GIS LEARNING TOOL FOR USA'S TALLEST SKYSCRAPERS AND THEIR CONSTRUCTION

A Thesis
Presented to the
Faculty of
San Diego State University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Kanaka Nethra Rajah Nagarajasetty
Fall 2014
SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the
Thesis of Kanaka Nethra Rajah Nagarajasetty:
GIS Learning Tool for USA’s Tallest Skyscrapers and Their Construction

Carl Eckberg, Chair
Department of Computer Science

William Root
Department of Computer Science

Ziad M. Bayasi
Department of Civil, Construction, and Environmental Engineering

Oct 30, 2014
Approval Date
Copyright © 2014
by
Kanaka Nethra Rajah Nagarajasetty
All Rights Reserved
DEDICATION

I dedicate my dissertation work to my beloved husband, friend, Santhosh Saibabu Karupakula for bearing with my mood swings. This work would not have materialized without your help and support. Thank you for giving me strength to reach for the stars and chase my dreams.

I also dedicate this dissertation to my family and friends. A special feeling of gratitude to my loving parents, Anuradha and Nagaraja Setty for their words of wisdom, endless love, support and encouragement and to my brother Nithin and his wife Sujitha for always being there for me.

I dedicate this work to my extended family Geetha and Saibabu Karupakala, Rashmi and NagaHarsha, Vrushank who deserve my wholehearted thanks as well.
In any city with lots of skyscrapers, lots of skyline, the moon seems bigger than it is. It’s called the Moon Illusion.

-- Neil Degrasse Tyson
ABSTRACT OF THE THESIS

GIS Learning Tool for USA’s Tallest Skyscrapers and Their Construction
by
Kanaja Nethra Rajah Nagarajasetty
Master of Science in Computer Science
San Diego State University, 2014

Urban development in the twenty-first century takes many forms, but for many none quite so interesting as the skyscraper. With swelling cities and growing concerns about the environment, vertical living has become the preferred way of life for millions of people around the world. But just how these tall buildings are designed, constructed and operated remains a mystery to many—even to those who live in them.

The motivation behind this application is to build an interactive and one-stop Geographic Information systems (GIS) learning tool that will help users learn about structural facts and geography of tallest skyscrapers around the metro cities of USA. For purpose of this application development, any building more than 700ft (213m) is considered as one of the tallest skyscrapers. The points displayed on USA map are the metro cities hosting these skyscrapers. When users click on cities, a brief description about the city along with a link to the top three skyscrapers is displayed. The links of the skyscrapers opens a HTML page that has a photo gallery, embedded video, facts, structural information etc., in a web browser.

Map Objects Java Objects (MOJO), a set of Java API’s provided by ESRI, is used to display a map of the United States of America and skyscrapers locations in the form of points. Along with MOJO, other technical languages used to develop this application are HTML5, CSS3, JavaScript and Java Swing.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>x</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>xi</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>xiv</td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2 TECHNOLOGY</td>
<td>4</td>
</tr>
<tr>
<td>2.1 Java</td>
<td>4</td>
</tr>
<tr>
<td>2.2 Map Objects Java Edition</td>
<td>5</td>
</tr>
<tr>
<td>2.3 HTML5, Java Script, CSS</td>
<td>6</td>
</tr>
<tr>
<td>2.4 Eclipse IDE</td>
<td>7</td>
</tr>
<tr>
<td>2.5 Development Platform</td>
<td>7</td>
</tr>
<tr>
<td>3 REQUIREMENTS</td>
<td>8</td>
</tr>
<tr>
<td>3.1 Data Requirements</td>
<td>8</td>
</tr>
<tr>
<td>3.2 Platform Requirements</td>
<td>9</td>
</tr>
<tr>
<td>3.3 Functional Requirements</td>
<td>9</td>
</tr>
<tr>
<td>3.4 User Interface Requirements</td>
<td>10</td>
</tr>
<tr>
<td>4 SOFTWARE DEVELOPMENT METHODOLOGY</td>
<td>11</td>
</tr>
<tr>
<td>5 SOFTWARE DESIGN</td>
<td>13</td>
</tr>
<tr>
<td>5.1 High Level Architecture Diagram</td>
<td>13</td>
</tr>
<tr>
<td>5.2 Classes</td>
<td>14</td>
</tr>
<tr>
<td>6 ECLIPSE AND MAP OBJECTS CONFIGURATION</td>
<td>16</td>
</tr>
<tr>
<td>7 LAUNCHING THE APPLICATION</td>
<td>20</td>
</tr>
<tr>
<td>7.1 Start Screen</td>
<td>20</td>
</tr>
<tr>
<td>7.2 The Map Window</td>
<td>21</td>
</tr>
<tr>
<td>8 MENU BAR</td>
<td>22</td>
</tr>
<tr>
<td>8.1 File</td>
<td>22</td>
</tr>
</tbody>
</table>
8.1.1 Add Layer .................................................................22
8.1.2 Print ..............................................................................23
8.1.3 Remove a Layer ...........................................................23
8.1.4 Legend Editor ..............................................................23
8.2 Theme ............................................................................25
  8.2.1 Open Attribute Table ...................................................25
  8.2.2 Create Layer from Selection .......................................25
8.3 Layer Control .................................................................27
  8.3.1 Promote Selected Layer ..............................................27
  8.3.2 Demote Selected Layer ...............................................28
8.4 By State ...........................................................................28
8.5 Extras .............................................................................29
  8.5.1 Fun with Civil .............................................................29
  8.5.2 WTC Revised Building Code .......................................29
  8.5.3 Demonstrate Historical Images ..................................30
  8.5.4 Mohawk Ironworkers ...............................................30
  8.5.5 Quiz ............................................................................32
8.6 Help ................................................................................33
  8.6.1 Help Topics ................................................................33
  8.6.2 Help Tool ..................................................................33
  8.6.3 Contact Us ................................................................33
  8.6.4 About MOJO Tool .....................................................35
9  MAPOBJECTS TOOLBARS ................................................37
  9.1 MOJO Zoom Pan Toolbar ..............................................37
    9.1.1 The Previous Extent Tool .........................................37
    9.1.2 The Next Extent Tool ...............................................38
    9.1.3 The Zoom to Active Layer Tool .................................38
    9.1.4 The Zoom to Full Extent Tool ....................................38
    9.1.5 The Zoom In Tool ....................................................38
    9.1.6 The Zoom Out Tool ..................................................38
    9.1.7 The Pan Layer Tool ..................................................39
    9.1.8 The Pan In One Direction Tool .................................39
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Classes Used for Development of This Application</td>
<td>15</td>
</tr>
<tr>
<td>12.1</td>
<td>Description of Shape File Header</td>
<td>54</td>
</tr>
<tr>
<td>12.2</td>
<td>Value/Shape Type Mapping</td>
<td>55</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 3.1. The requirements................................................................. 8
Figure 4.1. Software development life cycle/prototype model.......................... 12
Figure 5.1. High-level architecture design.................................................. 13
Figure 6.1. Download Eclipse................................................................. 17
Figure 6.2. Download map objects Java edition........................................... 17
Figure 6.3. Creating a new Java project..................................................... 17
Figure 6.4. Creating a new library............................................................. 18
Figure 6.5. Create a new user library called ESRI and click on import.............. 18
Figure 6.6. Browse to the location where map objects was installed and select all JAR files........................................................................... 19
Figure 6.7. Select the Java doc location and click on edit, link it with the java doc location from the installed map object folder............................. 19
Figure 7.1. The application launch screen - starting point of the application........ 20
Figure 7.2. Application's start................................................................... 21
Figure 8.1. When user clicks on add layer it allows the user to add a new shapefile layer to the map................................................................. 22
Figure 8.2. Remove a layer from the map..................................................... 23
Figure 8.3. Changing the legend.................................................................. 24
Figure 8.4. Depicting all states with a different color and labeling all the state names using the legend editor......................................................... 24
Figure 8.5. The attribute table for the “USAStates” layer............................... 25
Figure 8.6. Selecting the select features tool before creating a layer from selection. ....... 26
Figure 8.7. Selecting the points that we want to create a new layer from........... 26
Figure 8.8. Creating the new layer from the selected points. This layer is saved in your shapefiles folder................................................................. 27
Figure 8.9. Demonstrates the "promote selected layer" tool............................ 27
Figure 8.10. Demonstrates the "demote selected layer" tool........................... 28
Figure 8.11. State layer and point layer of selected state................................ 28
Figure 8.12. Demonstrate fun with civil..................................................... 29
Figure 8.13. Demonstrate WTC revised building code which opens a pdf file. ..........................30
Figure 8.14. Demonstrate historical images which opens a 40sec’s movie file. ......................31
Figure 8.15. Mohawk Ironworkers menu item. .................................................................32
Figure 8.16. Demonstrate quiz. Opens an online webpage where user can play quiz
and check his/her knowledge about skyscrapers of USA. ...........................................32
Figure 8.17. Demonstrates the table of contents help tool..................................................33
Figure 8.18. Demonstrates the legend editor help tool......................................................34
Figure 8.19. Demonstrates the layer control help tool......................................................34
Figure 8.20. Demonstrates the help tool. .........................................................................35
Figure 8.21. Demonstrates the contact us tool. .................................................................35
Figure 8.22. Demonstrates the about MOJO tool. ..............................................................36
Figure 9.1. Toolbar. ...........................................................................................................37
Figure 9.2. Zoom/Pan toolbar. ..........................................................................................38
Figure 9.3. Pan in one direction's drop down. .................................................................39
Figure 9.4. Identify tools table. Table displayed when using identify tool. .........................40
Figure 9.5. Selection toolbar. ..........................................................................................40
Figure 9.6. Find tool’s dialog. Dialog displayed when using the find tool with the
input string: “Texas”. The feature found is highlighted on the map.................................41
Figure 9.7. The query builder tool’s dialog is displayed when using the query Builder
tool with the SQL statement, we used the query STATE_NAME='California’
here. The result is highlighted on the map.................................................................41
Figure 9.8. Select features’ drop down list. .....................................................................42
Figure 9.9. Select features tool used to draw a rectangle. A rectangle shape is drawn
on the map using this tool. .........................................................................................42
Figure 9.10. Buffer tool’s dialog. The buffer dialog provides a way to input the buffer
distance based on the currently selected features. .....................................................43
Figure 9.11. Attributes table for the selected layer...........................................................43
Figure 10.1. Customized toolbar.....................................................................................44
Figure 10.2. The add layer dialog. This dialog allows the user to select the new layer
to be added to the current map. ................................................................................45
Figure 10.3. The California layer is added and appears on the TOC...............................45
Figure 10.4. The distance between two points. The selected distance is displayed as a
blue line and the distance appears on the status bar in miles and kilometers...............46
Figure 10.5. Example of CSV file. ..................................................................................47
Figure 10.6. Displays the layer “My Points1” added using the XY feature. ......................47

