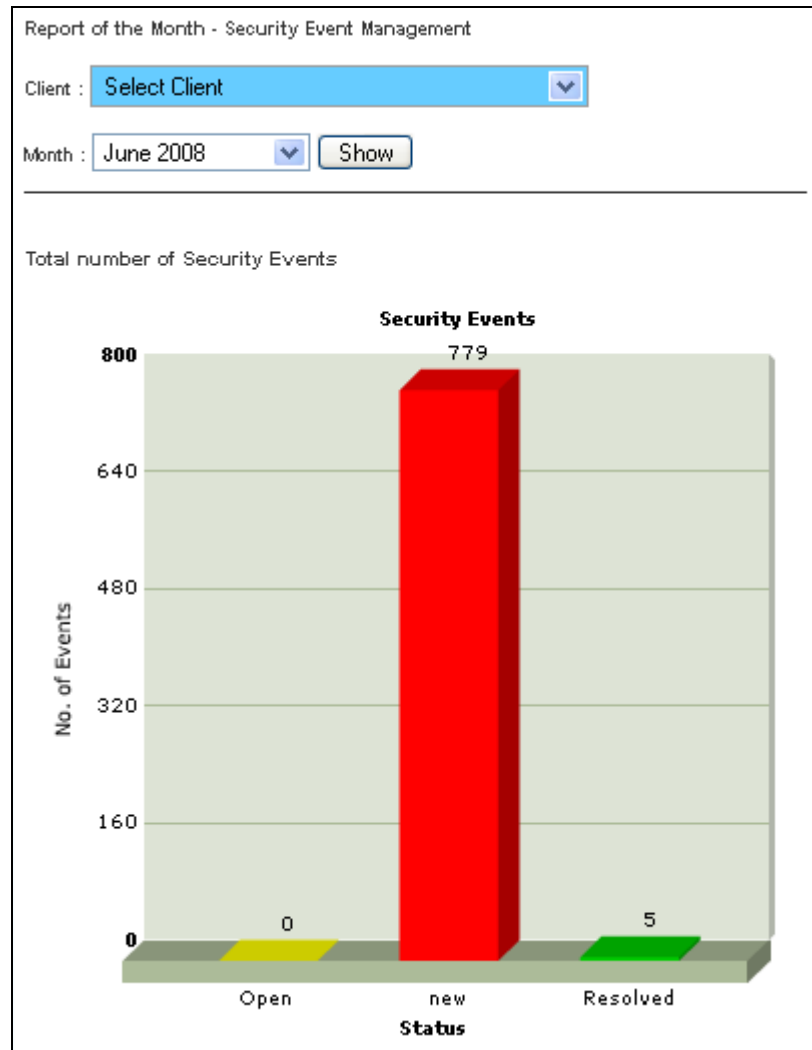


Figure 4.5 : Interface Report of the month

***Note:** The above chart type and interface design may revise depends on the Designers.

4.2.3.3 Use Case of Generate Report

This use case is initiated by the Security Consultant. It provides the capability for Security Consultant to generate report. Figure shows the Generate Report Use Case.

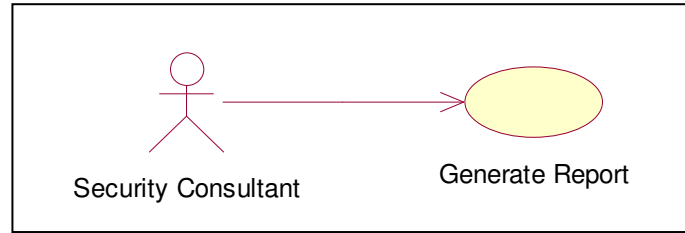


Figure 4.6 : Generate Report Use Case

Following figure shows the Interface Design Specification for Generate Report.

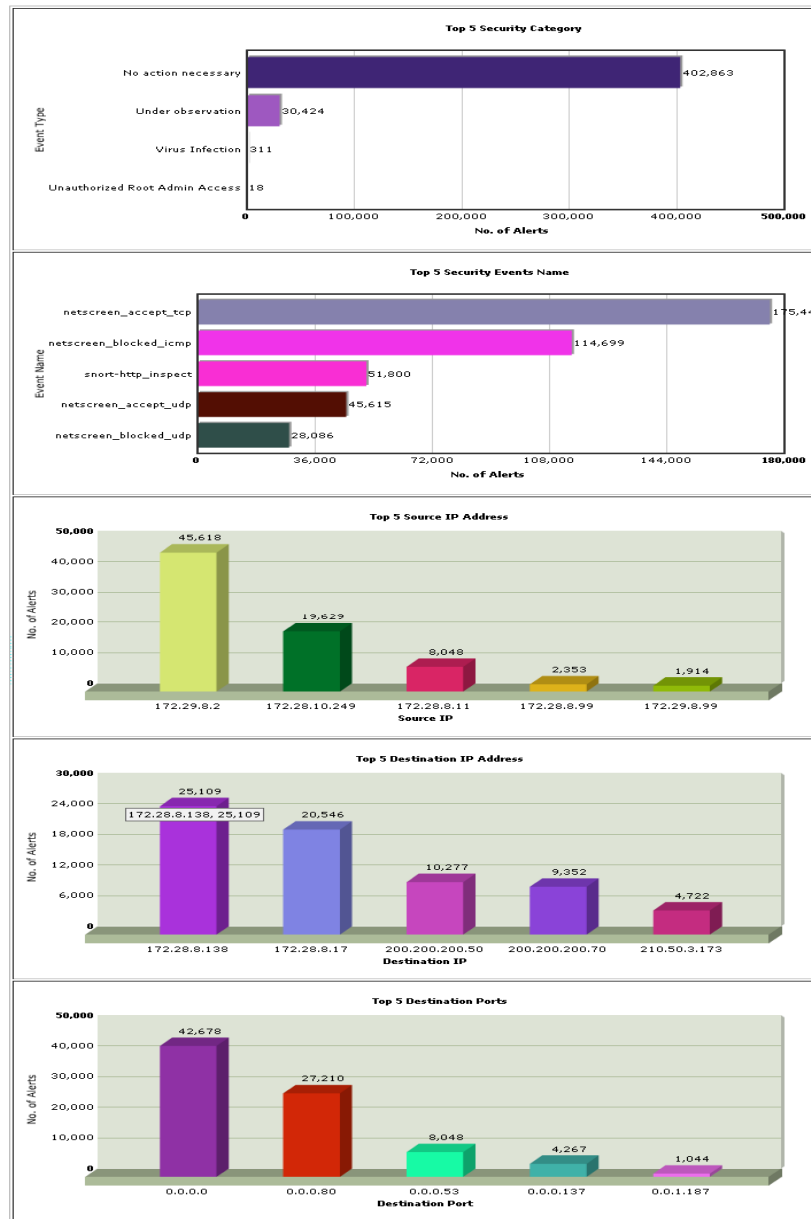


Figure 4.7 : Interface Top 5 Report

Client :	<input type="text" value="Select Client"/>
Month :	<input type="text" value="June 2008"/> <input type="button" value="Generate"/>

Figure 4.8 : Interface Generate Vulnerability Report

4.2.4 Class Diagram

Class diagram shows the whole structure of the system including classes, internal structure and also relationships between all the classes. In this project RM Module has been divided into three packages, the packages are, Generate Report, Upload Report and Get Report. Figure below shows the relationship among CSCs in RM Module.

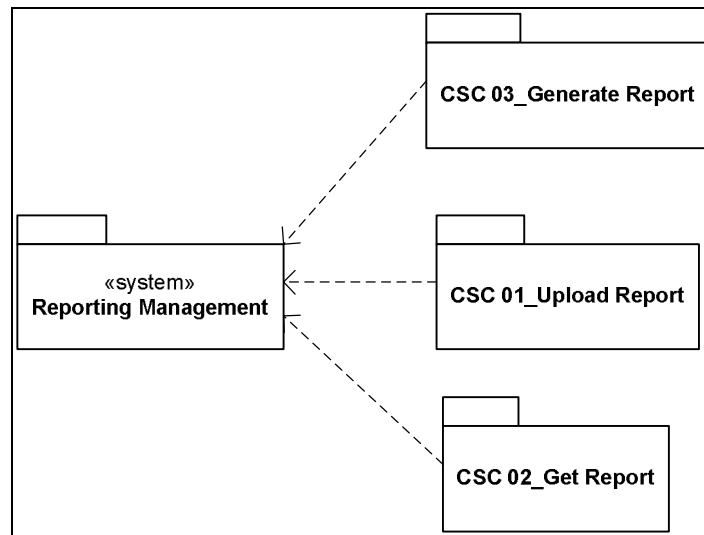


Figure 4. 9 : Relationship among CSCs in RM Module

The Class diagram of RM Module consists of ClientEmailList, FormList, StatisticalData, clientId, FormList, ReportList and Event. Figure below shows the Class Diagram for RM Module.

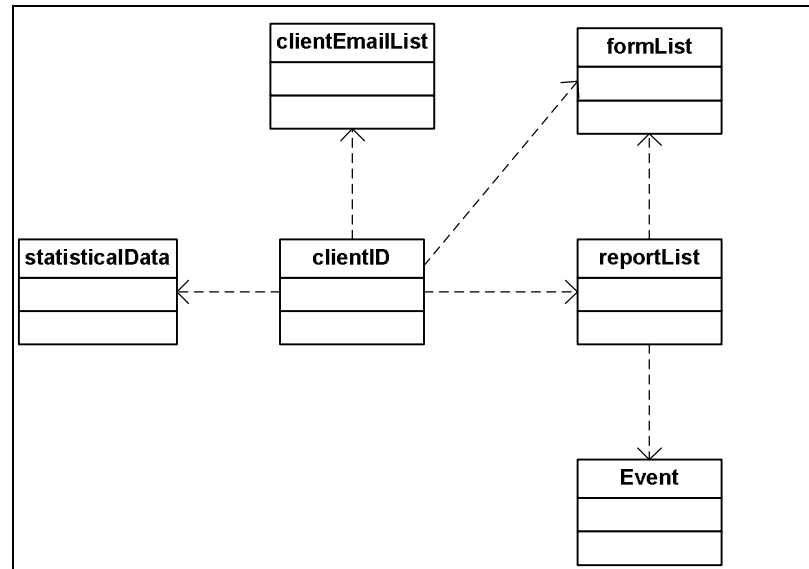


Figure 4. 10 : Class Diagram for RM Module

4.2.5 Sequence Diagram and Collaboration Diagram

Sequence diagram together with collaboration diagram has been used in order to explain the requirement workflow of the use case.

Appendix B shows the sequence diagram of every use case for RM Module and Appendix C shows the collaboration diagram of each use case in RM Module.

4.2.6 System Architecture

The system architecture contains the system's physical organization. It will not be explain in this section following the project scope.

4.2.7 Component Diagram

The physical module of the code can be interpreted in component diagram. The source code libraries and also the runtime files can be included in component diagram. Figure below shows the main component that has been required to implement the RM Module.

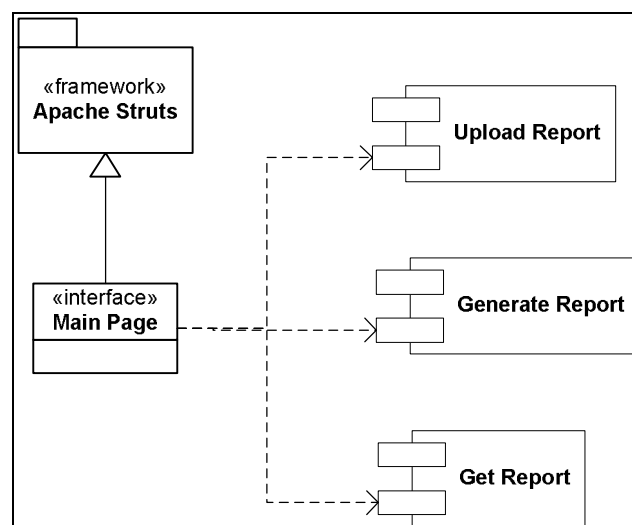


Figure 4.11 : Component Diagram

4.2.8 Deployment Diagram

Client use web browser to access RM Module on Tomcat application server. Each request must go through Apache web server. All the data been stored on MySQL database server and can be retrieve on user request

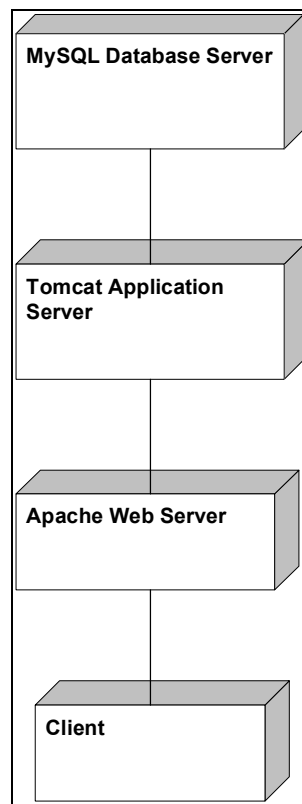


Figure 4.12 : Deployment Diagram

CHAPTER 5

CONCLUSION

This chapter discusses the overall conclusion of the analysis and design phase of the project. The conclusion covers aspect of the project achievement, project constraints, recommendation and lesson learn.

5.1 Project Constraints

There were several constrain faced by the author in order to complete the project. The following sections discuss the details of these constraints.

5.1.1 Lack of Knowledge in Pentaho and JAVA

Due to the lack of depth knowledge and understanding in using the Pentaho itself and also JAVA has causes the author forced to took a longer time than what was been estimated in the project schedule. Thus has effect the design process which also needs to be done by the author.

5.1.2 Lack of knowledge of information security

Since the project is closely related to the latest information security network monitoring it took the author took more time than expected to get the clear view and better understanding in order to proceed with the project. The author is new to the whole security network monitoring jargon including the software and terms that has been used in information security industry.

5.2 Lesson Learnt

There were valuable lessons that the author has learnt and some useful knowledge has been gained throughout the project development. The following section explains these in brief.

5.2.1 Exposure to Open Source Environment

The whole analysis and design during the project development automatically expose the author to open source environment and open source industry. There were various amounts of discussion and task given to the author in order to familiarize the open source environment and open source industry. A lot of comparison and study has been made and has given the author knowledge both in depth and in breadth in understanding the open source concept.

5.2.2 Other Lesson Learnt

Another lesson learnt by the author was the importance of following the whole processes and activity in software development life cycle in order to be able to produce a good quality product. Practicing the good software engineering ethics and principles is one of the major contributions to a successful project.

5.3 Recommendation

There are several recommendations that can be put forward to enhance the current development of the project. The following section will describe briefly on this matter.

5.3.1 Improve the Process flow activity

The author recommends that the implementation of the process flow activity is crucial and needed to be implemented. From the early analysis phases in data and requirement gathering the execution of a better process flow activity will help to save a lot of time cost and energy.

5.3.2 Smaller and Portable Device

As technology is progressively towards a better future, the system should be able to use by other device which is smaller and portable. The author recommends that the development team should integrate the system for suitability in smaller and portable device.

5.3.3 Improvement on Network Environment

Currently there is no server provided to the author for the project development. The machine was also not suitable for the development as accessing the repository is slow. By improving the network facilities it will help to increase the productivity of the team members.

5.4 Summary

RM module is a product that has been developed in order to replace the previous system by using a whole different technology. The author was required to understand and study the existing system before the project development. The author was also required to do a research on the suitability of Pentaho in RM module.

The author has become familiar with a various kind of the new technology in the market and gained a valuable experience during the whole project development. Being exposed to a new environment has broadened the author's mind and better understanding on the current application technology in terms of design, architecture, the methodology that has been used, the development tools and also the deployment tools.

With all the recommendations that have been stated earlier, it is hoped that the RM module will be able to be successfully implemented.

REFERENCES

1. Pentaho Corporation (2008), Product Overview of Pentaho Open Source Business Intelligence, USA:
<http://www.pentaho.com/>
2. Roger S. Pressman, (2001), “Software Engineering – A Practitioner’s Approach”
USA: McGraw Hill
3. MySQLLAB, (2005), “MySQL Language Reference”
USA: MySQL
4. Robert C Martin(1999), “Designing Object Oriented Application Using UML”, 2nd edition
Prentice Hall
5. Douglas Bell (2005), “Software Engineering for Student”
England: Addison – Wesley
6. Ian Sommerville (2001), “Software Engineering”
England: Addison – Wesley
7. University Teknologi Malaysia (2007), “UTM Thesis Manual”
Skudai: Universiti Teknologi Malaysia

APPENDIX A

PROJECT SCHEDULE

ID	Task Name	Start	Finish	Duration	May 2008		Jun 2008				Jul 2008				Aug 2008				Sep 2008					
					5/18	5/25	6/1	6/8	6/15	6/22	6/29	7/6	7/13	7/20	7/27	8/3	8/10	8/17	8/24	8/31	9/7	9/14	9/21	
1	Preparation	5/19/2008	5/30/2008	2w	■																			
2	Analysis (SRS)	5/30/2008	6/27/2008	4.2w	■																			
3	Design (SDD)	6/30/2008	7/31/2008	4.8w	■																			
4	Implimentation & Coding	8/1/2008	8/29/2008	4.2w	■																			
5	Testing (STD)	8/1/2008	9/19/2008	7.2w	■																			
6	Finalization Phase	9/1/2008	9/19/2008	3w	■																			

Gant Chart

APPENDIX B

SEQUENCE DIAGRAM

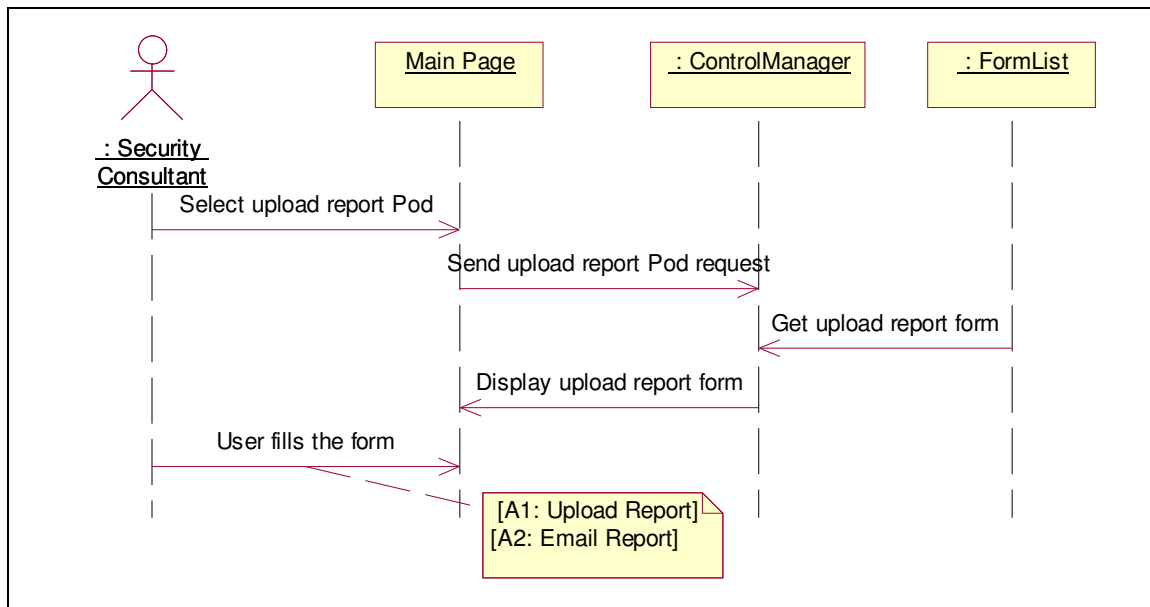


Figure B.13 : Sequence Diagram Upload Report: Basic Flow

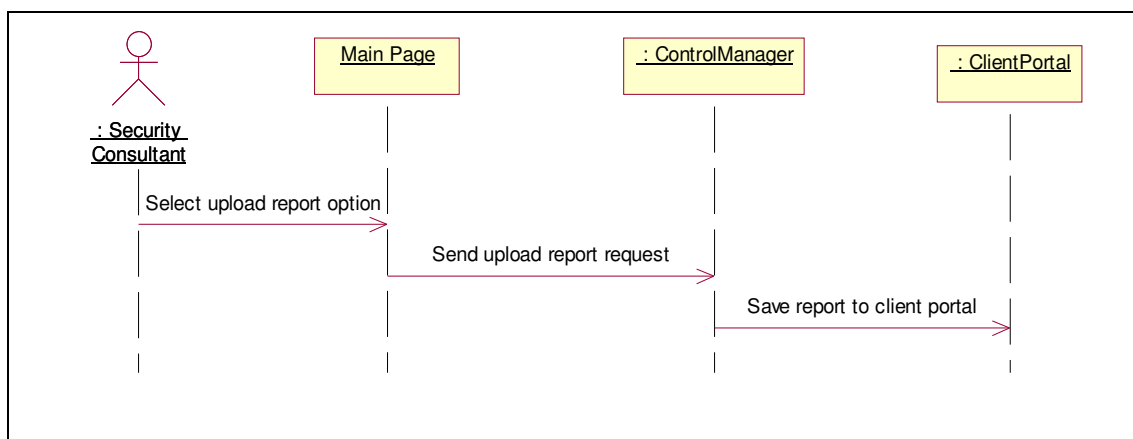


Figure B.14 : Sequence Diagram Upload Report: Alternative Flow [A1: Upload]

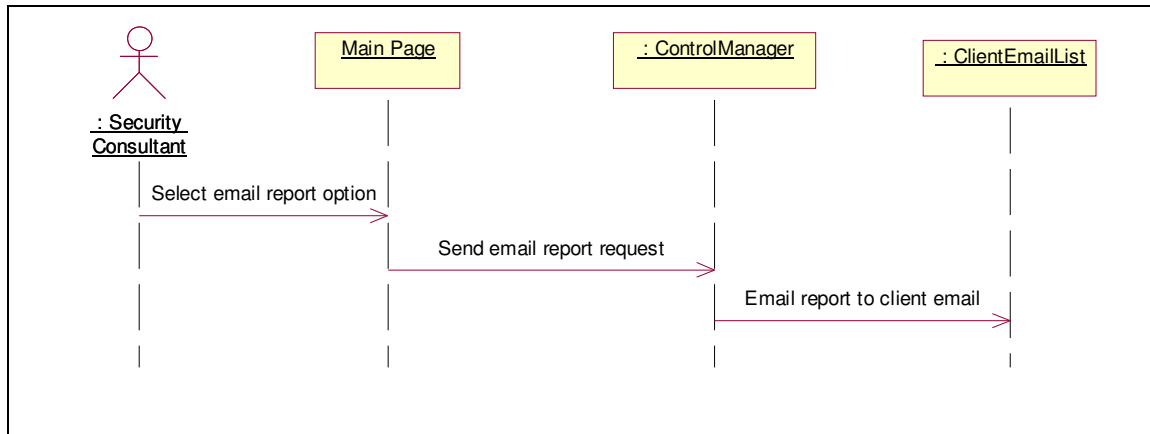


Figure B.15 : Sequence Diagram Upload Report: Alternative Flow [A1: Email]

