

Software Design Specification

I. Table of Contents

I. TABLE OF CONTENTS.....	1
1.0 INTRODUCTION.....	3
1.1 GOALS AND OBJECTIVES	3
1.2 SYSTEM STATEMENT OF SCOPE	3
1.2.1 General Requirements	3
1.3 SYSTEM CONTEXT	4
1.4 MAJOR CONSTRAINTS	4
2.0 DATA DESIGN	5
2.1 DATABASE DESCRIPTION	5
3.0 ARCHITECTURAL AND COMPONENT-LEVEL DESIGN.....	6
3.1 PROGRAM STRUCTURE.....	6
3.1.1 Overall.....	6
3.1.2 Create Inspection.....	7
3.1.3 During Inspection.....	7
3.1.4 Post-Inspection.....	7
3.1.5 Approval	7
3.2 DESCRIPTION FOR COMPONENTS	7
3.2.1 Switch User	7
3.2.2 Facility.....	8
3.2.3 Create/Modify Inspection – Step 1.....	8
3.2.4 Create/Modify Inspection – Step 2.....	9
3.2.5 File Results – Step 1	9
3.2.6 File Results – Step 2	10
3.2.7 Approval	10
3.2.8 Checklist Maintenance.....	11
3.2.9 Letter Maintenance.....	11
4.0 USER INTERFACE DESIGN.....	13
4.1 DESCRIPTION OF THE USER INTERFACE.....	13
4.1.1 Screen Images	13
Login Screen.....	13
Search Pages	17
Approval Queue	17
4.1.2 Objects and actions.....	17
➤ Menu Items	18
4.2 INTERFACE DESIGN RULES	23
4.3 COMPONENTS AVAILABLE	23
4.3.1 Intrinsic Controls.....	23
4.3.2 ActiveX Controls.....	25
5.0 RESTRICTION, LIMITATIONS, AND CONSTRAINTS.....	26
TIME.....	26
EMPLOYEE SKILLS	26

INSUFFICIENT RESOURCES	26
6.0 TESTING ISSUES.....	27
6.1 CLASSES OF TEST.....	27
6.2 PERFORMANCE BOUNDS	37
6.3 IDENTIFICATION OF CRITICAL COMPONENTS	37
7.0 APPENDICES	39
<i>Rechargeable batteries</i>	39
<i>Palm OS versus Windows CE</i>	39
<i>Screen Size</i>	39
<i>HotSyncing</i>	39
<i>Voice Recognition/cursive writing</i>	39
<i>Handwriting Input</i>	41
<i>Free Programs and Utilities</i>	41

1.0 Introduction

This section describes the design for the **Waste Management Inspection Tracking System (WMITS)**.

1.1 Goals and Objectives

The main purpose of WMITS is to help automate the entire process that the Department of Environmental Quality (DEQ) Waste Management Division (WMD) staff members perform throughout an inspection. The goals of WMITS are:

- To minimize the time span of any inspection
- To minimize the amount of paper work required
- To provide a searchable database of all past inspections
- To provide an automated channel for the public to request information (under Freedom of Information Act)

Critique: The specific goals and objective of the WMITS design should also be discussed.

1.2 System Statement of Scope

1.2.1 General Requirements

The following general requirements were laid out for our project named WMITS:

- A way in which DEQ could add new facilities to the database.
- A way in which DEQ could generate electronic checklists.
- A search on all electronic checklists.
- A way in which they could generate letters to be sent out to facilities based on inspection results.
- A way in which all letters and checklists could be stored electronically.
- A way to search for existing facilities.
- A way to print blank checklists and staff reports.
- A way in which they could view data which was entered into the database prior to our software.
- DEQ wanted a product that would allow them to easily add new checklists and letters or change existing checklists and letters.
- **Interface Enhancements**
Staff members of WMD have requested a lot of interface enhancements that will increase the usability of the product for the staff.

- **Database Administrative Interface**
There is current no documented interface for WMD staff members to maintain the checklist and letter templates. Should no such interface existed, Cyber Rovers will have to implement one from scratch.
- **Online Help**
To develop an extensive help menu for the users and also to setup the online help for the need of the help in the future.
- **Training**
The staff members have also requested throughout training for the entire staff for use with the software.

We will also implement a web-based helpdesk for WMD staff members to report bugs and request support. The helpdesk will be located at <http://www.cyberrovers.com>.

1.3 System Context

Eventually, multiple users will be using the product simultaneously. Therefore, concurrent connection will be an issue for implementation. In addition, this is a pilot product that hopefully, if successful, can be used in other locations as well. This leads to issues about future support for a larger user base.

1.4 Major Constraints

Time

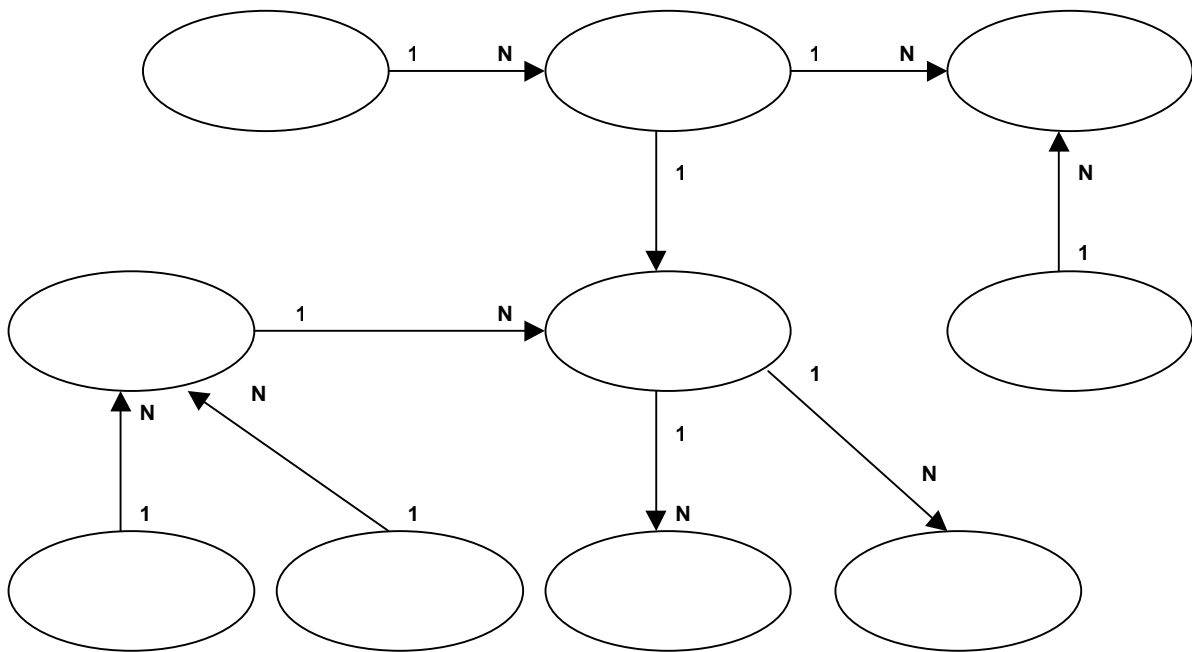
We only have about two months to finish all documentation, software creation and enhancements. We have a lot of ideas but cannot implement them due to time constraint. One of the major ones is to move the application to be completely browser-based.

Funding

To develop and implement the Palm Pilot integration, we will need funding to buy at least one Palm Pilot. We will request the funding from University of Michigan – Dearborn should we decided to pursue this function.

