SQL Based Development for the Management of FMEA and FMECA files

by

Juan G. Vázquez Ortiz

A final report submitted in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING
in
COMPUTER ENGINEERING

UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS
2012

Approved by:

_____	_____
Pedro I. Rivera, PhD Member, Graduate Committee	Date
_____	_____
Pedro Resto, PhD Member, Graduate Committee	Date
_____	_____
Manuel Rodríguez, PhD President, Graduate Committee	Date
_____	_____
Ana C. González Representative of Graduate Studies	Date
_____	_____
Pedro I. Rivera, PhD Chairperson of the Department	Date

ABSTRACT

This work presents the implementation of a system to improve the availability of excel based data specifically the FMEA and FMECA data used by the US Army RAM Engineering and System Assessment group by creating a web based solution. The web based solution includes a database design with tables to store the FMEA and FMECA information, a C# application used to read and insert into the database the FMEA and FMECA excel based information, an ASP.NET web application for displaying the FMEA and FMECA information and a group of reports integrated into the web applications created to provide exporting options to Excel and PDF. This work also presents the results of the verification of the use of the Visual Basic included in Excel (Visual Basic for Applications) to access the FMEA and FMECA information from the database and write it directly into the excel sheet.

RESUMEN

Este trabajo presenta la implementación de un sistema para mejorar la disponibilidad de información que solo esta disponible en Excel, específicamente la información de los FMEA and FMECA que usa el "US Army RAM Engineering and System Group" creando una solución web que permite un mejor acceso a la información. La solución web incluye el diseño de la base de datos con las tablas para guardar la información de los FMEA y FMECA, una aplicación en el lenguaje de programación C# que es usada para leer e insertar en la base de datos la información de los FMEA y FMECA, una aplicación web en ASP.NET para desplegar la información de los FMEA y FMECA y un grupo de reportes que están integrados en la aplicación web y que fueron creados para proveer las opciones de exportar a Excel y a PDF. Este trabajo también presenta el resultado de la verificación del uso del Visual Basic que esta incluido en Excel (Visual Basic para Aplicación) para obtener la información de los FMEA y FMECA de la base de datos y escribir esta información directamente en el Excel.

To my family . . .

ACKNOWLEDGEMENTS

I would like to thanks my advisors Dr. Manuel Rodriguez and Dr. Pedro Resto for giving me the opportunity to work in this project and to use this project as my graduation project and to the Dr. Pedro Rivera for his availability. I also would like to thank my family for their continuous support.

Table of Contents

ABSTRACT ..... **II**

RESUMEN ..... **III**

ACKNOWLEDGEMENTS ..... **V**

TABLE OF CONTENTS ..... **VI**

FIGURE LIST ..... **VIII**

1 INTRODUCTION **2**

1.1 JUSTIFICATION 2
1.2 DEVELOPED SOLUTION 3
1.3 WORK CONTRIBUTIONS 4
1.4 SUMMARY OF FOLLOWING CHAPTERS 4

2 LITERATURE REVIEW **5**

2.1 FMECA 5
2.2 WEB APPS: HTML 9
2.3 WEB APPS: JAVASCRIPT 11
2.4 WEB APPS: CASCADING STYLE SHEETS (CSS) 13
2.5 ASP.NET 14
2.6 DATABASES 15
2.7 SQL 16
2.8 SQL SERVER 17
2.9 SQL SERVER REPORTING SERVICES 18

3 PROJECT STRUCTURE **20**

3.1 PROJECT OBJECTIVES 20
3.2 PROJECT COMPONENTS 20
3.3 EXCEL-BASED FMEA AND FMECA FILES 21
3.4 C# APPLICATION 21
3.5 WEB APPLICATION 22
3.6 SQL SERVER REPORTING SERVICES 22
3.7 TECHNOLOGIES 23

4 FMEA AND FMECA FILES **25**

4.1 FAILURE MODES AND EFFECTS ANALYSIS (FMEA) FILES 25
4.2 FAILURE MODE, EFFECTS AND CRITICALITY ANALYSIS (FMECA) FILES 27

4.3 FMEA AND FMECA WUC REFERENCE LIST 29

5 DATABASE DESIGN 31

5.1 DATABASE SCHEMA 31
5.2 DATABASE TABLES 34

6 C# APPLICATION 41

6.1 PROGRAM STRUCTURE 41
6.2 MAIN CLASS 42
6.3 READ EXCEL FILE 45
6.4 FILE TYPES 46
6.5 DATABASE OPERATIONS 49

7 WEB APPLICATION 51

7.1 WEB APPLICATION STRUCTURE 51
7.2 WEB PAGES 52
7.3 SQL SERVER: STORED PROCEDURES 62
7.4 REPORTS: EXPORT DATA 65

8 EXCEL MACRO 71

8.1 DATABASE ACCESS 73
8.2 DELETE DATA 75

9 CONCLUSIONS AND FUTURE WORK 76

9.1 CONCLUSIONS 76
9.2 FUTURE WORK 77

REFERENCES 78

Figure List

Figures **Page**

Figure 1: Example of FMEA worksheet format [MIL-STD-1629A] 6
Figure 2: Example of Criticality Analysis (CA) worksheet format [MIL-STD-1629A]..... 8
Figure 3: Example HTML Page and HTML Code 11
Figure 4: Client-Side Web Application Interaction 13
Figure 5: Server-Side Web Application Interaction 15
Figure 6: Project Structure diagram (Project Components)..... 21
Figure 7: Export Request Interaction Diagram 22
Figure 8: Example of FMEA excel file provided by [USArmyRAM]. 26
Figure 9: Example of FMECA excel file provided by [USArmyRAM]. 28
Figure 10: Example of Reliability Data in the FMECA excel file provided by [USArmyRAM]. 29
Figure 11: Example of WUC Reference List excel file provided by [USArmyRAM]. 30
Figure 12: Database Schema design diagram 33
Figure 13: SQL Server Database Tables Diagram..... 40
Figure 14: C# Application Components diagram 41
Figure 15: C# application console 42
Figure 16: C# application console - Insert FMEA information..... 44
Figure 17: C# Application File Types class diagram..... 46
Figure 18: FMEA excel file format example..... 48
Figure 19: C# Application Database Operations class diagram 49
Figure 20: Web Application Structure diagram (Web Application Components) 51
Figure 21: Example FMEA data in Web Application 53
Figure 22: Example FMECA data in Web Application..... 54
Figure 23: Example System (Excel) List in Web Application 57
Figure 24: Example System and Mode Id (Excel Row) List in Web Application 58
Figure 25: Example Excel Row in Web Application..... 58
Figure 26: Example of the Reliability Data in Web Application..... 59
Figure 27: Example WUC List in Web Application..... 60
Figure 28: Example of FMEA data for a WUC in Web Application 61
Figure 29: Example Part Id List in Web Application 62
Figure 30: Example of SQL Query to concatenate List of WUC 64
Figure 31: FMEA Report with Header information 66
Figure 32: FMECA Report with Header Information..... 66
Figure 33: FMEA Report without header information 67
Figure 34: FMECA Report without header information..... 67

Figure 35: Reliability Data Report..... 68
Figure 36: Example Export to Excel..... 69
Figure 37: Example Export to PDF 70
Figure 38: Excel with Database Access Macro without Data..... 71
Figure 39: Excel with Database Access Macro with Data..... 72
Figure 40: Excel Macro Database Access source code. 74
Figure 41: Excel Macro Delete Data Source Code..... 75

1 INTRODUCTION

In many companies is common to use excel based information especially when the company legacy information is excel based and the use of such information is part of the culture of the company and their employees. This situation is typical in companies that do not have employees with knowledge of the benefits that could bring the use of a database to store the information and the use of a web application to make the data more accessible.

The use of a database to store the information not only brings many benefits like: providing a more controlled place for the data storage and making the data more accessible but also open new possibilities for the management and analysis of the data. The use of a web application for data access also provides many advantages over the use of an excel file like: easier access and the insurance that everyone is accessing the same data. The recognition of the advantages of the use of databases and web applications for data management and data presentation has leaded many companies to change from an Excel based approach to a web based approach for the access and presentation of the data of the company.

1.1 Justification

A Failure Modes and Effects Analysis (FMEA) is used for the analysis of the failures of a system. The analysis includes the failure mode information, the causes for the failure and the failures effects. The Failure Mode, Effects and Criticality Analysis (FMECA) extends the FMEA by including a Criticality Analysis. The criticality analysis includes information like the failure rate and failure effect probability. The US Army RAM Engineering and System

Assessment group have been developing Excel-based Failure Mode and Effects Analysis (FMEA) and Failure Mode, Effects and Criticality Analysis (FMECA) tools for the equipment in the maintenance program [Resto11]. The use of an Excel-based approach is not ideal when there is a need for quick and easy access of the information. There is a need to develop a more accessible solution to access the FMEA and FMECA information.

1.2 Developed Solution

The solution presented is a web based solution for the presentation of the FMEA and FMECA information combined with the use of a database for the data storage. The use of a database provides a centralized location for the FMEA and FMECA information and opens new possibilities for the analysis and combination of the data. The use of a web application provides a more accessible solution for accessing the information.

One of the benefits of the use of a database is that the database provides a way to easily manage all the information. The information in all the FMEA and FMECA files can be combined to analyze the data in ways that were not possible using Excel. The information stored in the database can be obtained using different parameters like a part number, it is not restricted to the information in a single excel file. The information for a single part number across all the FMEA and FMECA excel file can be obtained easily from a database by requesting all the excel row with the specified part number. This task using Excel would require to manually open all the FMEA and FMECA files and combine all the rows with the

specified part number across all the FMEA and FMECA excel files in a single excel file which is not feasible specially when the required part number can be changed.

1.3 Work Contributions

The work done contributed to the US Army RAM Engineering and System Assessment group with:

1) The design of the database for the FMEA and FMECA information.
2) The creation of a console based application for inserting the FMEA and FMECA information into a SQL Server database.
3) The creation of web pages for accessing the FMEA and FMECA information.
4) Implement solution to link the FMECA web pages to the applications used by RAM.

1.4 Summary of Following Chapters

We first discuss some of the theory behind the use of FMEA and FMECA in Chapter 2. Chapter 3 talks about the Project Structure. Chapter 4 presents the excel information provided. Chapter 5 presents the design of the database for the storage of the excel information. The C# application used to insert the excel information into the database is presented in Chapter 6. The Web Application developed to access the information is presented in Chapter 7. Chapter 8 discuss about the use of the Visual Basic for Application available in Excel to access information from the database. The conclusions and future work are presented in Chapter 9.

2 LITERATURE REVIEW

2.1 FMECA

Failure Mode, Effects and Criticality Analysis (FMECA) is a reliability analysis technique used to determine the reliability of a product. FMECA was part of the risk management techniques developed for the defense and nuclear industries in the 1940's [Herman11]. In the 1960's the NASA formally developed and applied a FMECA procedure to improve and verify the reliability of the hardware of the space programs. Some of the first commercial industries to adopt these techniques were the petroleum industry, the chemical process industry and the automotive industry. In the 1990's, in response to new regulations and standards of the FDA, the medical device industry began using the FMECA.

FMECA standards can be divided into military and civilian standards [Ying12]. The military FMECA standard is the MIL-STD-1629A. Civilian FMECA standards, such as QS9000 and IEC6082 are used primary by the automotive and electronics industry. Some industries like the medical industries have their own FMECA norms and manuals.

The Military Standard MIL-STD-1629A [MIL-STD-1629A] lists the procedures for performing a Failure Mode, Effects and Criticality Analysis. The purpose of the FMEA, as stated in the MIL-STD-1629A, is to study the results or effects of item failure on system operation and to classify each potential failure according to its severity. This standard defines the fields of the FMEA and FMECA files.

FAILURE MODE AND EFFECTS ANALYSIS

SYSTEM _____
INDENTURE LEVEL _____
REFERENCE DRAWING _____
MISSION _____

DATE _____
SHEET _____ OF _____
COMPILED BY _____
APPROVED BY _____

IDENTIFICATION NUMBER	ITEM/FUNCTIONAL IDENTIFICATION (NOMENCLATURE)	FUNCTION	FAILURE MODES AND CAUSES	MISSION PHASE/ OPERATIONAL MODE	FAILURE EFFECTS			FAILURE DETECTION METHOD	COMPENSATING PROVISIONS	SEVERITY CLASS	REMARKS
					LOCAL EFFECTS	NEXT HIGHER LEVEL	END EFFECTS				

101-11 TASK 101 24 November 1980 MIL-STD-1629A

Figure 1: Example of FMEA worksheet format [MIL-STD-1629A]

The FMEA worksheet fields specified in the MIL-STD-1629A are defined as follow:

1) **Identification Number** – a serial number or other reference designation identification number assigned for traceability purposes.
2) **Item/Functional Identification** – the name or nomenclature of the item or system function being analyzed for failure mode and effects.
3) **Function** – a concise statement of the function performed by the hardware item.
4) **Failure Modes and Causes** – identify and describe all predictable failure modes for each indenture level analyzed.
5) **Mission phase/operational mode** – a concise statement of the mission phase and operational mode in which the failure occurs.
6) **Failure Effect** – identify, evaluate and record the consequences of each assumed failure mode on item operation, function, or status.

7) **Local Effect** – local effects concentrate specifically on the impact an assumed failure mode has on the operation and function of the item in the indenture level under consideration.

8) **Next Higher Level** – next higher level effects concentrate on the impact an assumed failure has on the operation and function of the items in the next higher indenture level above the indenture level under consideration.

9) **End Effects** – end effects evaluate and define the total effect an assumed failure has on the operation, function, or status of the uppermost system.

10) **Failure Detection Method** – a description of the methods by which occurrence of the failure mode is detected by the operator.

11) **Compensating Provisions** – identify and evaluate the compensating provisions, either design provisions or operation actions, which prevent or mitigate the effect of the failure.

12) **Severity Classification** – a severity classification category assigned to each failure mode and item according to the failure effect.

13) **Remarks** – any pertinent remark corresponding to and clarifying any other column in the worksheet.

According to the MIL-STD-1629A the purpose of the criticality analysis (CA) is to rank each potential failure mode identified in the FMEA according to the combined influence of severity classification and its probability of occurrence based upon the best available data.

CRITICALITY ANALYSIS

SYSTEM _____ DATE _____

INDENTURE LEVEL _____ SHEET ____ OF _____

REFERENCE DRAWING _____ COMPILED BY _____

MISSION _____ APPROVED BY _____

IDENTIFICATION NUMBER	ITEM/FUNCTIONAL IDENTIFICATION (NOMENCLATURE)	FUNCTION	FAILURE MODES AND CAUSES	MISSION PHASE/ OPERATIONAL MODE	SEVERITY CLASS.	FAILURE PROBABILITY / FAILURE RATE DATA SOURCE	FAILURE EFFECT PROBABILITY (β)	FAILURE MODE RATIO (α)	FAILURE RATE (λ_p)	OPERATING TIME (t)	FAILURE MODE CRIT # $C_m = \beta\alpha\lambda_p t$	ITEM CRIT # $C_r = \Sigma(C_m)$	REMARKS

MIL-STD-1629A

Figure 2: Example of Criticality Analysis (CA) worksheet format [MIL-STD-1629A]

Criticality Analysis (CA) worksheet fields as specified in the MIL-STD-1629A:

1) **Failure Probability/Failure Rate Data Source** – the failure probability of occurrence level when failure modes are adjudicated in terms of probability of occurrence or the data source of the failure rates used in each calculation when failure rate data are to be used in the calculation of criticality numbers.
2) **Failure Effect Probability** – the conditional probability that the failure effect will result in the identified criticality classification, given that the failure mode occurs.
3) **Failure Mode Ratio** – the fraction of the part failure rate related to the particular failure mode under consideration.

4) **Part Failure Rate** – the part failure rate from the appropriate reliability prediction or as calculated using the procedure described in MIL-HDBK-217.
5) **Operating Time** – the operating time in hours or the number of operating cycles of the item per mission derived from the system definition.
6) **Failure Mode Criticality Number** – the calculated value of the failure mode criticality number.

The objective of the FMECA is to support the decision making process [MIL-STD-1629A]. The analysis needs to provide usable information at or before the decision point. The FMECA serves to verify design integrity, identify and quantify sources of undesirable failure modes, and document the reliability risks [MIL-STD-1629A]. FMECA results can be used to provide the reasons for changes in operating procedures for minimizing the effects or for detecting the start of the undesirable failure modes [MIL-STD-1629A]. The FMECA results are not only used to provide design guidance, but they are used for maintenance planning analysis, logistics support analysis, survivability and vulnerability assessments, safety and hazards analyses, and for fault detection and isolation design [MIL-STD-1629A].

2.2 Web Apps: HTML

In the 1960's the Internet began as an experiment, the goal was to create a resilient information network that could withstand the loss of several computers without preventing the others from communicating [MacDonald10]. The U.S. Department of Defense provided the initial funding driven by potential disaster scenarios [MacDonald10]. The early Internet was mostly limited to educational institutions and defense contractors [MacDonald10]. It

flourished as a tool for academic collaboration, allowing the share of information across the world [MacDonald10]. In the early 1990's modems were created that could work over existing phone lines, and the Internet began to open up to commercial users [MacDonald10]. In the 1993, the firs HTML browser was created, and the Internet revolution began [MacDonald10].

The first generation of websites consisted of mostly of fixed HTML pages that needed to be updated by hand [MacDonald10]. A basic HTML page is like a word-processing document, it contains formatted content that can be displayed on the browser but it doesn't actually do anything [MacDonald10].

An HTML document has two types of content: the text and the elements (or tags) that tell the browser how to format it [MacDonald10]. The elements are recognizable because they are designated with angled brackets (< >) [MacDonald10]. HTML defines elements for different levels of headings, paragraphs, hyperlinks, italic and bold formatting, horizontal lines, and so on [MacDonald10].

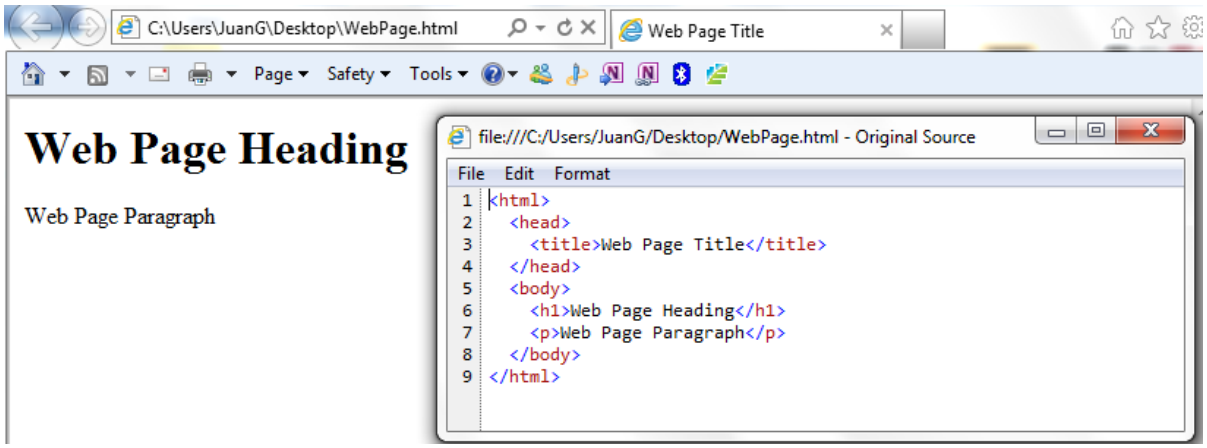


Figure 3: Example HTML Page and HTML Code

HTML 2.0 introduced the first seed of web programming with a technology called HTML forms [MacDonald10]. HTML forms expanded HTML so that it includes not only formatting tags but also tags for graphical widgets, or controls [MacDonald10]. These controls include common ingredients such as drop-down lists, text boxes, and buttons [MacDonald10]. HTML forms allow web developers to design standard input pages [MacDonald10]. When the user clicks a Submit button on a page all the data in the input control is patched together into one long string of text and sent to the web server [MacDonald10]. On the server side, a custom application receives and processes the data [MacDonald10].