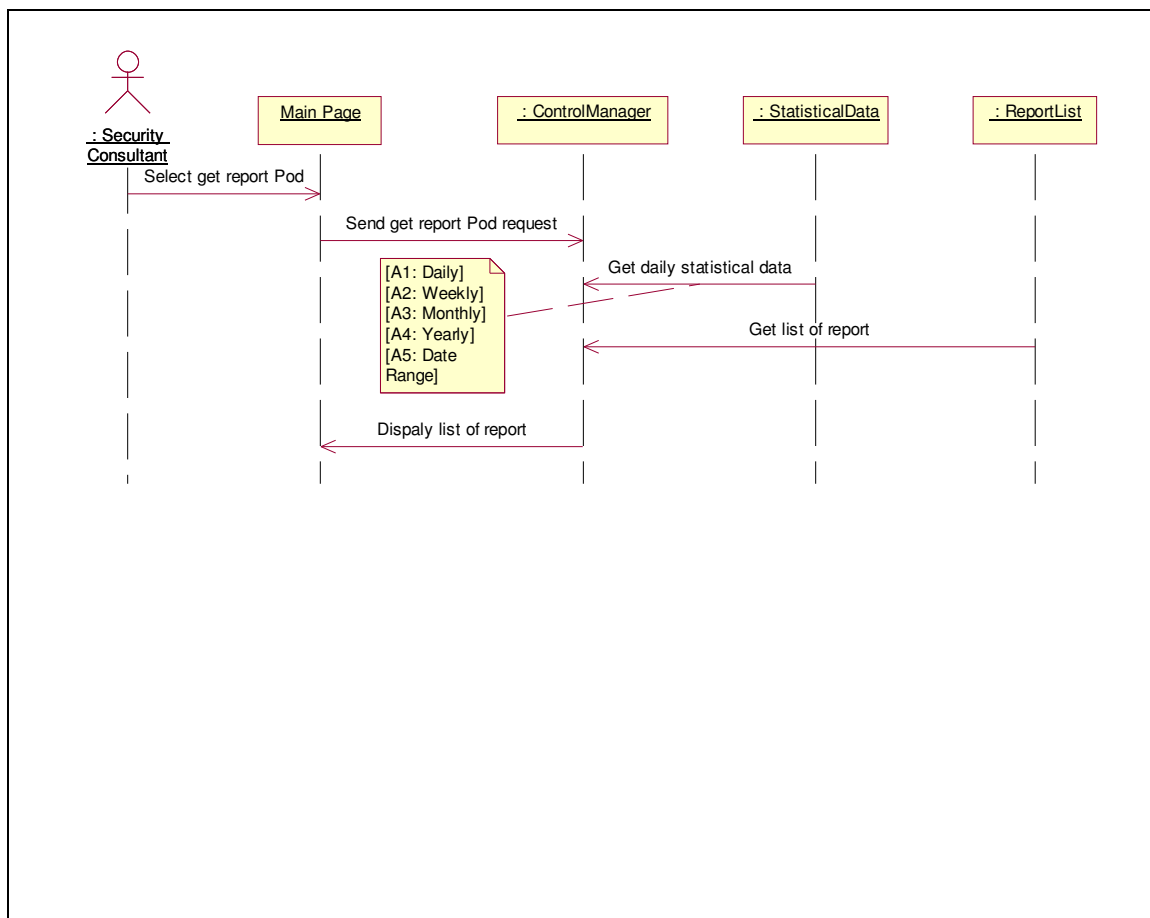


Figure B.16: Sequence Diagram Get Report: Basic Flow

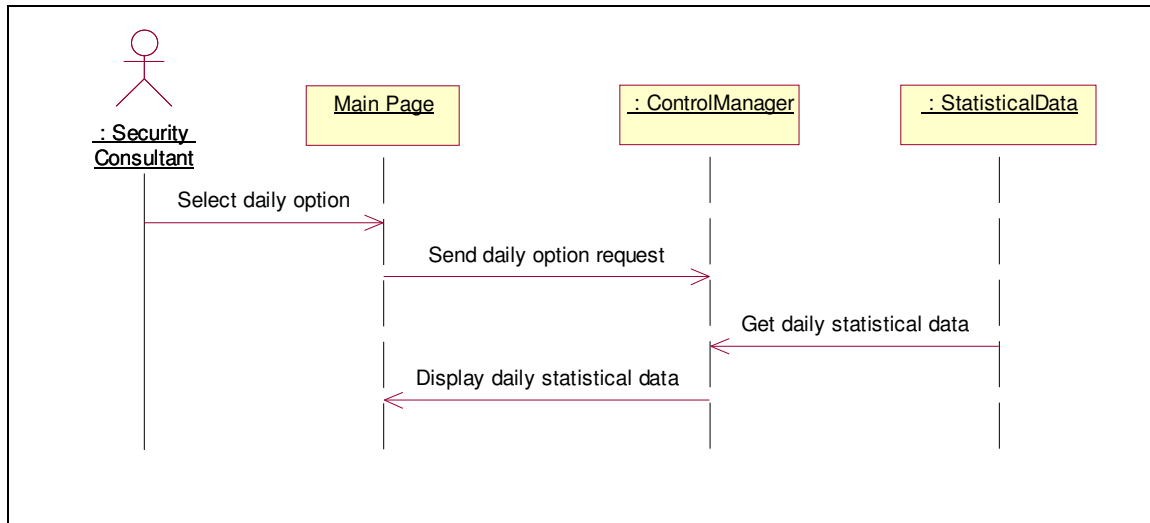


Figure B.17: Sequence Diagram Get Report: Alternative Flow [A1: Daily]

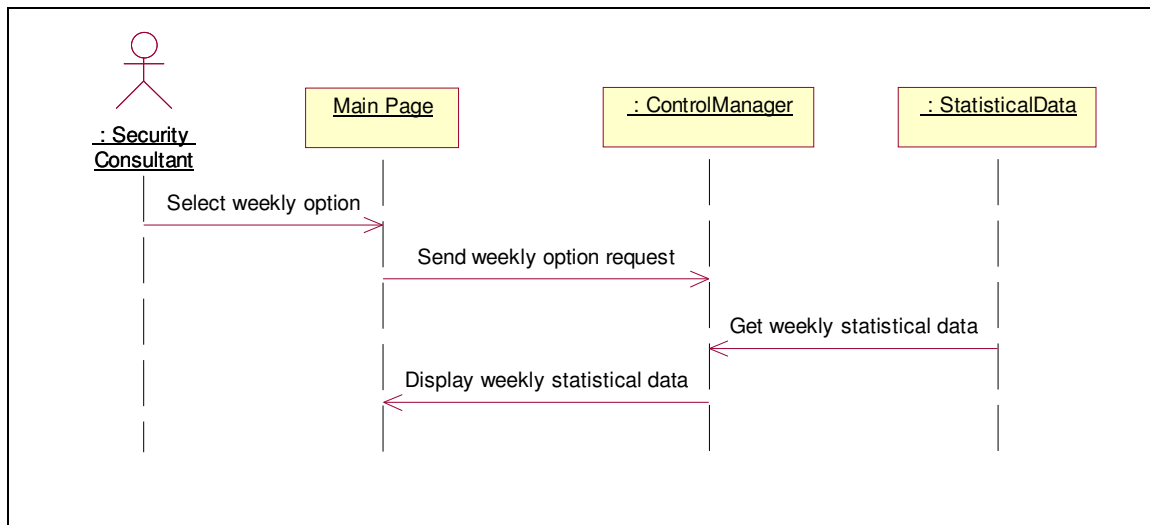


Figure B.18 : Sequence Diagram Get Report: Alternative Flow [A1: Weekly]

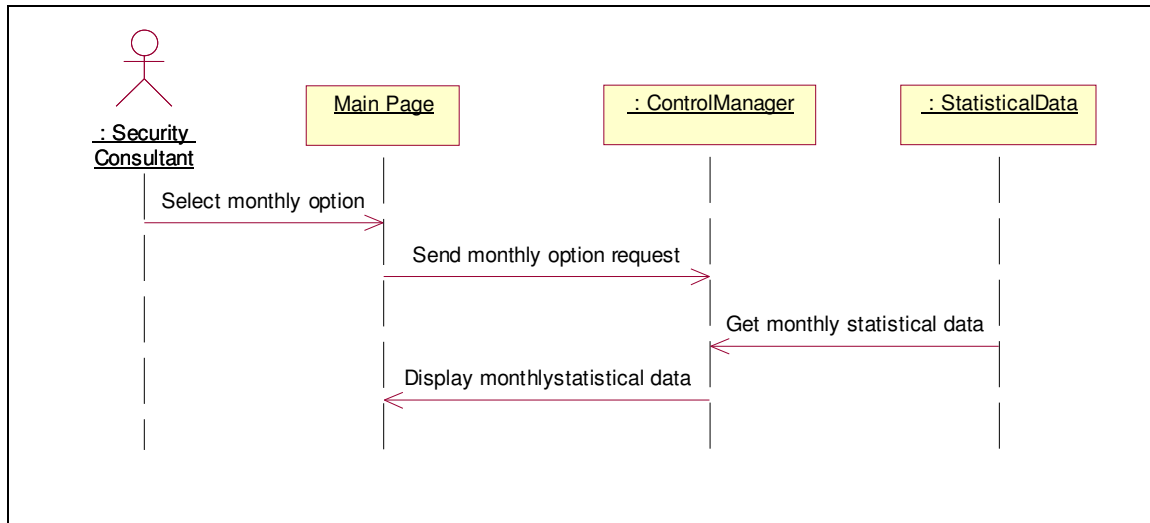


Figure B.19 : Sequence Diagram Get Report: Alternative Flow [A1: Monthly]

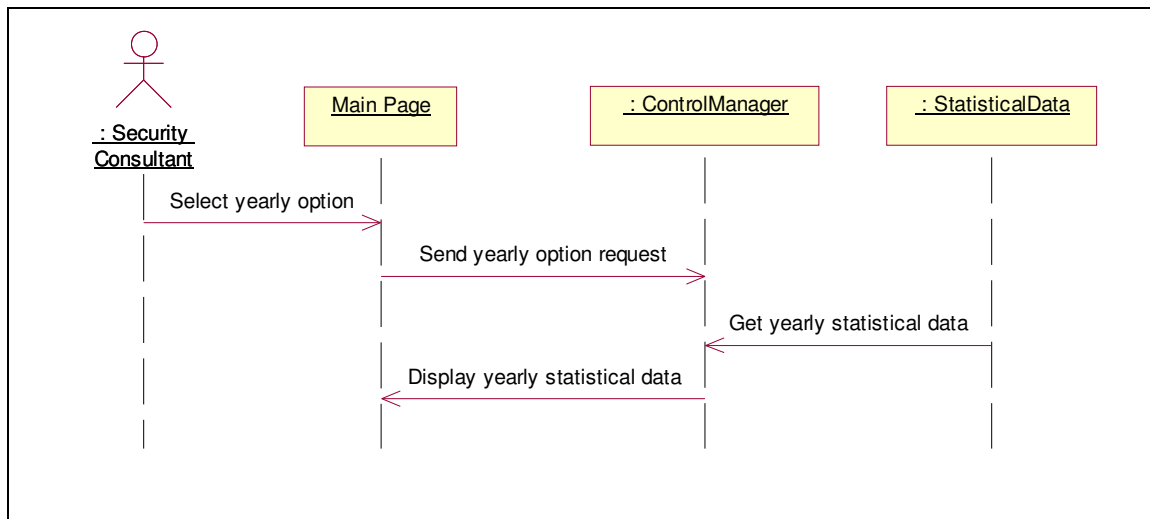


Figure B.20 : Sequence Diagram Get Report: Alternative Flow [A1: Yearly]

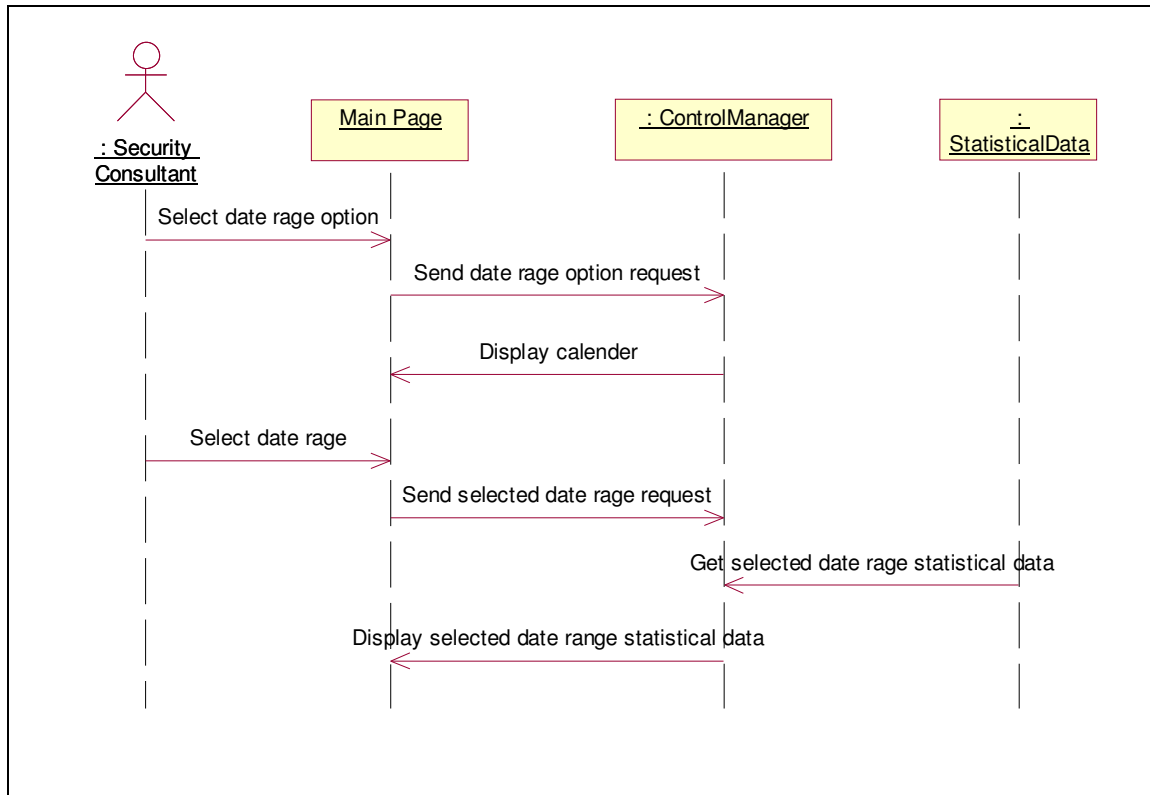


Figure B.21 : Sequence Diagram Get Report: Alternative Flow [A1: Date Range]

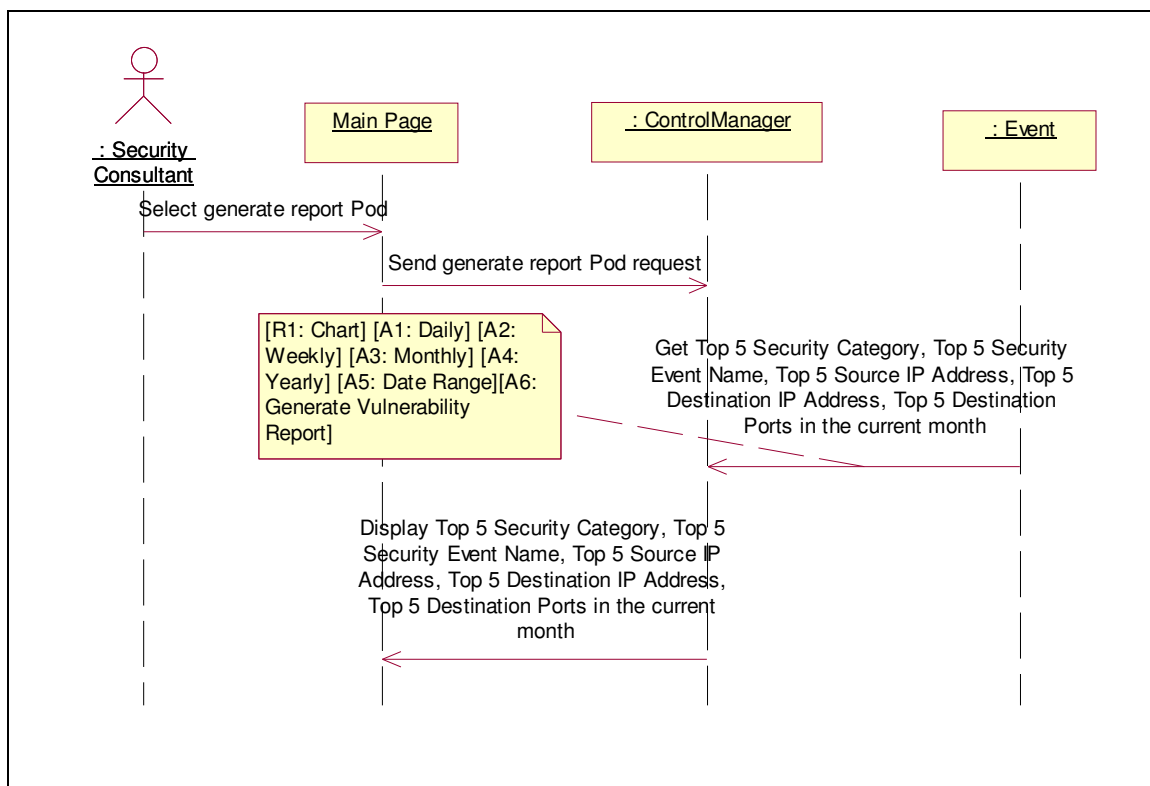


Figure B.22 : Sequence Diagram Generate Report: Basic Flow

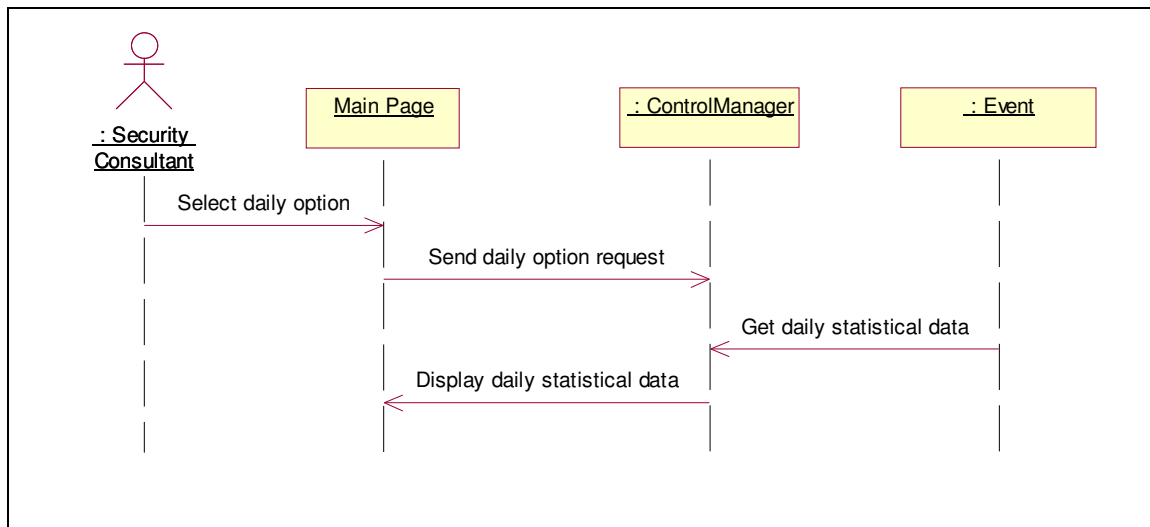


Figure B.23 : Sequence Diagram Get Report: Alternative Flow [A1: Daily]

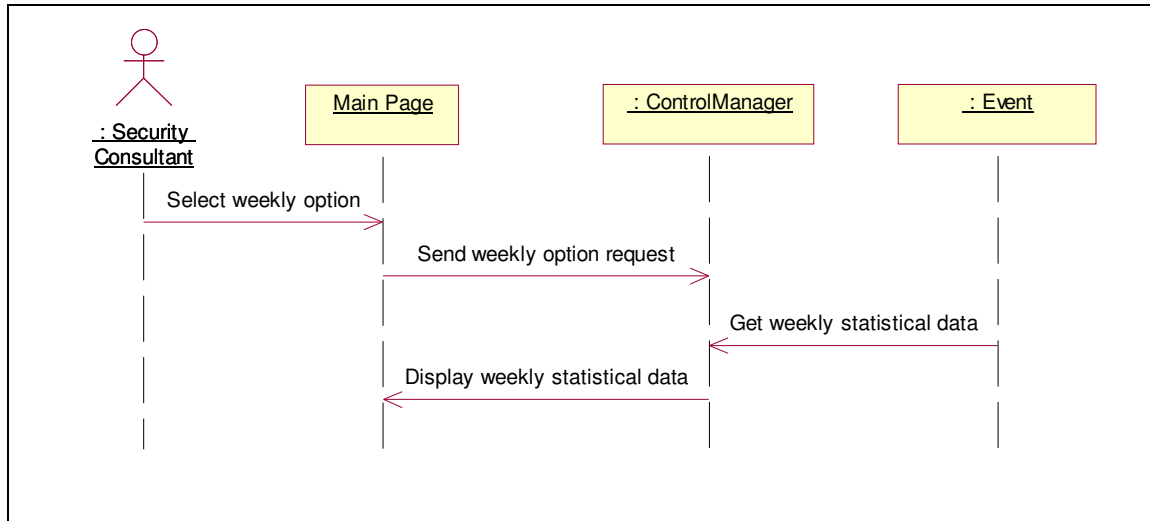


Figure B.24 : Sequence Diagram Get Report: Alternative Flow [A2: Weekly]

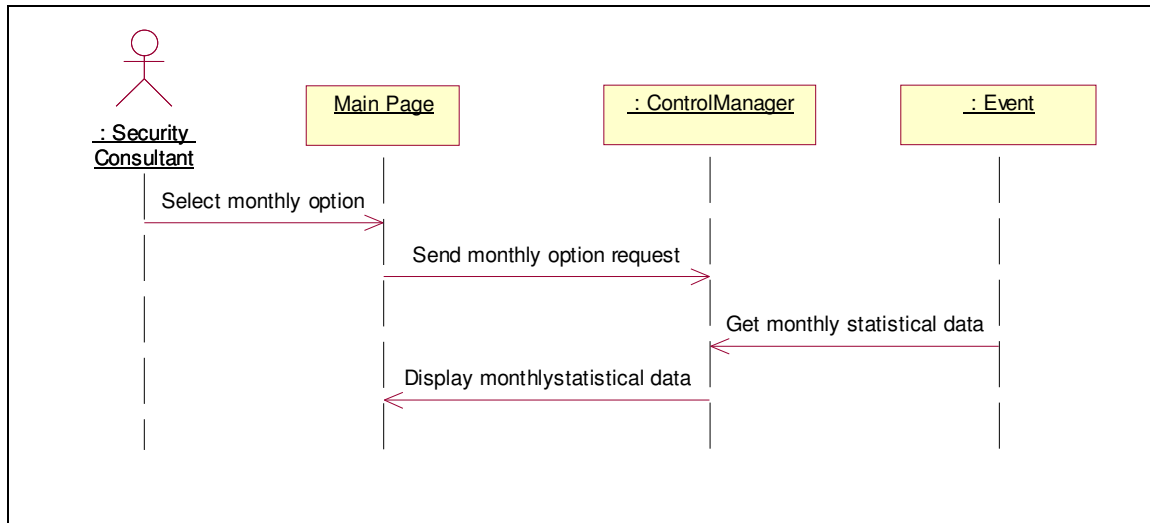


Figure B.25 : Sequence Diagram Get Report: Alternative Flow [A3: Monthly]