2.0 Data Design

2.1 Database Description

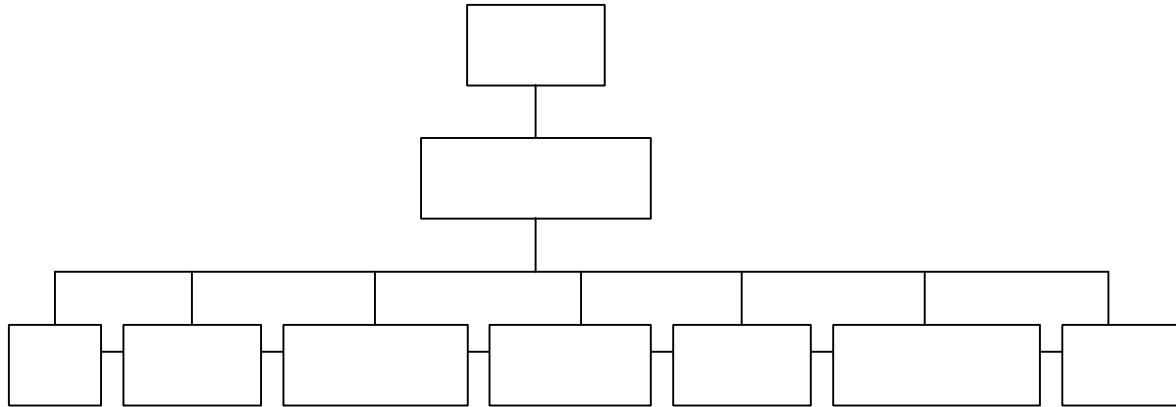


Entities	Each record in the entity represents a(n)...
TITLE	Staff title
STAFF	Staff member
ACCESS	Facility
MODULE	Module
FACILITY TYPE	Facility Type
EPA_CODE	EPA Code
INSPECTION	Inspection
INSPECTION_DETAIL	Inspection Checklist Item
INSPECTION_CHKLIST	Inspection Checklist

3.0 Architectural and Component-Level Design

3.1 Program Structure

3.1.1 Overall



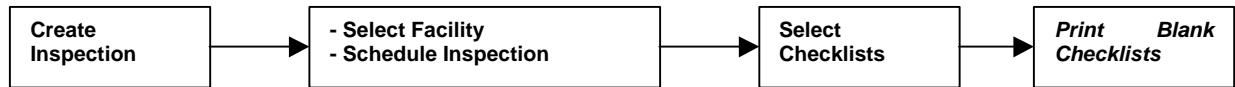
Menu Items

The following shows the architecture of the main menu:

- **File**
 - Switch User
 - Exit
- **Facility**
- **Inspection**
 - Create/Modify
 - File Results
 - Get Approval
 - Print Letters
 - View Schedule
- **Approval**
- **Reports**
 - Print Staff Report
 - Print Blank Checklists
 - Efficiency Report
 - ... (more TBD)
- **Maintenance**
- **Checklists**
- **Letters**
- **Users**
- **Options**
- **Help**
 - Contents
 - About



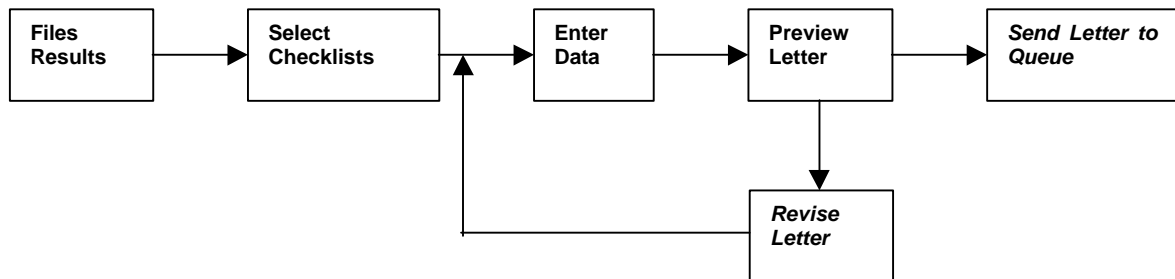
3.1.2 Create Inspection



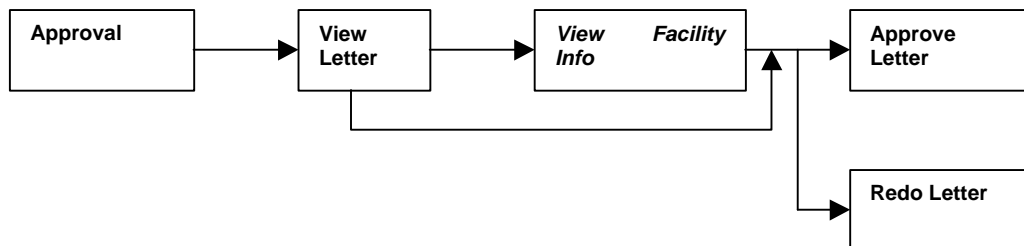
3.1.3 During Inspection



3.1.4 Post-Inspection



3.1.5 Approval



3.2 Description for Components

3.2.1 Switch User

Major Form(s) : *frmLogin*
Major Action(s) : Login

The form *frmLogin* will appear. User enters their username in *txtUserName* and password in *txtPassword*. Then click *cmdOkay*. User will be logged in if it is a valid username and password pair. If user clicks *cmdCancel* on this form, application will end if they confirmed their action.

3.2.2 Facility

Major Form(s) : *frmFacility*, *frmFacilityBrowse*, and *frmFacilitySearch*
Major Action(s) : Browse, Search, Save, Delete and View

Browse

Object Name: *cmdBrowse*

To browse all facilities, user can click on the bomb button next to the *txtFacilityID* field. *frmFacilityBrowse* will appear. User can highlight a facility in the grid, then click *cmdOkay*. All the information on *frmFacility* will be filled in.

Search

Object Name: *cmdSearch*

To search for a facility, user can click on the multi-page document button next to the *txtFacilityID* field. *frmFacilitySearch* will appear. User can highlight a facility in the result grid, the click *cmdOkay*. All the information on *frmFacility* will be filled in.

Save

Object Name: *cmdSave*

The Save button should be disabled unless the *txtFacilityID* field is filled in, and any there have been changes made to any field on the *frmFacility*. When the Save button is clicked, new record will be generated if the Facility ID does not exist in the system, otherwise, current record will be updated.

Delete

Object Name: *cmdDelete*

The Delete button should be disabled unless no historical data have been found for the facility. The Delete and View button should never be enabled at the same time.

View

Object Name: *cmdView*

The View button should be disabled unless historical data have been found for the facility. The Delete and View button should never be enabled at the same time.

3.2.3 Create/Modify Inspection – Step 1

Major Form(s) : *frmInspection*, *frmInspectionBrowse*, *frmInspectionSearch*,
frmFacilityBrowse, and *frmFacilitySearch*
Major Action(s) : Create Inspection, Modify Inspection (Inspection Browse,
Inspection Search), Next Step and Add/Edit/Delete Date

Create Inspection

To create an inspection, user needs to enter a new Inspection ID in the *txtInspectionID* field. To use an automatic number, click on the blank paper button, an automatic number will be generated and filled in the *txtInspectionID* field.

User also needs to enter the Facility ID. The browse and search functions are identical to the ones in the Facility module.

Modify Inspection

To modify an inspection, user needs to enter an existing Inspection ID in the *txtInspectionID* field. To browse all open inspections, user can click on the bomb button next to the *txtInspectionID* field. *frmInspectionBrowse* will appear. User can highlight an inspection in the grid, then click *cmdOkay*. All the information on *frmInspection* will be filled in.

Add Onsite Visit Date

User needs to click on the calendar icon, pick a date. *txtTime* can be left blank. User can then click *cmdDateAdd* to add the date to the grid.