2.3 Web Apps: JavaScript

Invented by Netscape back in 1995, JavaScript is nearly as old as the Web. JavaScript is a programming language that provides a way to add to the HTML: animations, interactivity and dynamic visual effects [Sawyer08]. The purpose of JavaScript is to define the behavior of a static HTML page. In the early days, JavaScript suffered from incompatibilities between

the two prominent browsers, Netscape Navigator and Internet Explorer [Sawyer08]. Because Netscape and Microsoft tried to outdo each other's browser by adding newer and better features, the two browsers often acted in very different ways, making it difficult to create JavaScript programs that worked well in both [Sawyer08]. In these days contemporary browsers like Firefox, Safari and Internet Explorer have standardized much of the way they handle JavaScript, making easier to write JavaScript programs that work for most everyone but there are still a few incompatibilities among current Web browsers [Sawyer08].

JavaScript isn't any good without the two other pillars of Web design: HTML and CSS [Sawyer08]. Many programmers talk about the three languages forming the "layers" of a Web page: HTML provides the structural layer, organizing content like pictures and words in a meaningful way; CSS (Cascading Style Sheets) provides the presentational layer, making the content in the HTML look good; and JavaScript adds a behavioral layer, bringing a Web page to life so it interacts with Web visitors [Sawyer08].

JavaScript is a client-side technology doesn't involve any server processing [MacDonald10]. Instead, the complete application is downloaded to the client browser, which executes it locally [MacDonald10].

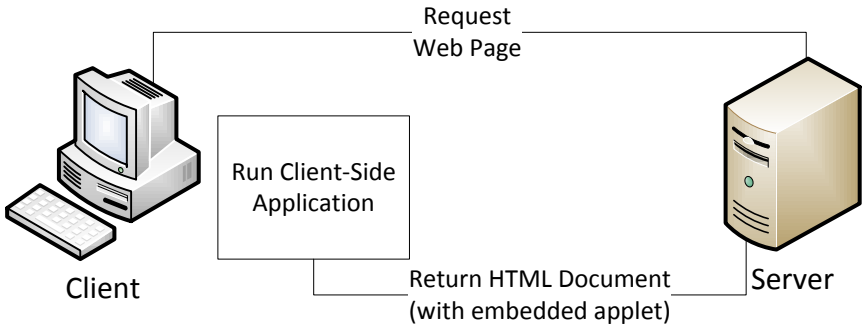


Figure 4: Client-Side Web Application Interaction

2.4 Web Apps: Cascading Style Sheets (CSS)

Large parts of HTML 3.2 and HTML 4.0 were devoted to presentational considerations [Meyer06]. The ability to color and size text through the font element, to apply background colors and images to documents and tables, to use table attributes (such as cellspacing), and to make text blink on and off [Meyer06]. For most sites, the markup was almost entirely comprised of tables and font elements, neither of which conveys any real semantic meaning as to what's being presented [Meyer06]. The problem of polluting HTML with presentational markup was not lost on the World Wide Web Consortium (W3C), which began searching for a quick solution [Meyer06]. In 1995, the consortium started publicizing a work-in-progress called CSS [Meyer06]. By 1996, it had become a full Recommendation, with the same weight as HTML itself [Meyer06].

CSS allows for much richer document appearance than HTML ever allowed [Meyer06]. CSS lets you set colors on text and in the background of any element; permits the creation of borders around any element, as well as the increase or decrease of the space around them;

lets you change the way text is capitalized, decorated, spaced, and even whether it is displayed at all; and allows you to accomplish many other effects [Meyer06]. The style sheets centralize the commands for certain visual effects in one place, instead of scattering them throughout the document [Meyer06]. Not only can you centralize all of the style information for a page in one place, but you can also create a style sheet that can be applied to multiple pages [Meyer06]. This is accomplished by a process in which a style sheet is saved to its own document and then imported by any page for use with that document [Meyer06]. Using this capability, you can quickly create a consistent look for an entire web site [Meyer06].

HTML is a structural language, while CSS is its complement: a stylistic language [Meyer06]. Recognizing this, the W3C, the body that debates and approves standards for the Web, is beginning to remove stylistic elements from HTML [Meyer06].

2.5 ASP.NET

ASP.NET is a web programming technology pioneered by Microsoft that allows developers to create dynamic web pages [Mitchell10]. The web pages whose content is determined dynamically based on user input or other information are called dynamic web pages [Mitchell10].

ASP.NET version 1.0 was released in January 2002 [Mitchell10]. In November 2005, Microsoft released the version 2.0 [Mitchell10]. Two years later, in November 2007,

Microsoft released ASP.NET version 3.5 and ASP.NET 4 was unveiled in April 2010 [Mitchell10].

ASP.NET is designed as a server-side technology [MacDonald10]. All ASP.NET code executes on the server [MacDonald10]. When the code is finished executing, the user receives an ordinary HTML page, which can be viewed in any browser [MacDonald10].

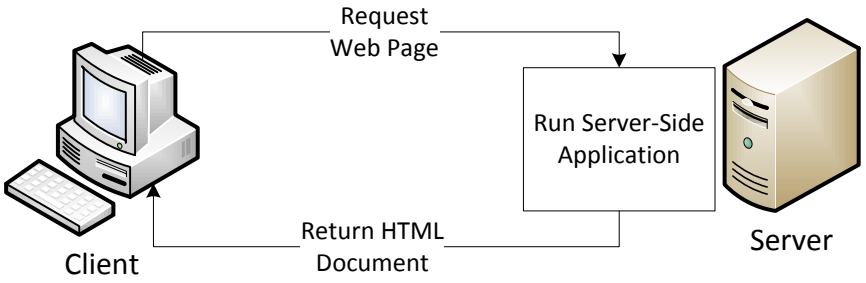


Figure 5: Server-Side Web Application Interaction

2.6 Databases

A database is nothing more than a set of related information [Beaulier09]. Because of the cumbersome nature of paper databases, some of the first computer applications developed were database systems, which are computerized data storage and retrieval mechanisms [Beaulier09]. Because a database system stores data electronically rather than on paper, a database system is able to retrieve data more quickly, index data in multiple ways, and deliver up-to-minute information to its user community [Beaulier09].

Early database systems managed data stored on magnetic tapes [Beaulier09]. Because there were generally far more tapes than tape readers, technicians were tasked with loading and unloading tapes as specific data was requested [Beaulier09]. Because the computers of the

era had very little memory, multiple requests for the same data generally required the data to be read from the tape multiple times [Beaulier09].

Over the first several decades of computerized database systems, data was stored and represented to users in various ways [Beaulier09]. In a hierarchical database system, for example, data is represented as one or more tree structures [Beaulier09]. Another common approach, called the network database system, exposes sets of records and sets of links that define relationships between different records [Beaulier09].

In 1970, Dr. E. F. Codd of IBM's research laboratory published a paper titled "A Relational model of Data for Large Shared Data Banks" that proposed that data be represented as sets of tables [Beaulier09]. Rather than using pointers to navigate between related entities, redundant data is used to link records in different tables [Beaulier09]. Each table in a relational database includes information that uniquely identifies a row in the table (known as the primary key), along with additional information needed to describe the entity completely [Beaulier09]. Some of the tables also include information used to navigate to another table ("redundant data") [Beaulier09].

2.7 SQL

Along with Codd's definition of the relational model, he proposed a language called DSL/Alpha for manipulation the data in relational tables [Beaulier09]. Shortly after Codd's paper was released, IBM commissioned a group to build a prototype based on Codd's ideas [Beaulier09]. This group created a simplified version of DSL/Alpha that they called

SQUARE [Beaulier09]. Refinements to SQUARE led to a language called SEQUEL, which was, finally, renamed SQL [Beaulier09].

In the mid-1980s, the American National Standards Institute (ANSI) began working on the first standard for the SQL language, which was published in 1986 [Beaulier09]. Along with refinements to the core language, new features have been added to the SQL language to incorporate object-oriented functionality, among other things [Beaulier09].

SQL goes hand in hand with the relational model because the result of an SQL query is a table [Beaulier09]. Thus, a new permanent table can be created in a relational database simply by storing the result set of a query [Beaulier09]. Similarly, a query can use both permanent tables and the result sets from other queries as inputs [Beaulier09].

2.8 SQL Server

SQL Server is a Relational Database Management System (RDBMS). A RDBMS not only stores the data, they also manage that data, restricting the kind of data that can go into the system, and facilitating getting data out of the system [Atkinson]. RDBMSs allows to go beyond the storage of the data into the realm of defining what that data should look like, or the business rules of the data [Atkinson]. A database is typically a group of constructs that include at least a set of table objects and, more often than not, other objects, such as stored procedures and views that pertain to the particular grouping of data stored in the database's tables [Atkinson]. An RDBMS may have multiple databases on just one server, or it may have only one [Atkinson].

2.9 SQL Server Reporting Services

Reports are generated and consumed by many different IT professionals as well as non-IT users [Krishnaswamy09]. Report developers, decision makers, power users, and end users consume reports in many formats on many different kinds of platforms [Krishnaswamy09]. Interactive reports are extensively used in enterprises from just look ups to making business decisions and forecasting [Krishnaswamy09].

Microsoft SQL Server Reporting Services (SSRS) is the reporting software from Microsoft [Krishnaswamy09]. SSRS is a dedicated server-based, one-stop platform that handles reporting related activities to deliver focused information where needed in a timely and flexible manner [Krishnaswamy09]. Microsoft has also integrated SSRS with Office and Share Point Server [Krishnaswamy09].

In SQL Server Reporting Services 2005 there were two services, one unattended windows service and an on demand web service [Krishnaswamy09]. The Windows service looked after reporting processing, scheduling and delivery, database maintenance and extensibility [Krishnaswamy09]. The web service worked with SOAP, URL access, Report Manager, Report processing, Report Models and extensibility [Krishnaswamy09]. In SSRS 2008, there is just one Window service and there is no dependence on IIS [Krishnaswamy09]. Everything that was in the web service in now present in the Windows service. This was achieved by replacing the functionality provided by IIS with native SQL Server components [Krishnaswamy09]. The reporting services (Windows service) are comprised of the web-based Report Manager Client, the Web Server Web Service and the background processing

[Krishnaswamy09]. While the Report Manager provides the ASP.NET and UI pages, the Report Server web service works with ASP.NET, SOAP, and URL interfaces [Krishnaswamy09]. The Report Server is at the heart of the Reporting Services which is implemented as a Windows service [Krishnaswamy09].

Visual Studio is tightly integrated with SQL Server, both of which leverage the .NET framework [Krishnaswamy09]. Visual Studio provides all the functionality available in the Business Intelligence Development Studio (BIDS) as well as the ability to integrate reports with both web and Windows programs [Krishnaswamy09]. Visual Studio also supports developing applications that can be hosted on Internet Information Services servers [Krishnaswamy09]. They can then be processed locally without the need for Report Server [Krishnaswamy09]. These client processed reports with the extension RDLC can be immediately deployed for intranet-wide distribution from IIS Servers [Krishnaswamy09]. This support is provided by making available the ReportViewer controls [Krishnaswamy09]. The usage of these controls makes it possible to create reports that can be viewed inside the application without going through the Report Server [Krishnaswamy09].

3 Project Structure

3.1 Project Objectives

The project objectives are:

1) Design database tables to insert the FMEA and FMECA information. Needs to be able to generate the FMEA and FMECA files.
2) Create console based C# application to read the FMEA and FMECA files and store the information into the database.
3) Create web pages to display the FMEA and FMECA information obtained from the database. Provide a way to link any page to the FMEA and FMECA web pages.
4) Provide exporting options to Excel and PDF.

3.2 Project Components

The project components are the Excel-based FMEA and FMECA files, the application that reads the Excel files and store them into the SQL database, the web application that display the FMEA and FMECA information and the reports that provide the exporting options.

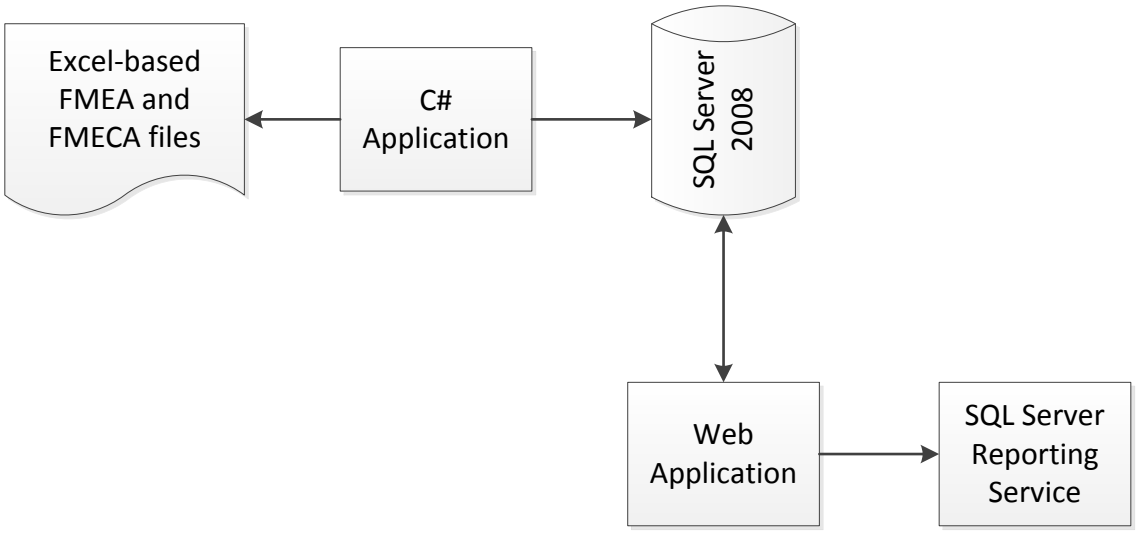


Figure 6: Project Structure diagram (Project Components)

3.3 Excel-based FMEA and FMECA files

The Excel-based FMEA and FMECA files are the excel files with the analysis of failures for the different systems that needs to be inserted into the SQL database.

3.4 C# Application

The C# Application is a console based application that reads the Excel-based FMEA and FMECA files and stores them into the SQL Server database. The application at least need to receive as an input the file or folder with the excel files and the type of the file. The type of the file is needed to identify the format of the selected excel files.

3.5 Web Application

The Web Application displays the selected FMEA or FMECA information in a web page. To provide the exporting to Excel and PDF options of the FMEA and FMECA the web application include a report for each file format that can be used to export to PDF or Excel the data of a selected FMEA or FMECA.

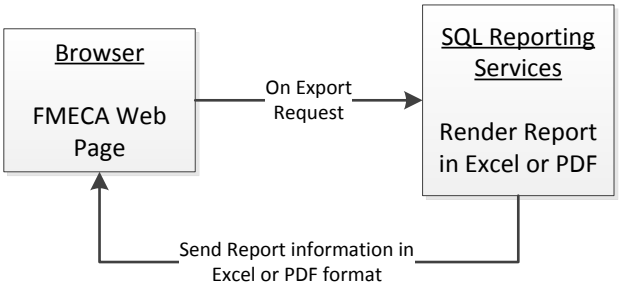


Figure 7: Export Request Interaction Diagram

3.6 SQL Server Reporting Services

The SQL Server Reporting Services are a group of reports used to export the FMEA and FMECA information to Excel or PDF. The reports were created locally in the web application. Visual Studio provides a way to create reports inside the web application without the need of using a Report Server. The reports are created inside the web application by creating a Report file, specified the data set with the report columns, drag and drop the columns to the report and applying any required formatting to create the required report.

3.7 Technologies

The implementation of the project required the use of the following technologies:

1) SQL (SQL Server) – query language used to perform database operations: insert, search, update.
2) C# - programming language selected to write the console application that inserts the FMEA and FMECA information into the database. C# is also the language selected for the code behind the ASP.NET web pages.
3) ASP.NET – the web application framework used to create the web pages.
4) HTML – language for creating the web pages. HTML is included in the ASP.NET web pages alongside the ASP.NET web controls. The ASP.NET web controls are translated in the server side to HTML and Java Script code that implement the web control functionality. Content layer of the web page.
5) CSS – used in the web application to change the appearance of the page. Presentation layer of the web page.
6) Java Script – used to create links to the pages with the FMEA and FMECA information. Some ASP.NET web controls generate Java Script code. Behavior layer of the web page.
7) SQL Server Reporting Services – used to create reports to provide exporting options.
8) T-SQL (Transact-SQL) – used to concatenate list of WUC.
9) Visual Basic for Applications (VBA) – visual basic available in Excel to use in macros. Used to access the database and obtain the FMEA information.

For the development of the project the following tools where used:

1) Visual Studio 2008 – used for developing the C# Application and the web application including the creation of reports.
2) SQL Server 2008 – used as database system. Two IDE include in SQL Server where used:
    a. SQL Server Management Studio – used as the IDE for the management of the database operations: creation of stored procedures and query testing.
    b. SQL Server Business Intelligence Development Studio – an IDE similar to Visual Studio but with options to create report project using the SQL Server Reporting Services. Used for the development of sample reports before creating the FMEA and FMECA report inside the web application.
3) Excel – the Visual Basic provided by Excel was used to access the database to obtain the FMEA information and to add the information to the excel file.