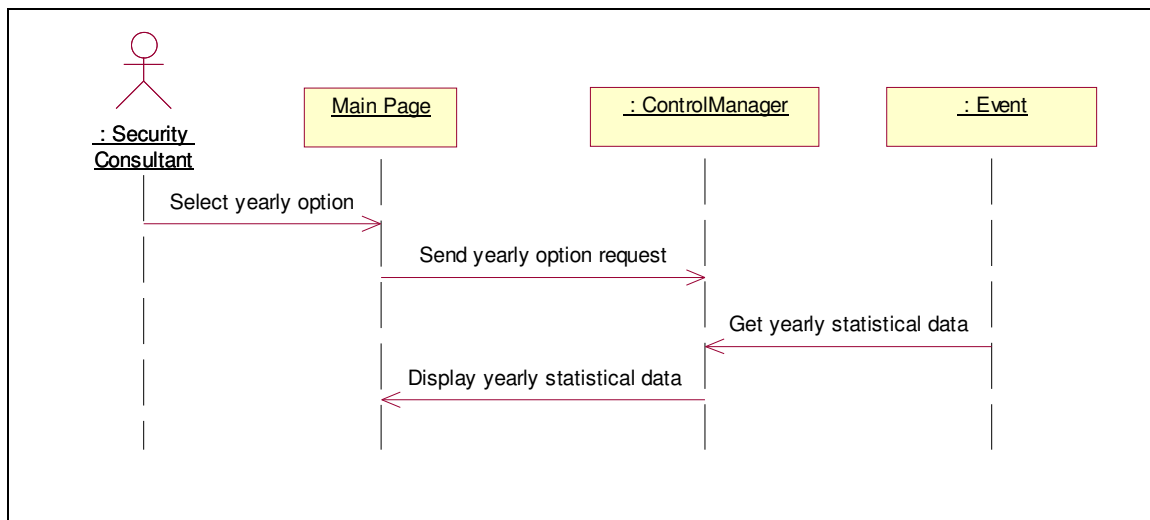


Figure B.26 : Sequence Diagram Get Report: Alternative Flow [A4: Yearly]

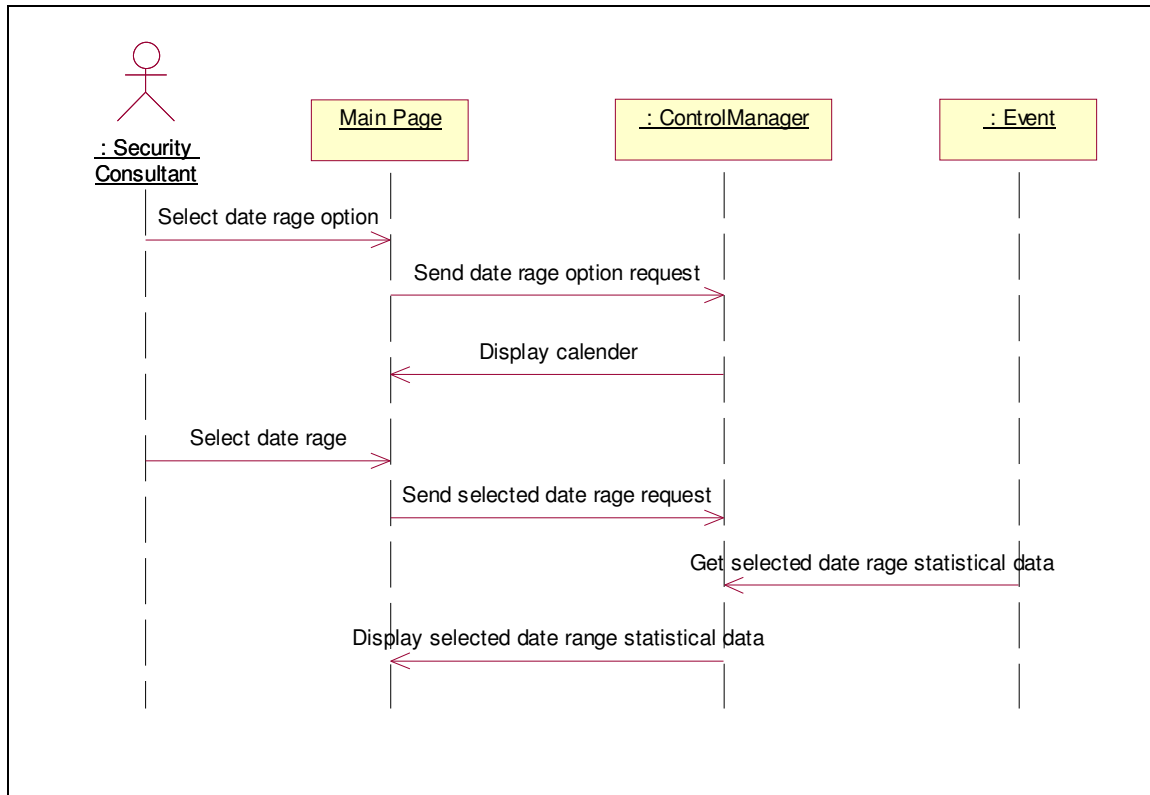


Figure B.27 : Sequence Diagram Get Report: Alternative Flow [A5: Date Range]

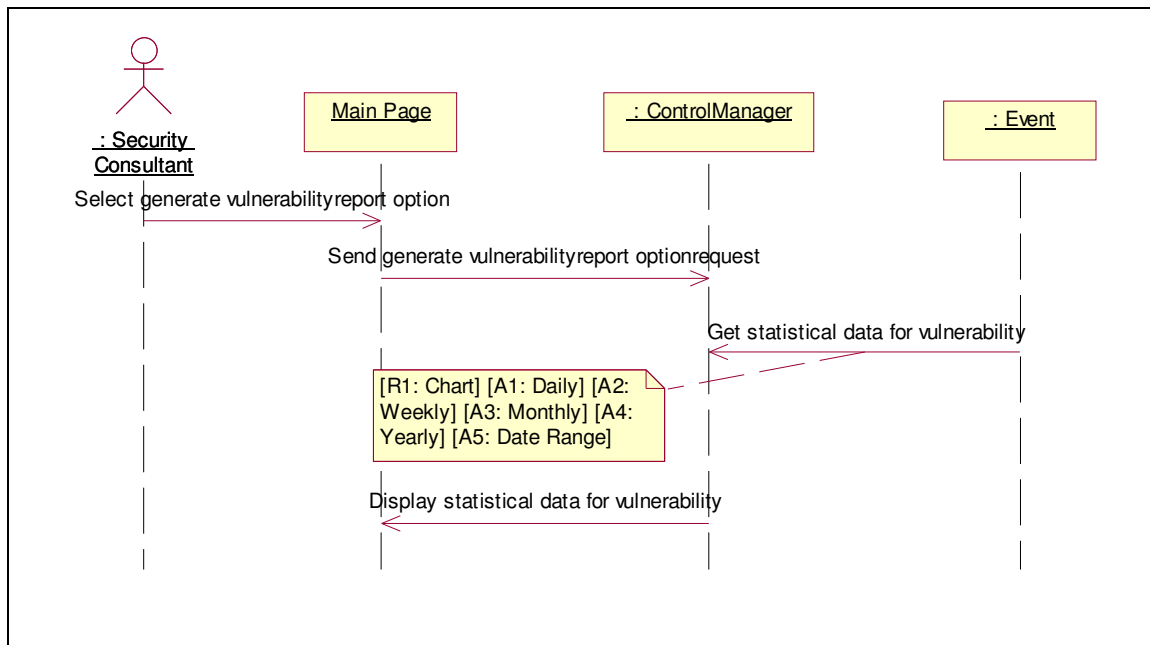
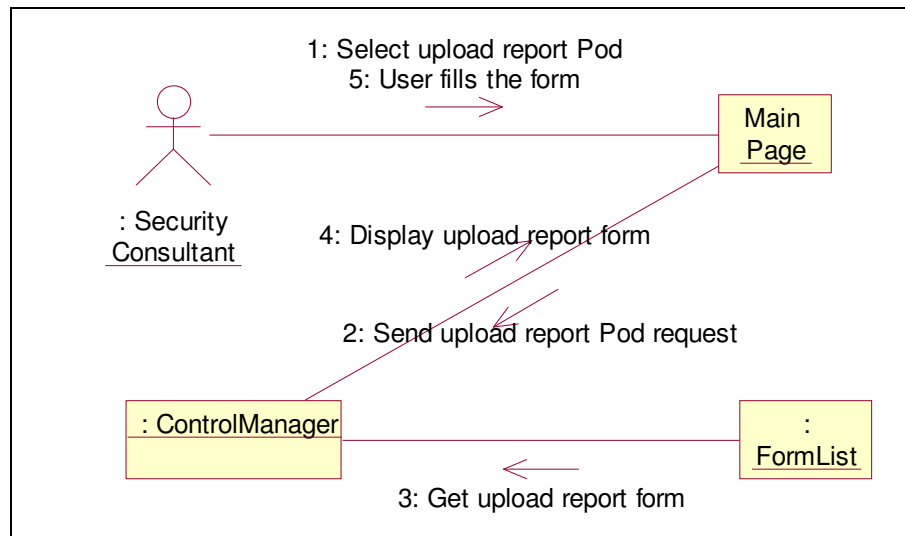


Figure B.28 : Sequence Diagram Get Report: Alternative Flow [A1: Generate Vulnerability Report]