Edit/Delete Onsite Visit Date

If a user highlights a row in the date grid, he or she can edit the contents in the *txtDate* (using the calendar control) and *txtTime* field. Then he or she can click on *cmdDateEdit* to update the record. The user can also click *cmdDateDelete* to remove the highlighted row.

Next Step

frmPickCheckList will appear. See Create/Modify Inspection – Step 2 for more info.

3.2.4 Create/Modify Inspection – Step 2

Major Form(s)	: <i>frmPickCheckList</i>
Major Action(s)	: Pick Checklist(s), Finish

For the second step user needs to pick the checklist(s) that they will be using for the inspection using the intuitive *cmdAdd* and *cmdRemove* buttons. Once they have clicked *cmdFinish*, the inspection will be created/modified in the database.

3.2.5 File Results – Step 1

Major Form(s)	: <i>frmFileResults1</i> , <i>frmInspectionBrowse</i> , and <i>frmInspectionSearch</i>
Major Action(s)	: Browse/Search Inspections, Next Step

To file inspection results, user needs to first select a previously created inspection. He or she can use the standard browse and search function to locate the inspection. After an inspection has been chosen, user needs to choose the checklist(s) that they want to file the results. Click *cmdNext* to proceed to Step 2.

3.2.6 File Results – Step 2

Major Form(s) : *frmFileResults2*
Major Action(s) : Add/Edit/Remove comments and status for each inspection item,
Preview Letter, Spell Check (2)

Add Comments

To add comment for an inspection item in the checklist, user needs to highlight an item in *lstRegulations*, enter comments in the *txtComments* and select status in *cboStatus*. Then he or she needs to click *cmdAdd*. The item will be added to *lstInspected*.

Remove Comments

To remove comments for an added inspection item. User needs to highlight an item in *lstInspected* then click *cmdRemove*.

Edit Comments

To edit comments for an added inspection item. User needs to highlight an item in *lstInspected*. The comments for that item will be shown to the right. The user can then update the comments and click *cmdEdit* to apply the changes.

Spell Check

User can click on *cmdSpellCheck1* (for Add) and *cmdSpellCheck2* (for Edit) to check the spellings in the respective comments box.

Preview Letter

When *cmdPreview* is clicked, the application will generate a letter in the right portion of the screen. Which letter to use will depend on the letter chosen in *cboLetters*.

3.2.7 Approval

Major Form(s) : *frmApproval*, *frmLetterDisplay*
Major Action(s) : View, Approve, and Deny

frmApproval will include a data grid that shows the approval requests submitted by the inspectors. The user has the option to view open or closed requests on the data grid. Once a row has been highlighted, the user can click *cmdView* to view the letter. *frmLetterDisplay* will appear. He or she then has the option to either approve (*frmLetterDisplay.cmdApprove*) or deny (*frmLetterDisplay.cmdDeny*) it. The user can also click on the *frmApproval.cmdApprove* and *frmApproval.cmdDeny* to perform those actions without viewing the letter.

3.2.8 Checklist Maintenance

Major Form(s) : *frmMaintainChecklists*
Major Functions : New, Browse/View, Save, View, Delete, Close

New

When *cmdNewChecklist* is clicked, *frmMaintainChecklists* will be cleared and the internal checklist ID, *intChecklistID* will be set to null as well.

Browse

When *cmdBrowseChecklists* is clicked, *frmBrowseChecklists* will appear. After the user made a selection in *frmBrowseChecklists* (see details in the *frmBrowseChecklists* component description), *txtChecklistID* will be filled in by the value returned by *frmBrowseChecklists*. *frmMaintainChecklists* will be refreshed with all the details of that checklist. The bottom data grid will be filled in with items from the checklist.

Remove

When the user click on *cmdDeleteChecklist*, a warning message will appear to confirm the user's action. If the user confirms the action, the appropriate checklist header record in CHKLIST_HEADER will be removed. Records in CHKLIST_ITEM that are related to that header will also be removed. Afterwards, *lstChecklists* will be refreshed.

Close

When *cmdCloseChecklist* is clicked, if changes were made to any field within the form, a warning message will appear to confirm user's action. If the user confirms, the form will be closed and no actions will be performed on the database. If the user does not confirm, user will simply be returned to the form.

Save

When the user clicks on save, if *intChecklistID* already exists in the database, the header record in CHKLIST_HEADER with CH_ID equals to *intChecklistID* will be updated. Also, all records in CHKLIST_ITEM with CI_CH_ID equals to *intChecklistID* will be updated as well. If *intChecklistID* does not exist, a header record will be recorded in CHKLIST_HEADER. Appropriate number of records will also be created in CHKLIST_ITEM.

3.2.9 Letter Maintenance

Major Form(s) : *frmMaintainLetters*
Major Functions : New, Browse/View, Save, View, Delete, Close



New

When *cmdNewLetter* is clicked, *frmMaintainLetters* will be cleared and the internal letter ID, *intLetterID* will be set to null as well.

Browse

When *cmdBrowseLetters* is clicked, *frmBrowseLetters* will appear. After the user made a selection in *frmBrowseLetters* (see details in the *frmBrowseLetters* component description), *txtLetterID* will be filled in by the value returned by *frmBrowseLetters*. *frmMaintainLetters* will be refreshed with all the details of that letter. The bottom data grid will be filled in with items from the letter.

Remove

When the user click on *cmdDeleteLetter*, a warning message will appear to confirm the user's action. If the user confirms the action, the appropriate letter header record in LETTER_HEADER will be removed. Records in LETTER_ITEM that are related to that header will also be removed. Afterwards, *lstLetters* will be refreshed.

Close

When *cmdCloseLetter* is clicked, if changes were made to any field within the form, a warning message will appear to confirm user's action. If the user confirms, the form will be closed and no actions will be performed on the databaswe. If the user does not confirm, user will simply be returned to the form.

Save

When the user clicks on save, if *intLetterID* already exists in the database, the header record in LETTER_HEADER with LTR_ID equals to *intLetterID* will be updated. Also, all records in LETTER_ITEM with LTRI_LTR_ID equals to *intLetterID* will be updated as well. If *intChecklistID* does not exist, a header record will be recorded in LETTER_HEADER. Appropriate number of records will also be created in LETTER_ITEM.



4.0 User Interface Design

There will be about 30 interfaces in the program. We can't design on the exact number of it yet. Because the clients still have to think over on several interfaces, to see rather they can be combined some of the forms or put some them in separated forms.