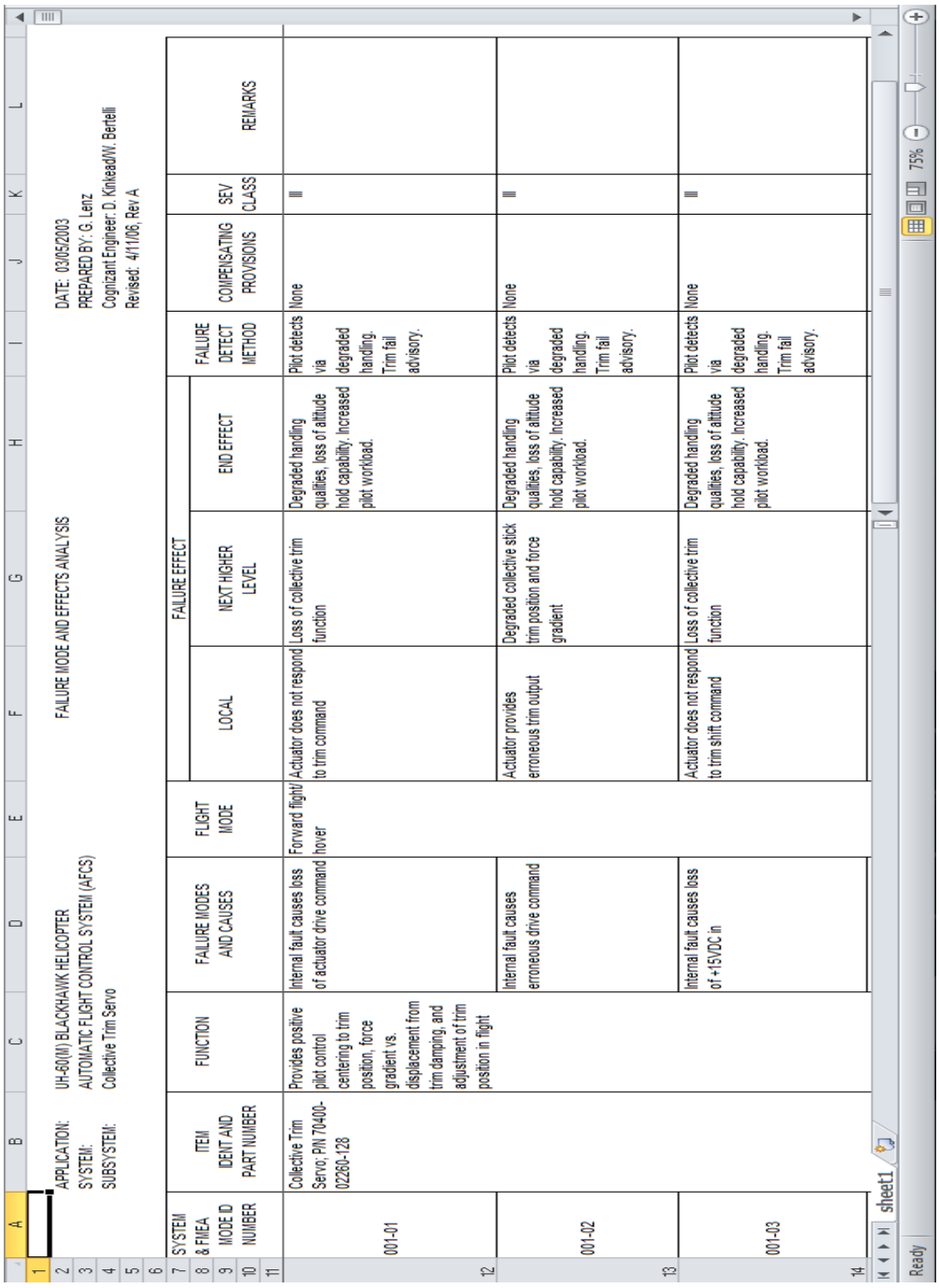
4 FMEA and FMECA Files

4.1 Failure Modes and Effects Analysis (FMEA) files

The FMEA files provided by the US Army RAM Engineering and System Assessment Group contain a single sheet with the FMEA information. The information provided in each file consists of:

1) A header with the Application, System, Subsystem, Date, R&M Engineer, Cognizant Engineer and Revised Date.

2) A table with the FMEA information: System & FMEA Mode ID Number, Item Identification and Part Number, Function, Failure Modes, Mission Phase/Operational Mode, Local Effect, Next Higher Level Effect, End Effect, Failure Detect method, Compensating Provisions, Severity Class and Remarks.



APPLICATION: UH-60(M) BLACKHAWK HELICOPTER
SYSTEM: AUTOMATIC FLIGHT CONTROL SYSTEM (AFCS)
SUBSYSTEM: Collective Trim Servo

FAILURE MODE AND EFFECTS ANALYSIS

DATE: 03/05/2003
PREPARED BY: G. Lenz
Cognizant Engineer: D. Kirkead/W. Bertelli
Revised: 4/11/06, Rev A

SYSTEM & FMEA MODE ID NUMBER	ITEM IDENT AND PART NUMBER	FUNCTION	FAILURE MODES AND CAUSES	FLIGHT MODE	FAILURE EFFECT			FAILURE DETECT METHOD	COMPENSATING PROVISIONS	SEV CLASS	REMARKS
					LOCAL	NEXT HIGHER LEVEL	END EFFECT				
001-01	Collective Trim Servo; P/N 70400-022260-128	Provides positive pilot control centering to trim position, force gradient vs. displacement from trim damping, and adjustment of trim position in flight	Internal fault causes loss of actuator drive command	Forward flight/ hover	Actuator does not respond to trim command	Loss of collective trim function	Degraded handling qualities, loss of altitude hold capability. Increased pilot workload.	Pilot detects via degraded handling. Trim fail advisory.	None	III	
001-02			Internal fault causes erroneous drive command		Actuator provides erroneous trim output	Degraded collective stick trim position and force gradient	Degraded handling qualities, loss of altitude hold capability. Increased pilot workload.	Pilot detects via degraded handling. Trim fail advisory.	None	III	
001-03			Internal fault causes loss of +15VDC in		Actuator does not respond to trim shift command	Loss of collective trim function	Degraded handling qualities, loss of altitude hold capability. Increased pilot workload.	Pilot detects via degraded handling. Trim fail advisory.	None	III	

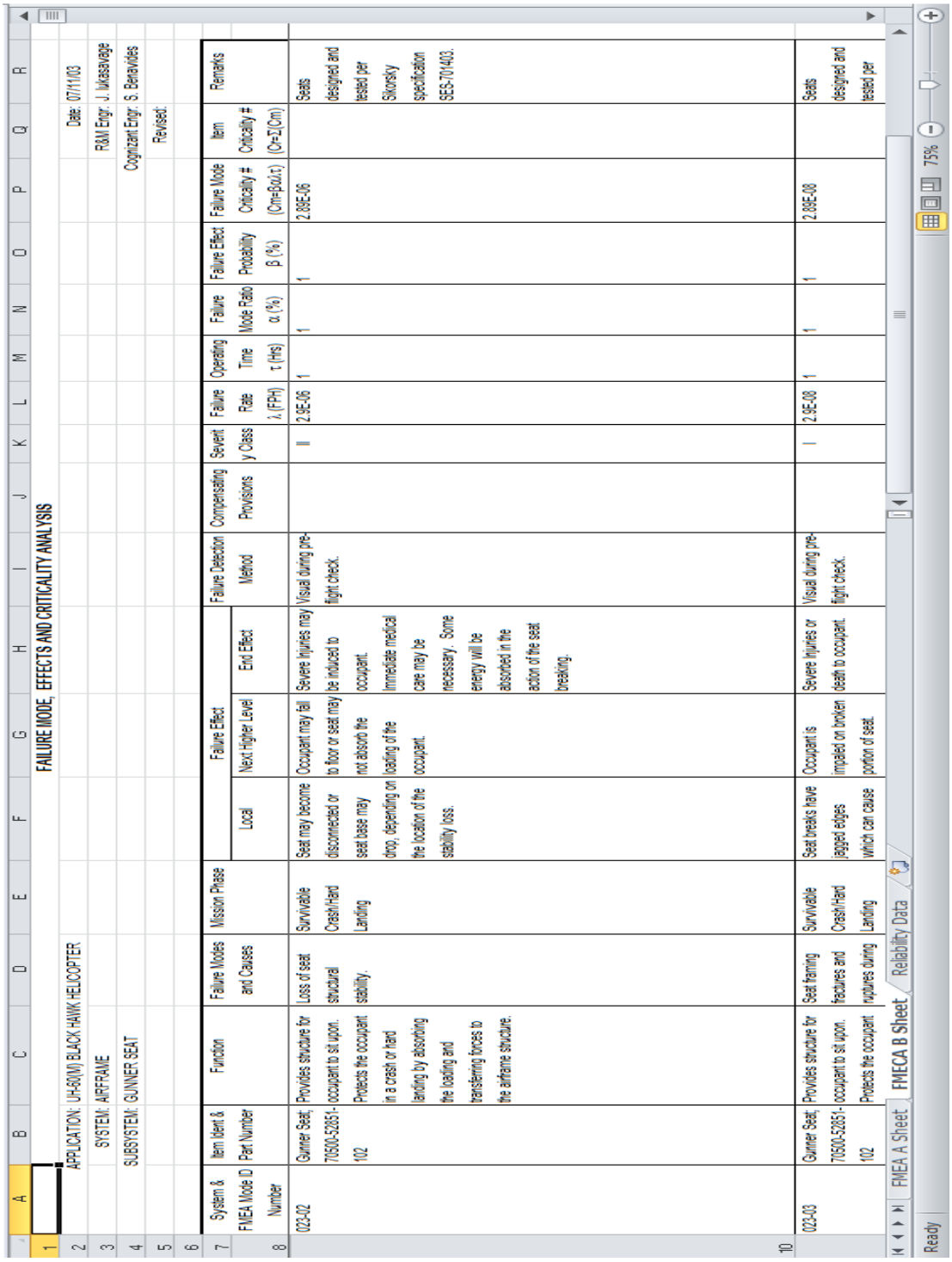
Figure 8: Example of FMEA excel file provided by [USArmyRAM].

4.2 Failure Mode, Effects and Criticality Analysis (FMECA) files

The FMECA files provided by the US Army RAM Engineering and System Assessment Group contain three sheets:

1) A sheet with the FMEA information.
2) A sheet with the FMECA information
3) A sheet with the Reliability Data

The header information and the FMEA information in the FMECA sheet comes directly from the FMEA sheet using a reference to the FMEA sheet. The information in the Reliability Data sheet is used to obtain some of the Criticality Analysis data that is contained in the FMECA sheet.



FAILURE MODE, EFFECTS AND CRITICALITY ANALYSIS

APPLICATION: UH-60(M) BLACK HAWK HELICOPTER
SYSTEM: AIRFRAME
SUBSYSTEM: GUNNER SEAT

Date: 07/11/03
R&M Engr: J. Iwasavage
Cognizant Engr: S. Benavides
Revised:

System & FMEA Mode ID Number	Item Ident & Part Number	Function	Failure Modes and Causes	Mission Phase	Failure Effect: Local	Failure Effect: Next Higher Level	Failure Effect: End Effect	Failure Detection Method	Compensating Provisions	Severity Class	Failure Rate λ (FPH)	Operating Time τ (Hrs)	Failure Mode Ratio α (%)	Failure Effect Probability β (%)	Failure Mode Criticality # (Cm=βαλτ)	Item Criticality # (Cr=Σ(Cm))	Remarks
023-02	Gunner Seat; 70500-52651-102	Provides structure for occupant to sit upon. Protects the occupant in a crash or hard landing by absorbing the loading and transferring forces to the airframe structure.	Loss of seat structural stability.	Survivable Crash/Hard Landing	Seat may become disconnected or seat base may drop, depending on the location of the stability loss.	Occupant may fall to floor or seat may not absorb the loading of the occupant.	Severe injuries may be induced to occupant. Immediate medical care may be necessary. Some energy will be absorbed in the action of the seat breaking.	Visual during pre-flight check.		II	2.9E-06	1	1	1	2.89E-06		Seats designed and tested per Sikorsky specification SES-701403.
023-03	Gunner Seat; 70500-52651-102	Provides structure for occupant to sit upon. Protects the occupant	Seat framing fractures and ruptures during	Survivable Crash/Hard Landing	Seat breaks have jagged edges which can cause	Occupant is impaled on broken portion of seat.	Severe injuries or death to occupant.	Visual during pre-flight check.		I	2.9E-08	1	1	1	2.89E-08		Seats designed and tested per

FMEA A Sheet | FMECA B Sheet | Reliability Data

Ready

Figure 9: Example of FMECA excel file provided by [USArmyRAM].

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
1	NOMENCLATURE	WUC	UMA/KFH	INH/KFH	IND/KFH	NODEF/KFH	EMA/KFH	MAF/KFH	MTTRu	MTTRe	MTTRu X UMA	MA X MTTRe	GA/KFH	AA/KFH	CS	MM
2	AIRFRAME	02	123.70028	93.41227	23.14586	7.14215	38.99266	8.54113	1.71	3.11	210.93509	121.09729	1.72323	0.36241	1.22	
3	FORWARD FUSELAGE SECTION	02A	26.90702	19.27171	5.67718	1.95813	8.84838	2.32839	2.11	4.00	56.87350	35.37630	0.34640	0.00059	1.24	
4	WINDOW/WINDSHIELD INSTALL	02A01	7.74239	6.39159	0.79071	0.56009	3.59115	0.82366	4.27	6.13	33.03336	22.00403	0.26357	0.00000	1.35	
5	PILOTS WINDSHIELD	02A01A	1.35080	1.21901	0.06589	0.06589	0.59303	0.19768	6.74	10.18	9.10966	6.03796	0.06589	0.00000	1.44	
6	CO-PILOTS WINDSHIELD	02A01B	1.74616	1.54848	0.00000	0.19768	0.79071	0.32946	5.68	8.37	9.90971	6.61617	0.09884	0.00000	1.47	
7	UPPER OVERHEAD WINDOW	02A01C	1.58142	1.28491	0.09884	0.19768	0.82366	0.03295	3.71	5.22	5.87103	4.29785	0.03295	0.00000	1.12	
8	LOWER NOSE WINDOW	02A01D	1.31785	1.12018	0.13179	0.06589	0.79071	0.03295	2.61	3.41	3.43465	2.69501	0.00000	0.00000	1.04	
9	CENTER PANEL WINDSHIELD	02A01E	1.08723	0.95544	0.09884	0.03295	0.46125	0.16473	3.39	4.02	3.69026	1.85296	0.06589	0.00000	1.61	
10	WINDOW/WINDSHIELD INST	02A0199	0.65893	0.26357	0.39536	0.00000	0.13179	0.06589	1.55	3.83	1.01804	0.50408	0.00000	0.00000	1.00	
11	FWD FUSELAGE DOOR INST	02A02	8.92845	6.09507	2.17446	0.65893	1.91089	0.39536	0.74	1.10	6.57279	2.09868	0.06589	0.00000	1.03	
12	NOSE ACCESS DOOR ASSY	02A02A	1.35080	0.98839	0.23062	0.13179	0.13179	0.00000	0.76	1.08	1.02793	0.14167	0.00000	0.00000	1.06	
13	PILOT DOOR ASSY	02A02B	4.34891	2.70160	1.21901	0.42830	1.05428	0.29652	0.68	0.55	2.96846	0.57986	0.03295	0.00000	1.03	
14	PILOT DOOR WINDOW	02A02B01	0.62598	0.52714	0.06589	0.03295	0.06589	0.03295	1.21	0.95	0.75447	0.06260	0.00000	0.00000	1.10	
15	PILOT DOOR LATCH ASSY	02A02B02	0.23062	0.09884	0.13179	0.00000	0.06589	0.00000	0.50	0.45	0.11531	0.02965	0.00000	0.00000	1.00	
16	PILOT DOOR JETTISON	02A02B03	0.59303	0.16473	0.42830	0.00000	0.49419	0.09884	0.56	0.55	0.32946	0.27345	0.00000	0.00000	1.00	
17	KEY AND LOCK SET	02A02B05	1.05428	0.69187	0.16473	0.19768	0.00000	0.00000	0.90	0.00	0.95215	0.00000	0.00000	0.00000	1.00	
18	PILOT DOOR ASSY (NOC)	02A02B99	1.84499	1.21901	0.42830	0.19768	0.42830	0.16473	0.44	0.50	0.81707	0.21415	0.03295	0.00000	1.03	
19	CO-PILOT DOOR ASSY	02A02C	3.09695	2.27330	0.72482	0.09884	0.69187	0.09884	0.81	1.93	2.52039	1.33762	0.03295	0.00000	1.00	
20	CO-PILOT DOORWINDOW	02A02C01	0.32946	0.29652	0.03295	0.00000	0.09884	0.00000	1.52	2.80	0.50078	0.27675	0.00000	0.00000	1.00	
21	CO-PILOT DOORLATCH	02A02C02	0.19768	0.09884	0.09884	0.00000	0.06589	0.00000	0.63	0.90	0.12520	0.05930	0.00000	0.00000	1.00	
22	CO-PILOT DOOR JETTISON MECH	02A02C03	0.39536	0.09884	0.29652	0.00000	0.23062	0.03295	0.38	0.49	0.15155	0.11202	0.03295	0.00000	1.00	
23	KEY AND LOCK SET	02A02C05	0.26357	0.19768	0.03295	0.03295	0.00000	0.00000	0.50	0.00	0.13179	0.00000	0.00000	0.00000	1.00	
24	CO-PILOT DOOR ASSY (NOC)	02A02C99	1.91089	1.58142	0.26357	0.06589	0.29652	0.06589	0.84	3.00	1.61108	0.88955	0.00000	0.00000	1.00	
25	FWD FUSELAGE DOOR INST (NOC)	02A0299	0.13179	0.13179	0.00000	0.00000	0.03295	0.00000	0.43	1.20	0.05601	0.03954	0.00000	0.00000	1.59	
26	PILOT/CO-PILOT SEAT	02A03	0.84394	0.63185	0.10319	0.10890	0.28745	0.14864	0.63	0.45	0.53148	0.12888	0.01517	0.00000	1.02	
27	SEAT RESTRAINT SYSTEM	02A03A	0.30410	0.19490	0.03206	0.07714	0.16843	0.10389	0.61	0.63	0.18642	0.10580	0.01517	0.00000	1.01	
28	LAP BELT	02A03A01	0.06797	0.05438	0.01359	0.00000	0.04078	0.02719	0.46	0.43	0.03127	0.01767	0.00000	0.00000	1.00	
29	SHOULDER HARNESS	02A03A02	0.10876	0.05438	0.00000	0.05438	0.05438	0.02719	0.70	0.68	0.07613	0.03671	0.00000	0.00000	1.00	
30	INERTIA REEL	02A03A03	0.09442	0.07296	0.00858	0.01288	0.06008	0.04292	0.61	0.76	0.05794	0.04549	0.00858	0.00000	1.02	

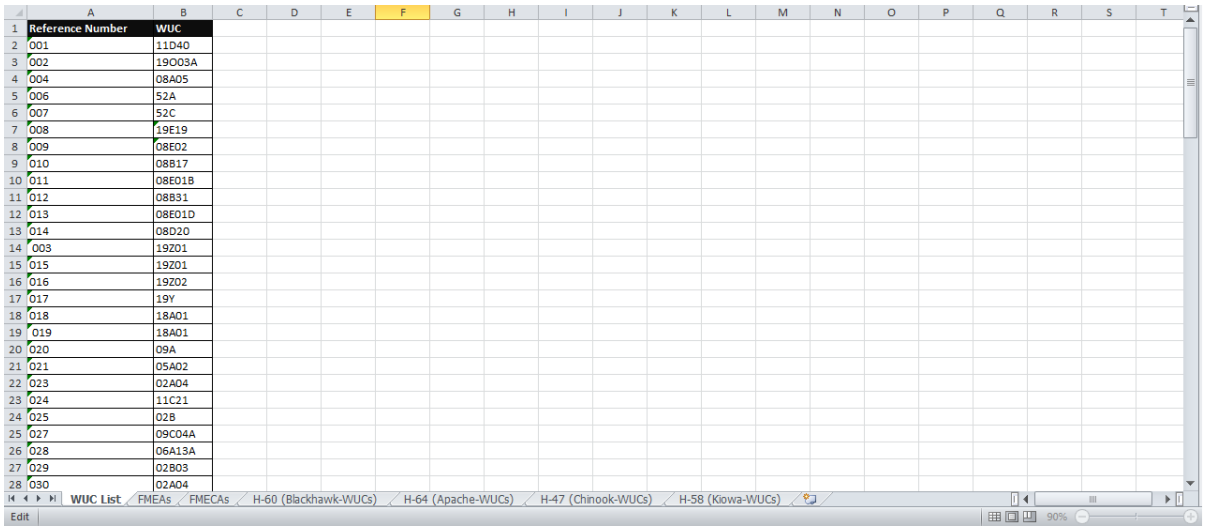
FMEA A Sheet | FMECA B Sheet | Reliability Data

Edit 100%

Figure 10: Example of Reliability Data in the FMECA excel file provided by [USArmyRAM].

