Chapter Six: System Testing and Implementation

6.1 Introduction
6.2 System Testing
   6.2.1 Unit Testing
   6.2.2 Integration Testing
   6.2.3 User Acceptance Testing
      6.2.3.1 User Testing and Analysis of the Questionnaires
6.3 System Implementation
   6.3.1 System Development
      6.3.1.1 Coding
   6.3.2 System Security
   6.3.3 CW-MSLD System Implementation
6.4 Summary
6.1 Introduction

This chapter illustrates the system testing and implementation phases. The testing phase involves some modification to the previous design phase and system testing has been done to minimize the programming and system error. At the implementation phase, system requirements such as hardware and software will be defined. Besides that, the system prototype interfaces and functionalities (module) will be fully demonstrated to users.

6.2 System Testing

Testing the system is a very important stage to ensure that all system requirements have been developed without errors. System testing can be done through some stages. The first stage is called unit testing or component testing and this testing done during the development of the system. Each component, script or module test isolates from other component or unit by checking the input and output for it.

The second stage is called integration testing. The integration between components will be tested and in case there are any errors the components will be tested again. The third stage is called user acceptance testing and this testing done by users who request to develop the system. The third stage is called security testing. The final stage is called user acceptance testing and this testing done by users who request to develop the system.

6.2.1 Unit Testing

Unit testing focuses on testing module, script or component that has been designed by PHP, JavaScript, or Rosa Applet. For example, the developer tested the map tools button functionality such as Zoom in on a map or obtain information when clicking on the map by using Identify button that is designed by using Rosa Applet.
6.2.2 Integration Testing

After the unit testing has been done with satisfaction for each component or script, the integration testing is started to ensure the CW-MSLD System components worked together smoothly. The functional and non-functional requirements were tested in this stage. One example for integration testing is to search the parcel model by entering the parcel ID and if the GIS database has the parcel requested, the system will display it and it can use the data given to register new real estate.

6.2.3 User Acceptance Testing

User acceptance testing is the final stage of testing before CW-MSLD System begins to be implemented by the user. The potential users evaluate the system to reveal the errors and omissions of system requirements that were defined in early stage developing the system. A total of 10 questionnaires were given out in Tripoli Cadastral Office to evaluate all the functions available in the system. The questionnaires were distributed to the staffs in early March 2007 and collected after test the system. The completed questionnaires revealed valuable information that enabled to evaluate the functional and nonfunctional requirements for the CW-MSLD system. User Acceptance Questionnaire is appended in appendix (VII).

6.2.4.1 User Testing and Analysis of the Questionnaires

The user guide was submitted to the users with the evaluation questionnaires after the users had used the system. The system was tested by six users and analyses of the questionnaires were done.

1) Analysis of User Interface Evaluation

Working with user interface any system is dependent on users’ computer background and understanding of the system environment. Based on the evaluation, the system was found to be easy to use. The highest rating mean of 4.2 indicates that searching on the map to get
information is easy. The results were converted into a bar chart in Figure 6.1 to show more clearly.

![User Interface Evaluation](image)

**Figure 6.1:** User interface evaluation bar chart

The bar chart in Figure 6.2 shows the evaluation for user interface satisfaction. The bar chart clearly indicates that the users are satisfied by using the help tools.

![User Interface Satisfaction](image)

**Figure 6.2:** User interface satisfaction bar chart

**ii) Analysis of Evaluation Pertaining to Features**

Figure 6.3 shows the evaluation of testing the accuracy of geographic data (map) as the data accuracy is the most important part of a successful GIS application. The bar chart indicates a good frequency for testing the accuracy for the parcel’s area, boundary, and location.
6.3 System Implementation

The results obtained from analysis of available features on the system is depicted in Table 6.1

Table 6.1: Available features on the CW-MSLD System

<table>
<thead>
<tr>
<th>No</th>
<th>Statement</th>
<th>Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Administrator able to add new user to login to the system</td>
<td>√</td>
</tr>
<tr>
<td>2</td>
<td>Access control for authorized user to login to the system</td>
<td>√</td>
</tr>
<tr>
<td>3</td>
<td>Administrator able to monitor the login users to system</td>
<td>√</td>
</tr>
<tr>
<td>4</td>
<td>Working with real coordinates map</td>
<td>√</td>
</tr>
<tr>
<td>5</td>
<td>Use the map to get requested parcel</td>
<td>√</td>
</tr>
<tr>
<td>6</td>
<td>The system able to register the parcel not included on the map</td>
<td>√</td>
</tr>
<tr>
<td>7</td>
<td>Updating certificate information before issuing certificate</td>
<td>√</td>
</tr>
<tr>
<td>8</td>
<td>The system able to register a whole cadastral (parcel/building)</td>
<td>√</td>
</tr>
<tr>
<td>9</td>
<td>The system able to register subdivides cadastral (parcel/building)</td>
<td>√</td>
</tr>
<tr>
<td>10</td>
<td>The system able to transfer ownership for registered real estate</td>
<td>√</td>
</tr>
</tbody>
</table>

6.3.1 System Development

This section describes the programming language and tools that have been used to develop CW-MSLD System. The system developed as web mapping page makes use of the map...
server and PHP as the core programming language techniques. Besides that, the JavaScript and Rosa Applet have been used to help the system interfaces become more interesting and easy to use. For example the Rosa Applet tools used image button tools that help the user interact with the map in easy ways such as ZOOM IN, ZOOM OUT … etc.