4.1 Description of the User Interface

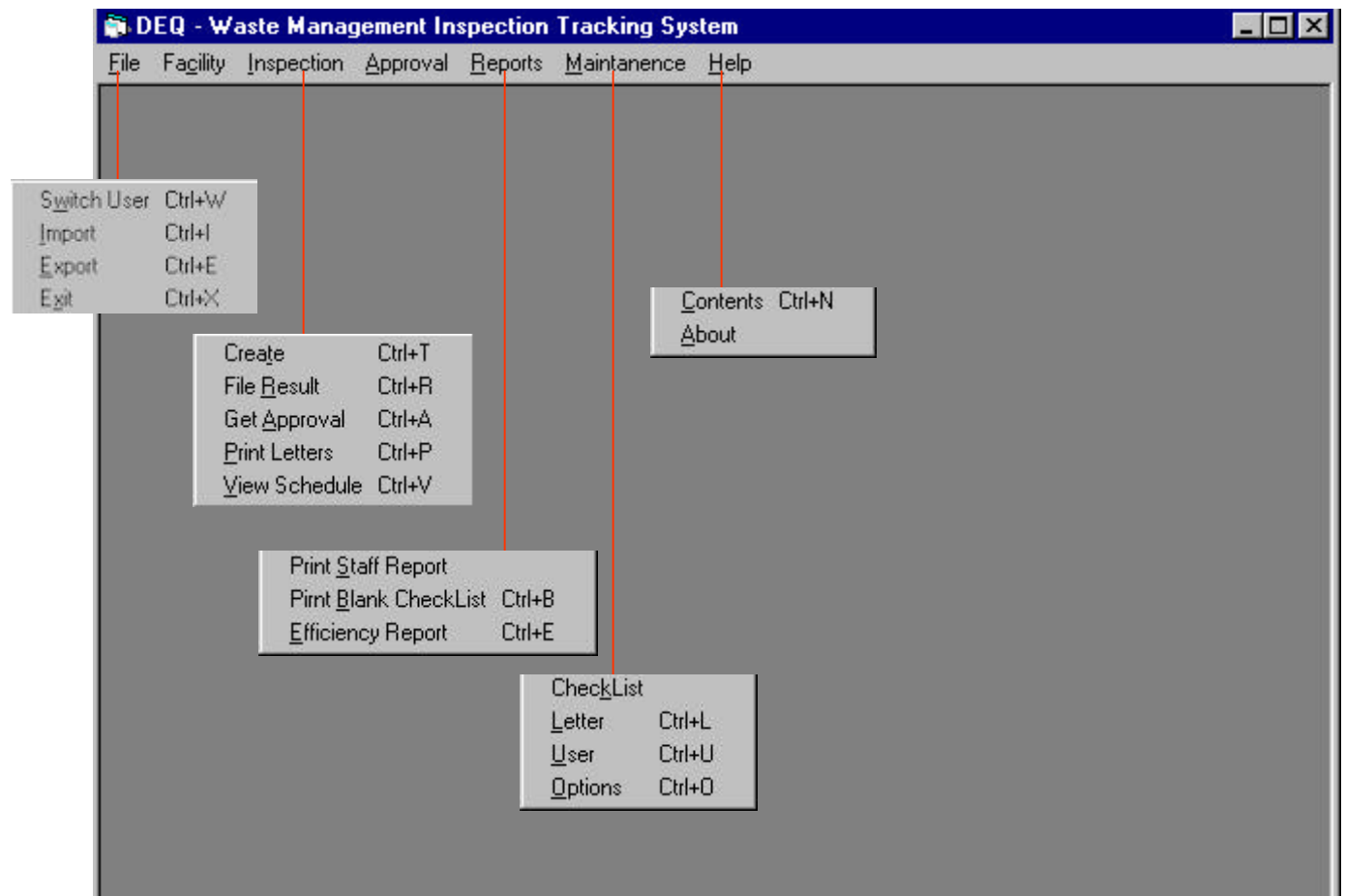
Below are some of the forms in the program. After fire up the program, the login screen will appear. If the users enter the right user name with the matching password, it will immediately take them to the main interface (mdiDEQ).

4.1.1 Screen Images

Login Screen

A screenshot of a Windows-style dialog box titled "Waste Management Division Login". It contains two text input fields: "User Name:" and "Password:". Below the fields are two buttons: "OK" and "Cancel". The dialog box has a standard Windows title bar with a close button (X) in the top right corner.

Main Interface





Generate Check List

Generate Check List (Letter)

Available Checklist Items

Select Check List
List

1
10
11
12
12a
12b
1a
3
5
5a

Determined if waste streams are hazardous waste? (Rule 302: 40 CFR 262.11)

Additional Comments

☐ Show in letter Compliance Status **Add**

Selected Checklist Items

Select Check List
List

Additional Comments for Internal

Additional Comments for Outgoing Letter

Remove ☐ Show in letter Compliance Status **Update**

Choice a Letter Format **Preview**

Generate Check List

Preview

Export **Cancel** **Help**



Facility Information

Facility Information					
Facility ID		EPA ID		Facility Type	
<input type="text" value="FC4802B"/>		<input type="text"/>		<input type="text"/>	
<input type="button" value="Key"/> <input type="button" value="Copy"/>				<input type="checkbox"/> TSD	
Name		Address		Contact Person	
<input type="text"/>		<input type="text"/>		<input type="text"/>	
City	County	Zip Code	Fax Number	Phone Number	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
Mailing Information					
Name		Address		Contact Person	
<input type="text"/>		<input type="text"/>		<input type="text"/>	
City	State	Zip Code	Fax Number	Phone Number	
<input type="text"/>	<input type="text" value="MI"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
Owner Information					
Company		Address		Owner	
<input type="text"/>		<input type="text"/>		<input type="text"/>	
City	State	Zip Code	Fax Number	Phone Number	
<input type="text"/>	<input type="text" value="MI"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
Comments					
<input type="text"/>					
<input type="button" value="Save"/> <input type="button" value="Cancel"/> <input type="button" value="Help"/>					



Search Pages

Facility Browse

Start At

Up arrow, Down arrow

OK Cancel View

Search Inspection ID

Start At

Up arrow, Down arrow

OK Cancel View

Approval Queue

Letters Waiting for Approval

View Approve Redo Help Cancel



4.1.2 Objects and actions

1. Login Screen

User Name

User name can be ranged from 6 to 20 letters (numbers), as the industry standard. No special characters, space. And mostly likely the users will use their DEQ user name for this system as well.

There are 10 people using this project in DEQ, as current status. Eight of them are inspectors, one manager, and one network administrator. Even with a small group like that, we are not going to use the drop down menu for the login name. Logically speaking, drop down menu do save time, and more convenience. But we change our mind after one of the inspector we corresponded with make an interesting comments, “if someone can’t even remember their login name, they would find a new job.” Further more, with a blind login name system, it will provide extra level of security. Because it’s harder to get a user’s user name and password compare to just get their password.

Password

Password can be ranged from 6 to 20 letters (numbers), as the industry standard. No special characters, space.

OK

If the users enter the right user name with the matching password, it will immediately take them to the main interface.

Cancel

If the user wishes to exit the program, hit the “Cancel” button.

2. Main Interface

➤ Menu Items

The followings show the architecture of the main menu:

- **File**
 - Switch User
 - Exit
- **Facility**
- **Inspection**
 - Create/Modify
- File Results
- Get Approval
- Print Letters
- View Schedule
- **Approval**



- **Reports**
 - Print Staff Report
 - Print Blank Checklists
 - Efficiency Report
 - ... (more TBD)
- **Maintenance**
 - Checklists
- Letters
- Users
- Options
- **Help**
 - Contents
 - About

➤ **Switch User**

Major Form(s) : *frmLogin*
Major Action(s) : Login

The form *frmLogin* will appear. User enters their username in *txtUserName* and password in *txtPassword*. Then click *cmdOkay*. User will be logged in if it is a valid username and password pair. If user clicks *cmdCancel* on this form, application will end if they confirmed their action.

➤ **Facility**

Major Form(s) : *frmFacility*, *frmFacilityBrowse*, and *frmFacilitySearch*
Major Action(s) : Browse, Search, Save, Delete and View

➤ **Browse**

Object Name: *cmdBrowse*

To browse all facilities, user can click on the bomb button next to the *txtFacilityID* field. *frmFacilityBrowse* will appear. User can highlight a facility in the grid, then click *cmdOkay*. All the information on *frmFacility* will be filled in.

➤ **Search**

Object Name: *cmdSearch*

To search for a facility, user can click on the multi-page document button next to the *txtFacilityID* field. *frmFacilitySearch* will appear. User can highlight a facility in the result grid, then click *cmdOkay*. All the information on *frmFacility* will be filled in.

➤ **Save**

Object Name: *cmdSave*

The Save button should be disabled unless the *txtFacilityID* field is filled in, and any there have been changes made to any field on the *frmFacility*. When the Save button is clicked, new record will be generated if the Facility ID does not exist in the system, otherwise, current record will be updated.