Figure 10.7. The dialog opens up when the user uses the hotlink tool and clicks on a point on the map. (We clicked on Seattle in this example). .................................48

Figure 10.8. Sample static web page with information about the skyscraper. ..............49

Figure 10.9. Help tip window. Window displayed when the Help Tool is used on the hotlink button. .................................................................51

Figure 11.1. Webpage displaying the top 20 famous skyscrapers of the world. ............52

Figure 12.1. Creating shape file by selecting features. (Theme->create layer from selection). .................................................................56

Figure A.1. Config.properties for Seattle. .................................................................62

Figure A.2. Hot link JDialog for Seattle. .................................................................63

Figure B.1. Application directory structure. .............................................................65
ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my advisor and Thesis chairperson Professor Dr. Carl Eckberg, Dept. of Computer Science, San Diego State University for giving me an opportunity to work on this Thesis project and always supporting me with my work with sufficient guidance. His warm encouragement and constructive comments have been an enormous help to accomplish this Thesis.

I thank Professor William Root, Dept. of Computer Science, San Diego State University for his acceptance and for being extraordinary committee member and for his help and support.

I also am obliged to Professor Zaid Bayasi, Dept. of Civil Engineering, San Diego State University for showing interest in knowing about the tool, its use and being part of my Thesis committee.

I would also like to take this opportunity to thank my mother, Anuradha father, Nagarajasetty and loving husband, Santhosh for all the support in every step of my life.
CHAPTER 1

INTRODUCTION

Whenever I visited metro cities like Chicago, New York City etc. of USA, I look up at the tall skyscrapers and I wonder how people made them SO BIG! Some of them are extremely fabulous, they are all made for lots of people to live in or work in on just a small piece of land. Building these tall buildings is not an easy task; people are so little, how do they make such big buildings? How does a skyscraper stand up? What’s inside of them? Who wants to climb all those stairs? Is sky the limit? All these questions inspired me to know more and research about the structural and technical details of how architects and engineers experiment with new styles and building taller and more innovative skyscrapers.

This application focuses on making learning about USA’s tallest skyscrapers and their construction, an interesting and interactive tool for Civil Engineering students and general users. It aims at creating a place where students/users can acquire knowledge about what is a skyscraper, what are the modern materials used in construction and provides useful background information on each skyscraper, making this a fun primer for anyone interested in architecture. This application is designed as a graphical user interface (GUI), which is map-based and has interactive webpages, embedded videos, fast facts and pictures to make learning more fascinating and fun.

Environmental Science Research Institute (ESRI) has an outstanding presence in the field of Geographic Information Science (GIS). Map Objects is a suite of Java-based developer components for creating client or server-side mapping and GIS applications [1]. The release of Map Objects Java Standard Edition allows a developer to build custom map applications using the open standard Java programming language. Software developers can almost arbitrarily extend the basic map functionality and provide customized capabilities to satisfy the end user. The java API package provides this high degree of customization.

This tool was developed using Map Objects Java Objects (MOJO) technology, which is a product of ESRI. The IDE used for developing this tool is Eclipse 3.5. Map Objects Java Edition is used as it can perform activities as labeling map features, thematic mapping,
panning and zooming through multiple map layers, querying spatial and attribute data, performing geometric operations, measuring distances, displaying real-time geographic data, and much more. It provides features by which geographic-based display can be implemented in designing applications. Map Objects supports such data types as shape files, layers, image formats such as BMP, TIFF, PNG, and JPG. The term MOJO is a non-ESRI term for Map Objects Java Objects, which is more easily remembered. The two great virtues of these technologies are:

- The GIS application can be customized to a very high degree.
- Deployment to a wide variety of platforms is made easy by Java executable jar files.

In addition to a catalog of USA skyscrapers and the world’s top twenty, this thesis had the goal of describing the architectural techniques used to make skyscrapers and their evolution over time. The architects of earlier skyscrapers did not anticipate their attack by King Kong or very large airplanes.

Professor M Ziad Bayasi, of the Civil Engineering department was kind enough to work with me at great length to allow me to including structural details of methods used for building tall and how these techniques evolved.

It is my hope that the resulting application will be of some value to Civil Engineering students, as well people in general geography or urban planning classes. Of course, some people are simply interested in tall building. The Empire State building has had millions of visitors, as have other skyscrapers.