APPENDIX C

COLLABORATION DIAGRAM

**Figure C.1** : Basic Flow: Upload Report

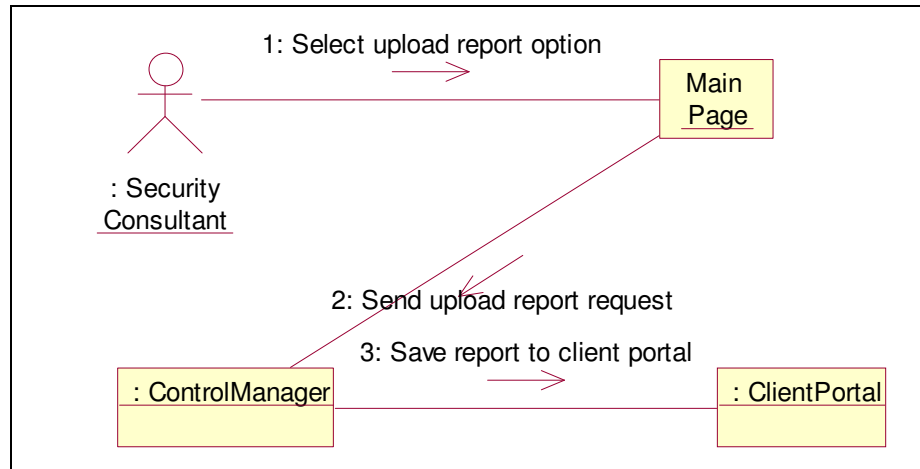


Figure C.2 : Alternative Flow Upload Report: A1 Upload Report

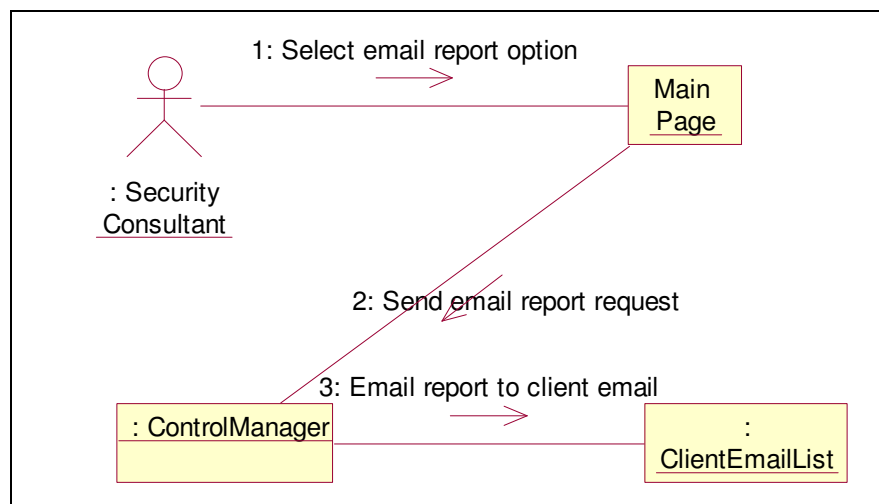


Figure C.3 : Alternative Flow Upload Report: A1 Email Report

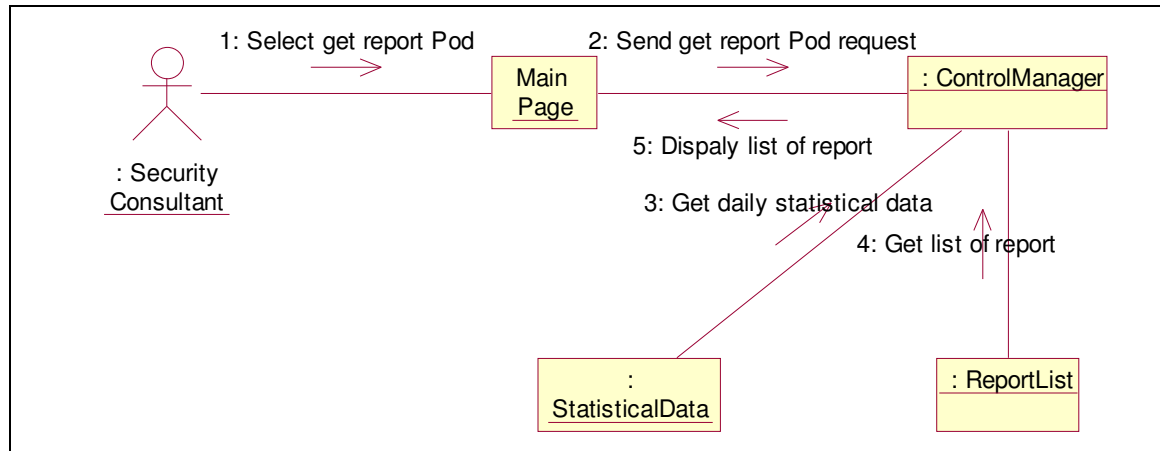


Figure C.4 : Basic Flow: Get Report

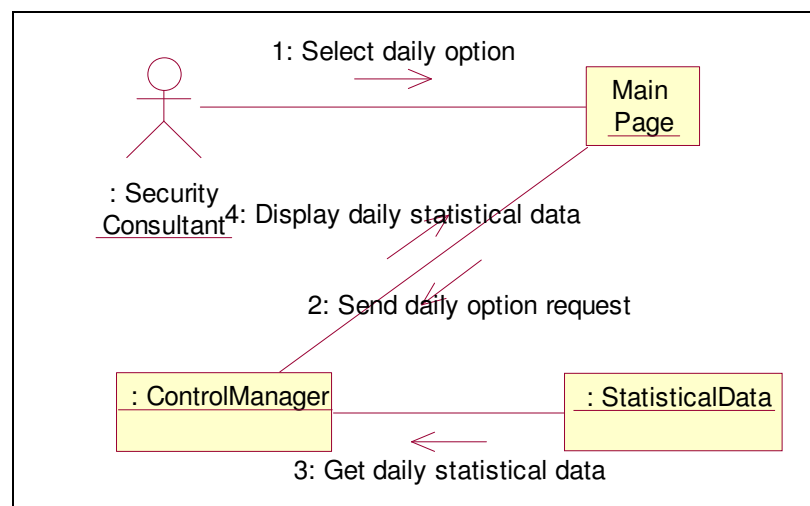


Figure C.5 : Alternative Flow Get Report: A1 Daily Report

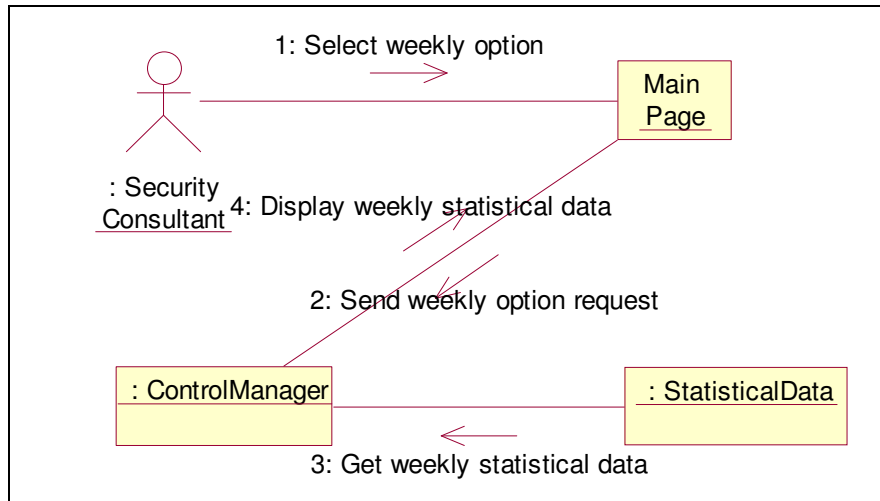


Figure C.6 : Alternative Flow Get Report: A2 Weekly Report

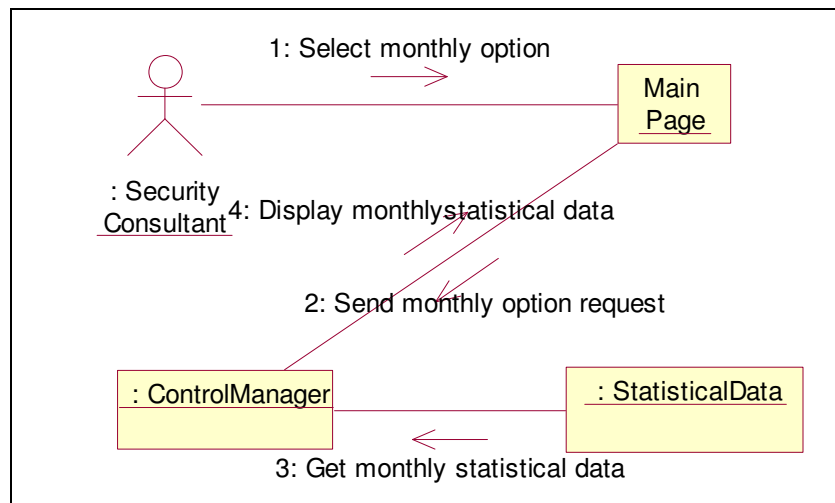


Figure C.7 : Alternative Flow Get Report: A3 Monthly Report

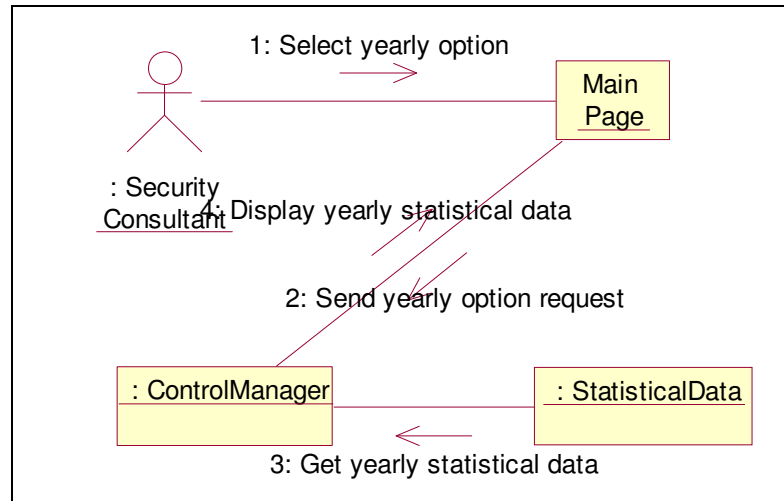


Figure C.8 : Alternative Flow Get Report: A4 Yearly Report

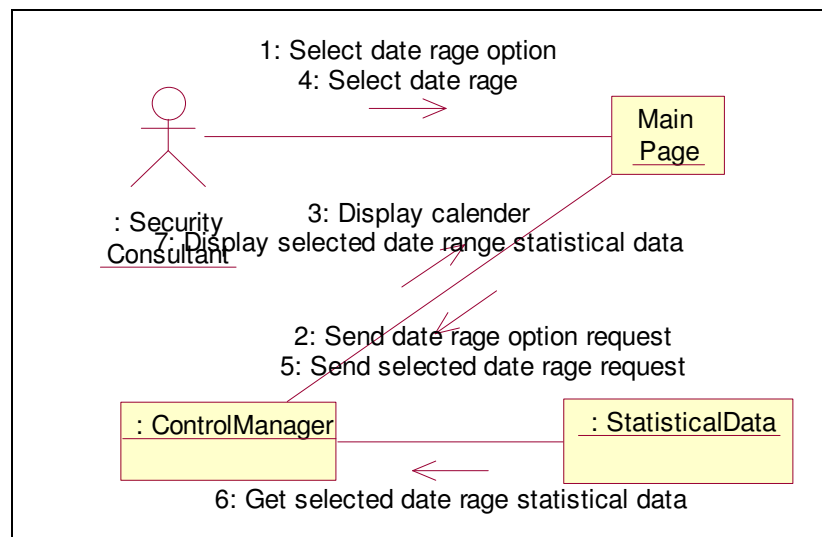


Figure C.9 : Alternative Flow Get Report: A5 Date Range Report

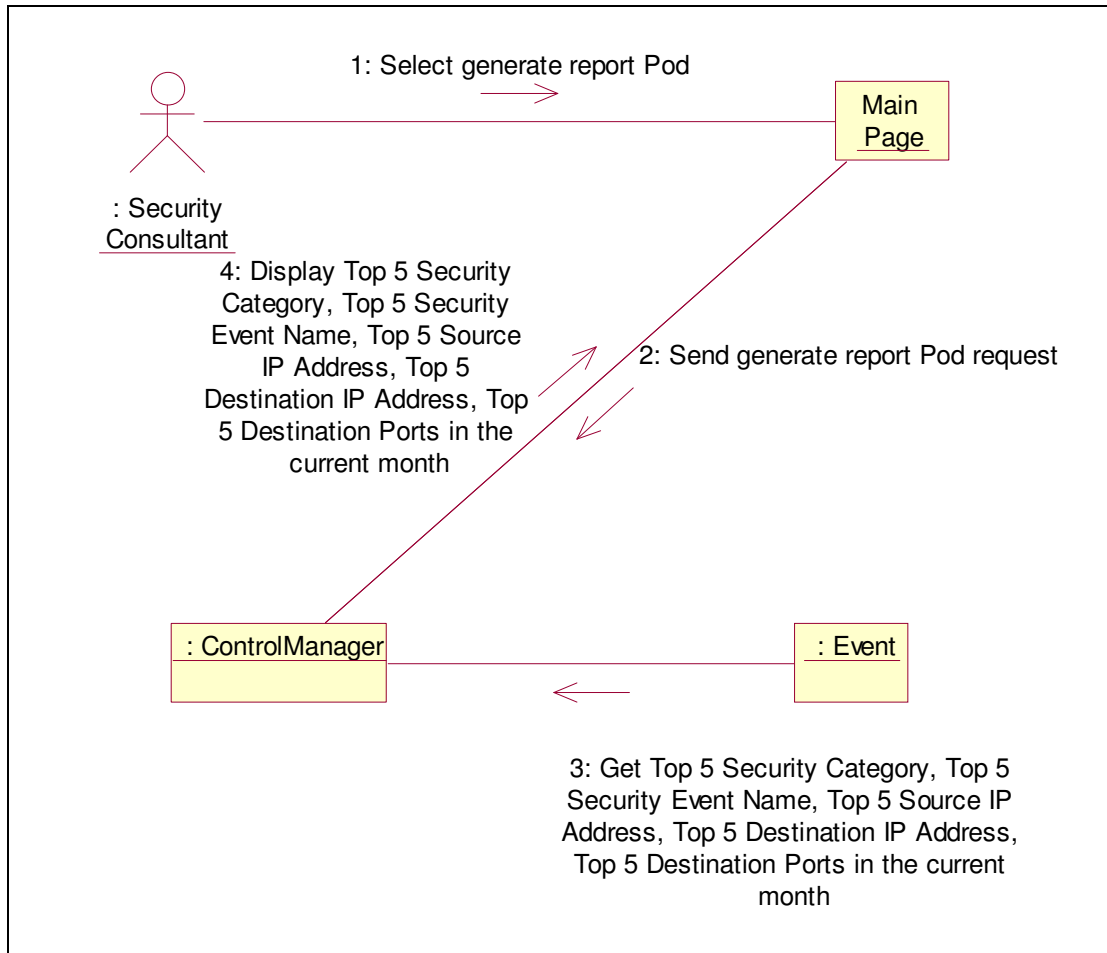


Figure C.10 : Basic Flow: Generate Report

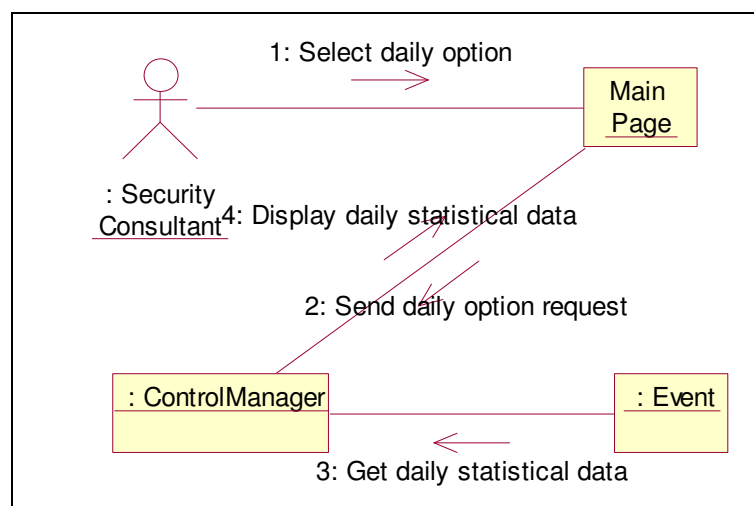


Figure C.11 : Alternative Flow Generate Report: A1 Daily

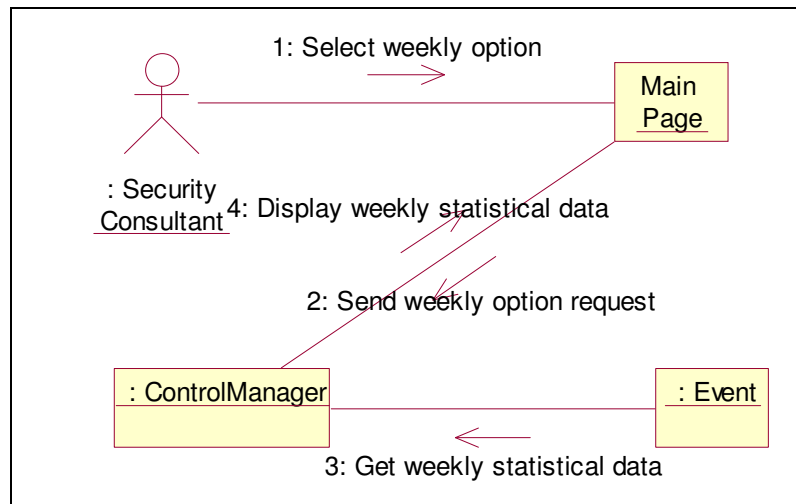


Figure C.12 : Alternative Flow Generate Report: A2 Weekly

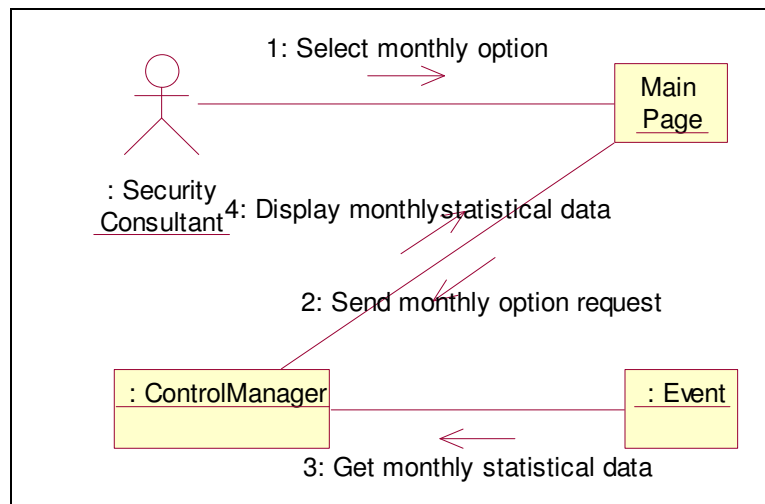


Figure C.13 : Alternative Flow Generate Report: A3 Monthly

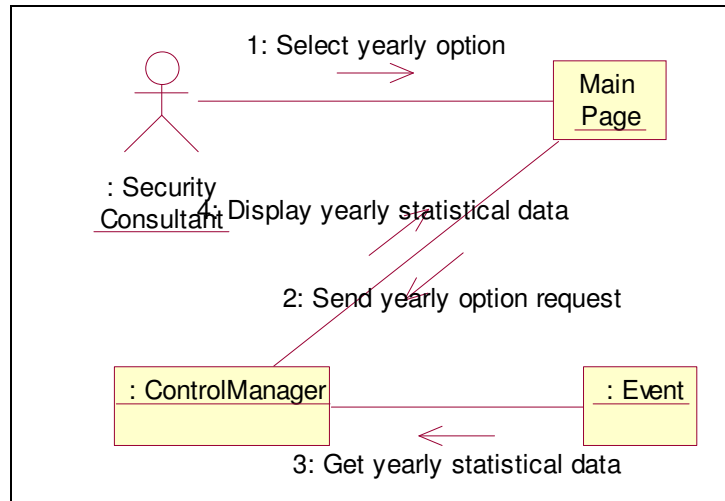


Figure C.14 : Alternative Flow Generate Report: A5 Date Range

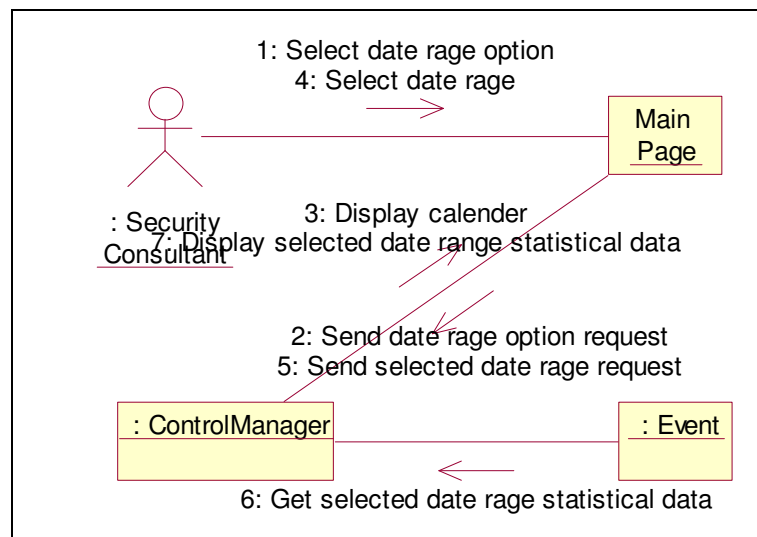


Figure C.15 : Alternative Flow Generate Report: A5 Yearly

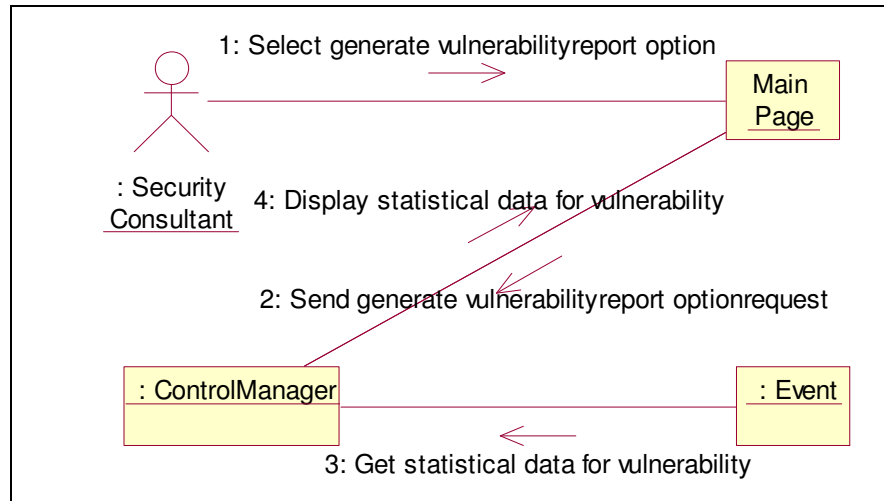


Figure C.16 : Alternative Flow Generate Report: A6 Generate Vulnerability Report

APPENDIX D**SOFTWARE REQUIREMENT SPESIFICATION**

NX

System Requirements Specification
Version V1.0

RESTRICTED

COPYRIGHT

© **SCAN Associates Berhad., Kuala Lumpur, 2008.**

Company Number: 525669-P

All rights reserved. No part of this publication may be reprinted, reproduced, stored in a retrieval system or transmitted, in any form or by any means, without the prior permission in writing from owners.

First published and distributed in June, 2008.

This edition published and distributed in June, 2008.

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SRS/001/D1.0	09.03.01	Restricted	D1.0	05/06/2008	PAGE i of iv

Document Authorization

NX

System Requirements Specification (SRS)

05th June 2008

For and on behalf of SCAN Associates Berhad:

Prepared By :

.....

**Name : Arztrie Delailah Binti Rumi
Designation : Software Engineer
Date :**

Approved By :

.....

**Name : Chong Sau Wei
Designation : Development Manager
Date :**

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SRS/001/D1.0	09.03.01	Restricted	D1.0	05/06/2008	PAGE ii of iv

Revision History

Date	Version	Description	Author

Distribution List:

- Members of NX Software Development Team
- Authorized Personnel of SCAN Associates Berhad

Any person(s) not listed above are not authorized to examine the contents of this document. Doing so would constitute a breach of security and shall be subjected to appropriate actions being taken against the offender.

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SRS/001/D1.0	09.03.01	Restricted	D1.0	05/06/2008	PAGE iii of iv

Table of Contents

1. INTRODUCTION	1
1.1 Purpose	1
1.2 Product Overview	1
1.3 Document Overview	2
2. PRODUCT USER	3
2.1 Reporting Management	3
3. REQUIREMENTS OVERVIEW	4
3.1 External Interfaces Requirements	4
3.1.1 Interface of Security Consultant	4
3.1.2 Interface of Management	5
3.2 Component	5
3.2.1 Reporting Management (SRS_RM_REQ)	6
3.3 Functionality	7
3.3.1 Reporting Management (RM)	7
3.4 Interface Framework	8
4. SOFTWARE REQUIREMENTS	9
4.1 Reporting Management	9
4.1.1 Use Case of Upload Report (SRS_RM_REQ_100)	9
4.1.2 Use Case of Get Report (SRS_RM_REQ_200)	11
4.1.3 Use Case of Generate Report (SRS_RM_REQ_300)	14
5. REFERENCES	17

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SRS/001/D1.0	09.03.01	Restricted	D1.0	05/06/2008	PAGE iv of iv

1. INTRODUCTION

The System Requirements Specification (SRS) focuses on the collection of software and user requirements which surrounding the NX project. In this section, the purpose, product overview and document overview will be detailed out. The System Requirements Specification is used for collecting project software and user requirements in a formal document.

1.1 Purpose

The SRS shall fully describe the external behavior of the application or subsystem of NX. It describes the functional and nonfunctional requirements, design constraints and other factors necessary to provide a complete and comprehensive description of the requirements for the software. This document shall serve as the primary communication medium between the project managers, the client, and other developers, upon which the features of the NX shall be described and implemented.

The following role shall utilize the System Requirements Specification for specific purposes:

1. The **System Analyst** shall create and maintain the SRS, as well as the individual use case and other components of the SRS package.
2. **Designers** shall use the SRS as a reference when defining responsibilities, operations, and attributes on classes, and when adjusting classes to the implementation environment.
3. **Implementers** refer to the SRS for input when implementing classes.
4. The **Project Manager** refers to the SRS for input when planning iterations.
5. **Testers** shall use the SRS to verify system compliance.

1.2 Product Overview

The NX is intended to provide a suite of solutions in order to prevent, detect and respond to the vulnerabilities, threats and events in real time as well as to support future MSS Security Operation Center (SOC) services. To achieve this, there are few modules are included as stated below:

1. Reporting Management (RM) – Business intelligence, analytics and reporting.

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NEXUS	NX /DEV/SRS/001/D1.0	09.03.01	Restricted	D1.0	05/06/2008	PAGE 1 of 15

1.3 Document Overview

This document specifies the requirements to be satisfied by the NX project. The requirements are captured in two (2) main formats: textual description and use-case model.

Textual description is used to capture functional and non-functional aspects, as well as describing the use case models. Use-case models are Unified Modeling Language (UML) notation for capturing the functional requirements of each module for the NX.

- Chapter 1 Describes the purpose, product overview and document overview.
- Chapter 2 Specifies the brief description of the target users of each module for the NX.
- Chapter 3 Specifies the brief descriptions of all use cases and actors, along with applicable diagrams and relationships
- Chapter 4 Specifies the software requirements detail workflows to enable the designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements
- Chapter 5 Referenced document.
- Chapter 6 Contains any general information that aids in understanding this specification.

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NEXUS	NX /DEV/SRS/001/D1.0	09.03.01	Restricted	D1.0	05/06/2008	PAGE 2 of 15

2. PRODUCT USER

This section of the SRS includes brief description of the target users of each module for the NX.

2.1 Reporting Management

The target users of Reporting Management (RM) module are:

1. Security Analyst
2. Senior Security Analyst
3. Security Expert
4. Management

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NEXUS	NX /DEV/SRS/001/D1.0	09.03.01	Restricted	D1.0	05/06/2008	PAGE 3 of 15

3. REQUIREMENTS OVERVIEW

This section divided into the following paragraphs which contain the use case diagrams of each module for the NX. This includes a list of names and brief descriptions of all use cases and actors, along with applicable diagrams and relationships.

3.1 External Interfaces Requirements

This section describes the external interfaces of NX as depicted in Figure 3-1.

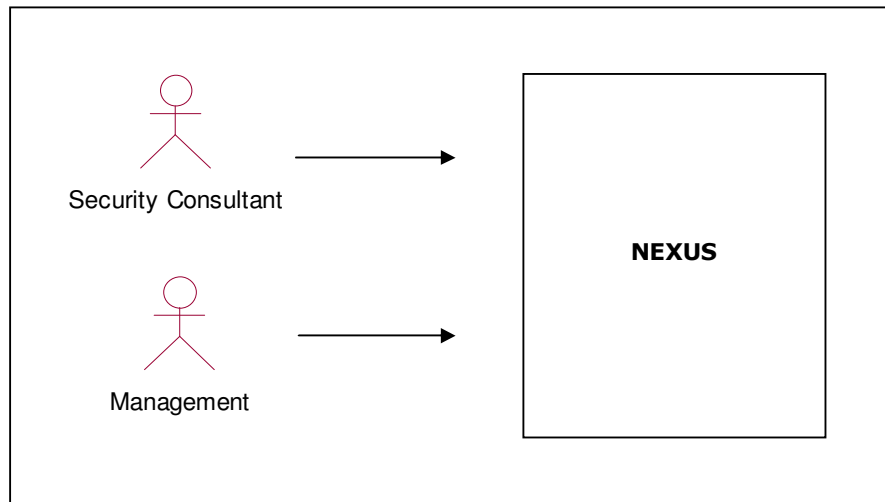


Figure 3-1: Use Case External Interface Diagram

3.1.1 Interface of Security Consultant

Interface Identification : Security Consultant

Interface Type : Person

Description:

The Security Consultant an abstract actor who represents the Security Analyst, Senior Security Analyst and Security Expert. The Security Consultant is able to monitor the security events and responds to the security alerts as well as view the reports.

Association:

This actor communicates with the Threat Management module and Reporting Management module.

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NEXUS	NX /DEV/SRS/001/D1.0	09.03.01	Restricted	D1.0	05/06/2008	PAGE 4 of 15

3.1.2 Interface of Management

Interface Identification : Management

Interface Type : Person

Description:

The Management is an abstract actor who represents the Head of Depart, Project Manager and Team Leader. The Management is able to view and manage the reports.

Association:

This actor communicates with the Reporting Management module.

3.2 Component

This section describes all the components or modules for NX development. There are two main components which are Threat Management and Reporting Management, as shown in Figure 3-2. The descriptions of each component are described in following sections.

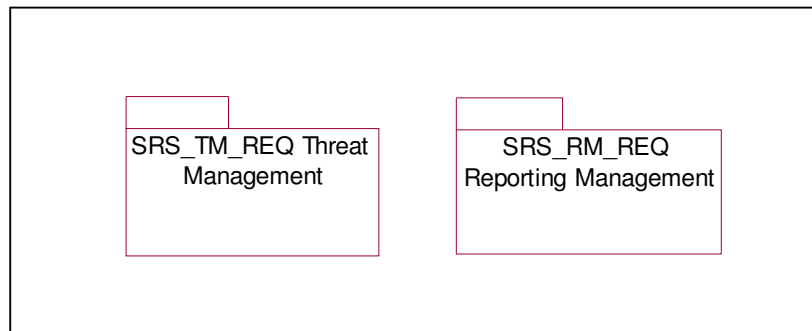


Figure 3-2: Main Components

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NEXUS	NX /DEV/SRS/001/D1.0	09.03.01	Restricted	D1.0	05/06/2008	PAGE 5 of 15

3.2.1 Reporting Management (SRS_RM_REQ)

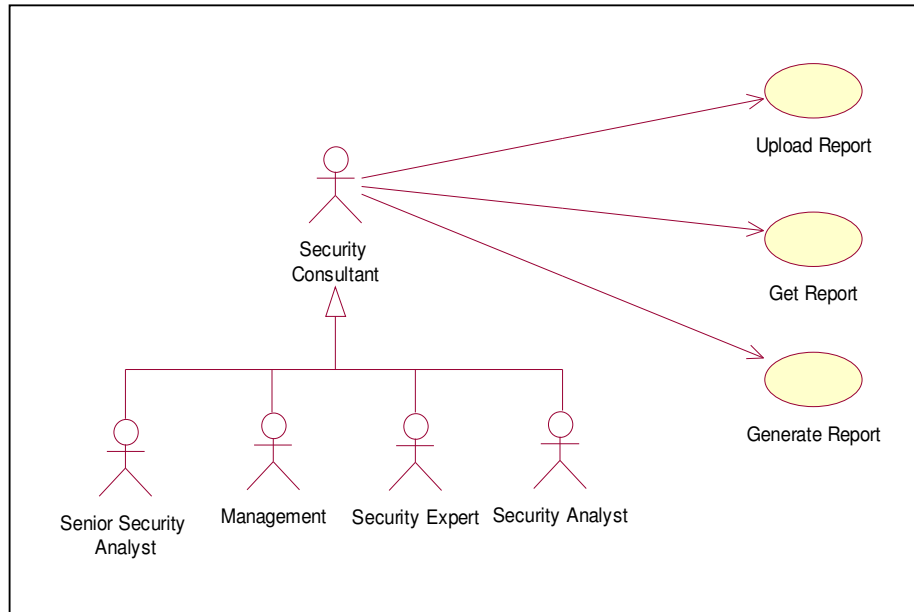


Figure 3-3: Use Case Diagram for Reporting Management

Description: Reporting Management will provide a mechanism for NX users to generate, view and publish reports. These reports can be configurable specific to the component such as TM, and can be categorized according to contents, such as technical report or management report.

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NEXUS	NX /DEV/SRS/001/D1.0	09.03.01	Restricted	D1.0	05/06/2008	PAGE 6 of 15

3.3 Functionality

This section describes the functional requirements of each component for the NX. In general, the system response time is not more than five seconds.

3.3.1 Reporting Management (RM)

- The reporting mechanism shall be flexible, configurable and customizable by users.
- The system shall be able to generate by default - daily, weekly and monthly technical and management reports, as well as other scheduled automatic report generation.
- The system shall be able to generate graphical reports and provides pre-defined report templates.
- The system shall have the capability to export reports to other types of user readable documents such as WORD, EXCEL or PDF.
- The system shall support archiving of all reports.
- Security Features
- All access to the system, remotely and locally, except to the publicly accessible web site of GSWP, shall require authorizations as defined and enforced by access control mechanism.
- All communication channels between components residing at the sites and the centre shall be secured in terms of authentication, confidentiality and integrity of information.
- All components of the system shall be installed on a secured and hardened operating system platform.

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NEXUS	NX /DEV/SRS/001/D1.0	09.03.01	Restricted	D1.0	05/06/2008	PAGE 7 of 15

3.4 Interface Framework

This section describes the main interface design of the NX. Figure 3-5 depicts the overall layout design with the tabs and pods. Each pod can be minimize and maximize.