➤ **Delete**

Object Name: *cmdDelete*

The Delete button should be disabled unless no historical data have been found for the facility. The Delete and View button should never be enabled at the same time.

➤ **View**

Object Name: *cmdView*

The View button should be disabled unless historical data have been found for the facility. The Delete and View button should never be enabled at the same time.

➤ **Create/Modify Inspection – Step 1**

Major Form(s) : *frmInspection*, *frmInspectionBrowse*, *frmInspectionSearch*,
frmFacilityBrowse, and *frmFacilitySearch*

Major Action(s) : Create Inspection, Modify Inspection (Inspection Browse, Inspection Search), Next Step and Add/Edit/Delete Date

➤ **Create Inspection**

To create an inspection, user needs to enter a new Inspection ID in the *txtInspectionID* field. To use an automatic number, click on the blank paper button, an automatic number will be generated and filled in the *txtInspectionID* field.

User also needs to enter the Facility ID. The browse and search functions are identical to the ones in the Facility module.

➤ **Modify Inspection**

To modify an inspection, user needs to enter an existing Inspection ID in the *txtInspectionID* field. To browse all open inspections, user can click on the bomb button next to the *txtInspectionID* field. *frmInspectionBrowse* will appear. User can highlight an inspection in the grid, then click *cmdOkay*. All the information on *frmInspection* will be filled in.

➤ **Add Onsite Visit Date**

User needs to click on the calendar icon, pick a date. *txtTime* can be left blank. User can then click *cmdDateAdd* to add the date to the grid.

➤ **Edit/Delete Onsite Visit Date**

If a user highlights a row in the date grid, he or she can edit the contents in the *txtDate* (using the calendar control) and *txtTime* field. Then he or she can click on *cmdDateEdit* to update the record. The user can also click *cmdDateDelete* to remove the highlighted row.

➤ **Next Step**

frmPickCheckList will appear. See Create/Modify Inspection – Step 2 for more info.

➤ **Create/Modify Inspection – Step 2**

Major Form(s) : *frmPickCheckList*

Major Action(s) : Pick Checklist(s), Finish

For the second step user needs to pick the checklist(s) that they will be using for the inspection using the intuitive *cmdAdd* and *cmdRemove* buttons. Once they have clicked *cmdFinish*, the inspection will be created/modified in the database.

➤ **File Results – Step 1**

Major Form(s) : *frmFileResults1*, *frmInspectionBrowse*, and *frmInspectionSearch*

Major Action(s) : Browse/Search Inspections, Next Step

To file inspection results, user needs to first select a previously created inspection. He or she can use the standard browse and search function to locate the inspection. After an

inspection has been chosen, user needs to choose the checklist(s) that they want to file the results. Click *cmdNext* to proceed to Step 2.

➤ **File Results – Step 2**

Major Form(s) : *frmFileResults2*

Major Action(s) : Add/Edit/Remove comments and status for each inspection item,
Preview Letter, Spell Check (2)

➤ **Add Comments**

To add comment for an inspection item in the checklist, user needs to highlight an item in *lstRegulations*, enter comments in the *txtComments* and select status in *cboStatus*. Then he or she needs to click *cmdAdd*. The item will be added to *lstInspected*.

➤ **Remove Comments**

To remove comments for an added inspection item. User needs to highlight an item in *lstInspected* then click *cmdRemove*.

➤ **Edit Comments**

To edit comments for an added inspection item. User needs to highlight an item in *lstInspected*. The comments for that item will be shown to the right. The user can then update the comments and click *cmdEdit* to apply the changes.

➤ **Spell Check**

User can click on *cmdSpellCheck1* (for Add) and *cmdSpellCheck2* (for Edit) to check the spellings in the respective comments box.

➤ **Preview Letter**

When *cmdPreview* is clicked, the application will generate a letter in the right portion of the screen. Which letter to use will depend on the letter chosen in *cboLetters*.

➤ **Approval**

Major Form(s) : *frmApproval, frmLetterDisplay*

Major Action(s) : View, Approve, and Deny

frmApproval will include a data grid that shows the approval requests submitted by the inspectors. The user has the option to view open or closed requests on the data grid. Once a row has been highlighted, the user can click *cmdView* to view the letter. *frmLetterDisplay* will appear. He or she then has the option to either approve (*frmLetterDisplay.cmdApprove*) or deny (*frmLetterDisplay.cmdDeny*) it. The user can also click on the *frmApproval.cmdApprove* and *frmApproval.cmdDeny* to perform those

View

View details of the report

Redo

Send back to the inspector, either ask the inspector redo it, or take it over on some details

Approval

Approval the report, and computer will generate a message, tells that the report has been approval. Once the inspector sees the message, he/she will print out the report, and send it to the facility.

4.2 Interface Design Rules

Interface design focuses on three areas of concern:

1. The design of interfaces between software modules;
2. The design of interfaces between the software and other nonhuman producers and consumers of information (i.e., other external entities);
3. The design of the interface between a human (i.e., the user) and the computer.

Easy to Learn

Readability

Easy navigate between interfaces

4.3 Components Available

Since we are using Visual Basic as our front-end development language, there are a lot of ready-made components that are available for us to use already. The following is a list of controls that we will be using for this software.

4.3.1 Intrinsic Controls

TextBox

A TextBox control, sometimes called an edit field or edit control, displays information entered at design time, entered by the user, or assigned to the control in code at run time.

Label

A Label control is a graphical control you can use to display text that a user can't change directly.

Line

A Line control is a graphical control displayed as a horizontal, vertical, or diagonal line.

Image

Use the Image control to display a graphic. An Image control can display a graphic from a bitmap, icon, or metafile, as well as enhanced metafile, JPEG, or GIF files.

ListBox

A ListBox control displays a list of items from which the user can select one or more. If the number of items exceeds the number that can be displayed, a scroll bar is automatically added to the ListBox control.

Scrollbars

Scroll bars provide easy navigation through a long list of items or a large amount of information. They can also provide an analog representation of current position. You can use a scroll bar as an input device or indicator of speed or quantity—for example, to control the volume of a computer game or to view the time elapsed in a timed process.

CommandButton

Use a CommandButton control to begin, interrupt, or end a process. When chosen, a CommandButton appears pushed in and so is sometimes called a push button.

Menu

A Menu control displays a custom menu for your application. A menu can include commands, submenus, and separator bars. Each menu you create can have up to four levels of submenus.

ComboBox

A ComboBox control combines the features of a TextBox control and a ListBox control—users can enter information in the text box portion or select an item from the list box portion of the control.

Checkbox

A CheckBox control displays an X when selected; the X disappears when the CheckBox is cleared. Use this control to give the user a True/False or Yes/No option. You can use CheckBox controls in groups to display multiple choices from which the user can select one or more. You can also set the value of a CheckBox programmatically with the Value property.

OptionButton

An OptionButton control displays an option that can be turned on or off.

Shape

The Shape control is a graphical control displayed as a rectangle, square, oval, circle, rounded rectangle, or rounded square.

4.3.2 ActiveX Controls

DataRepeater

The DataRepeater control functions as a scrollable container of data-bound user controls. Each control appears in its own row as a "repeated" control, allowing the user to view several data-bound user controls at once.

DateTimePicker

The DateTimePicker control enables you to provide a formatted date field that allows easy date selection. In addition, users can select a date from a dropdown calendar interface similar to the MonthView control.

MSFlexGrid

The Microsoft FlexGrid (MSFlexGrid) control displays and operates on tabular data. It allows complete flexibility to sort, merge, and format tables containing strings and pictures. When bound to a Data control, MSFlexGrid displays read-only data.