This document describes how this application was put together, from its requirements, to the tools and technologies used to build this application. It is divided into 14 chapters. Chapter 2 discusses the technologies used to build this application. Chapter 3 talks about the requirements, which were gathered to implement this application. These include the functional, data, platform and the user interface requirements. Chapter 4 focuses on the software development methodology used; the way prototyping was used to make this tool a success. Chapter 5 highlights the Software Design used to implement this application; it discusses the classes used and a high level architecture of the application. Chapter 6 describes the configuration and installation steps of important tools and software used to build this application. Chapter 7 gives us the steps for launching this application, what we see
when we start this application, and where to go from there. Chapter 8 defines the Menu bar and all its tools and features. Chapter 9 delves into the toolbars implemented in this tool; these include the Zoom/Pan toolbar, the Selection toolbar and the Custom toolbar. Chapter 10 further explains the custom toolbar, and goes deeper into its tools and features. Chapter 11 explains the world top 20 button and its functionality. Chapter 12 describes about the shape files and how to create them. Chapter 13 gives us some closing points in the form of conclusion and discusses the obstacles faced in implementing this application and finally, Chapter 14 talks about the probable future enhancements and improvements to the exiting application.
CHAPTER 2

TECHNOLOGY

The emphasis of this chapter is on the technologies used in the implementation of this application. There were many options available in terms of technologies to use for this application. I chose a java application with links to web pages using ESRI's Map Objects Java Edition along with Java, and other web technologies. Taking Dr. Carl Eckberg's suggestion it was determined that Map Objects Java Edition and Java are best suited for the implementation of this interactive learning tool for the USA’s tallest skyscrapers and their construction. Java being a platform independent programming language gels well with Map Objects. In the following section we will discuss the technologies used in detail.

2.1 JAVA

Java is a high-level programming language and computing platform first released by Sun Microsystems in 1995. It runs on a variety of platforms such as Windows, Mac OS, and various versions of UNIX [2] and this ability makes it one of the most popular and widely used languages. Java is a concurrent, class-based, simple object-oriented language. Java source code files are compiled into an intermediate code called byte code, which can then be executed by a Java interpreter. Compiled Java code can run on most computers because Java interpreters and runtime environments, known as Java Virtual Machines (JVM). Bytecode can also be converted directly into machine language instructions by a Just In Time (JIT) compiler.

Following are a few advantages of Java, which made it the preferred language for this application.

- **Java is easy to learn:** Java was designed to be easy to use and is therefore easy to write, compile, debug, and learn than other programming languages.
- **Java is object-oriented:** This allows you to create modular programs and reusable code.
- **Java is platform-independent:** One of the most significant advantages of Java is its ability to move easily from one computer system to another. The ability to run the same program on many different systems is crucial to World Wide Web
software, and Java succeeds at this by being platform-independent at both the source and binary levels.

- **Java is distributed**: Java is designed to make distributed computing easy with the networking capability that is inherently integrated into it. Writing network programs in Java is like sending and receiving data to and from a file.

- **Java is secure**: Java considers security as part of its design. The Java language, compiler, interpreter, and runtime environment were each developed with security in mind.

- **Java is robust**: Robust means reliability. Java puts a lot of emphasis on early checking for possible errors, as Java compilers are able to detect many problems that would first show up during execution time in other languages.

- **Java is multithreaded**: Multithreaded is the capability for a program to perform several tasks simultaneously within a program. In Java, multithreaded programming has been smoothly integrated into it, while in other languages, operating system-specific procedures have to be called in order to enable multithreading.

- Java is easily deployed.

Because of Java's robustness, ease of use, cross-platform capabilities and security features, it has become a language of first choice for this thesis project.

### 2.2 Map Objects Java Edition

Map Objects Java Edition, often referred to as MOJO, particularly at San Diego State University, is ESRI's brain child which provides a pure java solution for adding dynamic maps to an application. The MOJO suite comprises of more than 900 Java-based GIS and mapping developer components that can be used to build custom, cross-platform GIS applications or applets. It also includes prebuilt JavaBeans that are easy to use in your preferred IDE's. According to Victoria Kouyoumjian, ESRI’s Map Objects product manager, "Map Objects Java Standard Edition provides an extensive Java-based application programming interface for developers to build client geographic-based display, query, and data retrieval applications."

Significant features of Map Objects Java Standard Edition include the capability of combining multiple data sources (local, Internet, and Intranet) to craft customized maps that provide connectivity with ArcIMS, have a wide range of GIS capabilities, and are compatible with many data sources including shapefile, ArcSDE layers, and a variety of image formats such as BMP, TIFF, PNG, JPEG, and GIF. Developers can also make use of Map Objects
Java Standard Edition to create customized feature layers from their own custom data sources.

Map Objects Java Objects (MOJO) is a developer's tool kit. It comprises of a set of Java archive files, commonly known as the JAR files containing pure Java components that can be used to develop usable stand-alone GIS applications or to add map functionality to applications. Since it is written in Java, it becomes very easy to integrate Java classes and components with it, making it a cross platform tool [3, 4].

2.3 HTML5, JAVA SCRIPT, CSS

HTML5 is a markup language used for structuring and presenting content for the World Wide Web (WWW) and a core technology of the Internet. It is a new standard for Hypertext Markup Language (HTML). HTML5 is a cross-platform designed to deliver almost everything you want to do online without requiring additional plugins. It does everything from animation to apps, music to movies, and can also be used to build complicated dynamic applications that run in your browser.

Following are some of the additional features of HTML5 that were not seen in the previous versions.

- The Canvas element, which defines graphic drawing using JavaScript.
- Audio and video tags which define movie and music content.
- Header, footer and section tags which make it easy to define page section and elements.
- Application caches, this allows webpages to start storing more information locally on the visitor's computer.
- Required and placeholder tags which make validation easy.
- HTML5 is still a work in progress. However, all major browsers support many of the new HTML5 elements and APIs [5].

Now that we know how a document can be rendered, the next step would be to know how to make this document user friendly and interactive. This can be achieved using Java Script, which accesses all the contents on the rendered screen and dynamically updates and re-renders the screen contents. The JavaScript is usually written between tags `<script>....</script>` and this can be included in HTML file or can be written as a separate file with ‘.js’ extension.
The most important feature of a webpage is presentation; this is taken care by Cascading Style Sheets (CSS), like JavaScript this can also be part of the main page alongside HTML tags defining how it should be rendered (color, font, size etc.), or can be written into a separate ‘.css’ file which helps for better readability and brevity, avoiding duplication.

### 2.4 Eclipse IDE

An Integration Development Environment plays an important role in effective application development. The important features, which IDE provides, are:

- **Code editing capability** which provides you IntelliSense with code refactoring. IntelliSense corresponds to accessing the descriptions of the functions, their parameter list and it reduces the reference to the external documentation and hence speeds up the development time.

- **Debugger**: It helps you in debugging your code by adding the watch to the variables in the code, which helps you in monitoring the different values for the variables at different stages during the code execution. It also helps you in putting breakpoints in between the code execution so that it becomes easier to find bugs using tools like JUnit and Versioning.

I have used Eclipse as the IDE for this application. But for making eclipse available for the Map Objects Java Objects, library files (jar files) must be configured.

### 2.5 Development Platform

This part describes the various software applications used to develop this tool.

- **Map Objects Java Edition 2.0**: Map Objects comes in two editions: Windows and Java. We used the Java Edition for this application.

- **Java (SE) Development Kit 1.7**: This is intended for standard programming applications.

- **Eclipse IDE**: It is an IDE for writing, compiling, and running Java programs.

- **JavaScript, CSS3 and HTML5**: Used for creating all the web pages.
CHAPTER 3

REQUIREMENTS

A good understanding of the requirements for an application before starting the development phase is essential. The detail and the extent of correctness of the requirements have a large role to play in the overall success of the project. This chapter brings to light the various requirements gathered for this application. Figure 3.1 displays the requirements.