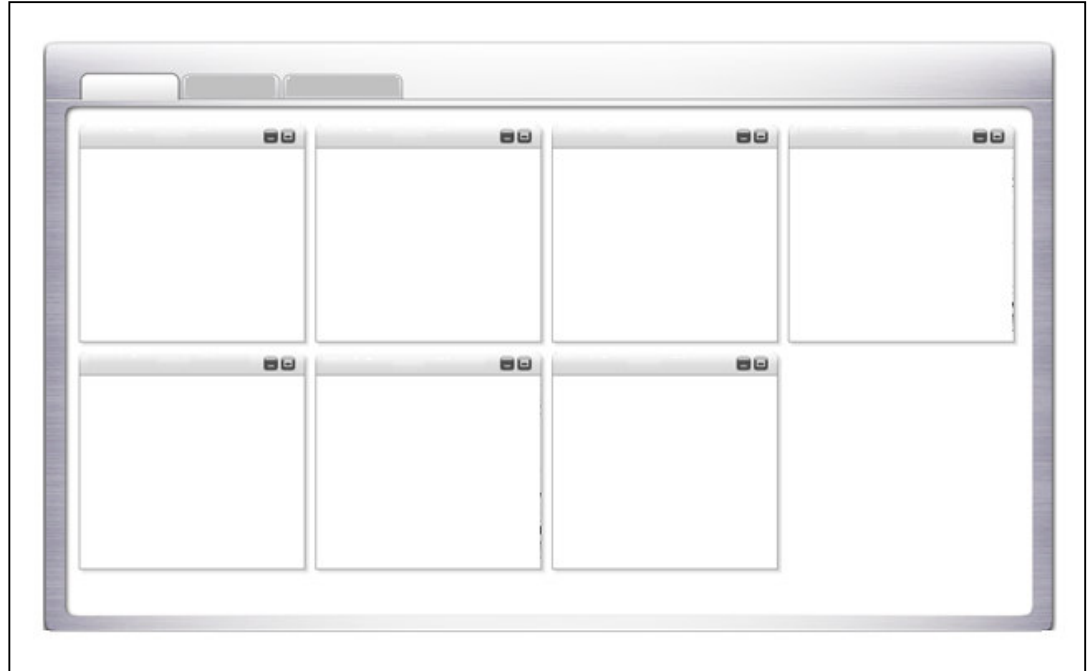


Figure 3-4: Main Interface Design Specification

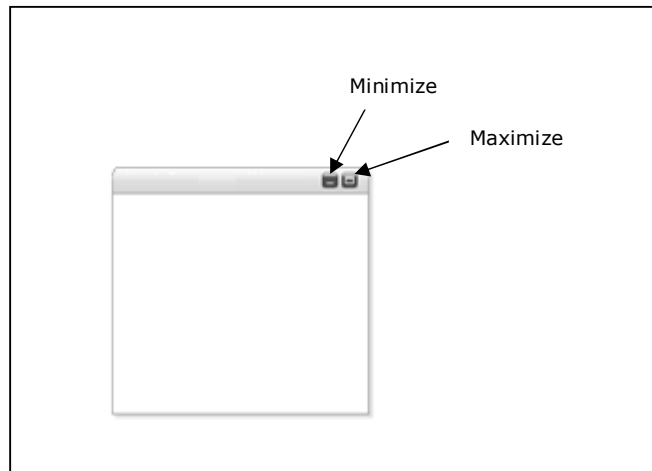


Figure 3-5: Pod

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NEXUS	NX /DEV/SRS/001/D1.0	09.03.01	Restricted	D1.0	05/06/2008	PAGE 8 of 15

4. SOFTWARE REQUIREMENTS

This section describes all the software requirements detail workflows to enable the designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. The majority of these requirements are captured in the use cases using use-case modeling approach.

4.1 Reporting Management

This section describes the requirements for Reporting Management (RM) module. There are three use cases in RM which are Upload Report, Get Report and Generate Report. Refer to section 3.2.2.

4.1.1 Use Case of Upload Report (SRS_RM_REQ_100)

This section describes the process of Upload Report use case.

4.1.1.1 Brief Description

This use case is initiated by the Security Consultant. It provides the capability for Security Consultant to upload report to customer.

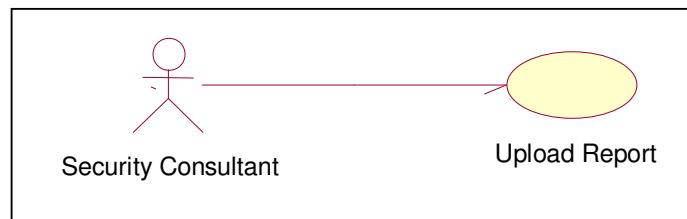


Figure 4-1: Upload Report Data Use Case

4.1.1.2 Characteristic of Activation

This event is driven by Security Consultant.

4.1.1.3 Pre-Condition(s)

The user is log into the system successfully.

4.1.1.4 Description

This sub paragraph shall describe the basic flow, alternative flow and exception flow of the View Statistical Data use case.

4.1.1.4.1 Basic Flow

1. The use case begins when the Security Consultant selects a pod to upload report.
2. The system provides the Upload Report form.

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NEXUS	NX /DEV/SRS/001/D1.0	09.03.01	Restricted	D1.0	05/06/2008	PAGE 9 of 15

3. The user fills the form **[A1: Upload] [A2: Email]**.
4. The use case ends.

4.1.1.4.2 Alternative Flow(s)

A1: Upload

1. The user selects an Upload option.
2. The system uploads the selected report to client portal.

A2: Email

1. The user selects an Email option.
2. The system emails the selected report to client email.

4.1.1.4.3 Exception Flow(s)

Not applicable.

4.1.1.5 Post Condition(s)

Not applicable.

4.1.1.6 Rule(s) and Constraint(s)

Not applicable.

4.1.1.7 Interface Design Specification

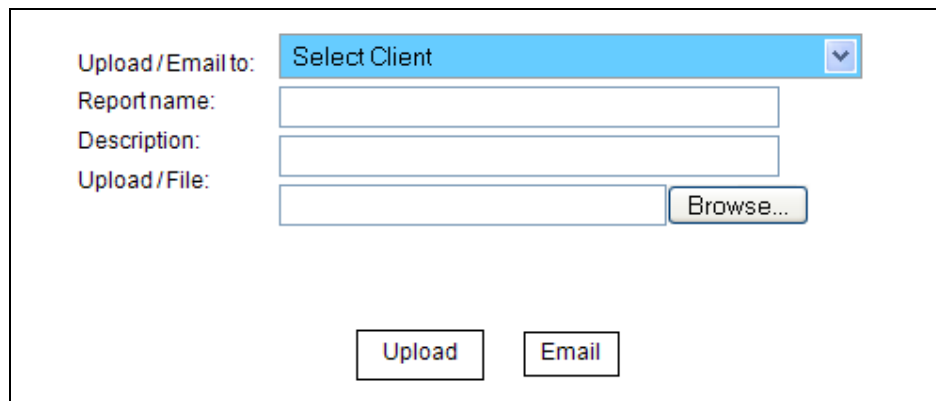


Figure 4-2: Upload Report

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NEXUS	NX /DEV/SRS/001/D1.0	09.03.01	Restricted	D1.0	05/06/2008	PAGE 10 of 15

4.1.2 Use Case of Get Report (SRS_RM_REQ_200)

This section describes the process of Get Report use case.

4.1.2.1 Brief Description

This use case is initiated by the Security Consultant. It provides the capability for Security Consultant to get report.

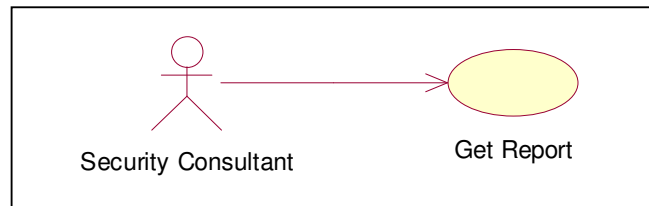


Figure 4-3: Get Report Data Use Case

4.1.2.2 Characteristic of Activation

This event is driven by Security Consultant.

4.1.2.3 Pre-Condition(s)

The user is log into the system successfully.

4.1.2.4 Description

This sub paragraph shall describe the basic flow, alternative flow and exception flow of the View Statistical Data use case.

4.1.2.4.1 Basic Flow

1. The use case begins when the Security Consultant selects a pod to get report [**R1: Chart**].
2. The system provides daily statistical data which according to the number of events based on status which is open, new and resolved [**A1: Daily**] [**A2: Weekly**] [**A3: Monthly**] [**A4: Yearly**] [**A5: Date Range**].
3. The system also provides the list of report.
4. The use case ends.

4.1.2.4.2 Alternative Flow(s)

A1: Daily

1. The user selects a daily option.
2. The system provides the statistical data in the current date.

A2: Weekly

1. The user selects a weekly option.
2. The system provides the statistical data in the current week.

A3: Monthly

1. The user selects a monthly option.
2. The system provides the statistical data in current month.

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NEXUS	NX /DEV/SRS/001/D1.0	09.03.01	Restricted	D1.0	05/06/2008	PAGE 11 of 15

A4: Yearly

1. The user selects a Yearly option.
2. The system provides the statistical data in current Year.

A5: Date Range

1. The user selects a Date Range option.
2. The user input the selected range of date.
3. The system provides the statistical data in between selected range of date.

4.1.2.4.3 Exception Flow(s)

Not applicable.

4.1.2.5 Post Condition(s)

Not applicable.

4.1.2.6 Rule(s) and Constraint(s)

R1: Chart

The system displays the statistical data in Bar Chart.

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NEXUS	NX /DEV/SRS/001/D1.0	09.03.01	Restricted	D1.0	05/06/2008	PAGE 12 of 15

4.1.2.7 Interface Design Specification

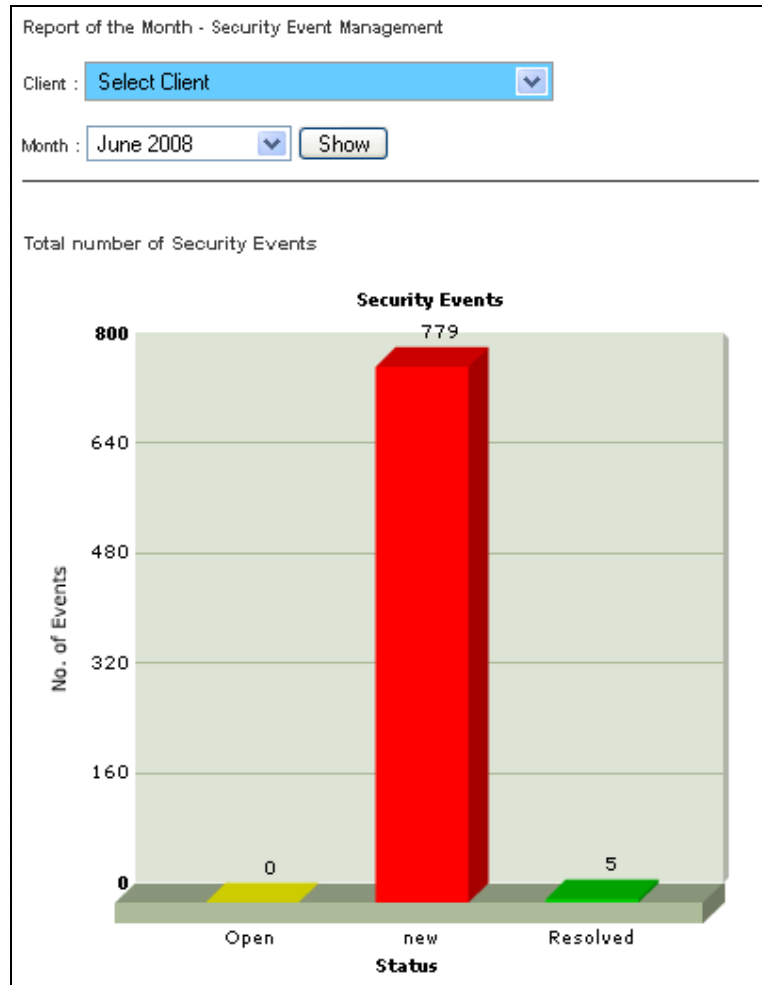


Figure 4-4: Report of the month

***Note:** The above chart type and interface design may revise depends on the Designers.

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NEXUS	NX /DEV/SRS/001/D1.0	09.03.01	Restricted	D1.0	05/06/2008	PAGE 13 of 15

4.1.3 Use Case of Generate Report (SRS_RM_REQ_300)

This section describes the process of Generate Report use case.

4.1.3.1 Brief Description

This use case is initiated by the Security Consultant. It provides the capability for Security Consultant to generate report.



Figure 4-5: Generate Report Use Case

4.1.3.2 Characteristic of Activation

This event is driven by Security Consultant.

4.1.3.3 Pre-Condition(s)

The user is log into the system successfully.

4.1.3.4 Description

This sub paragraph shall describe the basic flow, alternative flow and exception flow of the View Statistical Data use case.

4.1.3.4.1 Basic Flow

1. The use case begins when the Security Consultant selects a pod to generate Report.
2. The system provides the Top 5 Security Category, Top 5 Security Event Name, Top 5 Source IP Address, Top 5 Destination IP Address, Top 5 Destination Ports in the current month[**R1: Chart**] [**A1: Daily**] [**A2: Weekly**] [**A3: Monthly**] [**A4: Yearly**] [**A5: Date Range**][**A6: Generate Vulnerability Report**]
3. The use case ends.

4.1.3.4.2 Alternative Flow(s)

A1: Daily

1. The user selects a daily option.
2. The system provides the statistical data in the current date.

A2: Weekly

1. The user selects a weekly option.
2. The system provides the statistical data in the current week.

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NEXUS	NX /DEV/SRS/001/D1.0	09.03.01	Restricted	D1.0	05/06/2008	PAGE 14 of 15

A3: Monthly

1. The user selects a monthly option.
2. The system provides the statistical data in current month.

A4: Yearly

1. The user selects a Yearly option.
2. The system provides the statistical data in current Year.

A5: Date Range

1. The user selects a Date Range option.
2. The user input the selected range of date.
3. The system provides the statistical data in between selected range of date.

A6: Generate Vulnerability Report

1. The user selects the generate vulnerability report option **[A1: Daily] [A2: Weekly] [A3: Monthly] [A4: Yearly] [A5: Date Range]**.
2. The system provides the statistical data for the vulnerability.

4.1.3.4.3 Exception Flow(s)

Not applicable.

4.1.3.5 Post Condition(s)

Not applicable.

4.1.3.6 Rule(s) and Constraint(s)

R1: Chart

The system displays the statistical data in Bar Chart.

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NEXUS	NX /DEV/SRS/001/D1.0	09.03.01	Restricted	D1.0	05/06/2008	PAGE 15 of 15

4.1.3.7 Interface Design Specification

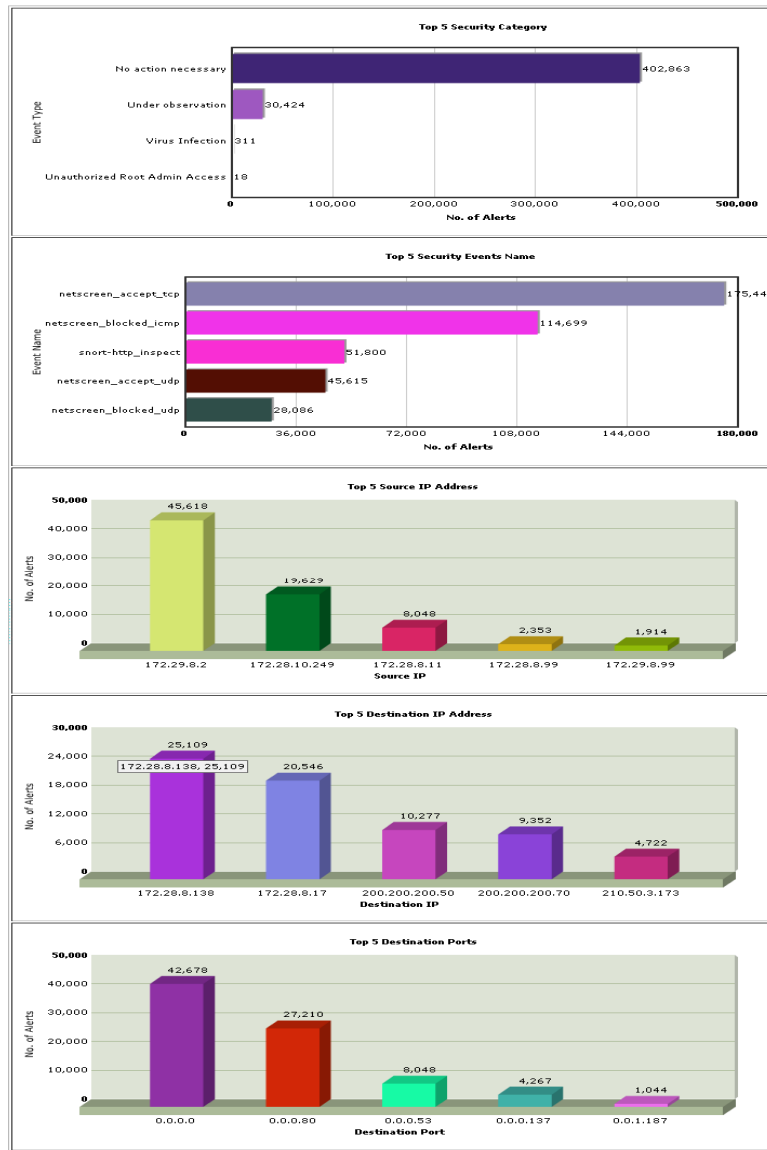


Figure 4-6: Top 5 Report

Client :

Month :

Figure 4-7: Generate Vulnerability Report

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NEXUS	NX /DEV/SRS/001/D1.0	09.03.01	Restricted	D1.0	05/06/2008	PAGE 16 of 16

5. REFERENCES

The following table outlines all documents that are referenced in the SRS.

No	Document Title	Document ID	Ver.	Reference
1.	GSCC Software Development Plan	GSCC/DEV/PLN/001/D1.0	1.0	External
2.	GSCC – Glossary	GSCC/DEV/PLN/002/D1.0	1.0	External
3.	Rational Unified Process	http://www.rational.com		Web
4.	Requirements Management Plan	GSCC/DEV/PLN/005/D1.0	1.0	External

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NEXUS	NX /DEV/SRS/001/D1.0	09.03.01	Restricted	D1.0	05/06/2008	PAGE 17 of 17

APPENDIX E**SYSTEM ARCHITECTURE DESIGN**

NX

System Architecture Design
Version V1.0

RESTRICTED

COPYRIGHT

© **SCAN Associates Berhad. Kuala Lumpur, 2008.**

Company Number: 525669-P

All rights reserved. No part of this publication may be reprinted, reproduced, stored in a retrieval system or transmitted, in any form or by any means, without the prior permission in writing from owners.

First published and distributed in July, 2008.

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	PAGE i of iv

Document Authorization

NX

System Architecture Design (SAD)

25th July 2008

For and on behalf of SCAN Associates Berhad:

Prepared By :

.....

**Name : Arztrie Delailah Binti Rumi
Designation : Software Engineer
Date :**

Approved By :

.....

**Name : Chong Sau Wei
Designation : Development Manager
Date :**

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	PAGE ii of iv

Revision History

Date	Version	Description	Author

Distribution List:

- Members of NX Software Development Team
- Authorized Personnel of SCAN Associates Berhad

Any person(s) not listed above are not authorized to examine the contents of this document. Doing so would constitute a breach of security and shall be subjected to appropriate actions being taken against the offender.

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	PAGE iii of iv

Table of Contents

1	INTRODUCTION	1
1.1	Purpose	1
1.2	Product Overview	1
1.3	Definitions, Acronyms and Abbreviations	1
1.4	Document Overview	2
2	HIGH LEVEL DESIGN	3
2.1	Product Perspective	3
2.2	Architectural Representation	3
2.3	Architectural Goals and Constraints	4
3	Use-Case View	5
3.1	Use Case Model: CSCI Reporting Management	5
3.1.1	Upload Report	6
3.1.2	Get Report	6
3.1.3	Generate Report	6
4	Logical View	7
4.1	Logical View: NX	7
4.2	Logical View: Reporting Management	8
4.3	Logical View: Interfaces for CSCI Reporting Management	9
4.4	Logical View: Class Diagram for CSCI Reporting Management	11
5	Process View	12
5.1	Process View: CSCI Reporting Management	12
5.1.1	Sequence Diagram: Use Case of Upload Report	12
5.1.2	Sequence Diagram: Use Case of Get Report	14
5.1.3	Sequence Diagram: Use Case of Generate Report	18
6	Component View	23
6.1	Component View: CSCI Reporting Management	23
7	Deployment View	24
7.1	Deployment View: CSCI Reporting Management	24
8	Size and Performance	25
8.1	Reporting Management (RM)	25
9	Quality	26
10	References	27

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	PAGE iv of iv

1 INTRODUCTION

The System Architecture Document (SAD) focuses on the whole architectural of the NX. It uses a number of different architectural representations to depict different aspects of the system. SAD captures and conveys the significant architectural decisions which have been made on the NX. In this section, the purpose, product overview and document overview will be detailed out.

1.1 Purpose

The SAD shall provide a shared understanding of the NX architecture across a broad range of people including developers, management and stakeholders. It describes the major components and functionalities of the NX. This document shall serve as the main reference document during all phases of development lifecycle for NX development team. Besides, it also used as a reference to track the progress of software development and assess the changes in requirements. In addition, it will be used as the basis for system maintenance, upgrades and enhancement.

1.2 Product Overview

The NX is intended to provide a suite of solutions in order to prevent, detect and respond to the vulnerabilities, Reporting and events in real time as well as generate reports and support future MSS Security Operation Center (SOC) services. To achieve this, there are few components or Computer Software Configuration Item (CSCI) of NX is stated below:

1. Reporting Management (RM) – Business intelligence, analytics and reporting.

1.3 Definitions, Acronyms and Abbreviations

CSC	– Computer Software Component
CSCI	– Computer Software Configuration Item: A group of software components treated as a single entity.
CSU	– Computer Software Unit
OS	– Operating System
RM	– Reporting Management
SAD	– System Architecture Design
RM	– Reporting Management
UML	– Unified Modeling Language

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 1 of 36

1.4 Document Overview

This document describes significant functionalities and interaction between the components of the NX. Unified Modeling Language (UML) notation will be used in this document to describe the NX components.

Chapter 1	Describes the purpose, product overview, definitions and document overview.
Chapter 2	Describes the high level design of the NX.
Chapter 3	Describes the functionalities and requirements of the NX.
Chapter 4	Describes the package diagram, class diagram and all the significant classes of each NX module.
Chapter 5	Describes the requirements workflows of each NX module.
Chapter 6	Describes the physical organization of each NX module.
Chapter 7	Describes the physical deployment of the NX.
Chapter 8	Describes the key sizing and timing requirements of the NX.
Chapter 9	Describes the quality of the project.
Chapter 10	Referenced document.

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 2 of 36

2 HIGH LEVEL DESIGN

This section describes the high level design of the NX. It describes the NX perspective and the entire project architectures.

2.1 Product Perspective

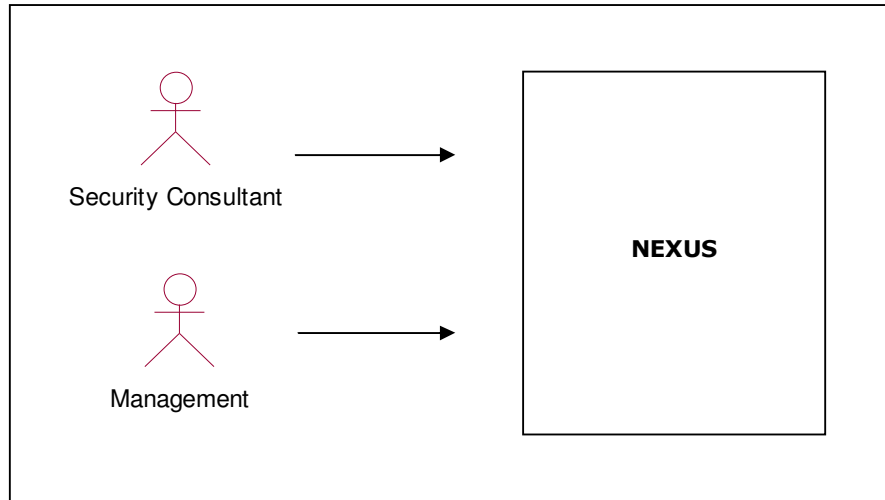


Figure 2-1: Use Case External Interface Diagram

NX is divided into two components that refer as CSCI, Reporting Management and Reporting Management. Figure 2-1 shows that two main interfaces, Security Consultant and Management, are interfacing with the NX CSCIs.

- Security Consultant is able to monitor and analyze the security events, responds to the security alerts as well as view the reports.
- Management is able to view and manage the reports.