TreeView

A TreeView control displays a hierarchical list of Node objects, each of which consists of a label and an optional bitmap. A TreeView is typically used to display the headings in a document, the entries in an index, the files and directories on a disk, or any other kind of information that might usefully be displayed as a hierarchy.

StatusBar

A StatusBar control provides a window, usually at the bottom of a parent form, through which an application can display various kinds of status data. The StatusBar can be divided up into a maximum of sixteen Panel objects that are contained in a Panels collection.

Toolbar

A Toolbar control contains a collection of Button objects used to create a toolbar that is associated with an application.

CommonDialog

The CommonDialog control provides a standard set of dialog boxes for operations such as opening and saving files, setting print options, and selecting colors and fonts. The control also has the ability to display help by running the Windows Help engine.



5.0 Restriction, Limitations, and Constraints

Time

Time is so far the biggest restriction or constraint for our project as we only have around three months to finish entire project. It is very important for us to watch the time we spend over every phase of the software development project. We could have included many more components to the software like online help menu but time restricts us from doing so.

Employee Skills

Employees programming and design skills is also one of the restriction. It does not have as big of an impact on the project as time but it sure does limit us from doing more addition to the projects.

Insufficient Resources

Not having all the necessary instruments also is a problem for our software. We planed to use latest equipment for the project like Hand Held PC with keyboard etc. but the employers cannot afford to provide such tools to all their employees so we had to abandon the plans.



6.0 Testing Issues

To validate the software we need to test the software. During the testing we will be concerned about the inputs and their expected outputs. We emphasize on the testing where we will input the data and will compare the output with the expected results. At this stage, we are not concerned about the process; we are only looking for the correct outputs.

6.1 Classes of Test

The software has many different additions and new interfaces/functions added to it. We will go through each of the new interface and other software function to describe different types of test performed on them.

Interface

We are creating new user interface using Visual Basic. This interface allows user to easily fill the checklist and thus enter data regarding inspection into the database. The older system makes inspector's work very tedious with use of several different windows to complete the form. The interface that we will design will enable user to complete the entire form with use of single window. The tests to be carried on these interface windows are described below.

Black Box testing

We are concern about inputs of data into the software and their expected outputs. So we will carry out black box testing where several different steps will be taken to test the software so that when we use the software with specific data all the given out puts will match our expected results. We will compare the output with our expected output list and will find out credibility of the software.

Login Window

We will make use several different names to log in to the system. We will use correct and incorrect User Names and Passwords to access the software and thus access database. We will not be allowed to log in using incorrect passwords and error message will be shown. When correct password is presented we will be able to log in be able to next window (DEQ- Microsoft Visual Basic window). We will also test OK and Cancel buttons on this window by performing test above.

DEQ – Microsoft Visual Basic [Design] Window

This is the main window that we will use to access the database using Visual Basic. We will have several different drop down menu in this window. File, Facility, Inspection, approve, Reports, Maintenance and Help are the drop down menu that will be available in this window we will try to use all the menus and than different options available in each of the window.



- File: When file button is clicked user will be shown two choices.

Switch User

We will test if the Switch User button works. We need to make sure that the login screen is presented when this choice is selected. We will test this by choosing the Switch User and then by logging in as another user in Login Window.

Exit

We want to make sure that user is logged out or is able to exit out when Exit button is selected. We will test it by first logging in as a user and then by utilizing Exit button to exit out.

- Facility:
When a Facility button is clicked we will be presented with Facility Screen. We can add or edit facility using the screen. We will test to make sure that the facility screen is presented when the button is clicked. We will do this by clicking on the button and making sure that the Facility Screen is presented.
- Inspection: When Inspection button is clicked user will be shown five choices.

Create/Modify

When click on this choice we will be presented with Inspection Step 1 Screen used to create or modify inspection. We want to make sure that this screen is presented when selection is made. We will test it by making the selection in drop down menu and by making sure that the correct window is presented.

File Results

When we select this choice we will be presented with File Result Window. We will test the button by clicking on it to go to File Result Window. We want to make sure that the proper window is presented.

Get Approval

When we select this choice we will be presented with Get Approval Window. We will test the button by clicking on it to go to Get Approval Window. We want to make sure that the proper window is presented.

Print Letters

When we select this choice we will be presented with Print Letter Window. We will test the button by clicking on it to go to Print Letter Window. We want to make sure that the proper window is presented

View Schedule

When we select this choice we will be presented with View Schedule Window. We will test the button by clicking on it to go to View Schedule Window. We want to make sure that the proper window is presented



- Approval:

This option is presented and works only for manger. We need to make sure that when someone other than inspector loges in the button is disabled. We will do this by logging in as both as inspector and as manger. We will make sure that manger is able to use the button while inspectors will not be able to use it. We also need to make sure that when selected by manager software will bring up window containing queued list of letters waiting to be approved.

- Reports:

Print Staff Reports

When we select this choice we will be presented with Report Window. We will test the button by clicking on it to go to Report Window. We want to make sure that the proper window is presented

Print Blank Checklists

When we select this choice we will be presented with Report Window. We will test the button by clicking on it to go to Report Window. We want to make sure that the proper window is presented

Efficiency Report

When we select this choice we will be presented with Report Window. We will test the button by clicking on it to go to Report Window. We want to make sure that the proper window is presented

- Maintenance:

Checklist

When we select this choice we will be presented with Checklist Window. We will test the button by clicking on it to go to Checklist Window. We want to make sure that the proper window is presented

Letters

When we select this choice we will be presented with Letters Window. We will test the button by clicking on it to go to Letters Window. We want to make sure that the proper window is presented

Users

When we select this choice we will be presented with Users Window. We will test the button by clicking on it to go to Users Window. We want to make sure that the proper window is presented

Options

When we select this choice we will be presented with Option Window. We will test the button by clicking on it to go to Option Window. We want to make sure that the proper window is presented



- Help:

Content

When we select this choice we will be presented with Content Window (help contents). We will test the button by clicking on it to go to the Window. We want to make sure that the proper window is presented.

About

When we select this choice we will be presented with About Window (window with software Info). We will test the button by clicking on it to go to Option Window. We want to make sure that the proper window is presented.

Inspection Generator Window

This window is to allow inspector to generate the inspection ID before actually going out for inspection. We will have several windows to select options form and also buttons to test from.

Inspection ID box automatically generates Inspection ID. We will make sure that the ID is automatically generated and that the ID dose not conflict with previous ID. We will test it by comparing the ID number with other automatically generated ID.

Facility ID box where user will select facility by clicking on facility from the facility list. We will make sure that all the facilities are available here and that correct facility information is associated with the facility.

We want to make sure that common information regarding that facility is shown in the Details window. We have to make sure that the information shown here is correct. To so we will use Facility ID known to us and this will bring up the information that we already know. Now we can check available information to us with information provided by software.

We want to make sure that all the boxes for the Scheduling information works properly. We will insert certain Date, Time and Comments regarding inspection. After details are filled we will use schedule button, which will create the schedule of the inspection in database with Date and Time specified.

We want to make sure that when Follow up inspection box is selected software will be able to bring up the original inspection checklist on which we are doing follow up. To perform this test we will select the button and will hit next which should bring us the old inspection.



We also want to make sure that when we click next after filling out all information in this window. We are presented with the print checklist window. From this window we will be able to select blank check list and print it out.

First window where user needs to make inspection form is the window from where inspector selects appropriate form from given choices. We will also check Add and Remove buttons to make sure that the selected checklist is

We will test Cancel button to make sure that when clicked it will cancel all the work done or will delete all the options selected. And will take us to main VB Window.

We will check the help button to see if the help menu is made available to the user when help button is selected. The menu should popup without any problems when help button is selected.