![Figure 3.1. The requirements.](image)

The requirements gathered for this tool are classified as follows:

- Data requirements.
- Platform requirements.
- Functional requirements.
- User Interface requirements.

3.1 DATA REQUIREMENTS

The data requirements were gathered from Prof. M. Ziad Bayasi in order to have accurate information pertaining to the structural details. The data represented in this tool plays a key role in the actual outcome of this application. As this application is designed for school children it was essential that all the data represented was well researched and easy to
understand. It was also required that all the data should be represented in a fun, pictorial and interactive way which would appeal to users and students. The display of data is covered in the Functional Requirements.

### 3.2 Platform Requirements

Platform requirements refer to the OS, development platform, the programming language and the other tools used in developing this tool. Although this tool is platform independent, the implementation was done using the Macintosh OS X platform. The decision to use Map Objects Java Edition was taken after carefully analyzing all the other options and by taking Professor Dr. Carl Eckberg’s suggestion. The decision to use web technologies was taken in order to make the user interface interactive. The decision was to make the user interface as easy to use and logical as possible. The user is also provided with lots of different ways of reaching the information by categorizing the data.

The different technologies used in developing this GIS tool are the following:

- JAVA as a programming language.
- Map Object Java Edition (MOJO) as GIS API.
- Web Technologies such as HTML5, CSS3, JavaScript.

### 3.3 Functional Requirements

A functional requirement describes what a software system should do. The GIS Learning Tool for USA’s Tallest Skyscrapers and Their Construction was designed based on the following functional requirements:

- The tool should run on any platform.
- The software must display the USA map on startup.
- The software must display the points on the USA map. These points represent the metro cities hosting the tallest skyscrapers. Information about the city and the tallest skyscrapers in that city is displayed when clicked on the point using the hotlink tool.
- All the graphical features and fonts should be neat and clean.
- The software shall provide tools to zoom in and zoom out and zoom to full extent.
- The software shall provide pan options.
- The software is capable of adding/removing layers, promoting and demoting layers, and adding more cities using CSV files.
• The places are identified with the help of an identification tool.
• The software comes with the print option.
• The software should be provided with a hotlink tool to get more information about particular places.
• There must be instructions provided to use the software.
• Hyperlink should allow the user to go to external webpage to get more information.
• The project should be nicely packaged in a JAR file and placed on a campus website so it can be easily downloaded and installed on any machine.

3.4 User Interface Requirements

The User Interface (UI) Requirements refer to the look and feel of the tool. The following describe the UI requirements:

• The tool should use easy to understand terminology and expressions.
• The tool should have high quality images and videos. The web pages should be interactive and interesting and must appeal to the end users.
• The tool must be easy to use and navigating from one page to another must be smooth and easy.
CHAPTER 4

SOFTWARE DEVELOPMENT METHODOLOGY

Prototyping is one of the most crucial parts of any software development life cycle. It is an early sample/model of the product to be built. Most software companies use prototyping in their life cycle of software development. Prototyping is essential because it ensures that the end users regularly work with the system and provide feedback which is incorporated in the prototype to result in a functional and usable system.

Prototyping is excellent for designing good human computer interface systems such as interactive tools and websites. Prototypes also give a clear picture to the developers on what the requirements are and how to go about developing them. Moreover, changes can be easily incorporated into the prototype at any point of software development, which makes it more appealing especially for a web application where one can have constant changes in design.

Prototyping was used as an essential part of this thesis project. Workflows were built before the actual code was written.

Figure 4.1 gives the basic idea about the Software Development Life Cycle [6] and describes the lifecycle and development steps for the USA’s tallest Skyscrapers and their construction tool. This model increases the flexibility of the development process by allowing the client to interact and experiment with a working representation of the product. The main focus of this model is to satisfy the client needs, and the developmental process only resumes once the requirements are satisfied with the functioning of the prototype.

Software development life cycle can be subdivided into six phases as shown in (Figure 4.1). Description of each phase is as follows:

- **ANALYSIS:** This is the requirement-gathering phase where we gather the data, take decision regarding the requirements of the thesis such as Data, Functional, Platform and other requirements.

- **DESIGN:** This is second phase of the development where we analyze and design the application in a broader sense. An initial draft that highlights the features and operations is created.
Figure 4.1. Software development life cycle/prototype model.

- DEVELOPMENT: In this phase the code is written keeping in mind the deliverables of the design phase. This is the longest phase of the SDLC. Unit testing was performed in order to ensure that there are fewer bugs and the code meets all the requirements.

- TESTING: During this phase, the implementation is tested against the requirements to ensure that it meets the needs, addressed and gathered during the requirement phase. Test cases were used to check whether the application meets all the requirements.

- VERIFICATION: In this step of the life cycle of the application it was verified and checked that the application meets all the requirements. Prof Dr. Carl Eckberg reviewed the application and if there were any additional requirements, changes were made to incorporate those requirements.

- MAINTENANCE: This step is about maintenance required once the application is deployed, however, in this case, it is out of the scope of this Thesis.
CHAPTER 5

SOFTWARE DESIGN

This chapter mainly focuses on the high-level software design; architecture and class diagrams used for implementing this application. This application is designed to make the installation easy and convenient for the end user.

5.1 HIGH LEVEL ARCHITECTURE DIAGRAM

Figure 5.1 shows a high-level architecture diagram for the application. It is composed of the following parts:

- USER INTERFACE
  Provide GUI to interact with tool/map

- MAP OBJECTS
  Renders data from shp/csv to print on map

- SHP/CSV
  Geometric and other information

Figure 5.1. High-level architecture design.
- **User Interface** - This describes the end user interaction with the application developed. It displays the home screen which is launched when the application starts and shows how the user uses the application to learn about various cities of USA’s tallest skyscrapers and their construction. Once the user click the launch app button on the startup screen, all the layers are displayed giving the user the ability to add/remove layers, color them using different schemes, label them, zoom in, zoom out, pan, zoom to active layer, help, select various features, open attribute tables and much more.

- **Map Objects and Java Classes** - These contain all the functionality and actions performed by the various toolbars like custom toolbar, selection toolbar, button, menus and other items on the main interface.

- **Shape/Resource/Text Files/Html Pages** - All the .CSV files (Comma Separated Values files) and text files having the location (latitudes/ longitudes) comes under this section. These .CSV files were converted to a shape file layer so that it is easy for the end user and are displayed on the UI Using MOJO and JAVA classes. The web pages (HTML files) and other resources such as images, CSS and Java Script [7] files and the shape files also fall into this category.

5.2 **Classes**

Class Diagrams [8] are very useful and helpful in organizing the entire code and its structure. It gives a clear understanding of how different classes interact with each other. It also accounts for the interdependencies of various classes used in this project for the application development. See Table 5.1 [4]. A Class Diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. It also helps to understand the interdependency of classes used for development.