2.2 Architectural Representation

There are several views used to represent NX architectures. They are use case view, process view (which includes sequence diagrams), logical view (which includes classes and sub-packages), component view (which includes source code libraries and run time files) and deployment view. These views are represented as Rational Rose Models and use the Unified Modeling Language (UML). The following sections describe each view in detail.

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 3 of 36

2.3 Architectural Goals and Constraints

There are some key requirements and system constraints that have significant bearing on the architecture. They are:

1. Security
 - All users to the RM console must be logged in before being able to use any of the RM application.
2. Performance
 - System response time in seconds and overall processing time not more than five seconds.
3. Logging and Auditing
 - All of the NX components must be able to support logging.
 - Supports at least five simultaneous sessions with different permission access.
4. Reporting
 - All of the NX components must be able to provide data in a format such as PDF and html that can be easily manipulated in order to generate reports.
5. Reliability
 - All of NX components must be able to run 24 x 7 x 364.

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 4 of 36

3 Use-Case View

This section describes use case model for each CSCI of the NX. The use case model identifies the users that will interact with the system. The use cases depict the important functionalities and requirements.

3.1 Use Case Model: CSCI Reporting Management

Use case Model of CSCI Reporting Management contains 3 use cases. The use cases are Upload Report, Get Report and Generate Report. The actor of this use cases are Security Consultant which represent Senior Security Analyst, Management, Security Expert and Security Analyst.

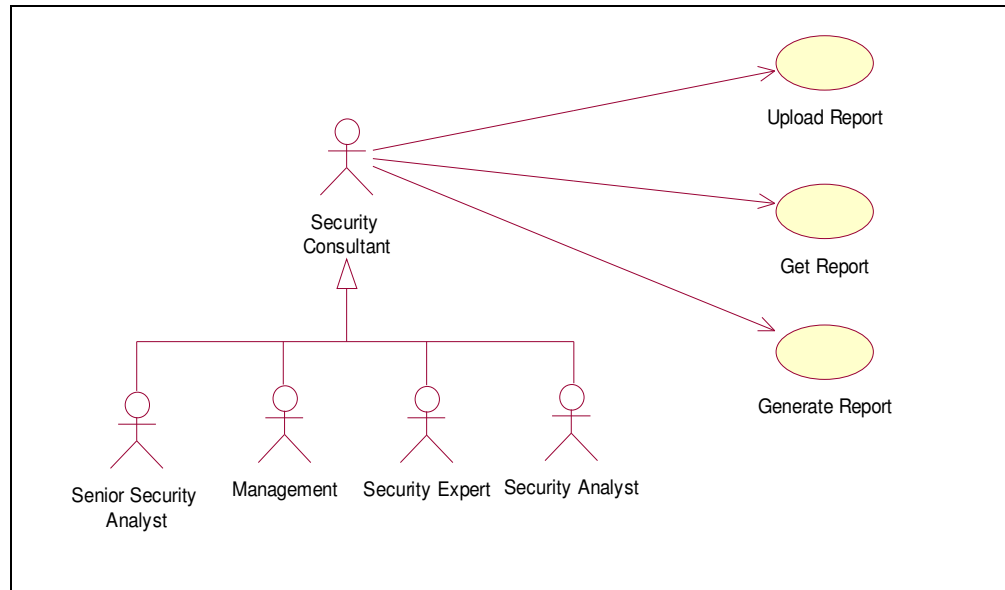


Figure 3-1: Use Case Diagram for Reporting Management

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 5 of 36

3.1.1 Upload Report

This use case is initiated by the Security Consultant. It provides the capability for Security Consultant to upload report to customer.

3.1.2 Get Report

This use case is initiated by the Security Consultant. It provides the capability for Security Consultant to get report.

3.1.3 Generate Report

This use case is initiated by the Security Consultant. It provides the capability for Security Consultant to generate report.

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 6 of 36

4 Logical View

This section is a description of the logical view of the architecture. It describes the most important packages and classes for each CSCI of the NX. It also describes the most important use-case realizations, for example, the dynamic aspects of the architecture. Class diagrams may be included to illustrate the relationships between architecturally significant classes.

4.1 Logical View: NX

This section packages the NX components in logical packages, as represented in the Figure 4-1. There are two main components which are Threat Management and Reporting Management.

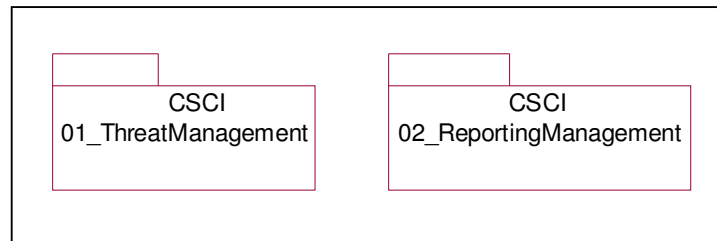


Figure 4-1: Main Packages for NX

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 7 of 36

4.2 Logical View: Reporting Management

CSCI RM is divided into 3 sub-packages which refer as CSC. Each CSC is visualized as a package as shown in Figure 5elow. Each package contains its own CSUs. There are 3 CSC such as Generate Report, Upload Report and Get Report.

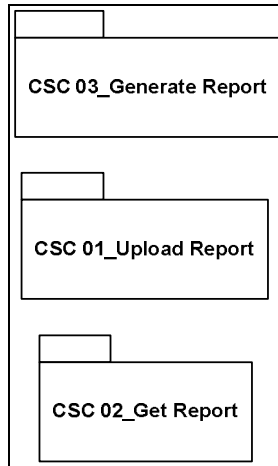


Figure 4-2: CSCs of CSCI Reporting Management

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 8 of 36

4.3 Logical View: Interfaces for CSCI Reporting Management

Interfaces show the whole structure of the system including classes, internal structure and also relationships between all the classes. In this project RM Module has been divided into three packages, the packages are, Generate Report, Upload Report and Get Report. Figure 5elow shows the relationship among CSCs in RM Module.

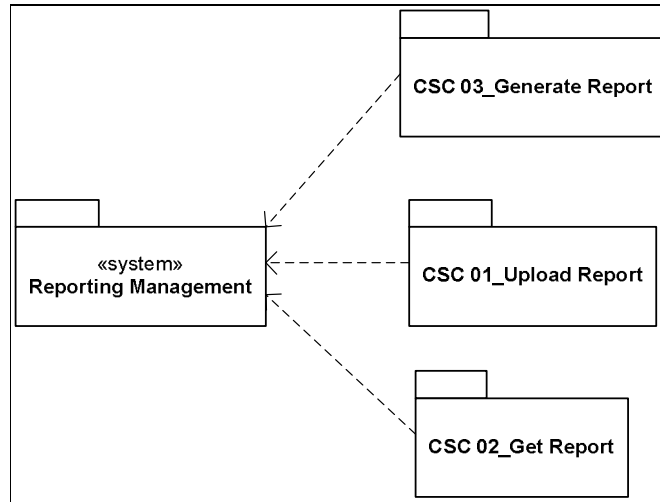


Figure 4-3: CSCs of CSCI Reporting Management

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 9 of 36

Each CSC or package plays different roles. Table 4-1 describes the list of CSCs for the RM and the roles of each CSC.

Table 4-1: Description of CSC for Reporting Management

No.	CSC Identification	CSC Name	CSC Description
1.	SAD_REQ_100	CSC 01_Upload Report	This package provides the capabilities to authenticate user upload report to the server or client portal.
2.	SAD_REQ_200	CSC 02_Get Report	This package provides the capabilities to manage report and to search the filtered reports.
3.	SAD_REQ_300	CSC 03_Generate Report	This package provides the capabilities for the user to generate various type of report.

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 10 of 36

4.4 Logical View: Class Diagram for CSCI Reporting Management

The Class diagram of RM Module consists of ClientEmailList, FormList, StatisticalData, clientID, FormList, ReportList and Event. Figure 5elow shows the Class Diagram for RM Module.

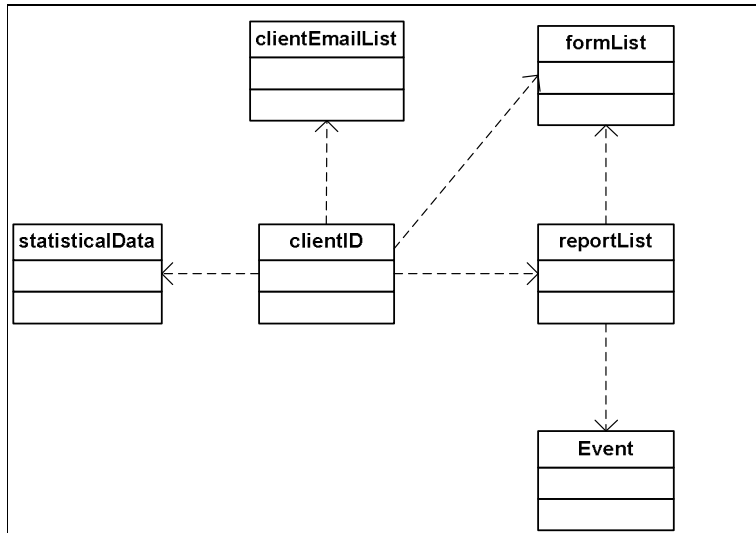


Figure 4-4: Class Diagram for CSCI Reporting Management

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 11 of 36

5 Process View

This section is a description of the process view of the architecture. Describes the tasks (processes and threads) involved in the system's execution, their interactions and configurations. Also describes the allocation of objects and classes to tasks.

5.1 Process View: CSCI Reporting Management

This section describes the requirements workflows of each use case model for the CSCI Reporting Management using Sequence Diagram. **Appendix A** shows the Collaboration Diagrams of each use case model for the CSCI Reporting Management.

5.1.1 Sequence Diagram: Use Case of Upload Report

This section depicts the scenarios for the use case of Upload Report which represented in Basic Flow, Alternative Flow (if any) and Exception Flow (if any) sequence diagrams.

5.1.1.1 Basic Flow: Upload Report

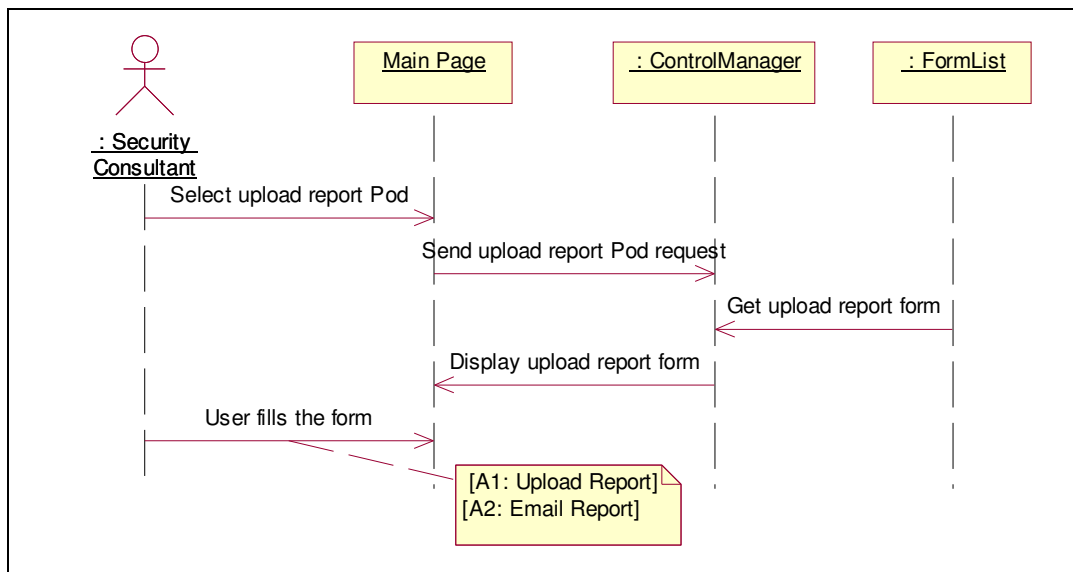


Figure 5.1 : Sequence Diagram Upload Report: Basic Flow

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 12 of 36

5.1.1.2 Alternative Flow

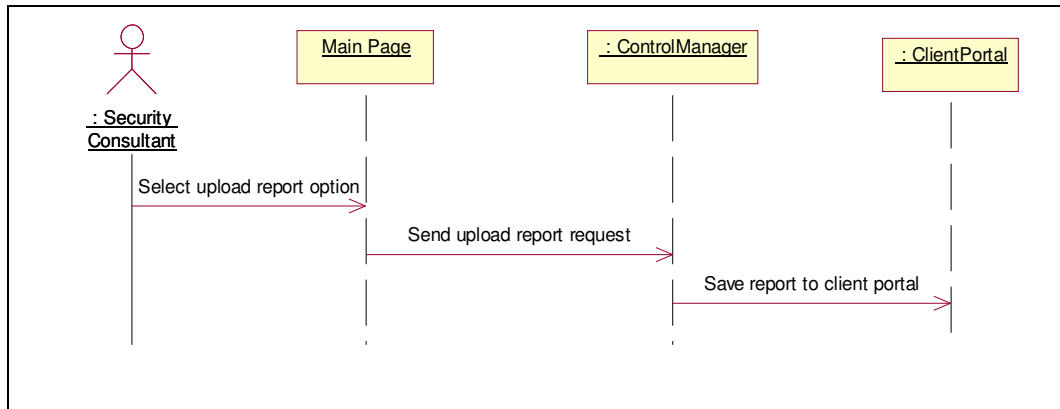


Figure 5.2 : Sequence Diagram Upload Report: Alternative Flow [A1: Upload]

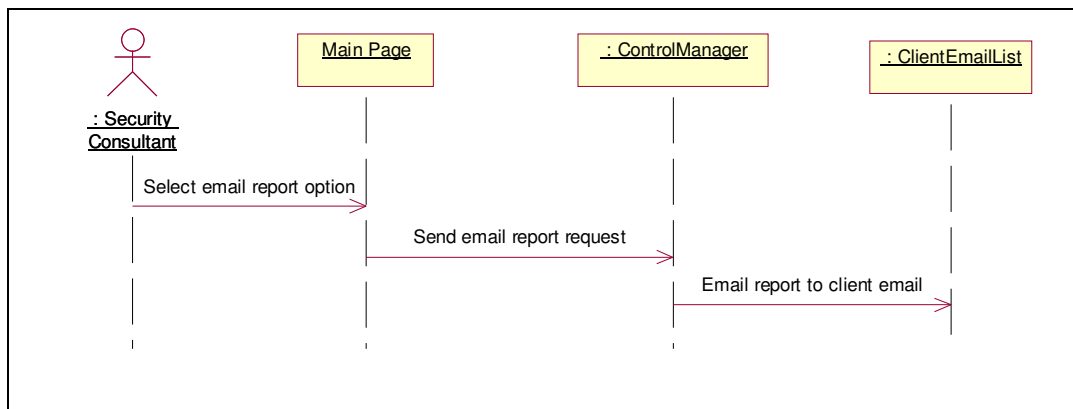


Figure 5.3 : Sequence Diagram Upload Report: Alternative Flow [A1: Email]

5.1.1.3 Exception Flow(s)

Not applicable.

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 13 of 36

5.1.2 Sequence Diagram: Use Case of Get Report

This section depicts the scenarios for the use case of Get Report which represented in Basic Flow, Alternative Flow (if any) and Exception Flow (if any) sequence diagrams.

5.1.2.1 Basic Flow: Get Report

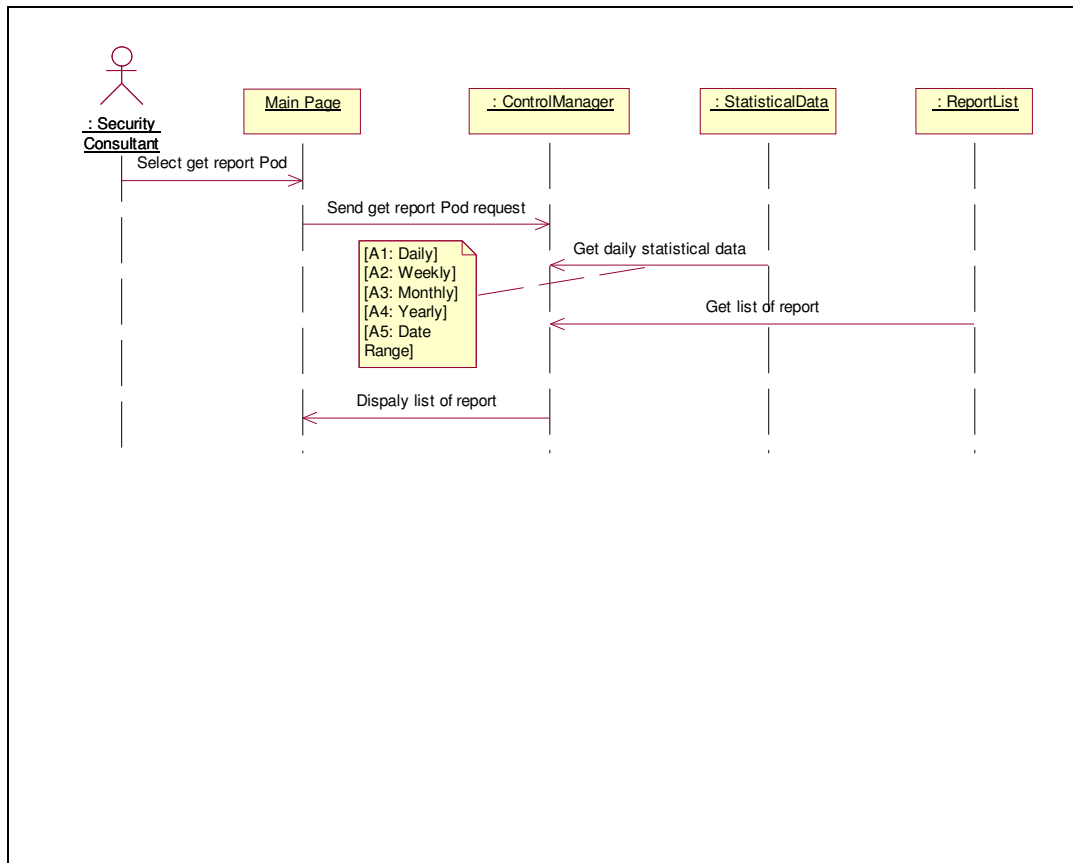


Figure 5.4: Sequence Diagram Get Report: Basic Flow

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 14 of 36

5.1.2.2 Alternative Flow

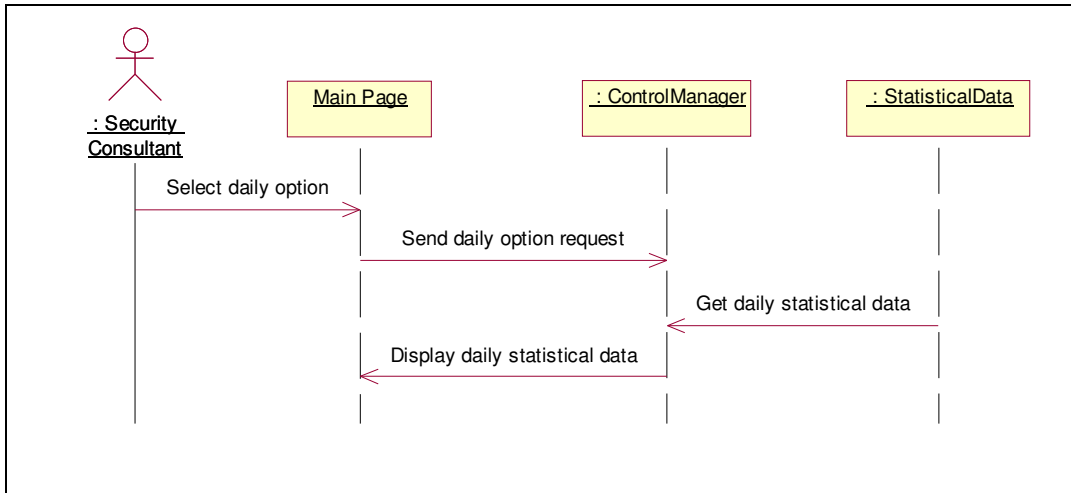


Figure 5.5: Sequence Diagram Get Report: Alternative Flow [A1: Daily]

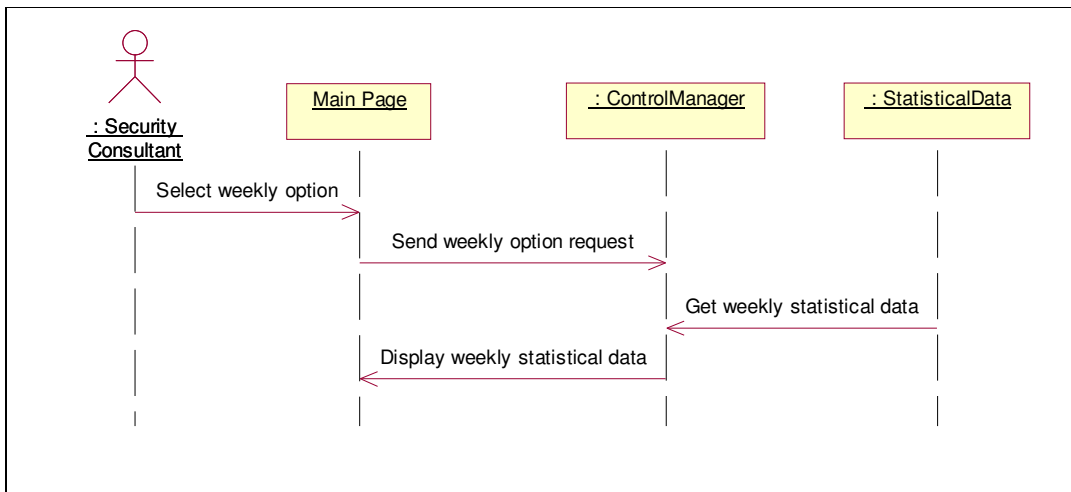


Figure 5.6 : Sequence Diagram Get Report: Alternative Flow [A1: Weekly]

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 15 of 36

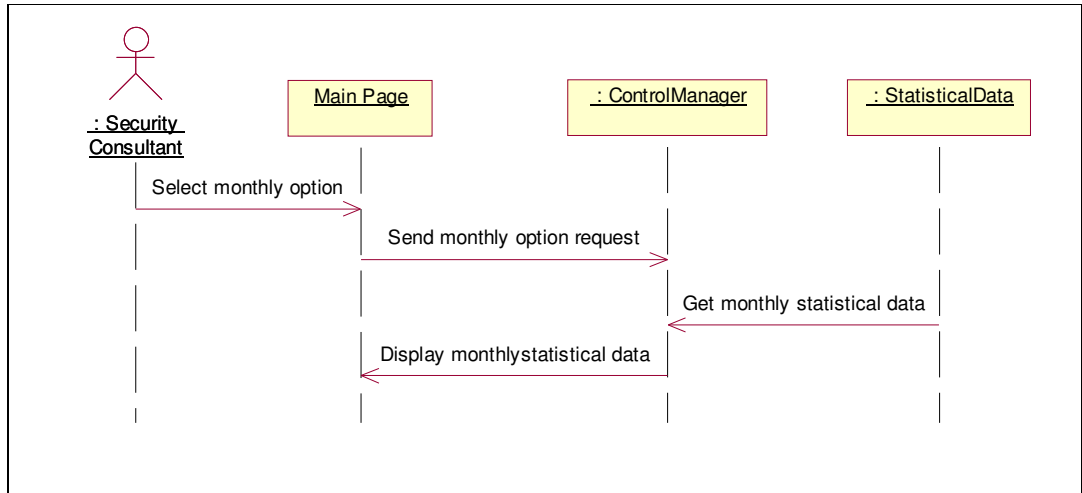


Figure 5.7 : Sequence Diagram Get Report: Alternative Flow [A1: Monthly]