The RDBMS used to build attribute data for this system is MySQL and for the map data (geographic data) are stored in three types of files that relate with each other to produce the vector map such as parcel layer. The merge between the attribute data and map data is done by using map server techniques.

### 6.3.1.1 Coding

Coding is the process of turning design of the web mapping for cadastral information system into specific instructions that the computer system can understand and execute (Gary et al., 2001). The initial unit testing and integration testing for a functional prototype model was done after having written the codes. As mention in Section 6.2.1, the system has been written using PHP script, MapServer, JavaScript, Rosa Applet and MySQL database management.

In this section, the author will present some parts from a mapfile. The mapfile defines a collection of mapping objects that together determine the appearance and behavior of the map as displayed in the web browser (Kropla, 2005).

In the code snippet below, the first line sets the name of the map to cis; the keywords UNITS define map unit where the map size that is displayed is defined in pixels by using keywords SIZE. Similarly, the keyword EXTENT sets the extent for the whole map (Xmin,
Ymin, Xmax, Ymax). The IMAGECOLOR keyword line 5 defines the background for the map image and the image format defines using IMGETYPE keywords line 6. The keyword SHAPEPATH line 7 tells MapServer where to find the shapefiles data.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Name “cis”</td>
</tr>
<tr>
<td>2.</td>
<td>UNITS METERS</td>
</tr>
<tr>
<td>3.</td>
<td>EXTENT 217400.00 3639200.00 218000.00 3639700.00</td>
</tr>
<tr>
<td>4.</td>
<td>SIZE 400 300</td>
</tr>
<tr>
<td>5.</td>
<td>IMAGECOLOR 255 255 255</td>
</tr>
<tr>
<td>6.</td>
<td>IMAGETYPE GIF</td>
</tr>
<tr>
<td>7.</td>
<td>SHAPEPATH &quot;/ms4w/var/www/htdocs/data/&quot;</td>
</tr>
<tr>
<td>8.</td>
<td>FONTSET &quot;/ms4w/var/www/htdocs/etc/fontset.txt&quot;</td>
</tr>
</tbody>
</table>

In the code snippet below, the lines from 94 to 99 define query map object which highlights the special results on the map image when querying special database so that the user is able to see the query result highlighted by using different colours. As seen on line 98, the query map is highlighted using yellow color.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>94.</td>
<td># Query Map</td>
</tr>
<tr>
<td>95.</td>
<td>QUERYMAP</td>
</tr>
<tr>
<td>96.</td>
<td>STATUS on # draw query map</td>
</tr>
<tr>
<td>97.</td>
<td>STYLE HILITE # highlight selected feature</td>
</tr>
<tr>
<td>98.</td>
<td>COLOR 255 255 0 # in yellow</td>
</tr>
<tr>
<td>99.</td>
<td>END</td>
</tr>
</tbody>
</table>

In the next code snippet, in lines 103 through 131 define the parcel layer which starts with Layer keyword at line 103 and finished with END keyword at line 131; the NAME keyword at line 103 sets parcel’s name that is used in MapServer; the METADATA defined in lines 105 through 108 allows the data to be stored as name value accessible by template tags file in line 129.

A polygon layer name parcel line 103 is specified in lines 109 through line 111. It retrieves its spatial data form a shapefile named parcel. The “PARCEL_ID” item in data set will be used to label the feature. The label properties are defined in lines 119 though 128 under
CLASS object. The layer style is defined using STYLE object in lines 155 through 188 which sets the filling colour and outline colour for parcel layer.

```
103. LAYER
104. NAME "parcel"
105. METADATA
106. "DESCRIPTION" "Cadastral"
107. "RESULT_FIELDS" "PARCEL_ID AREA PERIMETER"
108. END

109. STATUS on
110. TYPE polygon
111. DATA "parcel"

112. LABELCACHE on
113. LABELITEM "PARCEL_ID"
114.
115. CLASS
116. STYLE
117. COLOR 238 232 170
118. OUTLINECOLOR 222 184 135
119. END # end style

120. LABEL
121. TYPE truetype # use truetype font
122. FONT "arial" # use arial bold
123. SIZE 6 # use 6 point size
124. COLOR 0 0 0 # color text black
125. BACKGROUNDCOLOR 255 255 255 # render text on white background
126. MINDISTANCE 50 # labels > 50 pixels apart
127. POSITION cc # labels in center of feature
128. ANTIALIAS true # antialias the text
129. END # end label

130. TEMPLATE "/ms4w/var/www/htdocs/templates/parcel.html"
131. END # class
132. END #Layer
```

### 6.3.2 System Security

The system security is very essential to this system especially when the system is running on the internet environment network. Users are grouped into three groups. At the highest level is the system administrator and second level or group is the manager and the last group members are the registered staff users. Valid user ID and password are required whenever a user access the system. This is to prevent unauthorized users from using the system. The specific models for each group are explained in the user guide appendix on Page 106.
6.3.3 CW-MSLD System Implementation

This section presents the system interfaces for each module and the functionality for it. The system actually has three types of users which are manager, staff and administrator. Figure 6.4 shows the system modules tree based on user’s type.

To see the implementation function for each model, please refer to *appendix V: CW-MSLD System User Guide*. 

---

Figure 6.4: System Modules Tree
6.4 Summary

In summary, this chapter investigated the system testing and implementation. It presents the basic types of testing suitable for this system. The system testing started with test unit, followed by testing component or module. Next, integration testing between units or a component was done and finally the system test by the potential users to evaluate the system acceptance was carried out. The end phase in the system development is implementation where the user guide for the modules is shown step by step.