The classes as shown in Table 5.1 have been used to develop the functionality provided by this tool. In this project a couple of packages have been included to implement functionality provided by inbuilt classes. The MOJO classes in Table 5.1 are in MOJO Jar file.
### Table 5.1. Classes Used for Development of This Application

<table>
<thead>
<tr>
<th>JAVA COMPONENT</th>
<th>JAVA CLASSES</th>
</tr>
</thead>
<tbody>
<tr>
<td>import com.esri.mo2.ui.bean</td>
<td>TocAdapter, TocEvent, Map, Layer, Toc, Legend, Tool</td>
</tr>
<tr>
<td>import javax.swing</td>
<td>MenuBar, JMenuItem, AbstractTableModel, TableColumn, JButton, JPanel, JSplitPane, JToolBar, ImageIcon</td>
</tr>
<tr>
<td>import com.esri.mo2.ui.tb</td>
<td>ProjectToolBar, ZoomPanToolBar, SelectionToolBar</td>
</tr>
<tr>
<td>import com.esri.mo2.map.dpy</td>
<td>BaseFeatureLayer, Layerset, FeatureLayer</td>
</tr>
<tr>
<td>import com.esri.mo2.file.shp</td>
<td>ShapefileFolder, ShapefileWriter</td>
</tr>
<tr>
<td>import com.esri.mo2.map</td>
<td>DatasetLayer, TrueTypeMarkerSymbol, Layerset, featureLayer, BaseFeatureLayer</td>
</tr>
</tbody>
</table>
CHAPTER 6

ECLIPSE AND MAP OBJECTS CONFIGURATION

This project was developed using the java platform, edited and compiled in Eclipse IDE. Eclipse provides a rich Java development environment and plug-in system. It really goes well with Map Objects Java Edition when appropriate libraries are added. JAR files are added from the Map Objects [4] folder.

Following steps explain how to configure Eclipse IDE and Map Objects in detail.

- Download and install Eclipse IDE from the link: http://www.oracle.com/technetwork/developertools/eclipse/downloads/index.html. See Figure 6.1
- Install Map Objects Java Objects and Java Software Development Kit, see Figure 6.2
- Launch Eclipse and create a new Java Project. (Browse to File > New > Java Project), See Figure 6.3
- Click on libraries tab, click on Add Libraries, now select User Libraries and click next. See Figure 6.4
- It will open a preference window click User Libraries and then click new and enter the name of the library. See Figure 6.5
- Now click on Add Jars button, navigate to the folder where MOJO is installed and select all the JAR files. Make sure that the newly created library checkbox is selected. See Figure 6.6
- Next step is to attach Javadocs to the jar files. Expand one of the JAR file and click on Javadoc location and then click edit. In the Javadoc location path navigate to the Javadoc folder in ESRI documentation directory and then click OK. Repeat this step for all other JAR files. See Figure 6.7
- Now you can use MOJO with Eclipse.
Figure 6.1. Download Eclipse.

Figure 6.2. Download map objects Java edition.

Figure 6.3. Creating a new Java project.
Figure 6.4. Creating a new library.

Figure 6.5. Create a new user library called ESRI and click on import.
Figure 6.6. Browse to the location where map objects was installed and select all JAR files.

Figure 6.7. Select the Java doc location and click on edit, link it with the java doc location from the installed map object folder.
CHAPTER 7

LAUNCHING THE APPLICATION

This chapter talks about the starting point of the application, which is the startup screen and the application’s main map window which is the main entry point to all the features of this application.

7.1 START SCREEN

When the application first runs, a dialog (see Figure 7.1) will be displayed. This is from where the user can launch the application, read the instructions to use the application or exit from the application.

![Figure 7.1. The application launch screen - starting point of the application.](image)

When the user clicks on the launch app button, the map screen/main window will be displayed. If the user clicks on read me, the read me manual will be displayed and if the user clicks on exit, the application window will close. The purpose of this page is to provide the user with a home page or a launch page were he/she will have access to a Read Me file which provides the information about the knowledge a user can gain by using this tool.
7.2 The Map Window

After the user clicks on the launch app button, the "Map Screen" for the application is displayed. This is the applications main window and is divided into different parts as shown in Figure 7.2.

![Figure 7.2. Application's start.](image)

The "Menu Bar" contains the six main menus. The "Toolbars" section contains several buttons that will provide the user of this application tools to manipulate the different interactive maps displayed in the "Map Area".

The "Table of Contents" or the TOC appears on the left of the "Map Area" and it comprises of a list of all the map layers currently displayed. The "Status Bar" displays the coordinates of the map to which the cursor currently points to, and it updates dynamically as the mouse pointer moves. This is also used to display the distance between the points selected using the "Distance Tool".
CHAPTER 8

MENU BAR

A menu bar is the region of the application where there are drop down menus, the functionality of a menu bar is to provide easy access to common functionality of the tool. In this chapter we will look into the tools found in the menu bar of this application.

8.1 FILE

The "File" section in the menu tool consists of tools to add a layer, print the map, remove layer, and a legend editor.

8.1.1 Add Layer

The user can add a shapefile to the application by clicking on this menu item. It opens up a dialog where the user can select a shapefile saved on his/her computer. (See Figure 8.1) This can be used by students to learn about adding new layers and building up their maps, for example, they can add a world lakes shapefile which will highlight all the lakes of the world or add a particular country's shapefile such as USA's shapefile to learn more about that region.

Figure 8.1. When user clicks on add layer it allows the user to add a new shapefile layer to the map.
8.1.2 Print

When the user browses to File -> Print; this feature lets the user print the map on a printer connected to the user's computer.

8.1.3 Remove a Layer

If the user would like to remove a layer from the map he can either uncheck the little checkbox next to the layer name or navigate to File -> Remove layer (See Figure 8.2). This tool will delete the selected layer. This option is greyed out until the user selects a layer. The user can add the layer back to the map by selecting "Add Layer" from the menu.

![Figure 8.2. Remove a layer from the map.](image)

8.1.4 Legend Editor

This tool is among the most important tools in the GIS toolkit. It lets the user change various properties of the selected layer such as the symbol, colors, and label to name a few. It can dynamically alter the rendering of the map. The class com.esri.mo2.ui.ren.LayerProperties is used to provide this feature. When the user selects this item from the menu it opens up a dialog which has three tabs; symbols, Label and General (see Figure 8.3) The user can use these tabs to change the appearance of the map, for instance, change the symbol, color, display all the country labels on the map and so forth. This tool can be a handy tool to learn about different features of the shapefile, it helps in customizing the map based on the user's preferences, thereby, making learning more interactive and interesting.

Figure 8.4 displays an interesting use of the legend editor where we have depicted each state with a different color and labeled all the states. It has three tabs: Symbols, Labels and Generals. In the symbol tab we get option to select from single symbol, graduated symbol and unique symbol. By using these symbols the layer feature can be represented. By
using label tab we can provide labels to the layer feature and it can be labeled by different categories stored in the attribute table. By using general tab, users can select the option of when the feature layer should be shown.
8.2 Theme

The "Theme" section in the menu consists of tools such as "open attribute table" and "create layer from selection".

8.2.1 Open Attribute Table

This tool opens the attribute table associated with the selected layer. The attribute table gives the user important information about the layer saved in the dbf file. The dbf file has a row of non-locational attributes for each feature in a map layer. (See Figure 8.5).

![Figure 8.5. The attribute table for the “USAStates” layer.](image)

8.2.2 Create Layer from Selection

This tool can be used to create a new layer. To create a new layer, the user must first select a layer, then click on the "Select Features" tool on the selection tool bar and choose rectangle, circle, line or polygon and then click on the areas they want to make a new layer from. Once a region of the map is selected it will be highlighted in yellow, the user can then click on Theme -> Create Layer From Selection (See Figures 8.6, 8.7, and 8.8).
Figure 8.6. Selecting the select features tool before creating a layer from selection.

Figure 8.7. Selecting the points that we want to create a new layer from.
Figure 8.8. Creating the new layer from the selected points. This layer is saved in your shapefiles folder.

8.3 LAYER CONTROL

The "Layer Control" section in the menu consists of tools such as "promote selected layer" and "demote selected layer".

8.3.1 Promote Selected Layer

This tool promotes the layer selected. To use this tool the user must first select a layer to promote. If the layer is the topmost layer on the TOC, this tool will be greyed out. Figure 8.9 demonstrates how to use this tool.