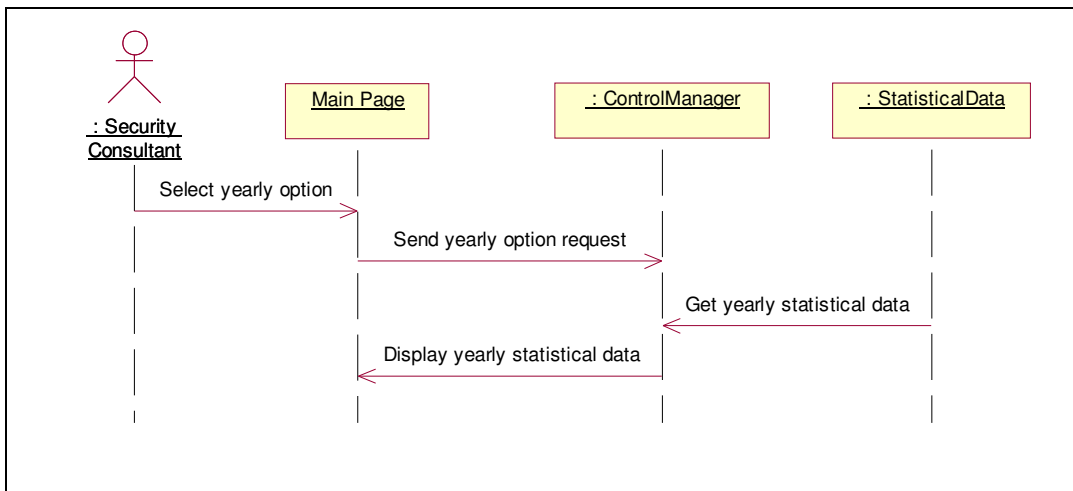


Figure 5.8 : Sequence Diagram Get Report: Alternative Flow [A1: Yearly]

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 16 of 36

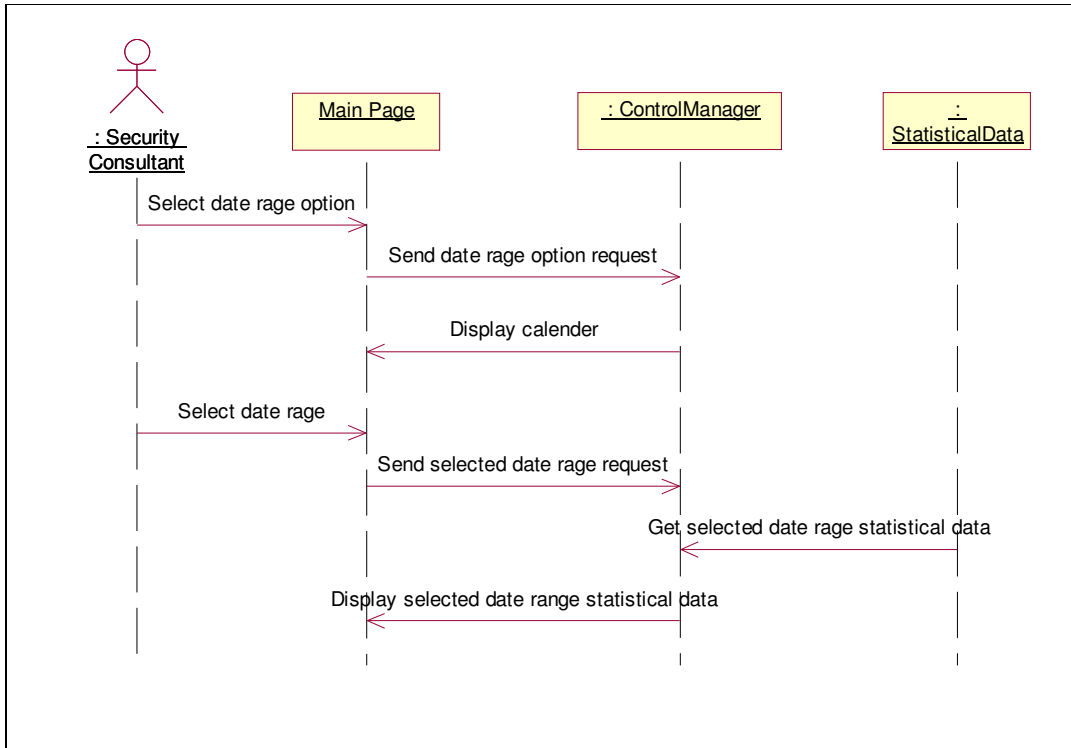


Figure 5.9 : Sequence Diagram Get Report: Alternative Flow [A1: Date Range]

5.1.2.3 Exception Flow(s)

Not applicable.

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 17 of 36

5.1.3 Sequence Diagram: Use Case of Generate Report

This section depicts the scenarios for the use case of Generate Report which represented in Basic Flow, Alternative Flow (if any) and Exception Flow (if any) sequence diagrams.

5.1.3.1 Basic Flow: Generate Report

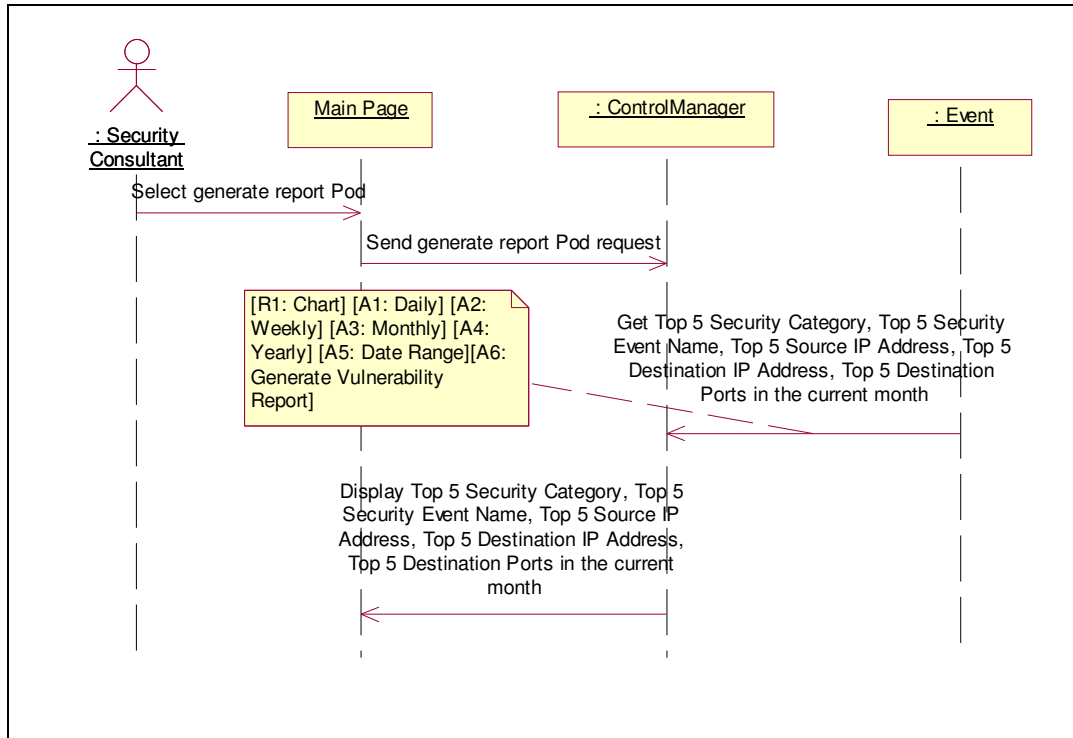


Figure 5.10 : Sequence Diagram Generate Report: Basic Flow

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 18 of 36

5.1.3.2 Alternative Flow

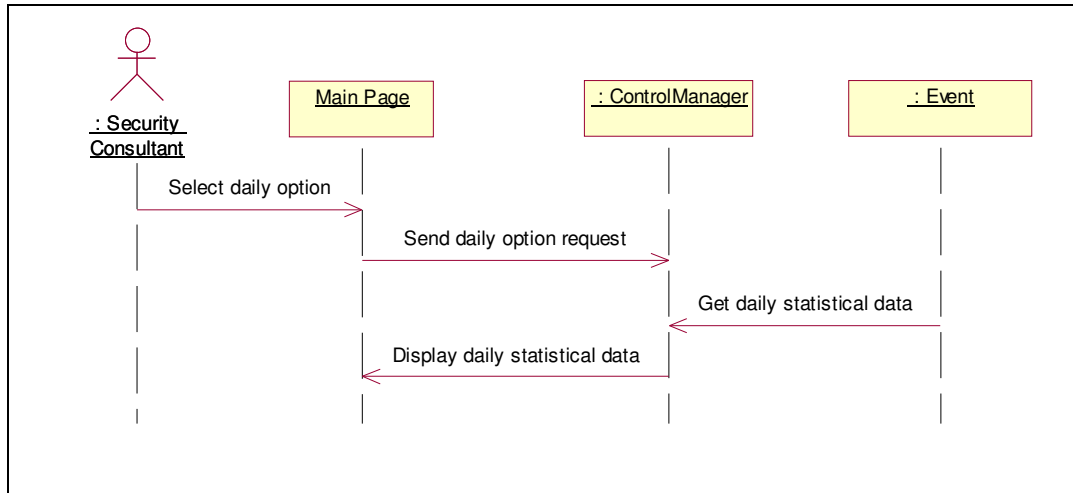


Figure 5.11 : Sequence Diagram Get Report: Alternative Flow [A1: Daily]

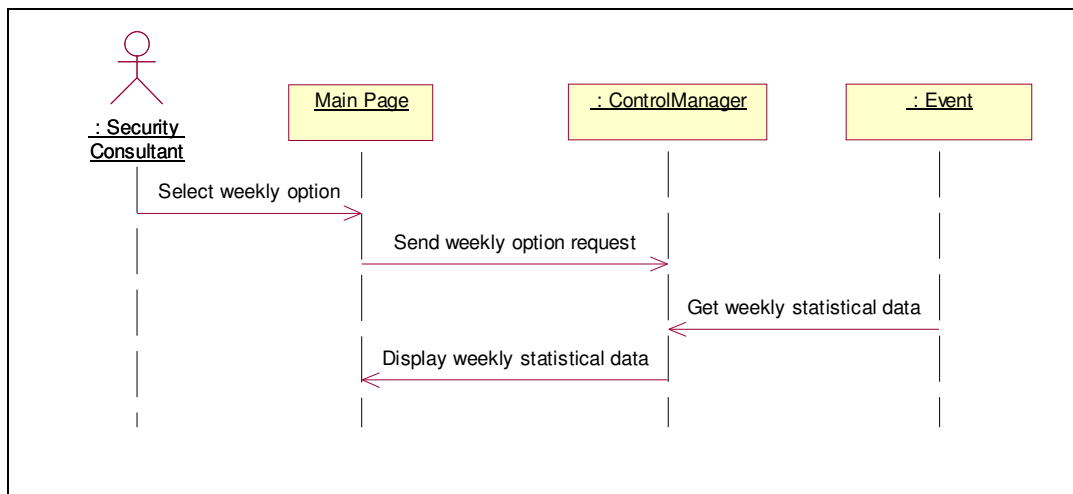


Figure 5.12 : Sequence Diagram Get Report: Alternative Flow [A2: Weekly]

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 19 of 36

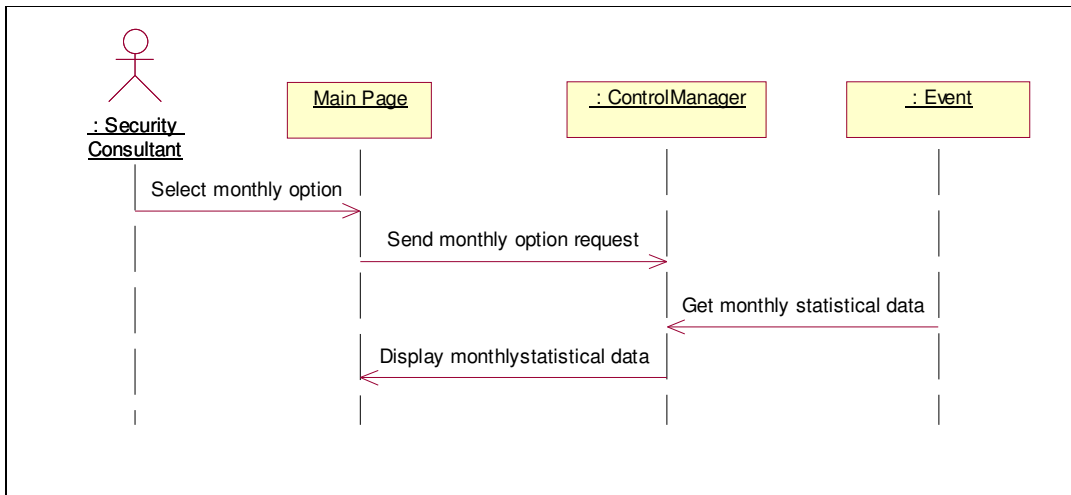


Figure 5.13 : Sequence Diagram Get Report: Alternative Flow [A3: Monthly]

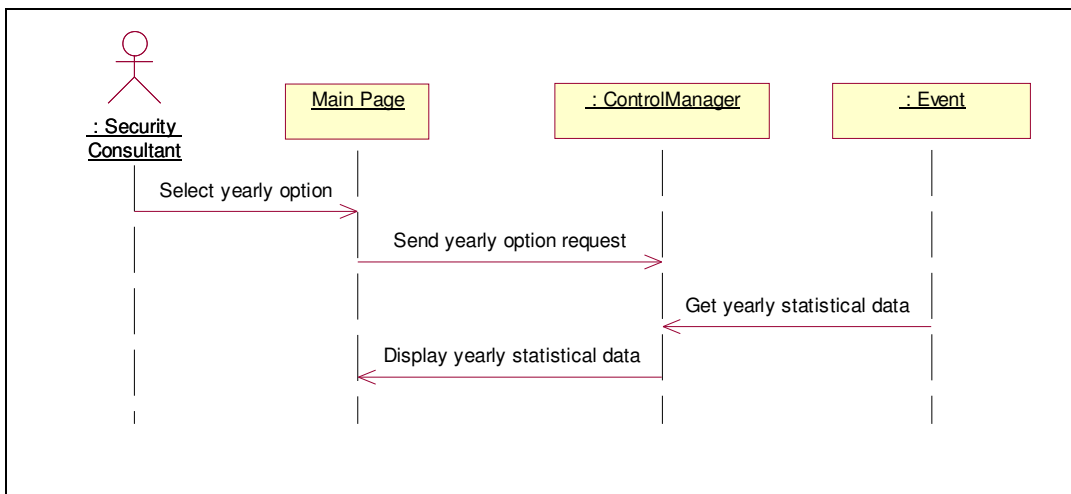


Figure 5.14 : Sequence Diagram Get Report: Alternative Flow [A4: Yearly]

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 20 of 36

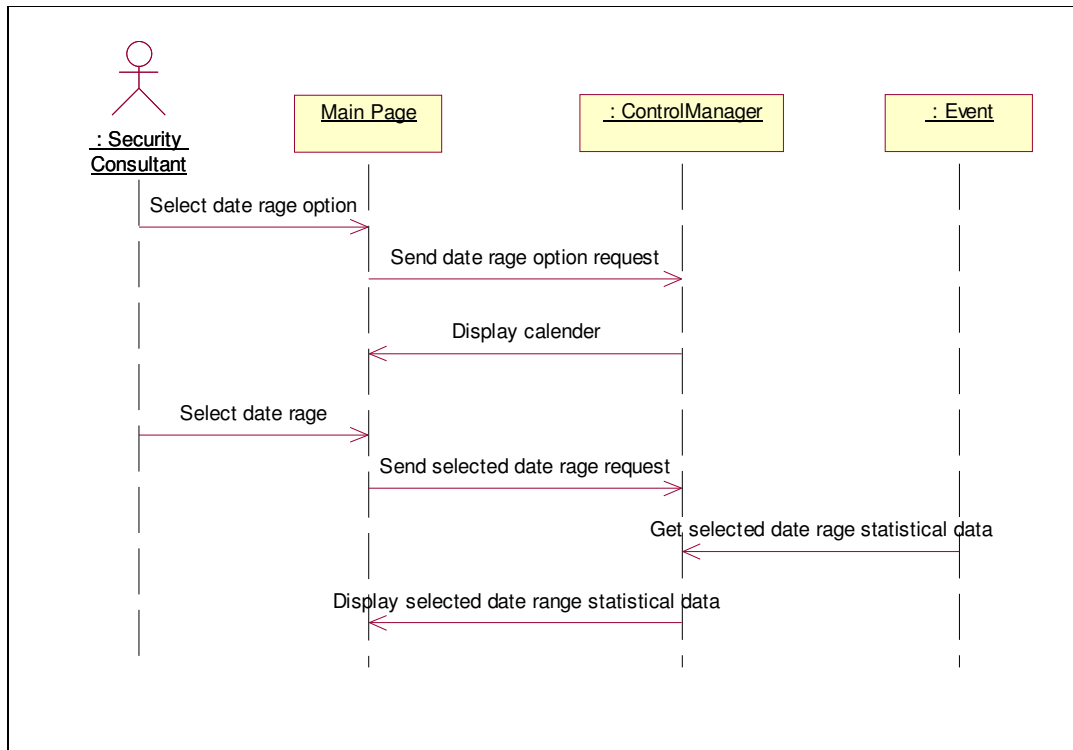


Figure 5.15 : Sequence Diagram Get Report: Alternative Flow [A5: Date Range]

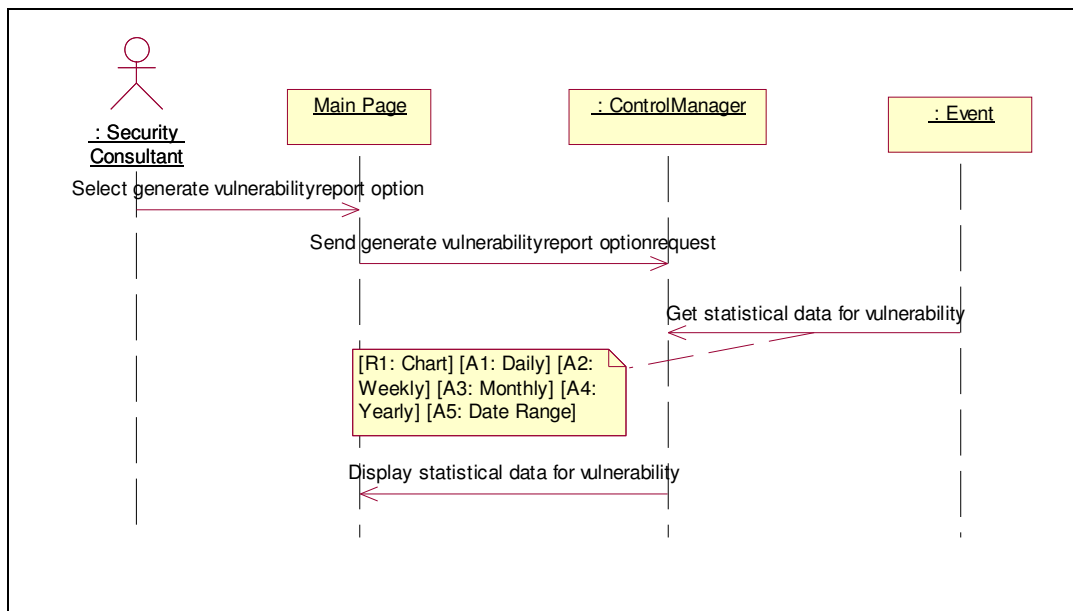


Figure 5.16 : Sequence Diagram Get Report: Alternative Flow [A1: Generate Vulnerability Report]

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 21 of 36

5.1.3.3 Exception Flow(s)

Not applicable.

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 22 of 36

6 Component View

This section describes the physical organization of each module for the NX system. The component view demonstrates the physical module of code. Each component can include both source code libraries and runtime files.

6.1 Component View: CSCI Reporting Management

Components which are involved in the development of CSCI RM are defined in this section. Figure 6-1 depicts the main components required to implement the CSCI RM. The physical module of the code can be interpreted in component diagram. The source code libraries and also the runtime files can be included in component diagram. Figure below shows the main component that has been required to implement the RM Module.

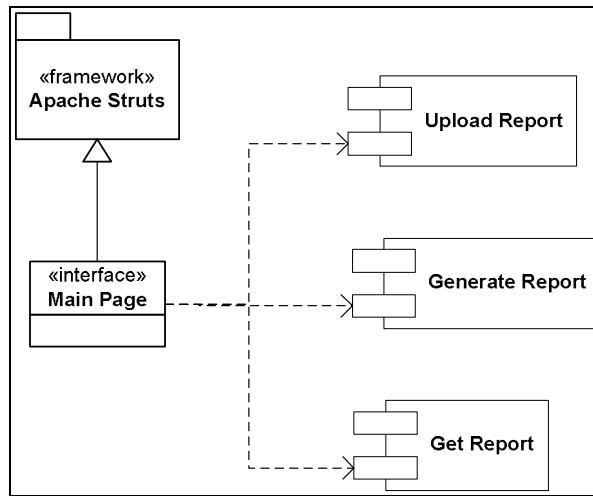


Figure 6-1: Main Component Diagram for CSCI Reporting Management

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 23 of 36

7 Deployment View

This section describes one or more physical nodes in which the NX components are installed and run. This view includes issues such as the network layout and the location of the components on the network. Due to the complexity of the NX system, the deployment view of each component will be visualized separately.

7.1 Deployment View: CSCI Reporting Management

The network management units which are involved in the development of CSCI RM are defined in this section. Figure 7-1 depicts the entire RM network layout. Client use web browser to access RM Module on Tomcat application server. Each request must go through Apache web server. All the data been stored on MySQL database server and can be retrieve on user request

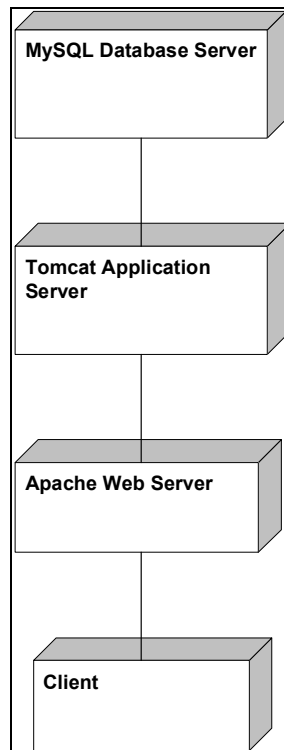


Figure 7-1: Deployment Diagram for CSCI Reporting Management

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 24 of 36

8 Size and Performance

The chosen system architecture supports key sizing and timing requirements. These include:

8.1 Reporting Management (RM)

1. The generation of each report will take no more than 5 seconds.
2. The database should be able to handle up to 2 TB (terabyte) of data after three months of operation.
3. The generation of each graph will take no more than 5 seconds.
4. All searches must not take more than 5 seconds.
5. RM component must be able to operate 24 hours a day, 7 days a week with minor interruption.