Print Inspection Form Window

This window will allow inspector to select and print blank print list. We want to make sure that all the available print lists are made available and that inspector is able to select them.

We want to use print button and to make sure that this will bring up print menu to send print job to desired printer. This should work without any problem.

We will test Cancel button to make sure that when clicked it will cancel all the work done or will delete all the options selected. And will take us to main VB Window.

Choosing Inspection Window

Inspector will use Inspection ID number generated automatically previously to bring up the inspection details. We will test to see if the software accepts the correct ID numbers and denies incorrect ID's. We will do this by inserting correct and incorrect ID into the software.

We will test Explode button (button with bomb), this should bring up the window that will allow us to search for the inspection with use of Inspection ID. We will test this by typing existing ID and then will try to view its content. This should work without any trouble.

When we select or type in Inspection ID it will show the checklist that we had selected when we created the inspection in Inspection Generator window. The list of checklist will be shown in Inspection Check List box. We will test the accuracy by comparing with our own data and list of checklist.

We want to test the box that allows us to select and Add/Remove inspection checklist from the Selected Check List box. We will do this by selecting checklist from given list and then by adding it to Selected Check List box.

We will test Cancel button to make sure that when clicked it will cancel all the work done or will delete all the options selected. And will take us to main VB Window.

We will test Next button to see if it takes us to Main Checklist Window. We will do this by filling in all the information and then clicking the next button.

Main Checklist Window

Available Checklist Items Window

We will use several different menus and buttons to select items from so we will be testing each of these buttons

We will try to select items from the available checklist by highlighting them

We will also test if we can add more than one item in to the inspected item boxes from the regulation box. If the software is written correctly we should be able to do this without any troubles.

We will also check for the description associated with regulation window in the details of the regulation window. The details of the regulation must match with the list of details and their regulation number provided by Detroit Environmental Quality department.

We will test the additional comment window to see if the user is able to insert text in to it. We will also check the correctness of the text from the letter generated by comparing it with what we have entered in the additional comments box. We will also check if the spell checks function works in this window by typing the incorrect spelling this should give us error or will draw line under the wrong text.

We will test “Show in letter checkbox” to see if it adds the content in the letter generated. We will select the box and upon completion of the form will see if the content are correctly inserted in letter. This should work without any problems.

We will have to test Compliance Status box to see if the selected compliance is inserted in to the letter. We will do this by choosing or highlighting an option from the Compliance status drop box menu and this will include the selected status into the letter.

We will use Add button to see if the selected information from the form is added to the letter and database. This should work without any troubles.

Selected Checklist Items

We will test if we are able to remove an item from inspected item box by using remove highlighted item button. We will make sure that only the selected item is removed in case the box contains more than one item in it. If the program is written correctly this should work.

We will test preview button to see if the letter is generated in the window to the right of the Generate checklist window. This window will contain the letter with listing of all the regulations selected. The letter should be generated without any troubles.

We will use update button to see if the changes made to the checklist are updated so that things to be removed are removed and things to be added are added.

We will test for the correctness of the letter generated, that is all the regulation are listed and their description presented must match the items we entered during the test and the list of details provided from DEQ listing the regulations and details associated with them. We should achieve desired results without any problems.

We will check Create letter button to see if the final version of the letter is generated. We will test this by clicking on the button, which should give us final letter.

We will check the cancel button to see if the user can exit out of the window without any problems when the button is selected. This should work without any problem.

We will check the help button to see if the help menu is made available to the user when help button is selected. The menu should popup without any problems when help button is selected.

At the end will test the save button to see if all the data is correctly recorded and the generated letter is saved. This should work without any troubles.

We will use Export button, which will pipe the letter to the word window. We will click the button while having letter in front of us and this will open the Microsoft word and letter will be piped in it. This should work without any troubles.

Inspection ID Explode Window

This window will come up when Explode button is selected from Choosing Inspection Window. It is created to allow user to find the Inspection ID.

We want to make sure that Start At box works, that is when first letter or character is typed of the Inspection ID we will get to the sorted list of all the Facility ID's starting with that character or letter.

We want to make sure that user is able to select or highlight the inspection and that Up and Down arrow keys works. We will do this by going up and down through Inspection ID list.

We will test OK button that should take us back to Choosing Inspection Window with the Inspection ID number in the box. We will do this by selecting the ID and then clicking on the OK button.



We will test Cancel button to make sure that when clicked it will cancel all the work done or will delete all the options selected. And will take us to main VB Window.

We will test view button which will able us to view the Inspection ID record. We will test this button by selecting ID and than clicking the View button that will present us with data. We will also test it the button by not selecting any Inspection ID and clicking View button. This will not bring us to any other window since no selection is made.

Creating Facility

This window is utilized to create a facility entry in the database if facility does not exist. User will have to simply complete from with text boxes and few drop down menus.

We will have the function that will check if the boxes are filled when we hit the save button after the completion of the form. This will not apply to the boxes that are in Mailing Information section or Owner Information section. If any other box from remaining section other than Fax Number and EPA ID is left blank error message will be shown and user will be told to fill the box to proceed ahead. We will test this by leaving some of the boxes empty that needs to be filled. We will leave EPA ID and Fax Number boxes empty. This should not give us any error messages.

Facility ID will box will contain the ID assigned to new facility. We will make sure that the box accepts all the characters and number in ID. We will test to make sure that the record is updated or added in the database. Also we will test to make sure that this box is not allowed to leave blank.

We will test Explode button (button with bomb), this should bring up the window that will allow us to search for the Facility ID.

We will also test search button (One with folder), which should bring us to historical data window. We will test it by clicking on it and checking if it takes us to historical data window.

EPA ID will box will contain the EPA ID assigned to new facility. We will make sure that the box accepts all the characters and number in ID. We will test to make sure that the record is updated or added in the database. We will also make sure that the record is saved even when this box is left blank. We will do so by keep the box empty while clicking save.

Facility Type box will contain facility type. We will make sure that user is able to put in data in the box and that the data is correctly inserted in the database. We test by filling in box with some numbers and clicking save after all the data is filled. We will also leave the box empty and when we click save it will display error message saying the box is empty.



Name box will contain the name of facility. We will allow name to be as long as it want to be. We will test several names to test this box and will make sure that data is updates in the database. We also want to make sure that this box is not allowed to be left blank. This can be done by leaving the box blank while clicking the save button which will give us error message.

Address box will contain the address of the facility. We will allow address to be as long as it want to be. We will use several different addresses to test this box and will make sure that data is updates in the database. We also want to make sure that the box is not allowed to be left blank. Test can be done by leaving the box blank while clicking the save button which will give us error message.

Contact box will contain the name of contact person at the facility. We will allow name to be as long as it want to be. We will use several different names to test this box and will make sure that data is updates in the database. We also want to make sure that the box is not allowed to be left blank. Test can be done by leaving the box blank while clicking the save button which will give us error message.

City box will contain the name of the city that facility is at. We will allow city name to be as long as it want to be. We will use several different names to test this box and will make sure that data is updates in the database. We also want to make sure that the box is not allowed to be left blank. Test can be done by leaving the box blank while clicking the save button which will give us error message.

County box will contain the name of the county that facility is at. We will allow the name to be as long as it want to be. We will use several different names to test this box and will make sure that data is updates in the database. We also want to make sure that the box is not allowed to be left blank. Test can be done by leaving the box blank while clicking the save button which will give us error message.

Zip Code box will allow only five letter and no characters. If letters typed are less than five letters error will be shown. Error will also be shown if the character is typed in. We will test this by typing in number that contains five letters. We will also insert incorrect data that contains characters and less than five letters. This will give us an error message.