Figure 8.9. Demonstrates the "promote selected layer" tool.
8.3.2 Demote Selected Layer

This tool demotes the layer selected. To use this tool the user must first select a layer to demote. If the layer is the bottommost layer on the TOC, this tool will be greyed out. Figure 8.10 that demonstrate how to use this tool.

Figure 8.10. Demonstrates the "demote selected layer" tool.

8.4 By State

The “By State” section in the menu consists of items such as “California”, ”Georgia”, Illinois”, ”Massachusetts”, ”Nevada” etc. When user clicks selects an item, it will display corresponding state layer along with the point layer (points present in that state). See Figure 8.11.

Figure 8.11. State layer and point layer of selected state.
## 8.5 Extras

The “Extras” section in the menu consists of four items namely, “Fun with civil”, “WTC Revised Building Code”, “Historical Images”, “Mohawk Ironworkers” and “Quiz”. When you click on each one of these it will take you to online webpage or show a document or play a short 40min movie or open a webpage, where user can learn about skyscrapers and their constructions.

### 8.5.1 Fun with Civil

This item under extras menu opens online hands on workshop where student/user can learn why structures like skyscrapers stand up and keep them from falling down. Users can simulate an experiment on load, forces, materials and shapes. (See Figure 8.12)

![Figure 8.12. Demonstrate fun with civil.](image)

### 8.5.2 WTC Revised Building Code

Before knowing about revised building code for world trade center after 9/11 attacks we should know about:

**WHAT IS A BUILDING CODE?**

Practically, it is the government’s official statement on building safety. Technically, it is a compendium of minimum safety standards arranged in a systematic manner (codified) for easy reference. It embraces all aspects of building construction—fire, structural, plumbing, electrical, and mechanical.

**WHO NEEDS BUILDING CODES?**
Yes, we all do whether in our homes, offices, schools, stores, factories, or places of entertainment and skyscrapers. We rely on the safety of structures that surround us in our everyday living. The public need for protection from disaster due to fire, structural collapse, and general deterioration underscores the need for modern codes and their administration.

**HOW RELIABLE ARE THEY?**

Most aspects of building construction electrical wiring, heating, sanitary facilities represent a potential hazard to building occupants and users. Building codes provide safeguards. Although no code can eliminate all risks, reducing risks to an acceptable level helps.

Above facts are true and the WTC revised Building code document tell us about Importance of strong building codes. (See Figure 8.13)

![World Trade Center Building Code Revision after 9/11 Attack](image)

**Figure 8.13. Demonstrate WTC revised building code which opens a pdf file.**

**8.5.3 Demonstrate Historical Images**

This menu item opens a short movie where student/user finds a famous black-and-white photograph’s taken during construction of skyscrapers in New York City, United States (see Figure 8.14).

**8.5.4 Mohawk Ironworkers**

Since the 1900s, the country's most iconic bridges and skyscrapers have been put up by men who risked life and limb to connect steel beams hundreds of feet in the sky. Ironworkers come from all backgrounds, but a small Indian reserve outside Montreal has supplied the U.S. with a proud lineage of Mohawk ironworkers.
Figure 8.14. Demonstrate historical images which opens a 40sec’s movie file.

Striding across a steel beam 26 stories in the sky, a lean ironworker grabs the corner column and pulls himself up. What could someday be a corner office on 55th Street in Manhattan is now nothing but open air. It's windless and the sun is bright, a perfect day for putting up steel. In 2 1/2 hours, the crew lays 68 pieces of steel [9].

New York City has a Mohawk community founded by construction workers for bridges and skyscrapers of Mohawk and other Iroquois origin. They worked from the 1930s to the 1970s on special labor contracts as specialists and participated in building the Empire State Building. The construction companies found that the Mohawk ironworkers did not fear heights or dangerous conditions. Their contracts offered lower than average wages to the First Nations people and limited labor union membership.

Approximately 200 Mohawk iron workers (out of 2000 total iron workers at the site) have contributed to rebuilding the One World Trade Center. They typically drive the 360 miles from the Kahnawake reserve on the St. Lawrence River in Quebec to work the week in lower Manhattan, and then return on the weekend to be with their families. A selection of portraits of these Mohawk ironworkers was featured in an online photo essay for Time Magazine in September 2012 [10].

Over many decades, Mohawk ironworkers played key roles in constructing most of the America’s famous skyscrapers and iconic bridges. To name few of their work are Empire State Building, the Willis Tower, the San Francisco Bay Bridge, the Rockefeller Center and the One World Trade Center.
The menu item “Mohawk Ironworkers” plays a short documentary video about Mohawk Ironworkers (see Figure 8.15).

![Figure 8.15. Mohawk Ironworkers menu item.](image)

8.5.5 Quiz

This menu item opens an online quiz where users have an opportunity to test his/her knowledge in recognizing the famous Skyscrapers (see Figure 8.16).

![Figure 8.16. Demonstrate quiz. Opens an online webpage where user can play quiz and check his/her knowledge about skyscrapers of USA.](image)
8.6 Help

The “Help” section in the menu bar consists of tools designed to help the user in navigating through this application and getting all the help needed to use this application.

8.6.1 Help Topics

The help topics consist of three sub tools, namely, the “Table of Contents”, “Legend Editor”, and “Layer Control”. Clicking on “Table of Contents” opens a dialog box which contains useful information about the TOC. The “Legend Editor”, opens up a dialog box describing the legend editor and how to use it. Finally, the “Layer Control” gives the user further information about the layer control. The Figures 8.17, 8.18, 8.19 show the tools in action.

8.6.2 Help Tool

This tool gives the user instructions on using the help tool of the custom toolbar. The help tool allows the user to learn about other tools. To use the help tool, the user left-clicks on the help tool in the toolbar and the right-click on any other tool to learn more information about that tool (See Figure 8.20).

8.6.3 Contact Us

This tool gives the user information about the author who built this software. For any king of enquires send an email to the given email id (see Figure 8.21).
Figure 8.18. Demonstrates the legend editor help tool.

Figure 8.19. Demonstrates the layer control help tool.
8.6.4 About MOJO Tool

This tool gives the user information about the Map Objects Java Objects software running on their system, it displays the MOJO version and a little information about the software (See Figure 8.22). When the user clicks on the “System Info” button, he/she will get information about the version of Java, the version of the Java virtual machine and the operating system running on the user’s computer.
Figure 8.22. Demonstrates the about MOJO tool.
CHAPTER 9

MAPOBJECTS TOOLBARS

Map Objects Java Edition also called Map Objects Java Objects (MOJO) particularly in San Diego State University, makes development easy using Eclipse, it has advantages such as a console output, easier formatting, library auto import and debugging. The ESRI Zoom/Pan Toolbar provides the basic functionalities such as Zoom in, Zoom out, Pan, etc. In this chapter our main focus is on integrating the toolbar into the GIS tool and the basic functionality of the tool bar.

The different categories of toolbars used are given below:

- JAVA Toolbar (Custom Toolbar).
- MOJO Zoom Pan Toolbar.
- MOJO Selection Toolbar.

Figure 9.1 shows the various toolbar used in this application.

![Figure 9.1. Toolbar.](image)

9.1 MOJO ZOOM PAN TOOLBAR

The “COM.ESRI.MO2.UI.TB.ZOOMPANTOOLBAR” library is used to implement the Zoom Pan Toolbar. It uses the functions provided by the “ZoomPanToolBarActions” class. The MOJO Zoom Pan Toolbar helps the end user interact with the map and work with its various features.

This toolbar consists of nine buttons (Refer to Figure 9.2.). We describe the functionality of each button below.

9.1.1 The Previous Extent Tool

This tool zooms the map to the previous extent stored in the extent history (almost like a ctrl + Z or undo for maps).
9.1.2 The Next Extent Tool

This tool zooms to the next extent stored in the extent history (almost like a redo for maps).