The partitioning of each NX component and the use of client-server architecture will ensure that the NX system will meet size and timing requirements.

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 25 of 36

9 Quality

All software developed will go through the appropriate review process to ensure quality. The purpose of review is to ensure that each software deliverable is of acceptable quality, using guidelines described in the Rational Unified Process review guidelines and checklists

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 26 of 36

10 References

The following table outlines all documents that are referenced in the SAD.

No	Document Title	Document ID	Ver.	Reference
1.	NX System Requirement Specification V1.1	NX/DEV/SRS/001/V1.1	1.1	External
2.	Use-Case Modeling Guidelines	PRISMA/DEV/UCM/006/V1.0	1.0	External
3.	Secure Development Guidelines	PRISMA/DEV/SDV/008/V1.0	1.0	External

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 27 of 36

Appendix A

RM Collaboration Diagrams

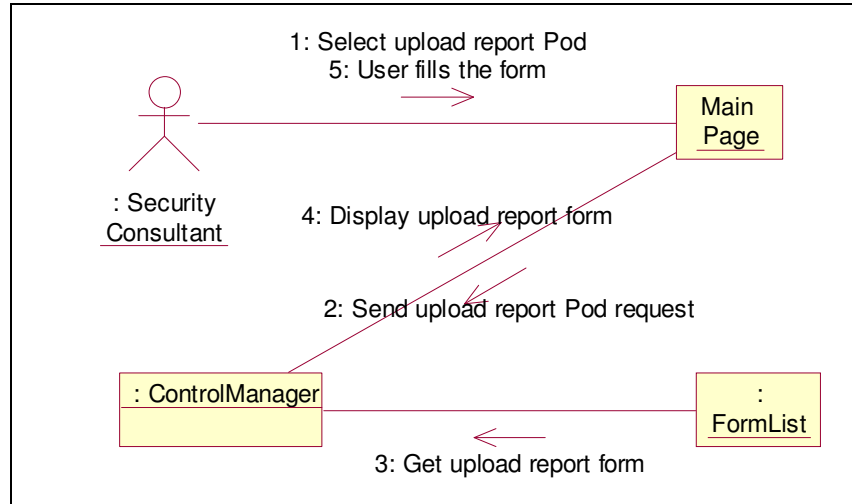


Figure A.1 : Basic Flow: Upload Report

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 28 of 36

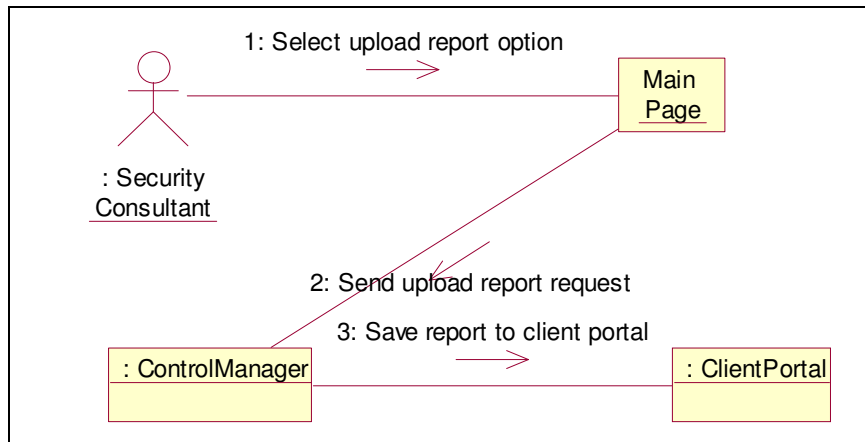


Figure A.2 : Alternative Flow Upload Report: A1 Upload Report

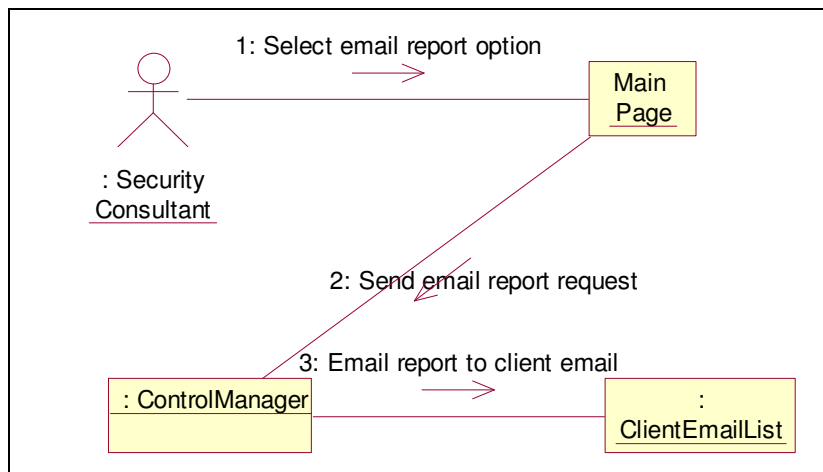


Figure A.3 : Alternative Flow Upload Report: A1 Email Report

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 29 of 36

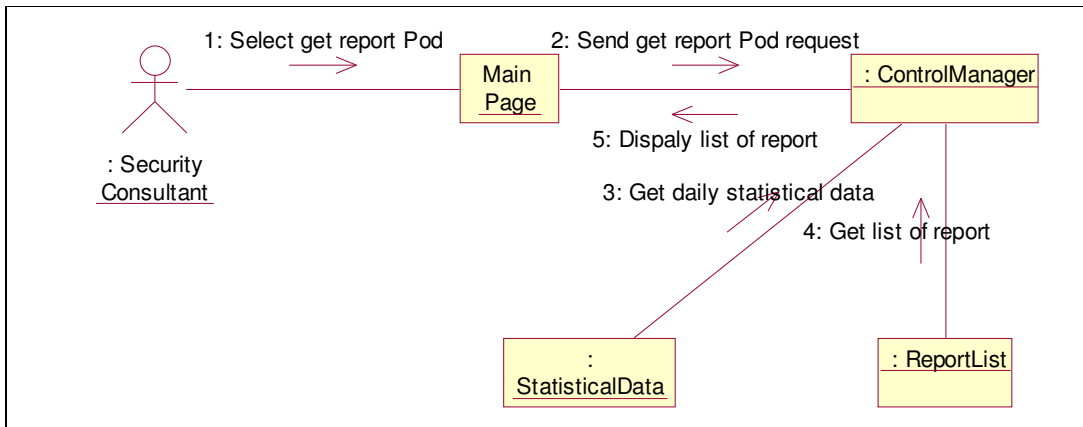


Figure A.4 : Basic Flow: Get Report

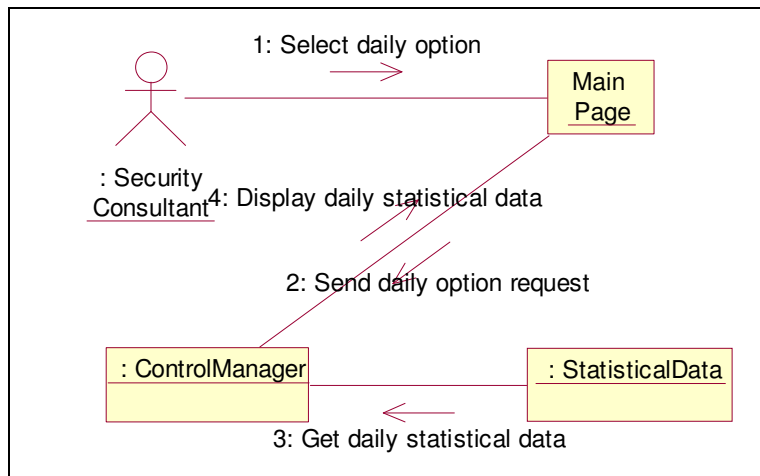


Figure A.5 : Alternative Flow Get Report: A1 Daily Report

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 30 of 36

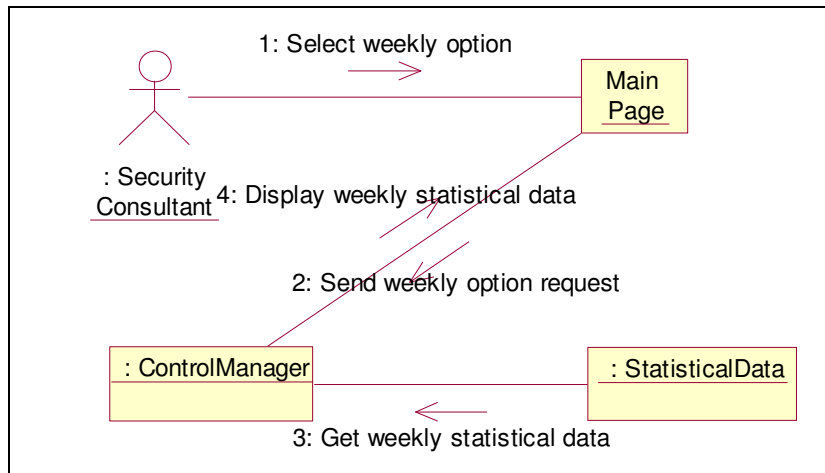


Figure A.6 : Alternative Flow Get Report: A2 Weekly Report

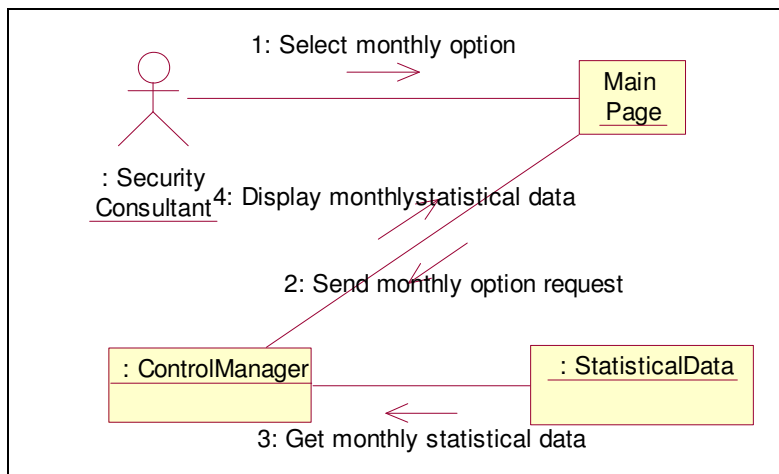


Figure A.7 : Alternative Flow Get Report: A3 Monthly Report

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 31 of 36

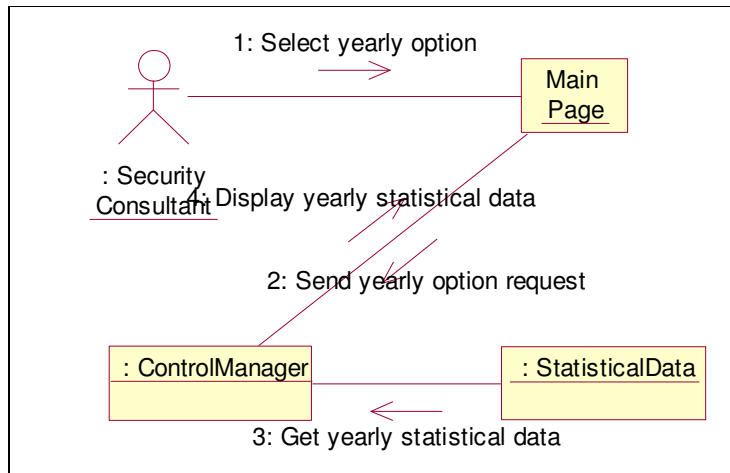


Figure A.8 : Alternative Flow Get Report: A4 Yearly Report

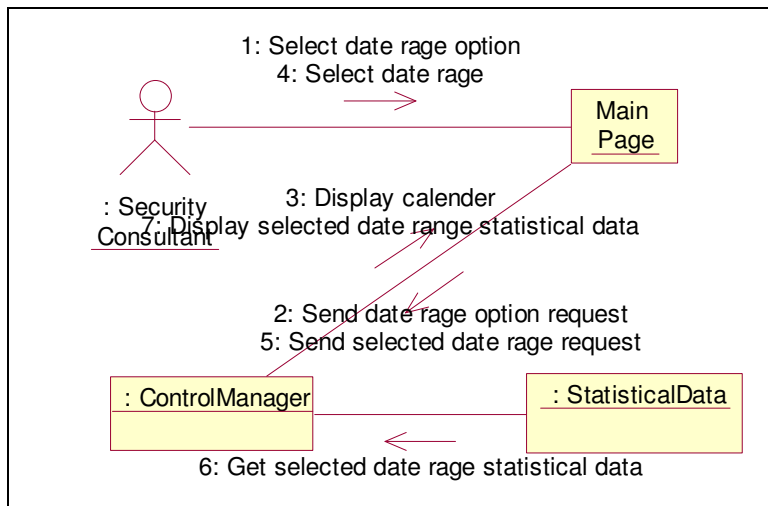


Figure A.9 : Alternative Flow Get Report: A5 Date Range Report

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 32 of 36

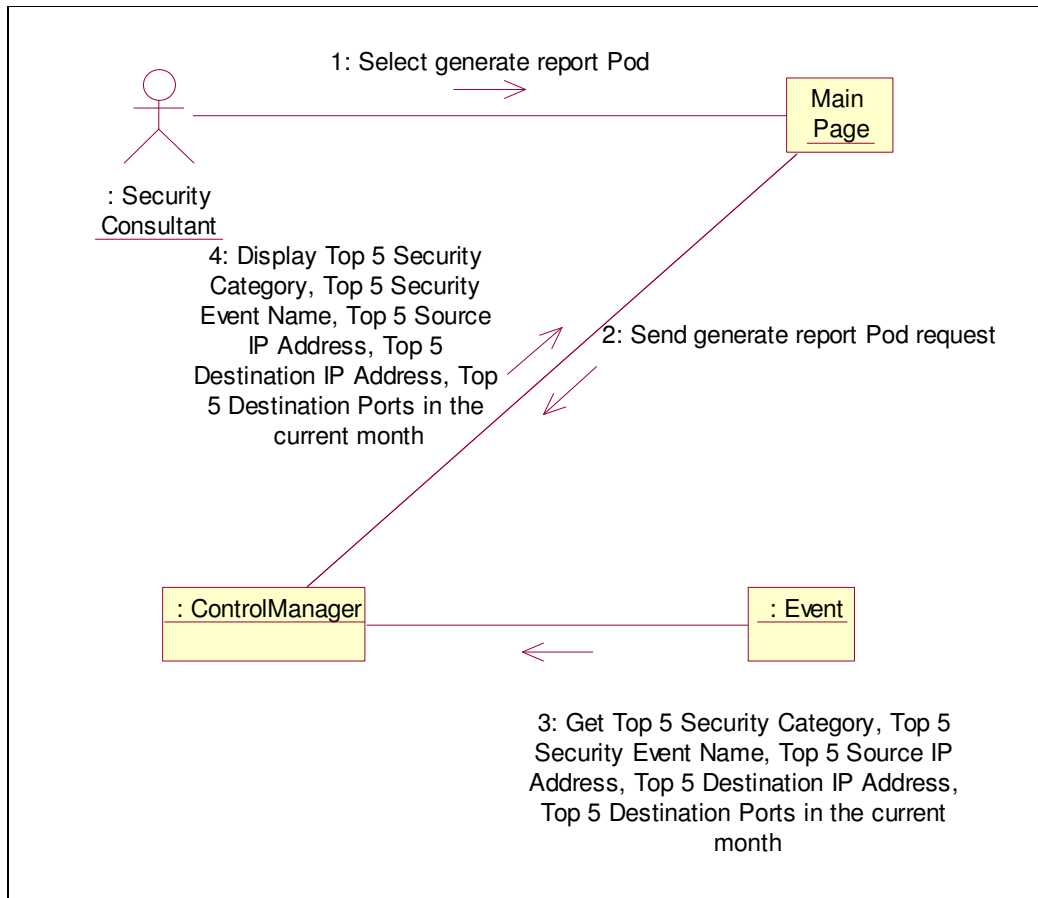
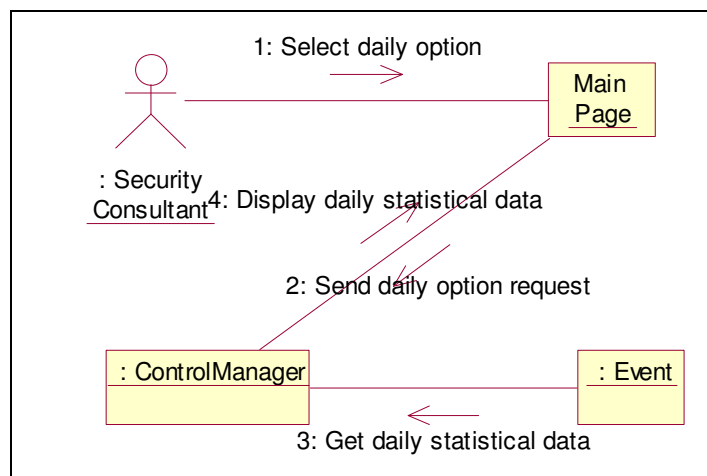


Figure A.10 : Basic Flow: Generate Report



PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 33 of 36

Figure A.11 : Alternative Flow Generate Report: A1 Daily

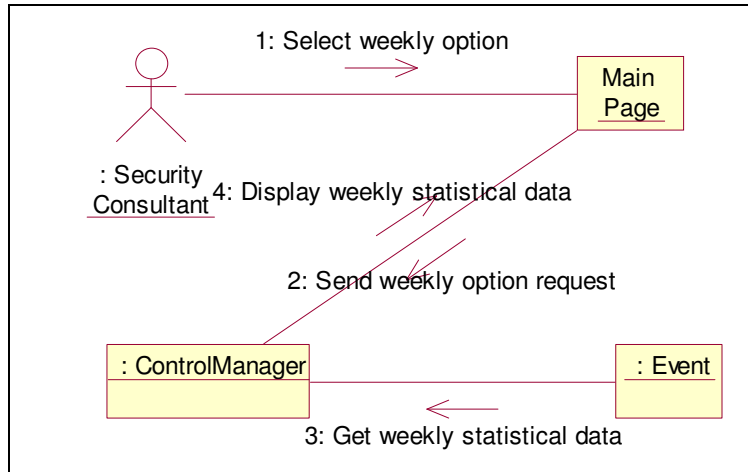


Figure A.12 : Alternative Flow Generate Report: A2 Weekly

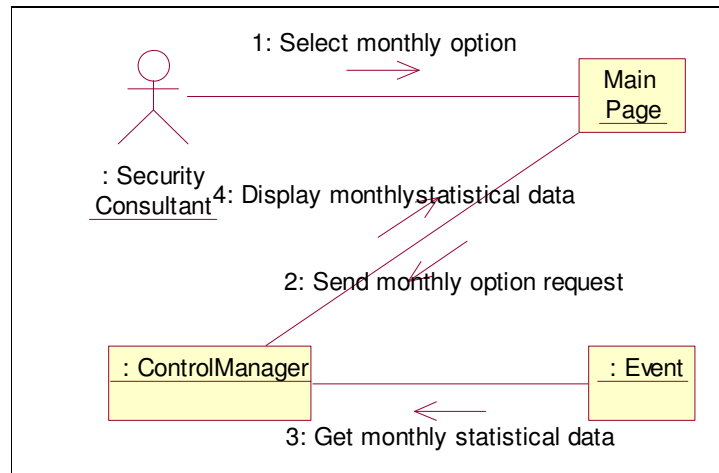


Figure A.13 : Alternative Flow Generate Report: A3 Monthly

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 34 of 36

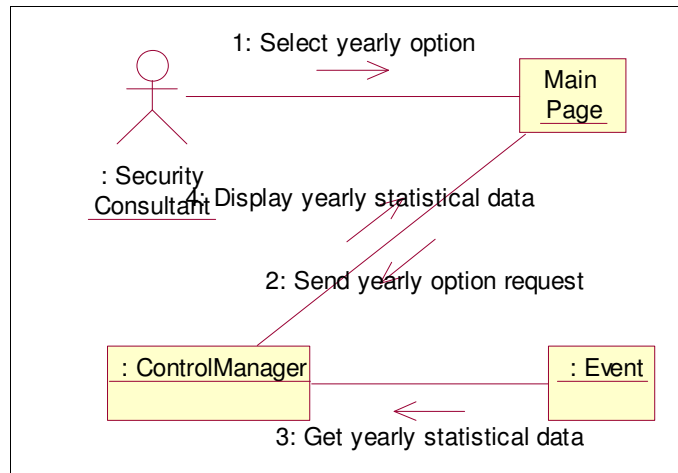


Figure A.14 : Alternative Flow Generate Report: A5 Date Range

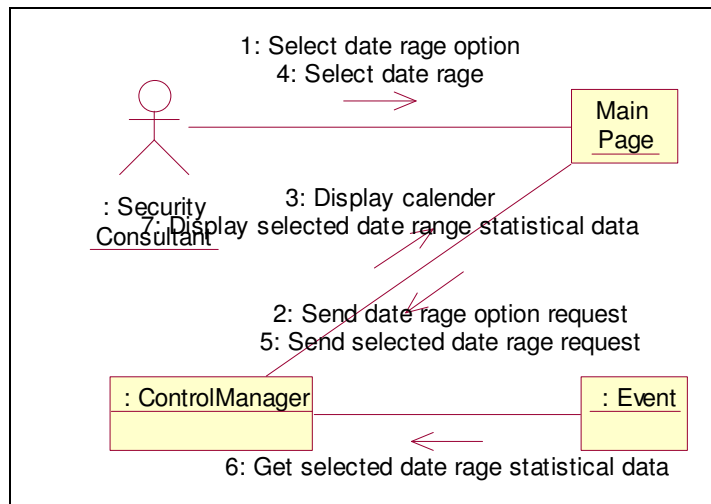


Figure A.15 : Alternative Flow Generate Report: A5 Yearly

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 35 of 36

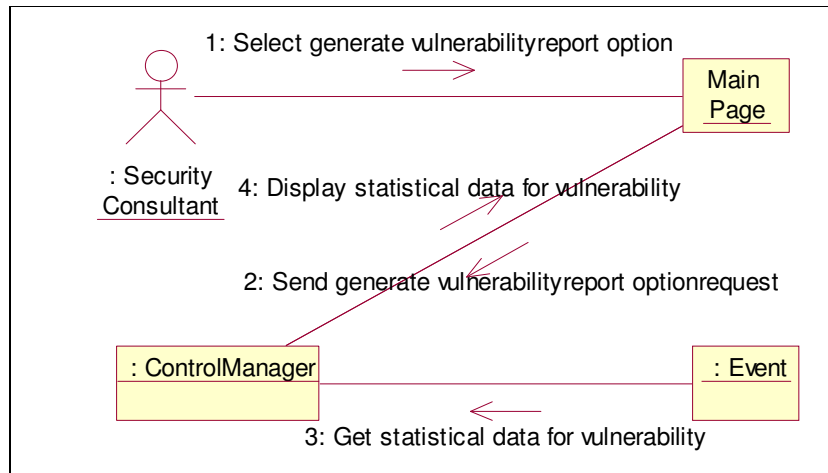


Figure A.16 : Alternative Flow Generate Report: A6 Generate Vulnerability Report

PROJECT	REFERENCE	LIBRARY	SECURITY ID	REVISION	DATE	PAGE
NX	NX/DEV/SAD/001/D1.0	09.03.01	Restricted	D1.0	14/07/08	Page 36 of 36