

User interface design mcs: pattern library

prepared by: nahyan-bin-khalid

Mineral Cadastre System

2014

Table of Contents

1. [Why Design Patterns 4](#_Toc387749239)
2. [User Interface Design Patterns 5](#_Toc387749240)

[2.1. Interaction Models 5](#_Toc387749243)

[2.1.1. Author 5](#_Toc387749244)

[2.1.2. Problem 5](#_Toc387749245)

[2.1.3. Solution 5](#_Toc387749246)

[2.1.4. Example 6](#_Toc387749247)

[2.1.5. Rationale 7](#_Toc387749248)

[2.1.6. Special Cases 7](#_Toc387749249)

[2.1.7. Supporting Research 7](#_Toc387749250)

[2.1.8. Related Standards 7](#_Toc387749251)

[2.1.9. Importance of Adherence 7](#_Toc387749252)

[2.2. Navigation / Orientation / Workflow 7](#_Toc387749253)

[2.2.1. Author 7](#_Toc387749254)

[2.2.2. Problem 8](#_Toc387749255)

[2.2.3. Solution 8](#_Toc387749256)

[2.2.4. Example 8](#_Toc387749257)

[2.2.5. Rationale 8](#_Toc387749258)

[2.2.6. Special Cases 8](#_Toc387749259)

[2.2.7. Supporting Research 8](#_Toc387749260)

[2.2.8. Related Standards 8](#_Toc387749261)

[2.2.9. Importance of Adherence 9](#_Toc387749262)

[2.3. Font Styles / Colours 9](#_Toc387749263)

[2.3.1. Author 9](#_Toc387749264)

[2.3.2. Problem 9](#_Toc387749265)

[2.3.3. Solution 9](#_Toc387749266)

[2.3.4. Rationale 10](#_Toc387749267)

[2.3.5. Special Cases 10](#_Toc387749268)

[2.3.6. Supporting Research 10](#_Toc387749269)

[2.3.7. Importance of Adherence 10](#_Toc387749270)

Table of Figures

[Figure 1: Dashboard 6](#_Toc387749284)

[Figure 2: Application Management – Showing list of applications with status 6](#_Toc387749285)

[Figure 3: New Application Form 6](#_Toc387749286)

[Figure 4: Map View 7](#_Toc387749287)

# Why Design Patterns

A design pattern describes a typical interaction problem, the solution, any rationale behind that solution, related patterns and other relevant details, such as the results of usability testing, reference case studies etc. that support the chosen solution. They describe desired behaviour - such as “Here is the optimal design that allows users to login” not just “Make sure labels are above the fields”.

The benefits of using a design pattern library over a style guide or templates are:

* The problem, solution (with examples) and rationales for common design solutions are captured in each pattern
* Alternates and special cases can be added where needed rather than having to break the pattern library.
* Optimal design elements can be mixed on any page, rather than having to specify separate page templates for every possible combination.

By re-using patterns, OpenM4S’s development team don’t have to re-invent the wheel each time new functionality is needed and while different design solutions may be equally appropriate for a given problem, re-using the one agreed solution means the users of the application only have to get used to one way of doing things to achieve their goals.

# User Interface Design Patterns



The MCS User Interface Design Patterns include

1. Interaction Models
2. Navigation / Orientation / Workflows
3. Font type / Sizes / Colours
4. Field Label Placement
5. Choice of Controls
6. Grouping Controls
7. Control Layout and Spacing Rules
8. Command Buttons
9. Tables
10. Error handling / Mandatory fields / Layered help
11. Feedback

## Interaction Models

### Author

Nahyan Bin Khalid Tuesday, 13 May 2014

### Problem

There are six common Interaction models available for most software applications and choosing the correct one early on is vital to the success of the project.

The six options are:

* Organizer/Workspace (used in Outlook, iTunes, Windows Explorer etc.)
* Hub & Spoke (best suited for PDAs, Phones & ATMs)
* Parallel Workspaces (Most websites or Online Applications)
* Wizards (best for complex tasks or infrequent users)
* Multiple Document Interfaces (Adobe Photoshop and other rich authoring tools)
* Single Document Interfaces (MS Word etc.)

Metro Interface (Windows 8 etc.)