Fax number box will only contain numbers and no characters. If the character is typed in it will show error message. We will test the box by typing in correct and incorrect data. Incorrect data will give us error. Correct data should be stored in database.

Phone Number box will contain the phone number of the facility. We will allow the number to be 7 or 10 digits long. We will use several different numbers to test this box and will make sure that data is updates in the database. We also want to make sure that the box is not allowed to be left blank. Test can be done by leaving the box blank while clicking the save button which will give us error message. We will use numbers that are

more or less than seven digits but not more than or equal to 10 this will show us error message and will no allow us to proceed forward.

We have similar boxes containing similar information in Mailing Information section and Owner Information section. We will test all the boxes in same way that we selected the boxes in test done above. None of the information in these boxes are mandatory so we can leave them blank and no error message will be shown because of these. We will test this by leaving each of the boxes blank during testing. This will not show any error messages.

IN Comments box user can insert unlimited lines of comments. We will test to see if comment is correctly inserted into the form and also in database. We will test it by filling it with data and then by clicking save. This should work without any problems.

Save button will save the data in database. We will test the button to see if the data is actually saved and that database is updated.

We will check the help button to see if the help menu is made available to the user when help button is selected. The menu should popup without any problems when help button is selected.

At the end will test the save button to see if all the data is correctly recorded and the generated letter is saved. This should work without any troubles.

White Box Testing:

For the white box testing on this product we will run the scripts through a script debuggers and watch the values that are coming in and going out.

We will have product run through and we will be looking for and at the break points

We will also be watching the variable values and changes in them as scripts run through sub-routines variable values.

Hand Held PC Integration:

We will be testing to make sure that all the checklists are present in Hand Held PC. For this we will open each and every one of the checklist and will carefully look at content of the form

We will have to make sure that no data is lost or manipulated during upload or download of the data from or to the PC from the Hand Held PC. To do this we will fill out the forms with the predetermined data and will import and export them. After import and export



also know as upload and download is complete we will go through the data and will make sure there are no changes in the data.

We will have to make sure that user is able to download new checklist to the Hand Held PC. To do so we will have to let user try to download the checklist and will have to look at content of the checklist to make sure that no information have been lost or that the list is not corrupted during the transfer.

We will have to make sure that data is correctly stored during the inspection. To test this we will fill out the list and will make sure that data that was filled in during the inspection process is stored in the PC.

6.2 Performance Bounds

We have setup certain bounds or criteria for our software so that by following that criteria we will be able to maintain quality and user friendliness of the software.

Response time of search function

Best Case Scenario -> Immediate

Worst Case Scenario -> 3 seconds

Response time of browse function

Next List of Records in 0.1 seconds

Import export to WMITS Mobile

Done within 0.5 second

Logon

User should be able to log on within 0.1 second

6.3 Identification of Critical Components

1. WMITS Mobile

We want to make sure that WMITS Mobile works perfectly. If any of the WMITS Mobile components don't work the entire section of the software that is focused on removing the need of paper form for inspection will be failure. This will be lose of big chunk of the research money and time. We want to make sure that WMITS Mobile Unit is fully functional and it removes the need of repeated work at once.

2. Letter Generator

We have to make sure that the letter is generated in correct format and without any corruption of data. If we are not careful letter with wrong data than once we selected could be generated because of mix-up during transfer of data.

3. Module User Access



We also need to make sure that Module User Access works correctly. The way software is designed we have allowed permissions to only certain users for some section of the software. We want to make sure that manager has access to all of the sections of the software, but at the same time we have to make sure that inspectors don't have access to some of the sections. We also need to make sure that one inspector cannot change the inspection report of another inspector.



7.0 Appendices

Palm Pilots/ Personal digital assistants (PDA)

Memory Size

- Need to have enough memory for multiple abbreviated checklists on them and enough memory for multiple inspections to be recorded in the same day. (If necessary we could abbreviate the checklists – for ex: 1) waste characterization; 2) EPA ID, 3) Manifests).

Palm has the least memory, but it runs very limited applications compare to CE. Which require more memory to run additional applications. In other words, memory size is one of a factor (not the only one) for the performance speed of the machine.

Rechargeable batteries

- Is it possible to recharge these in a car?
I can assume that is not included in the package, but additional utilities can be purchase separately.
- How long do the batteries last? (Could recharge at the office instead of the car if they lasted 8-10 hours)

Regular batteries last about 2 weeks (in a rate of using the machine 2 hr/day).

- No memories on the batteries (Where they have to be fully discharged prior to recharge.)

Yes, Palm III does. So I can assume that the rest of the Palm series do that as well.

Palm OS versus Windows CE

Which would be better for our application?

Read an advertisement for *Documents to Go*, which would create a file from various Word or other documents into a file for the Palm device. It can be updated by HotSyncing. www.dataviz.com/docsrg, info@dataviz.com.

Screen Size

Maximize the screen size that is available.

HotSyncing

Would like something that is easy to connect. Prefer one that is fast at downloading information both to and from the Palm Pilot. I've heard the Visor at www.handspring.com is faster than the Palm Series in this respect.

See <http://www.pdamd.com/install/palm/installpalm3.shtml> for more information.

Voice Recognition/cursive writing

Are their programs that recognize cursive writing? (I did not see any).

Voice Recognition would be a good feature.

Options we do not currently view as a necessity:



- Wireless connection to the Internet.
- Separate keyboard (it looks like all organizers have the option to either write or use an on-screen keyboard).
- Color monitor (heard it uses batteries faster)

Recommendation to stay away from Palm V because Lynne's son said there was problems with it.

Palm III: http://www.dataviz.com/products/documentstogo/jump/dxtg_jump.html

"Document to go"? – No. There is no word & excel are involved in the process, unless it can take care of Access.

Visor: <http://www.handspring.com/products/vdetails.asp>

"Faster download"? – Not noticeable with a small size of document download/uploads. Further more, according to the research I have done, Palm series takes lot less time on processing time. So it will actually save time by using Palm series overall.

Window CE: <http://www.cewindows.net/>

Handheld: <http://www.handheldmed.com>
<http://www.pdamd.com/tutorials.shtml> << **MUST READ**

Palm OS vs. Windows CE

Windows CE: <http://www.microsoft.com/windowsce/default.asp>

ZDnet CE download: <http://www.zdnet.com/downloads/ce/>

Palm size PC: <http://www.microsoft.com/windowsce/products/ppc/hwfeatures.asp>

- ✓ Instant-on and off
- ✓ Some feature black and white screens, while others now offer color
- ✓ RAM 4-16MB
- ✓ Screen size 240x320
- ✓ Microphone
- ✓ Industry standard infrared port
- ✓ AC adapter

To connect up with your corporate network and your PC, you can choose from many different types of connectivity:

- ✓ Serial and infrared wireless connections
- ✓ Compact Flash Type II, PCMCIA card support
- ✓ Ethernet LAN support
- ✓ Packet Data communications



- ✓ Wireless or cable-based corporate network (via appropriate Ethernet LAN PC card)
- ✓ Parallel ports
- ✓ Universal Serial Bus for adding peripherals
- ✓ E-mail (IMAP4 and POP4 protocols)
- ✓ Serial line interface protocols (SLIP) connectivity for wireless Cellular Digital Packet data

Handwriting Input

There are two ways to input handwritten information with your Palm-size PC, through the Note Taker feature. You can hand-write information and notes, and have CIC Jot character recognition software convert it to typewritten text for you. Or, you can choose to enter handwritten letters and freehand drawings. Sometimes the best, most useful communication is a quickly drawn picture or a graph-- or your handwritten comments.

Free Programs and Utilities

DiddleBug: alarms
Cipher: security, encrypted text
iSilo reader: converts HTML, text, and Palm DOC file to iSilo format.
LookDA: input text