4.3 FMEA and FMECA WUC Reference List

The FMEA and FMECA WUC Reference List is a file with the WUC (Work Unit Code) for each System or File. The information contained is the Reference Number that can be used to identify the FMEA or FMECA file and the WUC that is the value that needs to be added to the FMEA and FMECA files.



Reference Number	WUC
001	11D40
002	19O03A
004	08A05
006	52A
007	52C
008	19E19
009	08E02
010	08B17
011	08E01B
012	08B31
013	08E01D
014	08D20
003	19Z01
015	19Z01
016	19Z02
017	19Y
018	18A01
019	18A01
020	09A
021	05A02
023	02A04
024	11C21
025	02B
027	09C04A
028	06A13A
029	02B03
030	02A04

Figure 11: Example of WUC Reference List excel file provided by [USArmyRAM].

5 Database Design

5.1 Database Schema

The database schema consists of all the tables needed to generate the FMEA and FMECA files. To generate the files a way to identify a single excel file and a single row in the excel file is needed to be able to associate every column in a row to the row and to be able to identify if a row is already inserted. Every FMEA and FMECA files has a system, the system number can be used to identify the data of a single excel file and every row of data has a System and Mode Id that can be used to identify a single row in the excel file. The idea behind the design of the database schema is to use the System Number to identify the excel file and the System and Mode Id to identify a row in the excel file. To be able to generate the FMEA and FMECA files every column in the excel row needs to be associated to the System and Mode Id.

The database schema consist of the Aircraft for the System in the FMEA or FMECA file, the System that is used to identify a single FMEA or FMECA file, the information in the header of the files, the System and Mode Id for the rows of the files, the Criticality Analysis of the FMECA files that is calculated for the row of the files and the data in the rows of the files: Part, Function, Failure, Failure Effect, Flight Mode, Severity Class and Detect Method. There is also a table for the WUC (Work Unit Code) of the System with the relation that a single System can have more than one WUC.

The database schema was designed to be able to generate the excel files, this require that a way to identify a row and to associate each column in the excel file to that row. The database schema developed is not the typical schema on which the relations between the tables follow a specific order from top to bottom like: an Aircraft has Systems, a System has Part, a Part has Failure, a Failure has a Severity, etc. because this type of relations between the tables cannot be used to generate the excel files because with this type of relation between tables we can know information like all the defects of a Part but to be able to generate the files we need the Part in a specific row in the excel and the Defect in the same row. The basic idea is to be able to generate a single row of the data in the excel file.

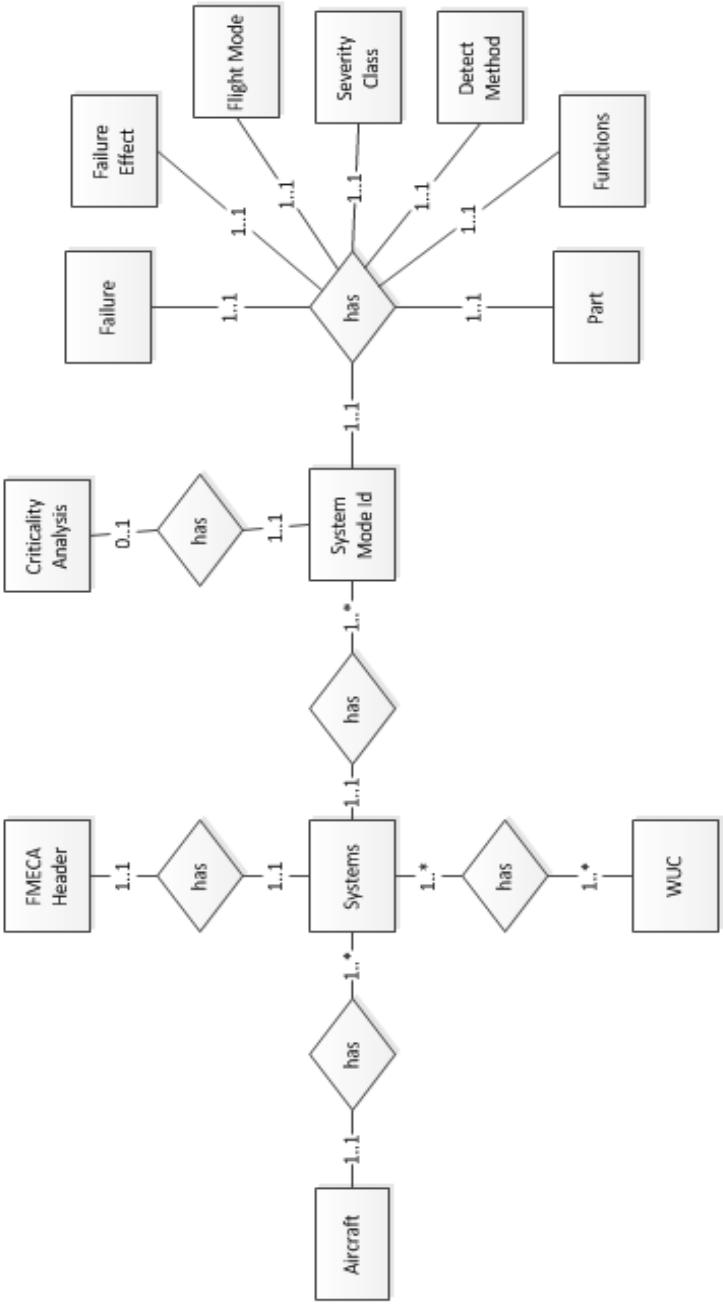


Figure 12: Database Schema design diagram

5.2 Database Tables

In the database schema the only relation many to many between tables is the relation between the System and the WUC Entities. This relation translate to a table for the relation, all the other relations do not need a table. The implementation of the defined schema translates to a table for each entity and a table for the relation between the System and WUC entities.

One important decision in the design of the tables is that to be able to generate the excel files each column in a row has to be associated to the row. The values in the columns are usually string values and some of them can contain paragraphs like the Function. The values of the columns can be repeated multiple times in the rows of the excel files. To avoid repetition columns like the Functions have their own table which is associated to the row using the table key. We used an auto increment integer value as the key of all tables. We made this design choice to avoid having a table key that is a large string like it would be the case for the Functions table.

The following are a description of the tables created and the list of the field of the tables:

1) **AIRCRAFT** – table with the name of the Aircraft of the System in the FMEA or FMECA file. Table Fields:
    a. AIRCRAFT_KEY – auto increment integer used as the key of the table.
    b. AIRCRAFT_NAME – the name of the aircraft.
2) **SYSTEMS** – the System of the FMEA or FMECA file. Used to identify a single excel file. Table Fields:
    a. SYSTEM_KEY – auto increment integer used as the key of the table.

b. SYSTEM – System Number

c. AIRCRAFT_KEY – foreign key from the Aircraft Table.

3) **FMECA_HEADER** – information in the header of the FMEA files. It was decides to put in a different table the header of the files instead that in the System table. Table fields:

a. SYSTEM_KEY – foreign key from the System Table. Used as table Key.

b. APPLICATION – application field in the header of the FMEA file.

c. SYSTEM – system name in the header of the FMEA file.

d. SUBSYSTEM – subsystem name in the header of the FMEA file.

e. DATE – date field in the header of the FMEA file.

f. RM_ENGR – RM engineer field in the header of the FMEA file.

g. CONGNIZANT_ENGR – Cognizant Engineer field in the header of the FMEA file.

h. REVISED – revised or updated date in the header of the FMEA file.

4) **SYSTEM_WUC** – relation between the System and the WUC (Work Unit Code). Table Fields:

a. SYSTEM_KEY – foreign key from the System Table.

b. WUC_KEY – foreign key from the WUC Table.

5) **WUC** – table with the WUCs. Table Fields:

a. WUC_KEY – auto increment integer used as the key of the table.

b. WUC – Work Unit Code

6) **SYSTEM_MODE_ID** – System and Mode Id from the rows in the FMEA and FMECA files. The System identify the excel file and the Mode Id a row in the file. Table Fields:

   a. SYSTEM_MODE_ID_KEY – auto increment integer used as the key of the table.

   b. SYSTEM_MODE_ID – system mode id in the format SYSTEM-MODE ID.

   c. SYSTEM – System number in the System Mode Id.

   d. MODE_ID – Mode Id number in the System Mode Id

   e. SYSTEM – foreign key from the System table.

7) **CRITICALITY_ANALYSIS** – criticality analysis data contained only in the FMECA files. Table Fields:

   a. SYSTEM_M ODE_ID_KEY – foreign key from the System Mode Id Table. Used as key of the table.

   b. FAILURE_RATE – failure rate field in the criticality analysis data.

   c. OPERATING_TIME – operating time field in the criticality analysis data.

   d. FAILURE_MODE_RATIO – failure mode ratio in the criticality analysis data.

   e. FAILURE_EFFECT_PROBABILITY – failure effect probability in the criticality analysis data.

   f. FAILURE_MODE_CRITICALITY – failure mode criticality field in the criticality analysis data.

g. ITEM_CRITICALITY – item criticality field in the criticality analysis data.

8) **SYSTEM_FAILURE** – relation between the System Mode Id that represents the rows in the FMEA and FMECA files and the columns of the row. Table Fields:

a. PART_KEY – foreign key from the Part table.

b. FUNCTION_KEY – foreign key from the Function table.

c. FAILURE_KEY – foreign key from the Failure table.

d. FAILURE_EFFECT_KEY – foreign key from the Failure Effect table.

e. SEVERITY_CLASS_KEY – foreign key from the Severity Class table.

f. DETECT_METHOD_KEY – foreign key from the Detect Method table.

g. FLIGHT_MODE_KEY – foreign key from the Flight Mode table.

h. SYSTEM_MODE_ID_KEY – foreign key from the System Mode Id table. Used as key of the table, more restrictive than using all table columns as key.

9) **PART** – part description. Table Fields:

a. PART_KEY – auto increment integer used as the key of the table.

b. PART_DESC – part description in the format Identification ; Part Number

c. IDENT – identification part of the part description

d. PART_NUMBER – part number of the part description

10) **FUNCTIONS** – part functions. Table Fields:

a. FUNCTION_KEY – auto increment integer used as key of the table.

b. FUNCTION_DESC – function description

11) **FAILURE** – failure information. Table Fields:

a. FAILURE_KEY – auto increment integer used as key of the table.

b. FAILURE_MODE – failure description.

c. REMARKS – failure comment.

d. COMPENSATING_PROVISION – failure compensating provision.

12) **FAILURE_EFFECT** – failure effect information. Table Fields:

a. FAILURE_EFFECT_KEY – auto increment integer used as key of the table.

b. LOCAL_EFFECT – failure local effect.

c. NEXT_LEVEL_EFFECT – failure next level effect.

d. END_EFFECT – failure end effect.

13) **FLIGHT_MODE** – flight mode information. Table Fields:

a. FLIGHT_MODE_KEY – auto increment integer used as the key of the table.

b. FLIGHT_MODE_DESC – flight mode in the format Mission Phase \ Operational Mode.

c. MISSION_PHASE – mission phase part of the flight mode.

d. OPERATIONAL_MODE – operational mode of the flight mode.

14) **SEVERITY_CLASS** – severity class information. Table Fields:

a. SEVERITY_CLASS_KEY – auto increment integer used as the key of the table.

b. SEVERITY_CLASS – severity class.

15) **DETECT_METHOD** – detect method information. Table Fields:

a. DETECT_METHOD_KEY – auto increment integer used as the key of the table.

b. DETECT_METHOD – detect method.

16) **RELIABILITY_DATA** – information in the Reliability Data information in the FMECA files. This information is used to obtain the criticality analysis information. Table Fields:

a. NOMENCLATURE – System name. Used as key of the table.

b. Reliability Data Fields: UMA_KFH, INH_KFH, IND_KFH, NODEF_KFH, EMA_KFH, MAF_KFH, MTTRU, MTTRE, MTTRU_X_UMA, EMA_X_MTTRE, GA_KFH, AA_KFH, CS, MMH_KFH, REM_KFH, REP_KFH, INHREM_KFH, NOTES, LINE_NUMBER, SEGMENT_NUMBER, MODE_IEFF, CFE_GFE, ROLL_UP, METH_2_ADDED.

In addition to the tables in the schema a table was added to store the Reliability Data used in the Criticality Analysis information. This information is located in all the FMECA files and is the same data in all the files. The information is some Reliability Data obtained for the systems that was include in the FMECA files to be used in the Criticality Analysis information.

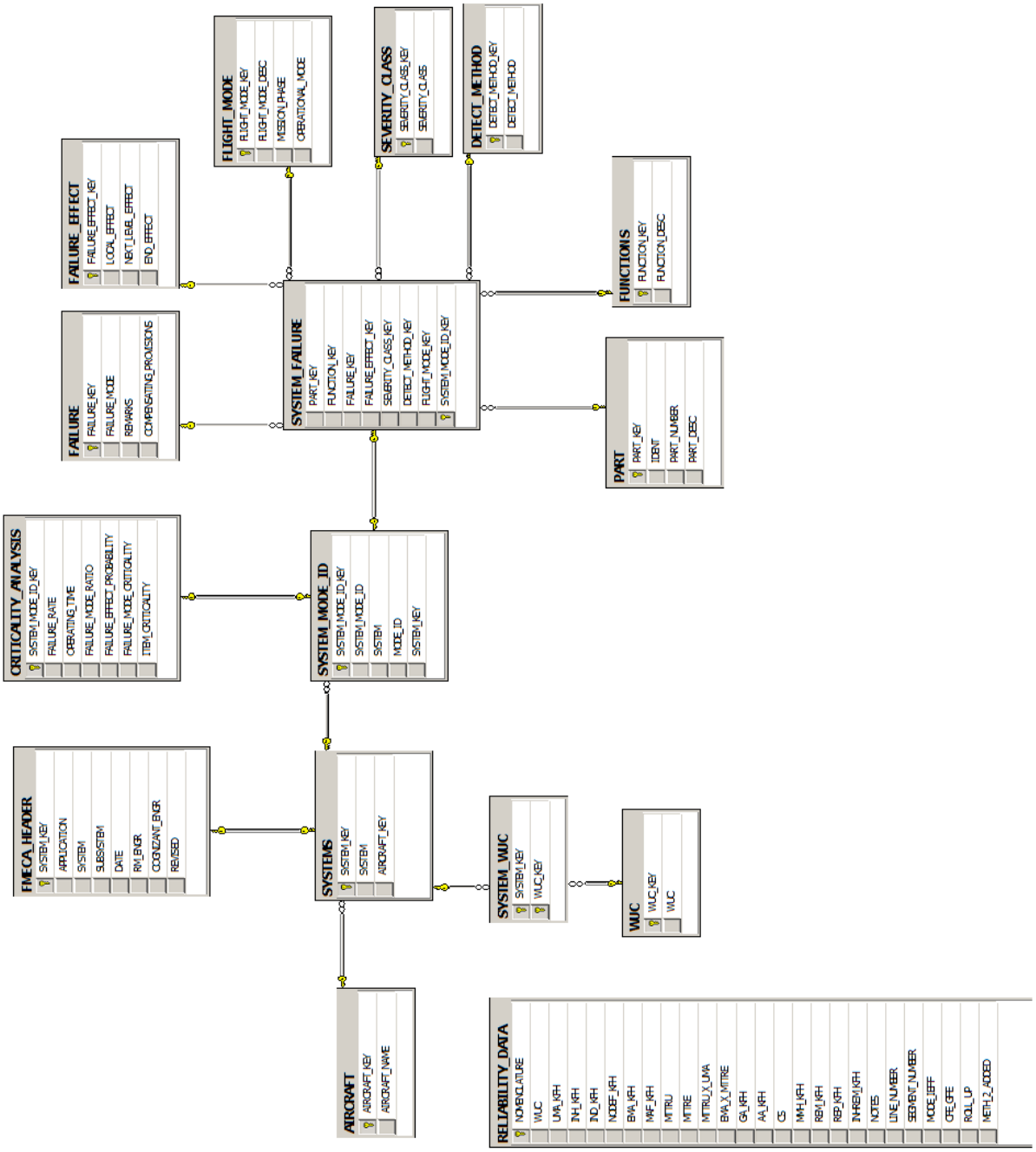


Figure 13: SQL Server Database Tables Diagram

6 C# Application

The C# Application is a console based program used to insert the FMEA and FMECA files into the SQL Server 2008 database.

6.1 Program Structure

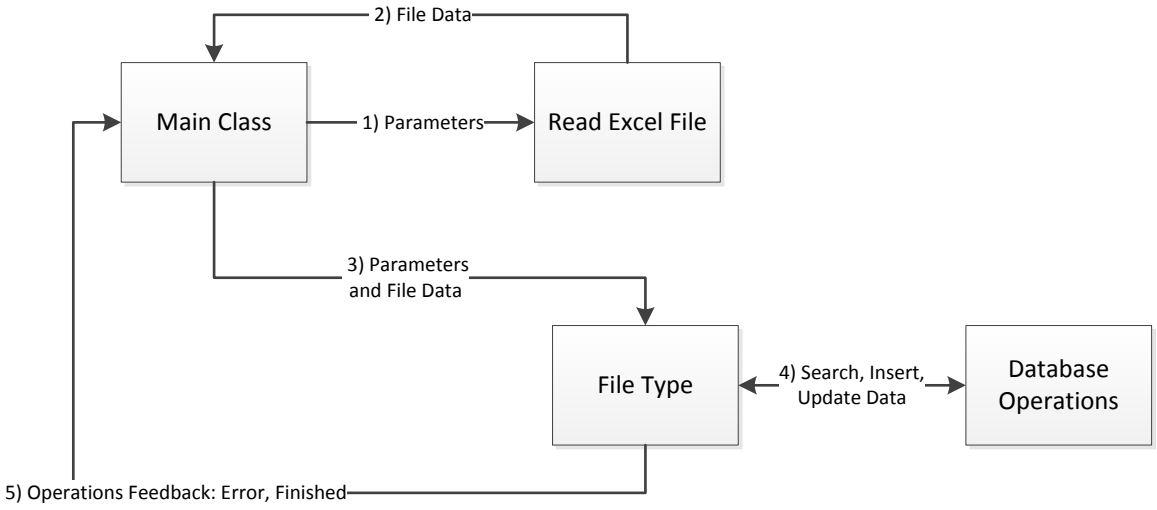


Figure 14: C# Application Components diagram

The Components of the console based C# Application are:

1) Main Class – main class that read the required parameters from the console and sent the information entered to the other components of the program.

2) Read Excel File – class used to read the information in a excel file. Receive form the main class the parameters needed to read the file information.

3) File Type – classes for the different types of files. Each file knows how to manage the data of a file with a specified format.

4) Database Operations – classes to perform database operations: search, insert, update.

6.2 Main Class

The Main Class display the console and read the parameters requires to read and insert the excel files in the database. The parameters required are:

1) The excel file or the directory with the excel files.

2) The type of the excel files: FMEA, FMECA, Reliability Data or WUC.

3) The name of the sheet in the excel file to be read.