### Solution

From analyzing your platform, toolset and target users, choose the appropriate Interaction model.

* In the case of the MCS this should be option 7 – a Metro Interface.
* Use affinity diagramming or other techniques to divide up planned functionality into manageable high level groups (Dashboard, Application, Property etc.)
* Provide access to these using grouped command buttons in blocks.

### Example

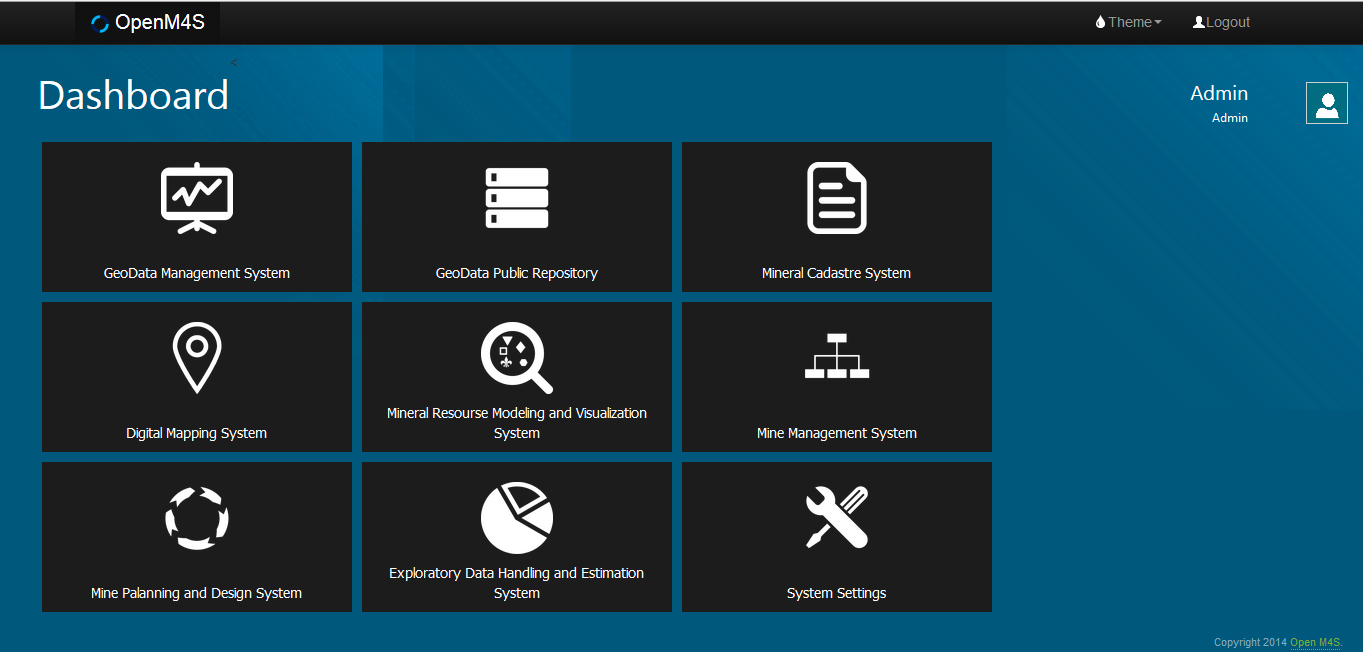