9.1.3 The Zoom to Active Layer Tool

This tool zooms the map to the extent of all the features that are part of the selected layer. Therefore, after this tool is selected, the map includes all the selected layers and features only.

9.1.4 The Zoom to Full Extent Tool

This tool zooms the map to the extent of all the layers within the map. Therefore, when this tool is selected we get a bird's eye view of the entire map.

9.1.5 The Zoom In Tool

This tool helps us zoom into the map, when we use the tool the cursor changes into a magnifying glass, and we need to select the arrow/cursor tool to unselect this tool. It also provides a tool for clicking or dragging a rectangle on the map in order to zoom in.

9.1.6 The Zoom Out Tool

The zoom out tool makes available a tool for clicking or dragging a rectangle on the map in order to zoom out. When we use this tool, the cursor changes to a zoom out magnifier to indicate that the tool is active. In order to unselect the tool, the arrow (cursor) tool button needs to be pressed.
9.1.7 The Pan Layer Tool

This tool allows us to drag the map to a new location without altering the map's scale. When we use this tool, the cursor changes to a hand shape to indicate that the tool is active. In order to unselect the tool, the arrow (cursor) tool button needs to be pressed.

9.1.8 The Pan In One Direction Tool

Pans the map in one of four directions, north, south, east, or west. When pressing this tool button the drop down list, seen in Figure 9.3, is displayed.

![Pan In One Direction Tool](image)

Figure 9.3. Pan in one direction's drop down.

9.1.9 Identify Tool

Performs and identify function on the features that are part of the currently selected layer. When using this tool, the cursor changes to a smaller pointer that has information sign on top, to indicate that the tool is active. In order to unselect the tool, the arrow (cursor) tool button needs to be pressed (Refer to Figure 9.4).

9.2 MOJO SELECTION TOOLBAR

The COM.ESRI.MO2.UI.TB.SELECTIONTOOLBAR library is used to develop the selection toolbar. This toolbar offers functions that allow the user to do selections based on the attribute table.

The Selection Toolbar contains seven standard buttons. The functionality of six of them is briefly described in this section. Figure 9.5 shows the toolbar.

9.2.1 The Find Tool

This tool opens a dialog to locate features on the current map whose attributes contain an end-user provided string. It doesn't have to be an exact match. As long as the provided string
9.2.2 The Query Builder

This tool opens a dialog for locating features based on a query that an end-user constructs using a subset of SQL logical and comparison operators provided in the same dialog. After the statement gets executed the result records are displayed in a table and the result features are highlighted on the map. Figure 9.7 shows what the "Query Builder" dialog looks like.
Find tool’s dialog. Dialog displayed when using the find tool with the input string: “Texas”. The feature found is highlighted on the map.

Figure 9.6. Find tool’s dialog. Dialog displayed when using the find tool with the input string: “Texas”. The feature found is highlighted on the map.

The query builder tool’s dialog is displayed when using the query Builder tool with the SQL statement, we used the query STATE_NAME=’California’ here. The result is highlighted on the map.

Figure 9.7. The query builder tool’s dialog is displayed when using the query Builder tool with the SQL statement, we used the query STATE_NAME=’California’ here. The result is highlighted on the map.

9.2.3 The Select Features Tool

This tool provides a way for selecting features by rubber banding a shape in the map. When the tool button is pressed, the drop down list seen in Figure 9.8 is displayed.

The list includes the options for using a rectangle, circle, and line or polygon shape to draw on the map in order to select the desired features (Refer to Figure 9.9). When the left-
42

Figure 9.8. Select features’ drop down list.

Figure 9.9. Select features tool used to draw a rectangle. A rectangle shape is drawn on the map using this tool.

click mouse button is released, the features within the drawn shape are automatically selected and highlighted in yellow.

9.2.4 The Clear All Selection Tool

This tool provides the ability to clear all the selected features that result from using the previously discussed tools that are part of the Selection toolbar.

9.2.5 The Buffer Tool

Pressing this toolbar button opens a dialog for constructing a buffer polygon around the currently selected features. This dialog can be seen in Figure 9.10.

9.2.6 The Attributes Tool

This tool displays attributes of the currently selected features. All the database records for the current layer are displayed in the table but only the records for the selected features are highlighted in blue. See Figure 9.11.
Figure 9.10. Buffer tool's dialog. The buffer dialog provides a way to input the buffer distance based on the currently selected features.

Figure 9.11. Attributes table for the selected layer.

9.3 CUSTOM TOOLBAR

This toolbar comprises of tools, which are customized according to the requirement of the application using JAVA. This toolbar allows users to add a layer, print the map image, view the distance between two points on the map, etc. this toolbar is discussed in detail in the next chapter.
CHAPTER 10

CUSTOM TOOLBAR

In addition to the toolbars discussed in chapter 9, these are additional tools customized according to the need of the project. Custom toolbars are created to make an easy interface for the students and to improve the usability of the tool. The snapshot in Figure 10.1 depicts the customized toolbar.

The Customized Toolbar contains seven buttons. Their functionality is briefly described in this section. See Figure 10.1.

![Customized toolbar](image)

**Figure 10.1. Customized toolbar.**

10.1 PRINT TOOL

This tool provides the printing functionality found in most user applications. It calls the native print dialog used in the operating system. It prints the current map on display.

10.2 ADD THEME TOOL

This tool opens a dialog to be able to browse for map layer (in SHP format) that one wants to be added to the map being currently displayed. See Figure 10.2 and 10.3.

10.3 POINTER TOOL

This tool provides a way for the user to unselect a map tool (that changes the shape of the cursor like Zoom In, Zoom Out, Pan, Identify, Hotlink and Help Tool) by changing the cursor back to the normal pointer
Figure 10.2. The add layer dialog. This dialog allows the user to select the new layer to be added to the current map.

Figure 10.3. The California layer is added and appears on the TOC.
10.4 **DISTANCE TOOL**

This tool displays the distance between any two points selected on the map. The user can select this tool and click on the starting point then drag the mouse to the end point. The distance is displayed on the status bar (Refer to Figure 10.4)

![Distance Tool Image](image)

**Figure 10.4.** The distance between two points. The selected distance is displayed as a blue line and the distance appears on the status bar in miles and kilometers.

10.5 **XY TOOL**

This tool allows the user to add a Comma Separated Values file to the map. On selecting this tool a dialog box opens where the user can select a CSV file to be added to the map. The CSV file must longitudes and latitudes. An example of CSV file is shown in Figure 10.5. When the XY tool user navigates to this file, a new layer will appear on the map. This file is displayed on the TOC as "MyPoints1" layer. The points are displayed as “buildings” ESRI font based image on the map. (See Figure 10.6)

10.6 **HOTLINK**

This button launches the hotlink tool which works with the active layer on the map and on clicking this tool the cursor changes to a lightning bolt and when the user clicks a point on the map with the center of the lightning bolt touching the point, it opens up a dialog box which provides the user additional information about the point clicked. The information
Figure 10.5. Example of CSV file.

Figure 10.6. Displays the layer “My Points1” added using the XY feature.

displayed on the dialog box is read from “config.properties” (see Appendix A). This dialog box provides a brief history of the city (New York City in Figure 10.7). The dialog also has three buttons, when you click on second button (See Figure 10.7) it launches a website with more information, details, pictures and videos about the top three tallest buildings in that city. Figure 10.8 describes the web page where the user could learn more about the skyscrapers in
Figure 10.7: The dialog opens up when the user uses the hotlink tool and clicks on a point on the map. (We clicked on Seattle in this example).

New York City. These webpages are found in the html directory of the application (see Appendix B).