4) Aircraft name when the file is a FMEA file: Black Hawk

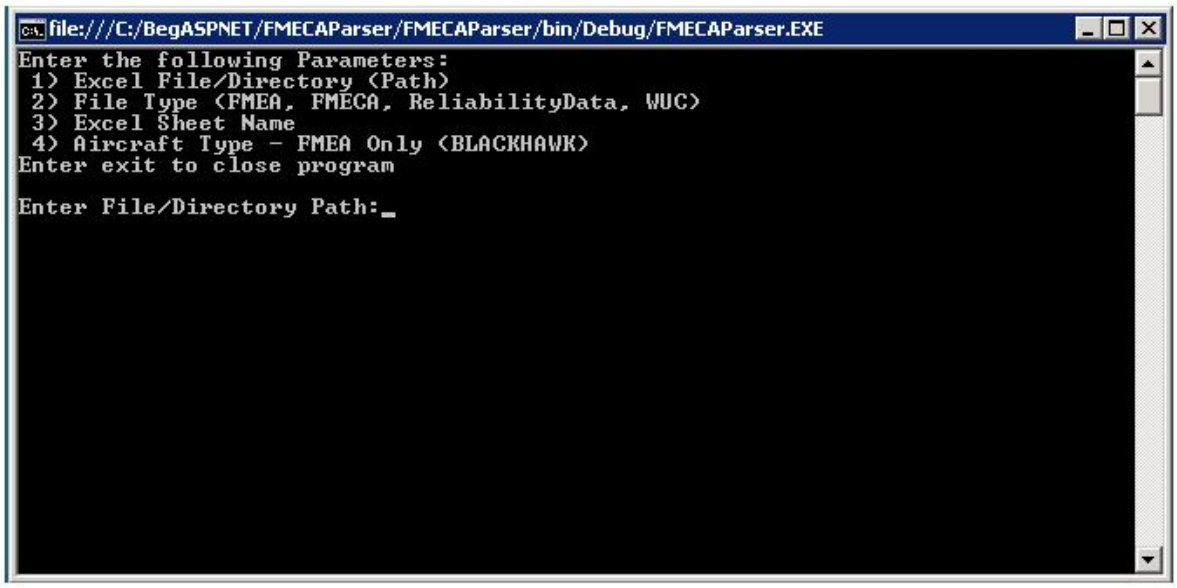


Figure 15: C# application console

The parameters are validated as they are entered. The following are the error displayed when an invalid value is entered.

1) File or Directory Path Parameter – Invalid File/Directory Path
2) File Type Parameter – Invalid File Type. Valid Values: FMEA, FMECA, ReliabilityData, WUC.
3) Sheet Name Parameter – Invalid Sheet Name (Null or Empty)
4) Aircraft Type Parameter – Invalid Aircraft Type. Valid Values: BLACKHAWK.

The functions of the Main Class is to maintain a loop requesting the required parameters and call the necessary classes to read the excel files and to insert the data into the database until the user enter the command exit. If the word exit is entered at any parameter the main class returns from the loop and end its execution.

```
C:\Windows\system32\cmd.exe
Enter the following Parameters:
 1) Excel File/Directory (Path)
 2) File Type (FMEA, FMECA, ReliabilityData, WUC)
 3) Excel Sheet Name
 4) Aircraft Type - FMEA Only (BLACKHAWK)
Enter exit to close program

Enter File/Directory Path:C:\Users\JuanG\Desktop\FMECA Project\FMECAFiles\FMEA
Enter File Type:FMEA
Enter Sheet Name:sheet1
Enter Aircraft Type:BLACKHAWK

Reading Excel File: 001 - Collective Trim Servo FMEA.xls
Inserting Excel File: 001 - Collective Trim Servo FMEA.xls

Reading Excel File: 002 - Avionics Relay Panel FMEA.XLS
Inserting Excel File: 002 - Avionics Relay Panel FMEA.XLS

Reading Excel File: 004 - Digital Clock FMEA.XLS
Inserting Excel File: 004 - Digital Clock FMEA.XLS

Reading Excel File: 011 - EFIS Junction Box FMEA.XLS
Inserting Excel File: 011 - EFIS Junction Box FMEA.XLS

Reading Excel File: 013 - MFSC-FMEA-ID 11-21-03.XLS
Inserting Excel File: 013 - MFSC-FMEA-ID 11-21-03.XLS

Reading Excel File: 016 - IDM FMEA.XLS
Inserting Excel File: 016 - IDM FMEA.XLS

Reading Excel File: 018 and 019 - AVC_FMEA-JL.XLS
Inserting Excel File: 018 and 019 - AVC_FMEA-JL.XLS

Reading Excel File: 028 - Gust Lock FMEA-ID Alicen-3-24-04.XLS
Inserting Excel File: 028 - Gust Lock FMEA-ID Alicen-3-24-04.XLS

Reading Excel File: 035 - TRGBX OIL LEVEL GAUGE FMEA-ID Alicen-3-24-04.XLS
Inserting Excel File: 035 - TRGBX OIL LEVEL GAUGE FMEA-ID Alicen-3-24-04.XLS

Reading Excel File: 037 - stormscope_kao_071603.XLS
Inserting Excel File: 037 - stormscope_kao_071603.XLS

Reading Excel File: 041 - IBU_kao071603.XLS
Inserting Excel File: 041 - IBU_kao071603.XLS

Reading Excel File: 044 - cvr-fdr_kao071603-2.XLS
Inserting Excel File: 044 - cvr-fdr_kao071603-2.XLS

Reading Excel File: 048 - FLIR FMECA-JL.XLS
Inserting Excel File: 048 - FLIR FMECA-JL.XLS

Reading Excel File: 050 - rev_sw_pnl_fmea_kao071503.XLS
Inserting Excel File: 050 - rev_sw_pnl_fmea_kao071503.XLS


Enter File/Directory Path:
```

Figure 16: C# application console - Insert FMEA information

6.3 Read Excel File

The read excel file component is a single class that read the data in a excel file. To read the data of a excel file a process similar to the database operations needs to be follow:

1) Define Excel connection string
2) Open Excel connection.
3) Read the data in the excel file.
4) Close Excel connection.

One of the objectives of the read excel file component is to be able to read the data of the excel files without knowing the format of the file. The idea behind this is that the Read Excel File component should be able to read the data in the excel file without knowing the format and a different class with knowledge of the format of the data in the file should receive the excel data and perform any require formatting or other operations. To accomplish this instead of reading the data in the excel file column by column and define the value in each column with the knowledge of the type of file specified, the data is read using the OleDbDataAdapter object. The OleDbDataAdapter object receive the connection to the excel file and the query used to read the data in the excel file, with this information the OleDebDataAdapter object can be used to fill a DataSet object with the data of the excel file. A DataSet object can contain multiple DataTable objects with data, in the case of the Read Excel file component the DataSet object created only contain a single DataTable object with the data in the excel

file. The DataTable object with the data in the excel file is the object returned by the Read Excel File component of the C# Application.

6.4 File Types

The File Types components is a group of classes that know how to manage the different types of excel files. The AbstractFile class is an abstract class from where the classes for the different types of files inherit. The principal property of the AbstractFile class is the data property which is a DataTable object with the data in the excel file and the principal abstract method of the AbstractFile class is the InsertFileInDataBase method. The classes that inherit from the AbstractFile class inherit the property data with the data read from a excel file and need to implement the InsertFileInDataBase method. The InsertFileInDataBase method implementation require the operations necessary to insert the data from the excel file into the database.

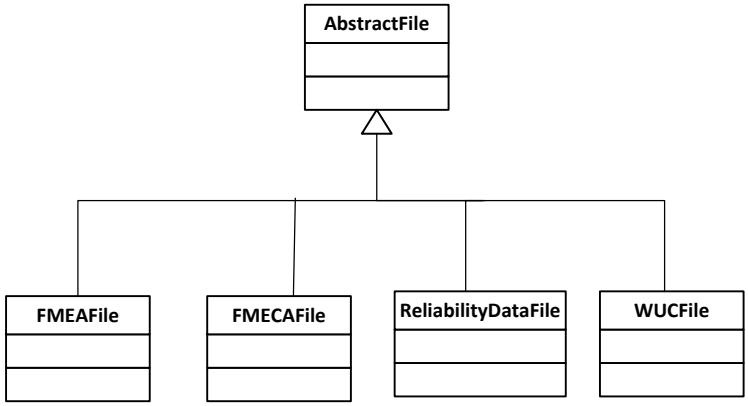


Figure 17: C# Application File Types class diagram

There is a class that inherits from the AbstractFile class for each type of excel file:

1) FMEAFile – to perform operations on the data of a FMEA excel file.

2) FMECAFile – to perform operations on the data of a FMECA excel file.

3) ReliabilityDataFile – to perform operations on the data of a ReliabilityData excel file.

4) WUCFile – to perform operations on the data of a WUC Reference List file.

These classes, in addition to insert the data into the database perform any operation required on the data like: removing invalid rows and fill an empty column with the previous value before inserting the data into the database.

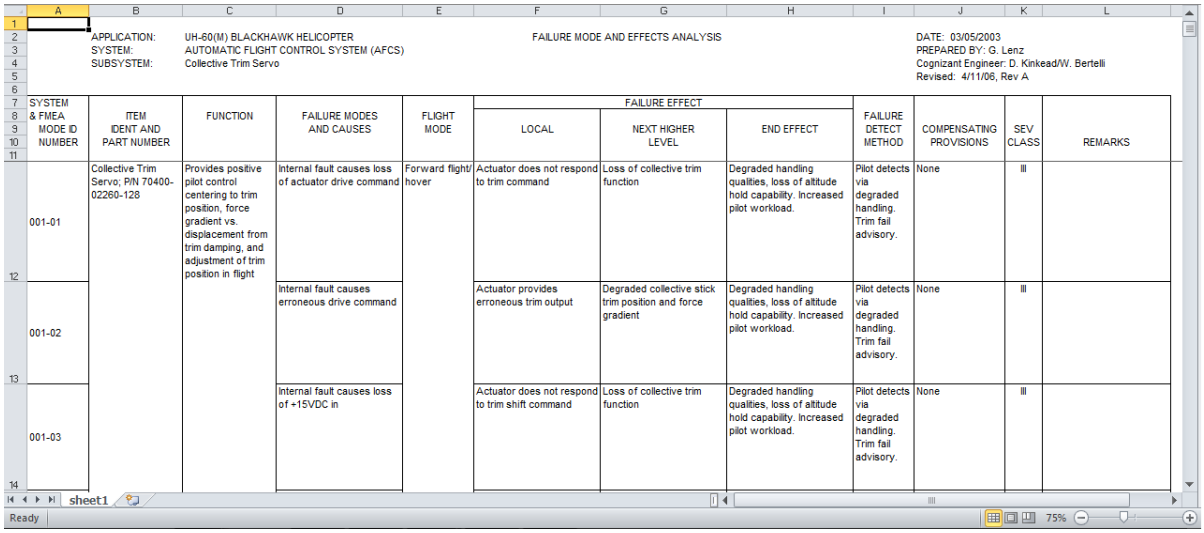
The FMEA excel files are formatted in a way on which for all the columns excluding the System and Mode ID and Remarks column, if the value of the column is empty the value of this column is the previous value in the column. This can be seen in most of the files in the values of columns like the Part Number and Function on where only the first row has a value and all the rows has those columns empty. To avoid having to format all the files, which is time consuming and is something that needs to be done in all the files the FMEAFile class perform this operation. The FMEAFile class performs the following operations before inserting the data into the database:

1) Separate header and data.

2) Obtain header information.

3) Remove any row from the data with the System and Mode Id empty – the System and Mode Id is a required value that must not be empty. This verification is also used to remove any empty row at the end of the file.

4) Fill data in empty columns with column previous value for the specified columns according to the file format.



APPLICATION: UH-60(M) BLACKHAWK HELICOPTER
SYSTEM: AUTOMATIC FLIGHT CONTROL SYSTEM (AFCS)
SUBSYSTEM: Collective Trim Servo

FAILURE MODE AND EFFECTS ANALYSIS

DATE: 03/05/2003
PREPARED BY: G. Lenz
Cognizant Engineer: D. Kinkead/W. Bertelli
Revised: 4/11/06, Rev A

SYSTEM & FMEA MODE ID NUMBER	ITEM IDENT AND PART NUMBER	FUNCTION	FAILURE MODES AND CAUSES	FLIGHT MODE	FAILURE EFFECT			FAILURE DETECT METHOD	COMPENSATING PROVISIONS	SEV CLASS	REMARKS
					LOCAL	NEXT HIGHER LEVEL	END EFFECT				
001-01	Collective Trim Servo; P/N 70400-02260-128	Provides positive pilot control centering to trim position, force gradient vs. displacement from trim damping, and adjustment of trim position in flight	Internal fault causes loss of actuator drive command	Forward flight/ hover	Actuator does not respond to trim command	Loss of collective trim function	Degraded handling qualities, loss of altitude hold capability. Increased pilot workload.	Pilot detects via degraded handling. Trim fail advisory.	None	III	
001-02			Internal fault causes erroneous drive command		Actuator provides erroneous trim output	Degraded collective stick trim position and force gradient	Degraded handling qualities, loss of altitude hold capability. Increased pilot workload.	Pilot detects via degraded handling. Trim fail advisory.	None	III	
001-03			Internal fault causes loss of +15VDC in		Actuator does not respond to trim shift command	Loss of collective trim function	Degraded handling qualities, loss of altitude hold capability. Increased pilot workload.	Pilot detects via degraded handling. Trim fail advisory.	None	III	

Figure 18: FMEA excel file format example

The FMECA excel file has the same information of the FMEA excel file but adding the criticality analysis data. The information in the FMECA files that is also contained in the FMEA is not directly written in the excel sheet with the FMECA but the information from the sheet with the FMEA is referenced in the sheet with the FMECA information. Because of this in many FMECA files the information of the FMEA is incomplete as consequence of a missing reference. The FMECAFile class only inserts the criticality analysis data of a

FMECA because its FMEA information can be incomplete. The FMEA information is inserted into the database using the FMEAFile class.

The RealiabilityDataFile and WUCFile only need to remove any invalid row or empty row at the end of the file. The excel files for the Realiability Data and WUC Reference List only contain data and when reading the excel files the connection string to the excel files can specify that the first row is the header with the columns names and that must be ignore. In both cases the DataTable object with the excel information will only contain the excel data.

6.5 Database Operations

The database operation components are the classes that perform database operations. The classes in the File Types component use these classes to insert the excel information into the database.

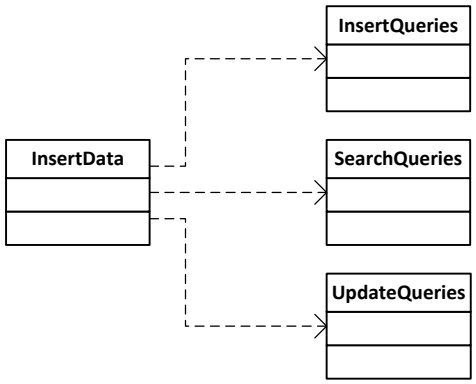


Figure 19: C# Application Database Operations class diagram

The basic operation for inserting the excel data into the database is as follow:

1) Before Inserting the data verify if information already exists.

2) If the information doesn't exist insert information.

3) If the information exists update information.

This logic is implemented in the InsertData class. This class uses object from the classes that implement the search, update and insert database operations.

1) InsertQueries class – implement insert database operations.

2) SearchQueries class – implement search database operations.

3) UpdateQueries class – implement update database operations.

7 Web Application

The Web Application is an ASP.NET web application that displays the FMEA and FMECA information from the database. The objective of the Web Application was to provide web pages with the FMEA and FMECA information and provide a way to link those pages to any web page.

7.1 Web Application Structure

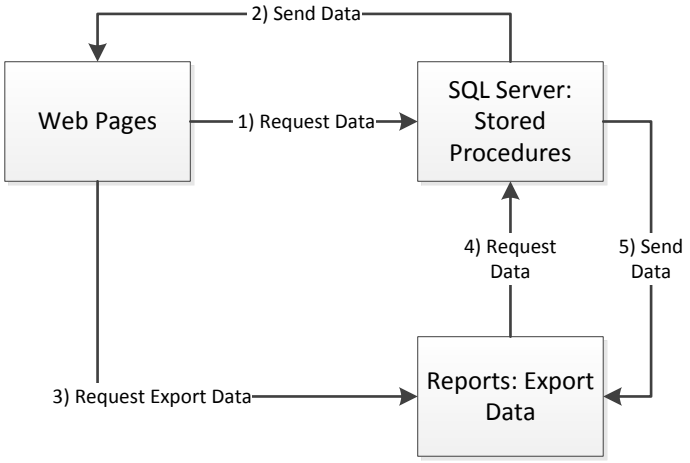


Figure 20: Web Application Structure diagram (Web Application Components)

The components of the Web Application are:

1) Web Pages – the ASP.NET web pages that display the FMEA and FMECA information.

2) SQL Server: Stored Procedures – stored procedures in the database that receive some parameters and return the requested information from the database.

3) Reports: Export Data – reports created using the Microsoft SQL Server Reporting Services that are used to export FMEA and FMECA information to excel and PDF.

7.2 Web Pages

The Web Pages component is a group of ASP.NET web pages that display the information of the FMEAs and FMECAs from the database. The information displayed in the web pages is the same information in the FMEA and FMECA excel files with the exception that in the web pages a column with the WUC was added. In the cases where the same System has more than one WUC the WUC column has the list of WUC for that System with the values separated by a dash or minus symbol (-).