Figure 1: Dashboard

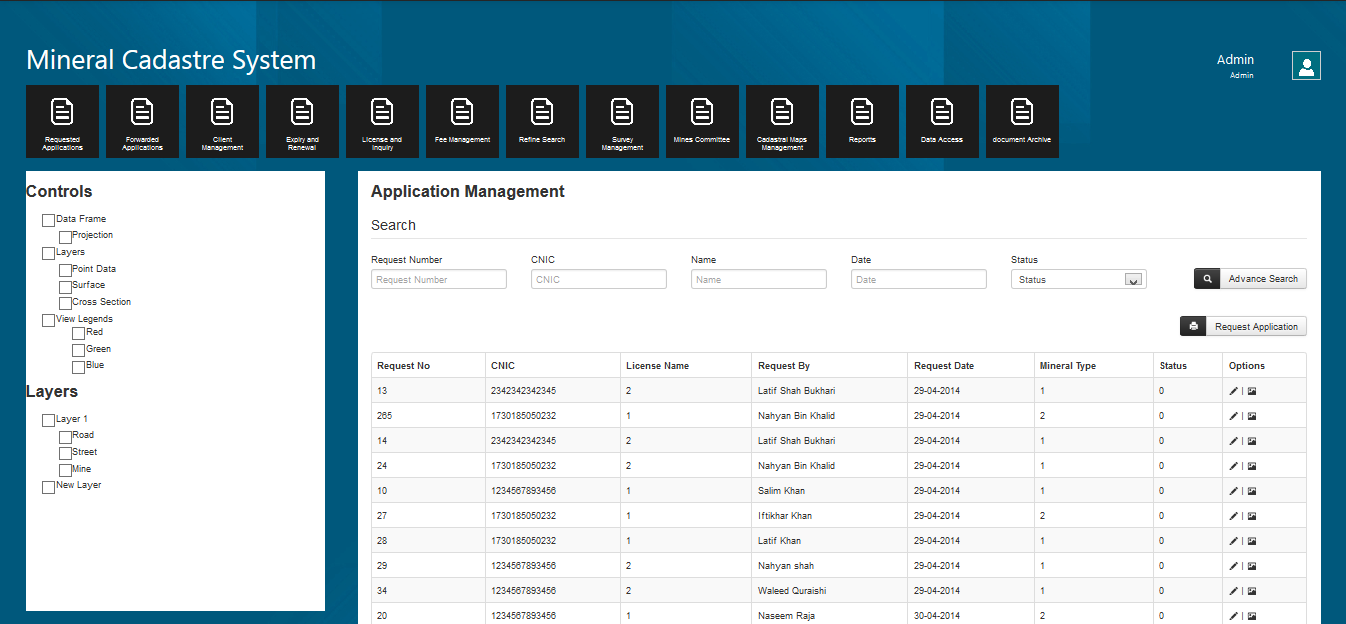


Figure 2: Application Management – Showing list of applications with status

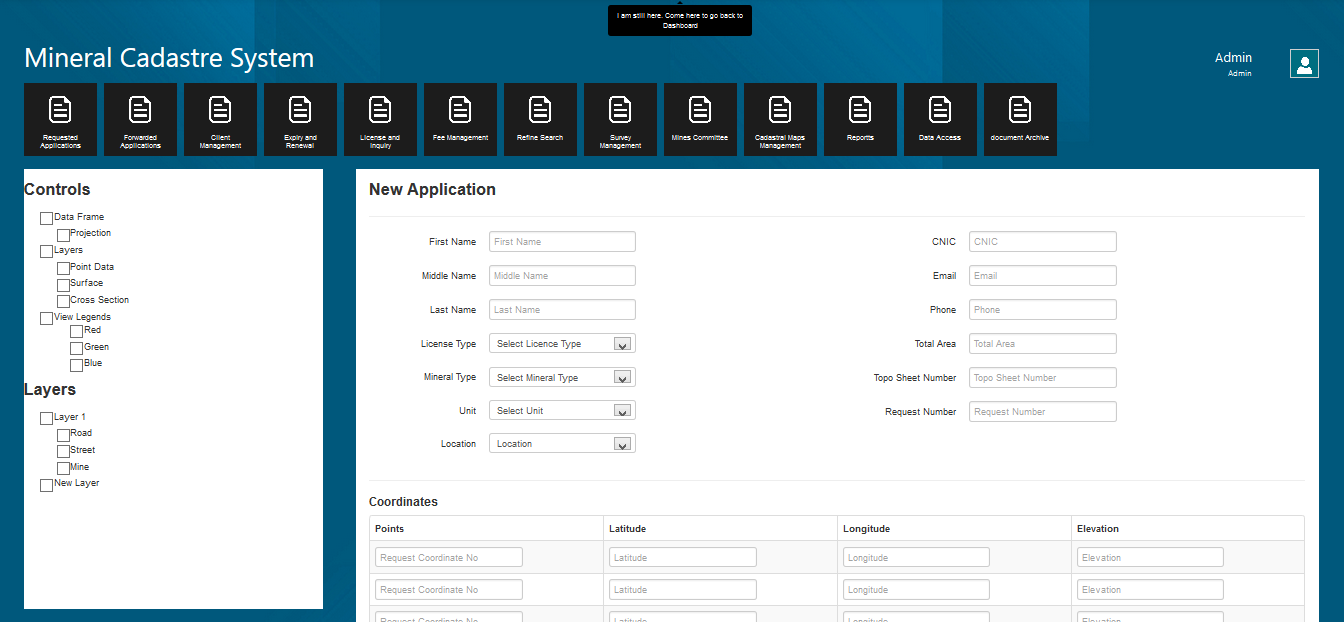


Figure 3: New Application Form

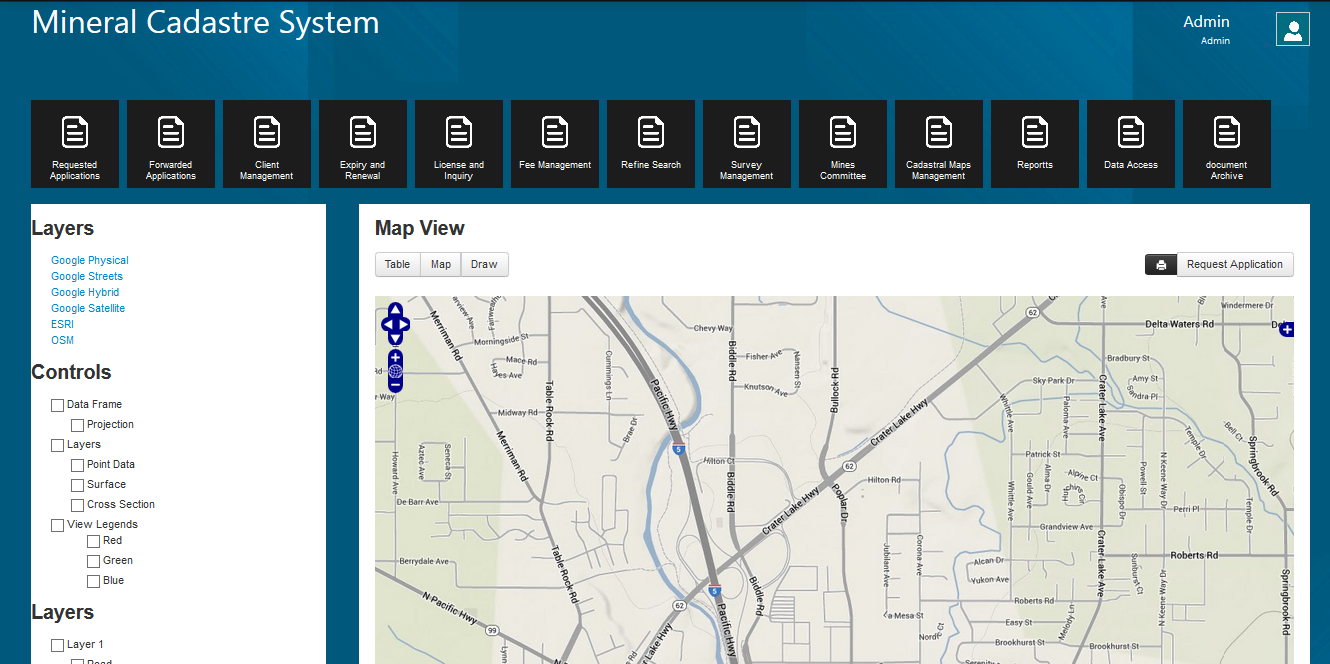


Figure 4: Map View

### Rationale

This application is following Metro Style User Interface. The benefit of using this modern interface is that it can be easily use on touch screens like tablets, smart phones along with conventional Desktop environment. This also enables us to deploy more information about Module in less space. It also prevents scrolling down more frequently. Each Module activates as a separate Application. The major pros is that now days most of the people are used to Windows 8 environment so they will have a look and feel of their native desktop.

### Special Cases

None identified.

### Supporting Research

* “The newest way to refer to Metro” http://www.zdnet.com/microsoft-design-language-the-newest-official-way-to-refer-to-metro-7000006526/
* “Why Metro now rules at Microsoft” http://www.cnet.com/news/why-metro-now-rules-at-microsoft/

### Related Standards

Navigation / Orientation / Workflow Pattern

### Importance of Adherence

High

## Navigation / Orientation / Workflow

### Author

Nahyan Bin Khalid Tuesday, 13 May 2014

### Problem

People with different levels of experience in both technology and MCS business domain need to identify where they can start and how to complete a workflow in order to achieve their business goals.

These workflows must be recognizable and need to provide navigation (how can I get there) and orientation (where am I) to achieve these.

### Solution

* Use affinity diagramming or other techniques to place information or form controls into logical groups
* The high level workflows (or major areas of functionality ) could be represented by an initial set of horizontal tabs or buttons on a toolbar
* On major pages use tabs to keep related groups of functions together
* Use dynamic panels or expanders to display extra functionality rather than presenting these in modal dialogs
* If appropriate, provide “Back” and “Next” command buttons on either side of “Save” to support a wizard style workflow

### Example

{Place here, a screen shot from MCS interface and explain why it is designed like this}

This is one way the Property screen could look, example shows “Rights” tab selected, “Lease” type displayed as an integrated panel rather than a modal dialog. Other types would be displayed dynamically in the same space.

### Rationale

* Tiles as links to modules as de-facto method used for metro UI.
* Tiles are a well understood pattern of separating groups of related functionality.

Keeping users on the same screen by not miss-using modal dialogs but presenting extra functionality by either expanders or dynamic panels reduces training and cognitive workload.

### Special Cases

None identified.

### Supporting Research

* Four reasons why I love tabs - page 79 “Don’t make me think” by Steve Krug
* Present Tabs Effectively – page 61 “Research-Based Web Design Guidelines” Koyani Baily Nall et al.
* Improving Navigation – page 148 “About Face 2.0” by Alan Cooper
* Selection dependent inputs – Dropdown list page 184 “Web Form Design” by Luke Wroblewski
* Additional inputs – best Practice – page 173 “Web Form Design” by Luke Wroblewski

### Related Standards

Navigation / Orientation / Workflow Pattern

### Importance of Adherence

High

## Font Styles / Colours

### Author

Nahyan Bin Khalid Tuesday, 13 May 2014

### Problem

Provide users with a visual interface that supports their goals.

### Solution

#### Body Text / Field Labels

Font: Segoe UI, Century Gothic, Courier New

Size: 14pt

Weight: Normal

Colour: Black

#### Command Button Labels

Font: Segoe UI, Century Gothic, Courier New

Size: 14pt

Weight: Normal

Colour: Black

Height: 20 pixels recommended (may depend on operating system)

#### Combo boxes / Input fields

Font: Segoe UI, Century Gothic, Courier New

Size: 14pt

Weight: Normal

Colour: Black

Height: 20 pixels recommended (may depend on operating system)

#### Section Dividers

Font: Segoe UI, Century Gothic, Courier New

Size: 14pt

Weight: Normal

Colour: Black

Height: 20 pixels recommended (may depend on operating system)

#### Sub Section Divider

Font: Segoe UI, Century Gothic, Courier New

Size: 14pt

Weight: Normal

Colour: Black

Height: 20 pixels recommended (may depend on operating system)

#### Footer

Font: Segoe UI, Century Gothic, Courier New

Size: 8pt

Weight: Normal

Colour: White

Height: 20 pixels recommended (may depend on operating system)

#### Feedback dialog

Font: Segoe UI, Century Gothic, Courier New

Size: 14pt

Weight: Normal

Colour: Black

Background: #faf4a3

Height: 20 pixels recommended (may depend on operating system)

### Rationale

* As screen sizes and resolutions increase each year, application’s default text sizes need to increase – currently 14pt is recommended
* Keep font sizes / weights to a minimum to reduce visual “noise”
* Use fonts specifically designed for screen display such as Tahoma, Verdana or Sans Serif.

### Special Cases

None identified.

### Supporting Research

* Text Appearance – page 95 “Research-Based Web Design Guidelines” Koyani Baily Nall et all.
* Chapter 19 : Designing Look and Feel – page 225 to 246 “About Face 2.0” by Allen Cooper

### Importance of Adherence

Medium to High