One design decision to avoid the multiplication of the quantity of the web pages was to pass a parameter to the web page indicating the format of the file: FMEA or FMECA and according to the value of this parameter the web page display the FMEA or FMECA for the specified System or the parameter required by the Web Page. The multiplication of the quantity of web pages can occur if to display the FMEA and FMECA of a given parameter there are 2 separate web pages. This is that there are two pages for displaying the FMEA and FMECA information of a System, two more pages to display the FMEA and FMECA information of a WUC, etc. The use of two separate web pages: one for the FMEA and another for the FMECA multiply by 2 the number of required pages compared to the use of a single page to implement both and using a parameter to specify the desire format.

localhost:49302/FMECAWebSite/FMECABySystemID.aspx?fileType=FMEA&SystemID=023

FAILURE MODE AND EFFECTS ANALYSIS (FMEA)

Excel Export

APPLICATION: UH-60(M) BLACK HAWK HELICOPTER FAILURE MODE AND EFFECTS ANALYSIS DATE: 7/11/03

SYSTEM: AIRFRAME R&M ENGR: J. lukasavage

SUBSYSTEM: GUNNER SEAT COGNIZANT ENGR: S. Benavides

REVISED: 12/11/03

System & FMEA Mode ID Number	WUC	Item Ident and Part Number	Function	Failure Mode	Mission Phase/Operational Mode	Local Effect	Next Level Effect	End Effect	Failure Detect Method	Compensating Provisions	Severity Class	Remarks
023-01	02A04	Gunner Seat; 70500-52851-102	Provides structure for occupant to sit upon. Protects the occupant in a crash or hard landing by absorbing the loading and transferring forces to the airframe structure.	Loss of seat structural stability.	All Mission Phases/Flight and Hover	Seat loose (not properly secured to floor and/or ceiling or seat itself is not structurally sound.)	Loss of affected utility seat stability.	Occupant experiences movement of seat. Occupant can move to another seat if available. No effect in flight. No effect on mission function. Seat can possible drop occupant in an aggressive maneuver, depending on the nature of stability loss. Significant reduction in safety margins. Seat needs to be secured before next takeoff.	Occupant feel.	Pre-flight check	IV	Seats designed and tested per Sikorsky specification SES-701403.
			Provides structure for occupant to sit			Seat may	Occupant					Seats

Figure 21: Example FMEA data in Web Application

localhost:49302/FMECAWebSite/FMECABySystemID.aspx?fileType=FMECA&SystemID=023

FAILURE MODE, EFFECTS AND CRITICALITY ANALYSIS (FMECA)

Excel Export

APPLICATION: UH-60(M) BLACK HAWK HELICOPTER FAILURE MODE, EFFECTS AND CRITICALITY ANALYSIS

SYSTEM: AIRFRAME

SUBSYSTEM: GUNNER SEAT

DATE: 7/11/03

R&M ENGR: J. lukasavage

COGNIZANT ENGR: S. Benavides

REVISED: 12/11/03

System & FMEA Mode ID Number	WUC	Item Ident and Part Number	Function	Failure Mode	Mission Phase/Operational Mode	Local Effect	Next Level Effect	End Effect	Failure Detect Method	Compensating Provisions	Severity Class	Failure Rate (FPH)	Operating Time (Hrs)	Fail Mo Rat (%
023-01	02A04	Gunner Seat; 70500-52851-102	Provides structure for occupant to sit upon. Protects the occupant in a crash or hard landing by absorbing the loading and	Loss of seat structural stability.	All Mission Phases/Flight and Hover	Seat loose (not properly secured to floor and/or ceiling or seat itself is not structurally sound.)	Loss of affected utility seat stability.	Occupant experiences movement of seat. Occupant can move to another seat if available. No effect in flight. No effect on mission function. Seat can possible drop occupant in an aggressive maneuver, depending on the nature of stability loss. Significant reduction in	Occupant feel.	Pre-flight check	IV			

Figure 22: Example FMECA data in Web Application

To implement the capability of displaying the FMEA or FMECA in the same web page a MultiView was used. A MultiView is an ASP.NET web control that can contain multiple Views but only one view is visible at a time. In the case of the web pages for displaying the FMEA or FMECA information two Views where created one to display the FMEA information and the other to display the FMECA information. Each View contains an ASP.NET GridView web control to display the information in a table. The parameter passed with the format required: FMEA or FMECA is used to control the view that is visible and information like the page title. The information presented in the web page is only requested for the visible view; the data for the view that is not visible was not requested because the parameter passed to the page indicated that the data required was the data of the other View. This is necessary to avoid requesting data that is not visible and increase the load time of the web page.

Another design decision was to use stored procedure for retrieving the data from the database instead of executing database queries directly from the web pages. The information displayed in the Web pages comes directly from stored procedures that are stored in the database. The stored procedures receive a parameter and return the requested FMECA information. The database connection and execution of the stored procedure is managed automatically by the ASP.NET SqlDataSource web control. The SqlDataSource web control needs the specification of the connection string and the name and value of the parameters to be passed to the stored procedure. The data obtained from the SqlDataSource web control is added to the table (GridView) in the visible view. The value of the connection string is defined and

obtained from the Web.config file. To change the value of the connection string in all the pages in the web application the value needs to be changed only in the Web.config file.

The testing of the capability of linking a column in a table to a FMEA or to a FMECA information was implemented and tested using a web page with a the list of Systems that identify a single excel file and a web page with the list of the System and Mode Id of a System that identify a single row in the excel page. An additional web page was created to display a single row in a excel file. This page receives parameters for the format of the file: FMEA and FMECA and the System and Mode Id that identify the row of an excel file.

The page with the list of System Id contains a table with the System Id in one column and additional columns with links to display the FMEA of the System Id, the FMECA of the System Id or the list of Mode Ids of the System.

SYSTEM ID			
001	View FMEA	View FMECA	View Mode IDs
002	View FMEA	View FMECA	View Mode IDs
004	View FMEA	View FMECA	View Mode IDs
006	View FMEA	View FMECA	View Mode IDs
007	View FMEA	View FMECA	View Mode IDs
008	View FMEA	View FMECA	View Mode IDs
009	View FMEA	View FMECA	View Mode IDs
010	View FMEA	View FMECA	View Mode IDs
011	View FMEA	View FMECA	View Mode IDs
012	View FMEA	View FMECA	View Mode IDs
013	View FMEA	View FMECA	View Mode IDs
014	View FMEA	View FMECA	View Mode IDs
015	View FMEA	View FMECA	View Mode IDs
016	View FMEA	View FMECA	View Mode IDs
017	View FMEA	View FMECA	View Mode IDs
018	View FMEA	View FMECA	View Mode IDs
020	View FMEA	View FMECA	View Mode IDs
021	View FMEA	View FMECA	View Mode IDs
023	View FMEA	View FMECA	View Mode IDs
024	View FMEA	View FMECA	View Mode IDs
025	View FMEA	View FMECA	View Mode IDs
027	View FMEA	View FMECA	View Mode IDs
028	View FMEA	View FMECA	View Mode IDs
029	View FMEA	View FMECA	View Mode IDs
030	View FMEA	View FMECA	View Mode IDs
031	View FMEA	View FMECA	View Mode IDs
033	View FMEA	View FMECA	View Mode IDs

Figure 23: Example System (Excel) List in Web Application

The page with the list of the Mode Id of a System contains columns to display the FMEA and the FMECA of the System and Mode Id.

localhost:49302/FMECAWebSite/ListOfModelIDs.aspx?SystemId=001

FMECA/FMEA Site

SYSTEM MODE ID		
001-01	View FMEA	View FMECA
001-02	View FMEA	View FMECA
001-03	View FMEA	View FMECA
001-04	View FMEA	View FMECA
001-05	View FMEA	View FMECA
001-06	View FMEA	View FMECA
001-07	View FMEA	View FMECA
001-08	View FMEA	View FMECA
001-09	View FMEA	View FMECA
001-10	View FMEA	View FMECA
001-11	View FMEA	View FMECA
001-12	View FMEA	View FMECA
001-13	View FMEA	View FMECA
001-14	View FMEA	View FMECA
001-15	View FMEA	View FMECA
001-16	View FMEA	View FMECA
001-17	View FMEA	View FMECA

Figure 24: Example System and Mode Id (Excel Row) List in Web Application

Selecting the view option of the FMEA or FMECA of a System and Mode Id displays a web page with the information of the row with the selected System and Mode Id.

localhost:49302/FMECAWebSite/FMECAByModeID.aspx?fileType=FMEA&ModeID=001-01

FAILURE MODE AND EFFECTS ANALYSIS (FMEA)

System & FMEA Mode ID Number	WUC	Item Ident and Part Number	Function	Failure Mode	Mission Phase/Operational Mode	Local Effect	Next Level Effect	End Effect	Failure Detect Method	Compensating Provisions	Severity Class	Remarks
001-01	11D40	Collective Trim Servo; P/N 70400-02260-128	Provides positive pilot control centering to trim position, force gradient vs. displacement from trim damping, and adjustment of trim position in flight	Internal fault causes loss of actuator drive command	Forward flight/ hover	Actuator does not respond to trim command	Loss of collective trim function	Degraded handling qualities, loss of altitude hold capability. Increased pilot workload.	Pilot detects via degraded handling. Trim fail advisory.	None	III	

Figure 25: Example Excel Row in Web Application

A web page was also created for the Reliability Data contained in the FMECA files to obtain the criticality analysis.

localhost:49302/FMECAWebSite/ReliabilityData.aspx

RELIABILITY DATA (FMECA)

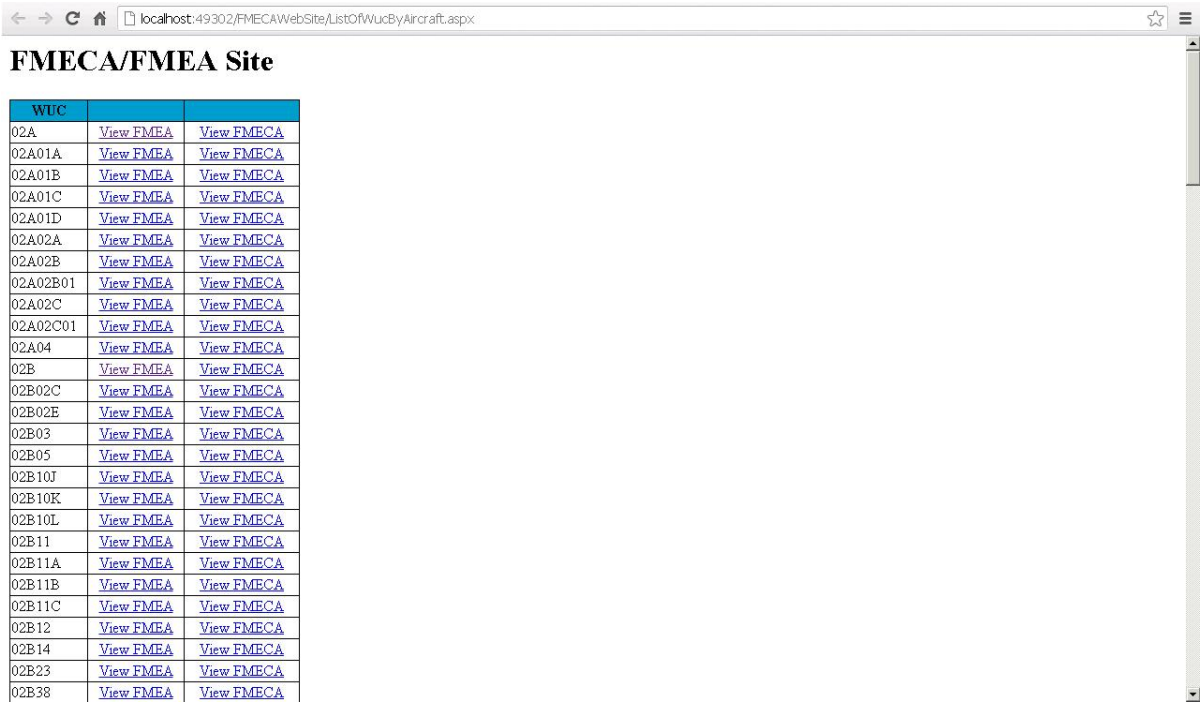
Excel Export

NOMENCLATURE	WUC	UMA/KFH	INH/KFH	IND/KFH	NODEF/KFH	EMA/KFH	MAF/KF
#1 ELECTRICAL JUNCTION	09A01E	0.152732304883212	0.127942561958333	0.02470974285	7.99999998785904E-05	0.124074724977996	0.09019823591
#2 ELECTRICAL JUNCTION	09A01F	0.236861533508212	0.187362047733333	0.049419485775	7.99999998785904E-05	0.158784467902996	0.06548849306
#3HYDRAULIC SYSTEM	07B	3.1299007655	2.1744573739	0.5600875052	0.395355886	1.5484772206	0.2965169146
#3HYDRAULIC SYSTEM (NOC)	07B99	0.1976779431	0.1317852954	0.0329463238	0.0329463238	0.0329463238	0
60B STAB ASSY CENTER	02C15B10	1.47499275197867	1.26543094021692	0.0886607665023525	0.120901045234929	0.773766689557365	0.09672083619
60B STAB ASSY OUTBOARD LH	02C15B12	2.12015084972236	1.81892723172153	0.127440761444309	0.173782856521361	1.11221028181407	0.13902628523
60B STAB ASSY OUTBOARD RH	02C15B11	2.29945708970033	1.97275826820472	0.138218732152149	0.188480089305319	1.20627257163795	0.150784071459
701D HOT SECTION MODULE	04A02	0.1153121334	0.0823658096	0	0.0329463238	0.0768747555805556	0
AC GENERATOR	09A01	2.33342694549142	1.92967447829167	0.205914524025	0.197837942999757	1.89722906148099	1.01229114898
AC GENERATOR (NOC)	09A0199	1.2849066301	1.0542823632	0.0658926477	0.1647316192	1.0213360393	0.5600875054
AC GENERATOR CONTROL	09A01A	0.658926477	0.5600875054	0.0658926477	0.0329463238	0.5930338293	0.2965169146
AC MOTOR PUMP	07B03	0.2306242669	0.1317852954	0.0658926477	0.0329463238	0.0329463238	0
AC SYSTEM	09A	3.61833357559142	2.85217154599167	0.238860847825	0.527301181399757	2.58910186228099	1.30880806358
AC SYSTEM (NOC)	09A99	0.6259801531	0.4941948577	0	0.1317852954	0.3953558862	0.2306242669
ACCELERATION CONTROL	15C02	0.0329463238	0.0329463238	0	0	0.0329463238	0.0329463238

Figure 26: Example of the Reliability Data in Web Application

The WUC value was also used to obtain the FMEA or FMECA information. The WUC value is associated to a System. A System Id can have more than one WUC and one WUC can be associated to more than one System Id. When the FMEA and FMECA information is obtained using a WUC the result is all the rows of the Excels with a System Id associated to the specified WUC. The web pages for the FMEA and FMECA for a specific WUC follow the same approach used for the pages with the FMEA and FMECA of a System Id, the only difference is that in this case the header information is not displayed because the header is associated to a single System Id and a WUC is associated to multiple System Id. For the

FMEA and FMECA of a WUC two web pages where created: a page with the list of WUC to test the link to the page with the FMEA and FMECA of the WUC and the page that display the FMEA or FMECA of the WUC in the same way that was done for the FMEA and FMECA of a System Id.



WUC		
02A	View FMEA	View FMECA
02A01A	View FMEA	View FMECA
02A01B	View FMEA	View FMECA
02A01C	View FMEA	View FMECA
02A01D	View FMEA	View FMECA
02A02A	View FMEA	View FMECA
02A02B	View FMEA	View FMECA
02A02B01	View FMEA	View FMECA
02A02C	View FMEA	View FMECA
02A02C01	View FMEA	View FMECA
02A04	View FMEA	View FMECA
02B	View FMEA	View FMECA
02B02C	View FMEA	View FMECA
02B02E	View FMEA	View FMECA
02B03	View FMEA	View FMECA
02B05	View FMEA	View FMECA
02B10J	View FMEA	View FMECA
02B10K	View FMEA	View FMECA
02B10L	View FMEA	View FMECA
02B11	View FMEA	View FMECA
02B11A	View FMEA	View FMECA
02B11B	View FMEA	View FMECA
02B11C	View FMEA	View FMECA
02B12	View FMEA	View FMECA
02B14	View FMEA	View FMECA
02B23	View FMEA	View FMECA
02B38	View FMEA	View FMECA

Figure 27: Example WUC List in Web Application

localhost:49302/FMECAWebSite/FMECAByWUC.aspx?fileType=FMEA&HELICOPTER=&WUC=02B

FAILURE MODE AND EFFECTS ANALYSIS (FMEA)

Excel Export

System & FMEA Mode ID Number	WUC	Item Ident and Part Number	Function	Failure Mode	Mission Phase/Operational Mode	Local Effect	Next Level Effect	End Effect	Failure Detect Method	Compensating Provisions	Severity Class	Remarks
025-01	02B	Cabin Section-Framing	Main cabin load and propulsion system load carrying member.	Buckling, Major	All	Permanent deformation of aircraft main load carrying ability of frame members. Flight envelope loads exceeded.	Redistribution of dynamic loading to adjacent structural members.	None, redundant load paths.	Visual inspection.	Multiple load paths. Qualified strength. Phase Inspections. Single occurrence severity is II.	I-R	
025-02	02B	Cabin Section-Framing	Main cabin load and propulsion system load carrying member.	Crack, Major	All	Permanent metal separation reduces load carrying ability of frame member.	Redistribution of dynamic loading to adjacent structural members.	None, redundant load paths.	Visual inspection.	Multiple load paths. Life Limit Design. Qualified life. Phase Inspection. Single occurrence severity is II.	I-R	
025-03	02B	Cabin Section-Framing	Main cabin load and propulsion system load carrying	Corrosion, minor	All	Reduced load carrying capacity.	Redistribution of dynamic loading to adjacent structural	None, redundant load paths.	Visual inspection.	Multiple load paths Corrosion prevention and control (CPC). Phase	III	

Figure 28: Example of FMEA data for a WUC in Web Application

In addition to the WUC, the other parameter used to obtain the FMEA and FMECA information was the Part Identification. A FMEA or FMECA of a specific Part Id is all the rows in all the FMEA or FMECA files with the specified Part Id.

localhost:8196/FMECAWebSite/ListOfIdent.aspx

FMECA/FMEA Site

PART ID		
Abrasion strip (titanium)	View FMEA	View FMECA
Active Vibration Control Computer (AVCC)	View FMEA	View FMECA
Active Vibration Control System	View FMEA	View FMECA
Air Data Computer	View FMEA	View FMECA
Airframe Accelerometers	View FMEA	View FMECA
Area Microphone/Preamp, Smiths Aero	View FMEA	View FMECA
Armored Crew Seat	View FMEA	View FMECA
AUX Fuel Management Panel	View FMEA	View FMECA
Aux Fuel Signal Conditioner	View FMEA	View FMECA
Avionics Relay Panel	View FMEA	View FMECA
Buffer Management Unit (BMU)	View FMEA	View FMECA
Cabin Section-Aluminum Skin.	View FMEA	View FMECA
Cabin Section-Floor.	View FMEA	View FMECA
Cabin Section-Framing	View FMEA	View FMECA
CEFS Relay Panel	View FMEA	View FMECA
Collective Grip Assy	View FMEA	View FMECA
Collective Trim Servo	View FMEA	View FMECA
Controllable Search Light	View FMEA	View FMECA
Crashworthy External Fuel System (CEFS), Auxillary Fuel System	View FMEA	View FMECA
Crashworthy External Fuel System (CEFS), external tanks and transfer pump package.	View FMEA	View FMECA
Cuff blade attachment bolts.	View FMEA	View FMECA
CVR/FDR	View FMEA	View FMECA
Cyclic Grip Assy	View FMEA	View FMECA
Cyclic Grip Assy, P/N 70400-01260-108	View FMEA	View FMECA
Data Concentrator Units (DCUs)	View FMEA	View FMECA

Figure 29: Example Part Id List in Web Application

7.3 SQL Server: Stored Procedures

The SQL Server Stored Procedures component contains the stored procedures that are used to obtain the data displayed in the web application and the data passed to the reports created to export the data to Excel and PDF. The following is the list of stored procedures with the parameters required (if any):

1) FMEAHeader – return the data in the header of a excel file. Parameters: System Id.
2) FMECAByItemIdent – returns the FMECA information of the specified Part Id. Parameters: Part Id
3) FMECAByModeID – return the FMECA data in the row with the specified System and Mode Id, including the WUC. Parameters: System and Mode Id.

4) FMECABySystemID – returns the FMECA data in the excel file with the specified System Id, including the WUC. Parameters: System Id.
5) FMECAByWuc – returns the FMECA information of the specified WUC and Aircraft. Parameters: WUC, Aircraft Name
6) GetListOfIdent – return the list of part identification in the database. Do not require a parameter.
7) GetListOfModeID – returns the list of System and Mode Id of the specified System. Parameters: System Id.
8) GetListOfSystemIDs – return the list of System Id in the database. Do not require a parameter.
9) GetListOfWUCByAircraft – return the list of WUC for the specified Aircraft. Parameters: Aircraft Name
10) ReliabilityData – return the reliability data used to obtain the criticality analysis information. Do not require a parameter.

To avoid the need of creating stored procedures for both the FMEA and FMECA information a single stored procedure was created with the FMECA information for the required parameters. The FMECA information also contains the FMEA information.

The query in the stored procedures for the data in the FMECA files includes the WUC of the System. A System can have more than one WUC and in those cases the different WUC values needs to appear separated by the dash or minus symbol (-). To implement this functionality the clause FOR XML was used to return the result of the query that return the

WUCs of a System in XML format and the T-SQL (Transact-SQL) STUFF function was used to concatenate the different WUC of a System in a single row. The following is a description of the clause and function used:

1) T-SQL STUFF function – inserts a string into another string.
2) FOR XML clause – return query result as XML. The FOR XML mode can be specified as one of the following: RAW, AUTO, EXPLICIT or PATH. The mode determines the shape of the resulting XML.

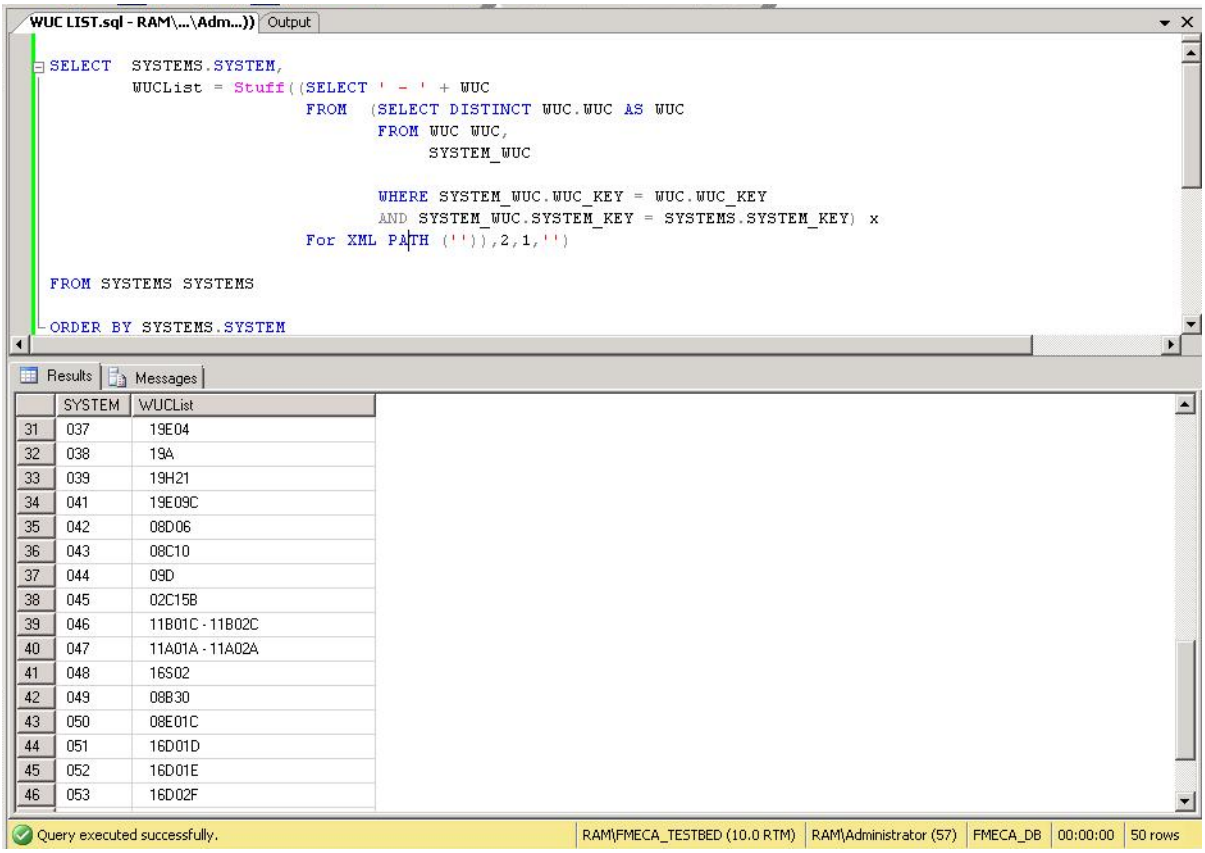


Figure 30: Example of SQL Query to concatenate List of WUC

7.4 Reports: Export Data

The Reports Export Data component is a group of reports created using the SQL Server Reporting Services that are used by the web application to export the data in the web page to excel or PDF. The data exported by the reports comes directly from the SqlDataSource used to display the FMEA or FMECA information in the web page. The reports were created as part of the web application and do not require the use of a Report Server. The following is the list of reports created:

1) FMEAReport – report that display the data of a FMEA excel file without the information in the header.
2) FMEASystemReport – report that display the data of a FMEA excel file with the header information.
3) FMECAReport – report that display the data of a FMECA excel file without the information in the header.
4) FMECASystemReport – report that display the data of a FMECA excel file with the header information.
5) ReliabilityDataReport – report that display the Reliability Data information in the FMECA excel files.

The reports created display data with the same columns of a FMEA, FMECA and ReliabilityData excel sheet. The data do not need to be the data of a single excel file but the format of the data needs to be the same.

The information in the report was specified by creating datasets using the stored procedures for the required information as the source, which are the same stored procedures used to display the information in the web pages. After specifying the datasets to be used in the report the columns in the data of the selected datasets can be added to the report. To pass the information to the reports the web pages need to pass data to the datasets defined in the report.

In the case of the reports with the header information, the data of these reports needs to be of a single excel file to display the header and data of the FMEA excel file.

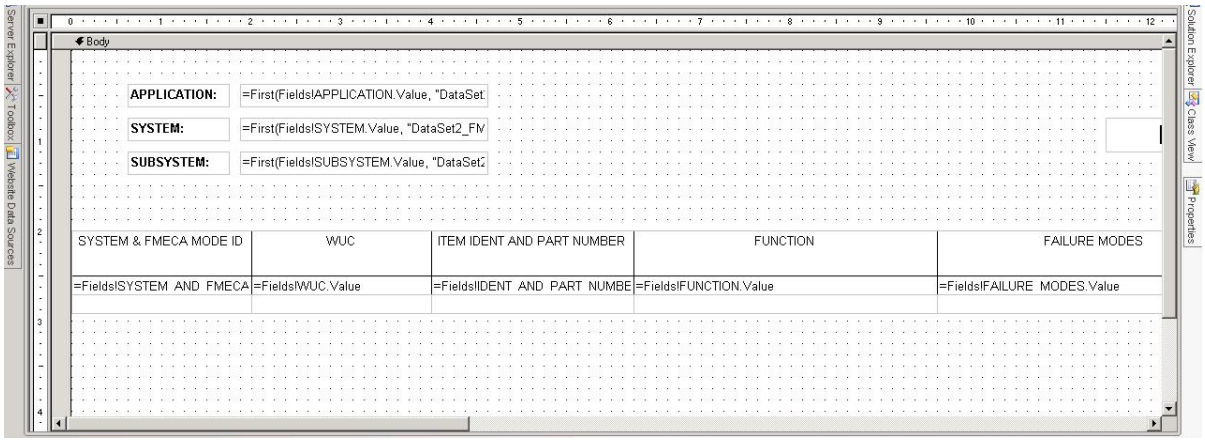


Figure 31: FMEA Report with Header information

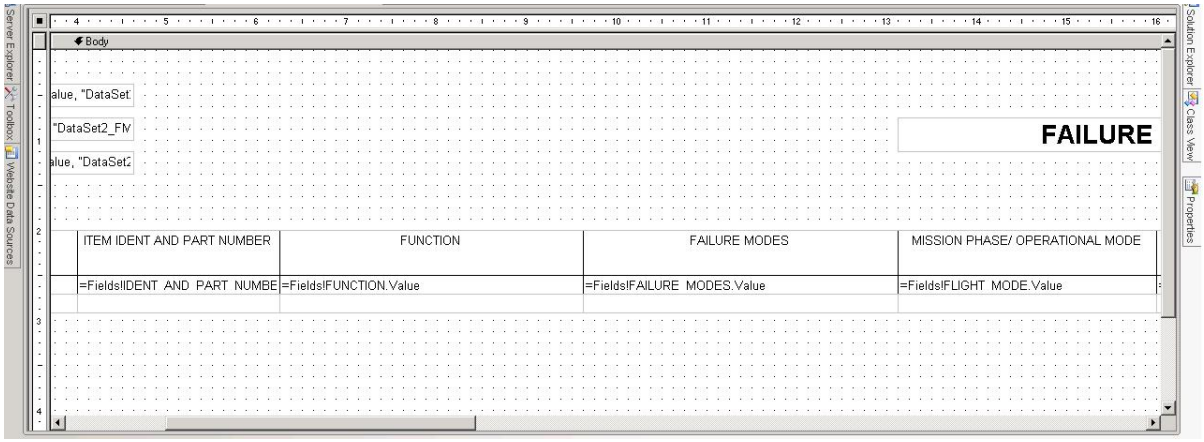


Figure 32: FMECA Report with Header Information

The reports without the header are for cases where the data included comes from different excel files and the header information is not required.

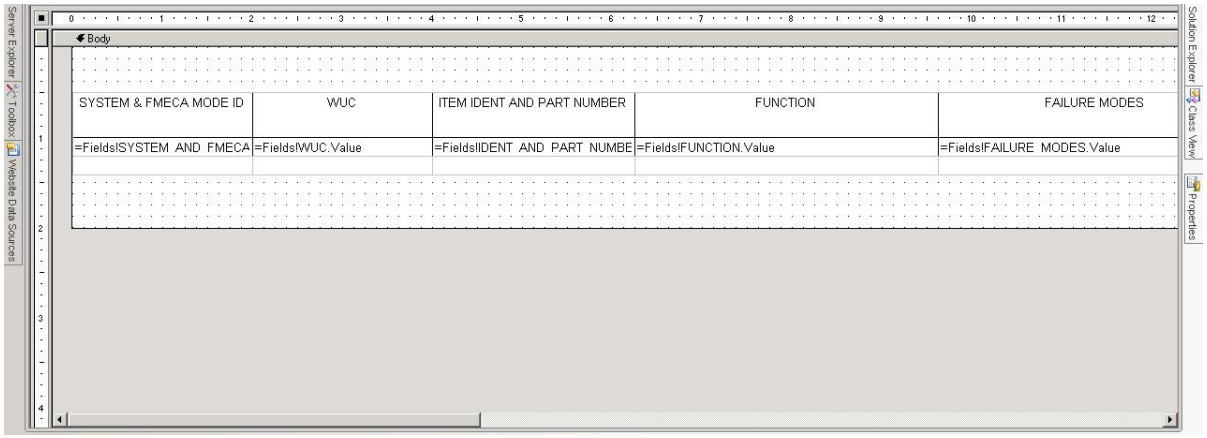


Figure 33: FMEA Report without header information

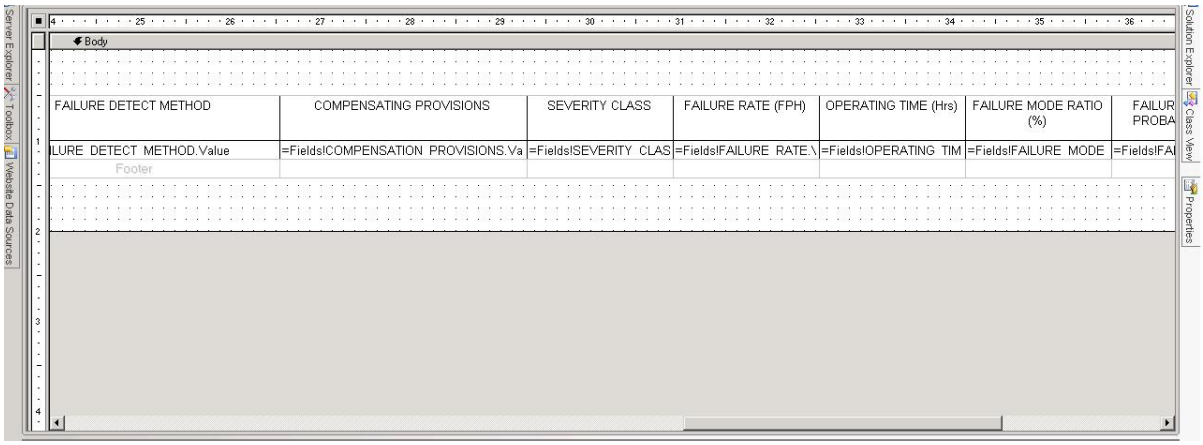


Figure 34: FMECA Report without header information

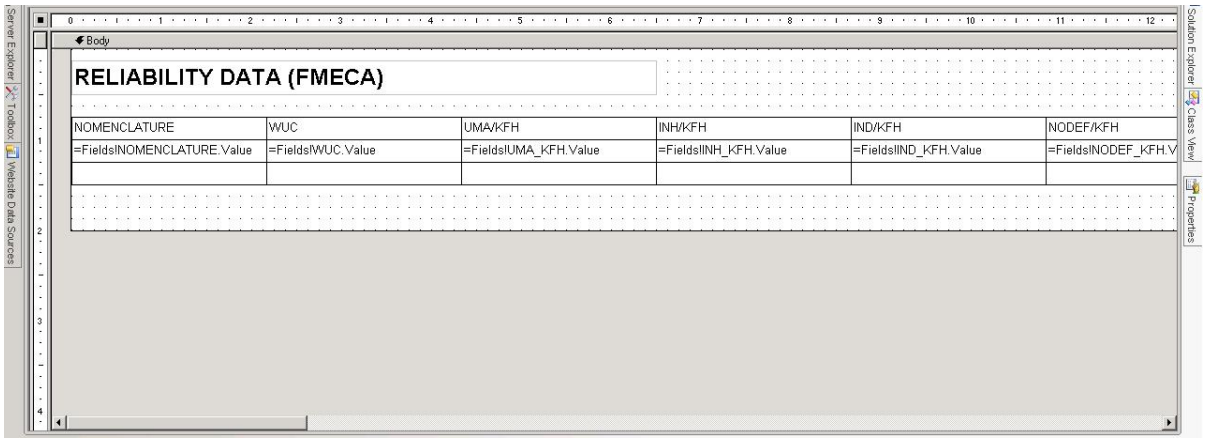


Figure 35: Reliability Data Report

The reports created are used to export the data to excel or PDF. The following are the step used to export the reports to Excel or PDF:

1) Create a LocalReport class object.
2) Set the path of the report in the LocalReport class object to the desired report in the web application. The method Server.MapPath(string pathInWebApp) is used to obtain the path of the report by specifying the location of the report in the web application.
3) Create ReportDataSource class objects for all the DataSet defined in the reports. The name of the ReportDataSource must be the name of the DataSets specified in the report.
4) Add the defined ReportDataSource to the LocalReport object.
5) Call the method Render of the LocalReport object. The Render method return an array of bytes with the report information and need the specification of the parameters:

a. Export Type – PDF or Excel

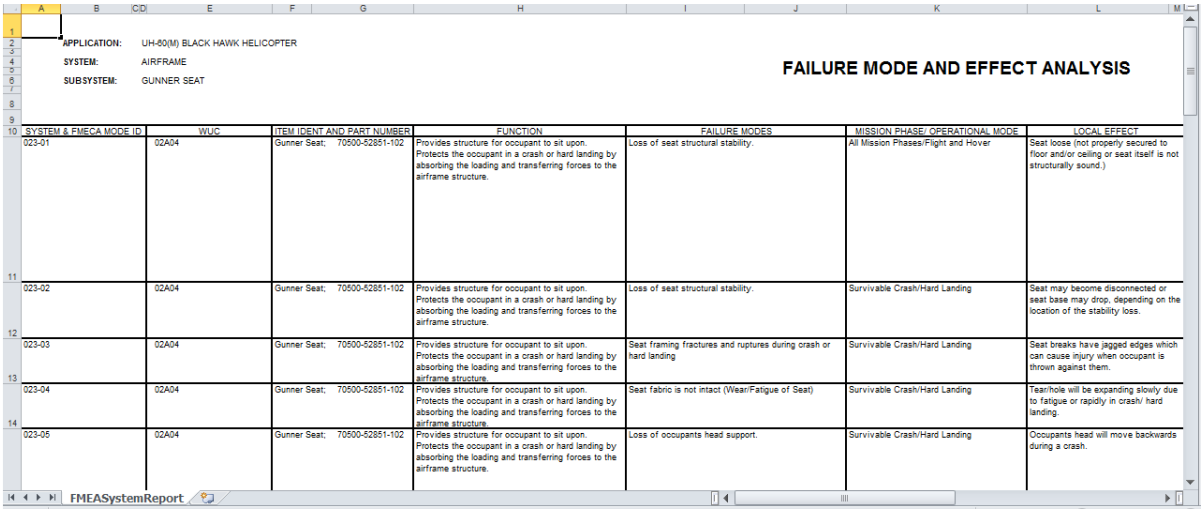
b. Device Info parameters

i. PDF: width, height, margins

ii. Excel: SimplePageHeader – true or false.

6) On the Response object to the export request specify the parameters: mimeType, filename and Content-Disposition and write the array of bytes to the Response stream by calling the method Response.BinaryWrite(byte[] byteArray).

7) After sending the array of bytes to the browser a window will appear asking to open or save the PDF or Excel file (Ex. Internet Explorer) or the file will be downloaded automatically (Ex. Google Chrome).



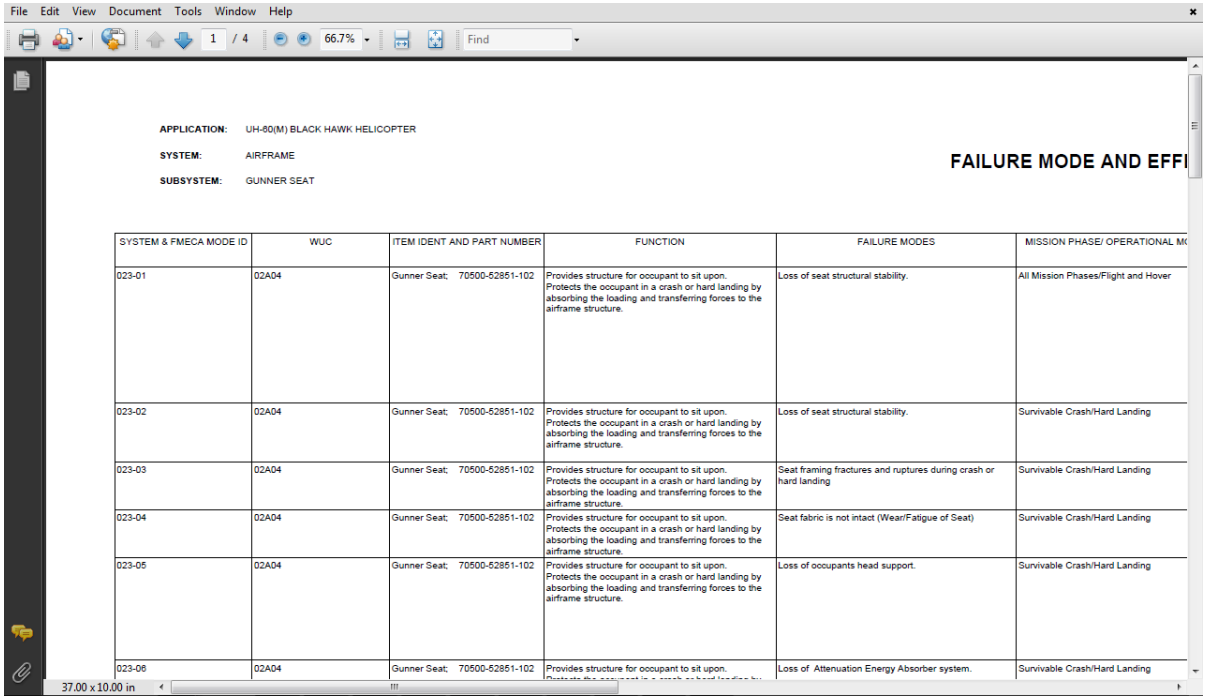
APPLICATION: UH-60(M) BLACK HAWK HELICOPTER
SYSTEM: AIRFRAME
SUBSYSTEM: GUNNER SEAT

FAILURE MODE AND EFFECT ANALYSIS

SYSTEM & FMECA MODE ID	WUC	ITEM IDENT AND PART NUMBER	FUNCTION	FAILURE MODES	MISSION PHASE/ OPERATIONAL MODE	LOCAL EFFECT
023-01	02A04	Gunner Seat; 70500-52851-102	Provides structure for occupant to sit upon. Protects the occupant in a crash or hard landing by absorbing the loading and transferring forces to the airframe structure.	Loss of seat structural stability.	All Mission Phases/Flight and Hover	Seat loose (not properly secured to floor and/or ceiling or seat itself is not structurally sound.)
023-02	02A04	Gunner Seat; 70500-52851-102	Provides structure for occupant to sit upon. Protects the occupant in a crash or hard landing by absorbing the loading and transferring forces to the airframe structure.	Loss of seat structural stability.	Survivable Crash/Hard Landing	Seat may become disconnected or seat base may drop, depending on the location of the stability loss.
023-03	02A04	Gunner Seat; 70500-52851-102	Provides structure for occupant to sit upon. Protects the occupant in a crash or hard landing by absorbing the loading and transferring forces to the airframe structure.	Seat framing fractures and ruptures during crash or hard landing	Survivable Crash/Hard Landing	Seat breaks have jagged edges which can cause injury when occupant is thrown against them.
023-04	02A04	Gunner Seat; 70500-52851-102	Provides structure for occupant to sit upon. Protects the occupant in a crash or hard landing by absorbing the loading and transferring forces to the airframe structure.	Seat fabric is not intact (Wear/Fatigue of Seat)	Survivable Crash/Hard Landing	Tear/hole will be expanding slowly due to fatigue or rapidly in crash/ hard landing.
023-05	02A04	Gunner Seat; 70500-52851-102	Provides structure for occupant to sit upon. Protects the occupant in a crash or hard landing by absorbing the loading and transferring forces to the airframe structure.	Loss of occupants head support.	Survivable Crash/Hard Landing	Occupants head will move backwards during a crash.

FMEASystemReport

Figure 36: Example Export to Excel



File Edit View Document Tools Window Help

1 / 4 66.7% Find

APPLICATION: UH-60(M) BLACK HAWK HELICOPTER

SYSTEM: AIRFRAME

SUBSYSTEM: GUNNER SEAT

FAILURE MODE AND EFFI

SYSTEM & FMECA MODE ID	WUC	ITEM IDENT AND PART NUMBER	FUNCTION	FAILURE MODES	MISSION PHASE/ OPERATIONAL M
023-01	02A04	Gunner Seat; 70500-52851-102	Provides structure for occupant to sit upon. Protects the occupant in a crash or hard landing by absorbing the loading and transferring forces to the airframe structure.	Loss of seat structural stability.	All Mission Phases/Flight and Hover
023-02	02A04	Gunner Seat; 70500-52851-102	Provides structure for occupant to sit upon. Protects the occupant in a crash or hard landing by absorbing the loading and transferring forces to the airframe structure.	Loss of seat structural stability.	Survivable Crash/Hard Landing
023-03	02A04	Gunner Seat; 70500-52851-102	Provides structure for occupant to sit upon. Protects the occupant in a crash or hard landing by absorbing the loading and transferring forces to the airframe structure.	Seat framing fractures and ruptures during crash or hard landing	Survivable Crash/Hard Landing
023-04	02A04	Gunner Seat; 70500-52851-102	Provides structure for occupant to sit upon. Protects the occupant in a crash or hard landing by absorbing the loading and transferring forces to the airframe structure.	Seat fabric is not intact (Wear/Fatigue of Seat)	Survivable Crash/Hard Landing
023-05	02A04	Gunner Seat; 70500-52851-102	Provides structure for occupant to sit upon. Protects the occupant in a crash or hard landing by absorbing the loading and transferring forces to the airframe structure.	Loss of occupants head support.	Survivable Crash/Hard Landing
023-06	02A04	Gunner Seat; 70500-52851-102	Provides structure for occupant to sit upon.	Loss of Attenuation Energy Absorber system.	Survivable Crash/Hard Landing

37.00 x 10.00 in

Figure 37: Example Export to PDF

8 Excel Macro

Additional work was done to verify how to access the information from the database using the Visual Basic for Application that is available in Excel and that can be used by creating macros.

The excel file created has two buttons:

1) FMEA – when is pressed the FMEA information of the system id specified is obtained from the database and displayed in the excel file.

2) Clear – delete the FMEA data from the excel file.

The System Id parameter for the database query is specified in the column B of the first row of the excel file. Because the System Id usually start with 0: 001, 023, etc. and the excel file remove the leading zero by default the value of the parameter System Id needs to be in single quotes: ‘001’, ‘023’, etc.

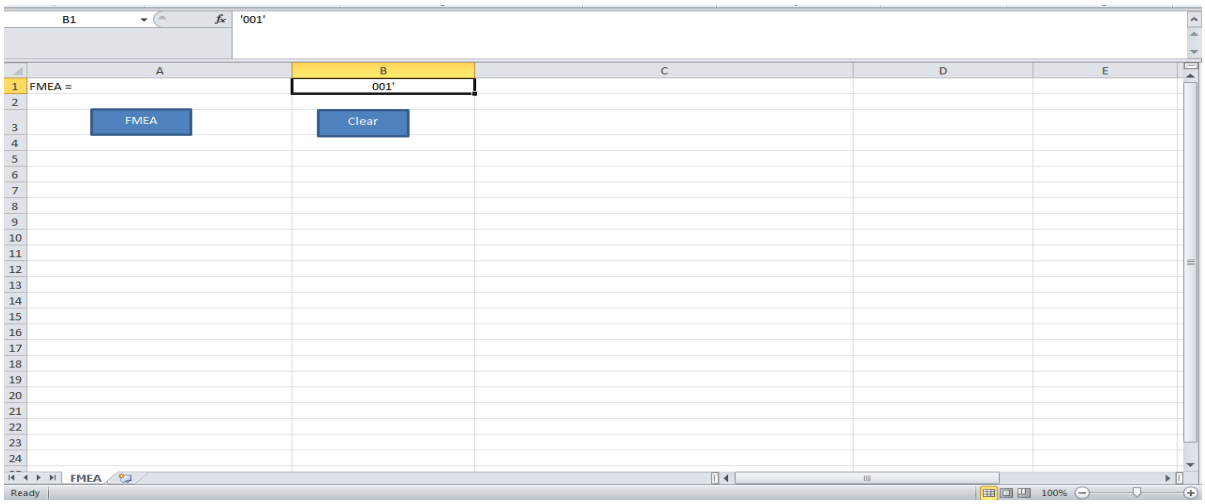


Figure 38: Excel with Database Access Macro without Data

After specifying the System Id of the FMEA excel file to be shown in the excel and pressing the FMEA button the information of the FMEA with the System Id specified is retrieved from the database and written in the excel file after removing any previous information.

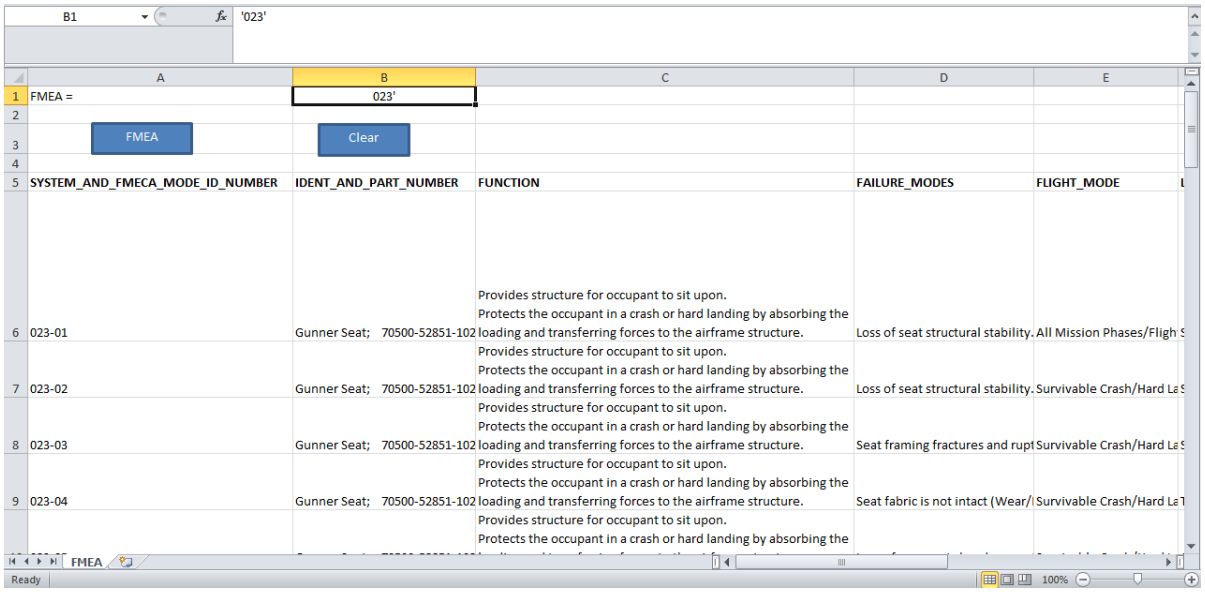


Figure 39: Excel with Database Access Macro with Data

Some of the functions or instructions used in the macros for displaying the FMEA information in the excel file are the following (replacing variables with value for better understanding):

1) Cells(5, “C”) = “FUNCTION” – write in the Row 5 and Column C the word FUNCTION.

2) Sheet1.Range(“A5:L5”).Font.Bold = True – define a range from the row 5 column A to the row 5 column L and set the Font value to Bold. This instruction put in bold the row with the columns name of the FMEA information.

8.1 Database Access

The code to access the SQL Server database from the excel file using Visual Basic for Application follow a pattern similar to writing a C# application, although the data types of the objects are different:

1) Create ADODB.connection object for the database connection.
2) Define connection string to the SQL Server database.
3) Open database connection using defined connection string.
4) Create a RecordSet object for the data.
5) Set database connection as the RecordSet Active Connection and obtain the data from the database calling the method Open of the RecordSet and passing as a parameter the query to the database.
6) Copy the data from the RecordSet in the excel file using the method CopyFromRecordset and passing as a parameter the RecordSet defined.
7) Close the RecordSet.
8) Close the database connection.
9) Set to null the database connection and the record set.

```
' Create a connection object.
Dim dbConnection As ADODB.connection
Set dbConnection = New ADODB.connection

' Provide the connection string.
Dim connectionString As String
connectionString = "Provider=SQLOLEDB;Data Source=LOCALHOST;Initial Catalog=FMECA;Integrated Security=SSPI;"
'connectionString = "Provider=SQLOLEDB;Data Source=RAM\FMECA_TESTBED;Initial Catalog=FMECA_DB;Integrated Security=SSPI;"

' Open the connection.
dbConnection.Open connectionString

' Create a recordset object.
Dim dataSet As ADODB.recordSet
Set dataSet = New ADODB.recordSet

With dataSet
    ' Assign the Connection object.
    .ActiveConnection = dbConnection
    ' Extract the required records.
    .Open "SELECT  SYSTEM_MODE_ID.SYSTEM_MODE_ID AS SYSTEM_AND_FMECA_MODE_ID_NUMBER, " & _
                   "PART.PART_DESC AS IDENT_AND_PART_NUMBER, " & _
                   "FUNCTIONS.FUNCTION_DESC AS [FUNCTION], " & _
                   "FAILURE.FAILURE_MODE AS FAILURE_MODES, " & _
                   "FLIGHT_MODE.FLIGHT_MODE_DESC AS FLIGHT_MODE, " & _
                   "FAILURE_EFFECT.LOCAL_EFFECT AS LOCAL_EFFECT, " & _
                   "FAILURE_EFFECT.NEXT_LEVEL_EFFECT AS NEXT_LEVEL_EFFECT, " & _
                   "FAILURE_EFFECT.END_EFFECT AS END_EFFECT, " & _
                   "DETECT_METHOD.DETECT_METHOD AS FAILURE_DETECT_METHOD, " & _
                   "FAILURE.COMPENSATING_PROVISIONS AS COMPENSATION_PROVISIONS, " & _
                   "SEVERITY_CLASS.SEVERITY_CLASS AS SEVERITY_CLASS, " & _
                   "FAILURE.REMARKS AS REMARKS " & _
            "FROM    SYSTEMS, SYSTEM_MODE_ID, PART, SYSTEM_FAILURE, FUNCTIONS, FAILURE, DETECT_METHOD," & _
                   "FAILURE_EFFECT, FLIGHT_MODE, SEVERITY_CLASS " & _
            "WHERE SYSTEMS.SYSTEM_KEY = SYSTEM_MODE_ID.SYSTEM_KEY " & _
            "AND     SYSTEM_MODE_ID.SYSTEM_MODE_ID_KEY = SYSTEM_FAILURE.SYSTEM_MODE_ID_KEY " & _
            "AND     PART.PART_KEY = SYSTEM_FAILURE.PART_KEY " & _
            "AND     FUNCTIONS.FUNCTION_KEY = SYSTEM_FAILURE.FUNCTION_KEY " & _
            "AND     FAILURE.FAILURE_KEY = SYSTEM_FAILURE.FAILURE_KEY " & _
            "AND     DETECT_METHOD.DETECT_METHOD_KEY = SYSTEM_FAILURE.DETECT_METHOD_KEY " & _
            "AND     FAILURE_EFFECT.FAILURE_EFFECT_KEY = SYSTEM_FAILURE.FAILURE_EFFECT_KEY " & _
            "AND     FLIGHT_MODE.FLIGHT_MODE_KEY = SYSTEM_FAILURE.FLIGHT_MODE_KEY " & _
            "AND     SEVERITY_CLASS.SEVERITY_CLASS_KEY = SYSTEM_FAILURE.SEVERITY_CLASS_KEY " & _
            "AND     SYSTEM_MODE_ID.SYSTEM = " & systemID & " " & _
            "ORDER BY SYSTEM_MODE_ID.SYSTEM_MODE_ID "

    ' Copy the records into the Excel Sheet1 (FMEA)
    Sheet1.Range(startData).CopyFromRecordset dataSet

    ' Close RecordSet
    .Close
End With

dbConnection.Close
Set dataSet = Nothing
Set dbConnection = Nothing
```

Figure 40: Excel Macro Database Access source code.

8.2 Delete Data

To delete the information of the excel file a Range starting from the row with the header and ending with the last row in the excel file is defined and the method Delete of the range is called to delete those rows.

```
Sub Delete()

    'Set the range to be deleted
    Dim rngRange As Range

    With Sheet1
        Set rngRange = .Range(.Cells(5, 1), .Cells(.Rows.Count, 1)).EntireRow
    End With

    'Delete this range
    rngRange.Delete

End Sub
```

Figure 41: Excel Macro Delete Data Source Code

9 CONCLUSIONS AND FUTURE WORK

9.1 Conclusions

The use of databases to store information and the use of web applications for accessing the information is replacing old methods like the use of Excel base information. Companies are realizing the benefits obtained by the use of databases and web applications and are searching ways to make use of these technologies.

The use of Excel based information has serious limitations that diminish the productivity of the company. One limitation that affects the productivity is that the information is not in a single place but in different Excel files. The combination of the information of different Excel files is time consuming and error prone.

The system provided offers a solution without the limitations of the Excel based information and at the same time opens new possibilities in the analysis of the FMEA and FMECA information. The database created permits the combination of the FMEA and FMECA information in ways that were not possible using excel. The information stored in a database can be filtered and organized in different ways providing different views of the data that can be used for different purposes. The FMEA and FMECA information is no longer limited to the information of a single Excel file. The information of the different FMEA and FMECA stored in the database can be combined to obtain easily information like all the rows from all the FMEA and FMECA with a specific Part Id. This task would not be feasible using Excel based information. The use of the web pages created provides easy and quick access to the

FMEA and FMECA information from the database. The information is no longer only contained in Excel files that need to be passed to the employees for the analysis of the information. The use of the web pages created removes this limitation by providing a way on which all the employees can access the FMEA and FMECA information by simply accessing a web application in the web browser. The use of web pages also provides the security that all the employees are accessing the same information and removes the possibility of a change in the FMEA and FMECA information, this is important because the use of Excel based information permits the modification of the data by all the employees with access to the Excel files which involves the unwanted risk of a change in the information.

The use of databases and web application for the data storage and data presentation is the approach to follow to make good use of the data available. The use of excel based information as the primary source of information must be left in the past in favor of the use of databases for information storage and a web application for information presentation.

9.2 Future Work

Some improvements that can be done in the future are:

1) Add support for other file formats.
2) Add support for other Aircrafts.
3) Integrate links of the work done in other web applications.

REFERENCES

[Ying12] Ying Chen; Cui Ye; Bingdong Liu; Rui Kang; , "Status of FMECA research and engineering application," Prognostics and System Health Management (PHM), 2012 IEEE Conference on , vol., no., pp.1-9, 23-25 May 2012

[Herman11] Herman, R.M.; Janasak, K.M.; , "Using FMECA to design sustainable products," Reliability and Maintainability Symposium (RAMS), 2011 Proceedings - Annual , vol., no., pp.1-6, 24-27 Jan. 2011

[MIL-STD-1629A] MIL-STD-1629A, "Military Standard Procedures for Performing a Failure Mode, Effects and Criticality Analysis", Nov. 1980

[Resto11] Resto, Pedro; Rodriguez Manuel, "SQL Based Development for the Integration of FMECA Tools and the ASAP Database", Jan. 2011

[MacDonald10] MacDonald, Matthew; "Beginning ASP.NET 4 in C# 2010", Apress, August 2010

[Sawyer08] Sawyer MacFarland, David; "JavaScript The Missing Manual", Pogue Press O'Reilly, July 2008

[Meyer06] Mayer, Eric A; "CSS The Definitive Guide 3rd Edition", O'Reilly, November 2006.

[Mitchell10] Mitchell, Scott; "Sams Teach Yourself ASP.NET 4 in 24 Hours", Sams, June 2010

[Beaulier09] Beaulier, Alan; "Learning SQL 2nd edition", O'Reilly, April 2009

[Krishnaswamy09] Krishnaswamy, Jayaram; "Learning SQL Server 2008 Reporting Services", PACKET Publishing, March 2009

[Atkinson] Atkinson, Paul; Viera, Robert; "Microsoft SQL Server 2012 Programming", WILEY, 2012

[USArmyRAM] US Army RAM Engineering and System Assessment Group