Description about the webpage is as follows (The numbers in the description is mapped to the numbers on webpage in Figure 10.8):

1. **Photo Gallery**: A slide show of the attractive images of the skyscraper is displayed here.

2. **Tallest buildings**: This section of the page lists tallest buildings in the city. This is linked to wikipage of that building.

3. **Video**: A short video about the skyscraper is embedded on this page.

4. **Find Me**: It is a simple point on a USA map pointing the building.

5. **Facts**: Latest facts about this building are listed here. The facts include issues faced during construction, entertainment in the building like rides, glass floor etc., current usage of the building and much more.

6. **Identification**: Provides the name, alternate name, official website link and also the architect/construction company designed/built the structure.

7. **Location**: Provides the physical address of the building.

8. **Structure in General**: This portion provides more civil engineering related about the building, which includes Structural information, Façade information and architectural style.

   a. **Structural System**: The term structural system in structural engineering refers to load-resisting sub-system of a structure. The structural system transfers loads through interconnected structural components or members. The structural system of high-rise buildings was introduced in 1969 and was
Figure 10.8. Sample static web page with information about the skyscraper.
extended to incorporate interior and exterior structures. The following interior structures are possible: Hinged frame, Rigid frame, Braced frame and Shear-walled frame, Outrigger structures. The following exterior structures are possible: Tube, Diagrid, Space Truss, Exoskeleton, and Super frame.

b. **Structural Material:** A steel tall building is defined as a building where the main vertical and lateral structural elements and floor systems are constructed from steel. A concrete tall building is defined as one where the main vertical and lateral structural elements and floor systems are constructed from concrete. A composite tall building utilizes a combination of both steel and concrete acting compositely in the main structural elements, thus including a steel building with a concrete core. A mixed-structure tall building is any building that utilizes distinct steel and concrete systems above or below each other. There are two main types of mixed structural systems: A steel/concrete tall building indicates a steel structural system located above a concrete structural system, with the opposite true of a concrete/steel building. Additional Notes: (1) If a tall building is of steel construction with a floor system of concrete planks on steel beams, it is considered a steel tall building. (2) If a tall building is of steel construction with a floor system of a concrete slab on steel beams, it is considered a steel tall building. (3) If a tall building has steel columns plus a floor system of concrete beams, it is considered a composite tall building.

c. **Facade System:** A facade is generally one exterior side of a building, usually, but not always, the front. The word comes from the French language, literally meaning frontage or face. In architecture, the facade of a building is often the most important aspect from a design standpoint, as it sets the tone for the rest of the building. From the engineering perspective of a building, the facade is also of great importance due to its impact on energy efficiency. Facade often referred to as Cladding, Envelope, Exterior Wall, or Curtain Wall.

d. **Facade Material:** The primary material(s) of a structure's exterior surface, except for windows. The facade material of any built object is any substance, which forms a significant portion (at least 10%) of its outward-facing surface, not including windows, doorways, or openings. More than one facade material may be assigned, as long as each one constitutes at least ten percent of the facade surface area. Facade materials do not include any underlying or hidden material such as insulation or structural walls, unless those are exposed to the outside. They may, however, be covered by paint, varnish, or very thin protective coatings. Very thin coatings on a facade are not considered facade materials themselves, although thick coatings like stucco are. Glass is only considered a facade material if it covers something opaque, such as a column, spandrel, or wall. Windows are not considered part of the facade for purposes of this definition. The actual arrangement or form of facade materials is described separately by the facade system. Built objects can also be classified according to facade color.

e. **Architectural Style:** It is characterized by the features, which make a building or other structure notable and historically identifiable. A style may include such
elements as form, method of construction, building materials, and regional character.

9. **Technical Data**: This section describes about the height, number of floors, cost incurred for construction, and current usage of the building

**10.7 Help Tool**

With this tool, you are able to right-click on a tool button and see a help window pop up with information on how that tool works. See Figure 10.9. When using this tool the cursor will change its shape into a pointer with a question mark to the side. To unselect the Help Tool, and go back to the normal cursor shape, the pointer tool button needs to be pressed.

![Figure 10.9. Help tip window. Window displayed when the Help Tool is used on the hotlink button.](image)
CHAPTER 11

WORLD TOP 20 SKYSCRAPERS

A World Top 20 Skyscrapers button is displayed on the top right of the application main page. This button takes us to the world’s famous top 20 Skyscrapers web page. This page is designed to depict flash cards. When the page is launched, user sees the pictures of the world’s top 20 skyscrapers. When he places the mouse on the picture, it flips to display the name and other information about respective skyscraper (see Figure 11.1).

Figure 11.1. Webpage displaying the top 20 famous skyscrapers of the world.
CHAPTER 12
SHAPE FILE

This chapter describes shape files and how to create them.

12.1 SHAPE FILE DEFINITION

A shape file is a digital vector (non topological) storage format used for storing geometric locations and their associated attribute information. The shapefiles handle single features that overlap or that is noncontiguous. They typically require less disk space and are easier to read and write.

Shapefiles [11] have advantage over other data sources such as faster drawing speed and edit ability; since a shapefile is non topological it does not maintain spatial relationship information such as connectivity, adjacency, and area definitions.

12.2 HOW SHAPE FILES CAN BE CREATED

Ways to create a Shape File are as follows:

- **Export:** Using software like ARC/INFO, PC ARC/INFO, Spatial Database Engine (SDE), and ARC View GIS Shapefiles can be created by simply exporting any data source to a shapefile.
- **Digitize:** With the help of ArcView GIS feature creation tools; Shape Files can be created by digitizing shapes.
- **Programming:** Using Avenue (ArcView GIS), Map Objects, Arc macro Language (AML) or Simple Macro Language (SML) software, we can create a shapefile within our programs.
- **Usage of Tool:** Write directly to the shapefile specifications by creating a program.

The shapefiles comprises of three mandatory and several optional files. Three mandatory files are as follows.

- **.shp** - This file is used to store the feature geometry and when added displays the geometry.
- **.shx** - This file is used to store the index of the feature geometry, helpful for navigation.
- **.dbf** - This file is used to store attributes for different shapes.
12.3 Shape File

Shapefile is denoted by .shp extension. This file stores the feature geometry. A shapefile has its header fixed with the variable length records. The main header is 100 bytes in length, which contains about 17 fields comprising 9 fields of 4 bytes each and 8 fields of eight bytes each. Tables 12.1 and 12.2 show the representation of the shapefile. A variable length record header is 8 bytes in length and mainly consists of two fields that contain data for record number and content length. There is 1:1 correspondence between the shape type in the main file header and its contents in the variable length record [11].

Table 12.1. Description of Shape File Header

<table>
<thead>
<tr>
<th>Position</th>
<th>Field</th>
<th>Value</th>
<th>Type</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte 0</td>
<td>File Code</td>
<td>9994</td>
<td>Integer</td>
<td>Big</td>
</tr>
<tr>
<td>Byte 4</td>
<td>Unused</td>
<td>0</td>
<td>Integer</td>
<td>Big</td>
</tr>
<tr>
<td>Byte 8</td>
<td>Unused</td>
<td>0</td>
<td>Integer</td>
<td>Big</td>
</tr>
<tr>
<td>Byte 12</td>
<td>Unused</td>
<td>0</td>
<td>Integer</td>
<td>Big</td>
</tr>
<tr>
<td>Byte 16</td>
<td>Unused</td>
<td>0</td>
<td>Integer</td>
<td>Big</td>
</tr>
<tr>
<td>Byte 20</td>
<td>Unused</td>
<td>0</td>
<td>Integer</td>
<td>Big</td>
</tr>
<tr>
<td>Byte 24</td>
<td>File Length</td>
<td>File Length</td>
<td>Integer</td>
<td>Little</td>
</tr>
<tr>
<td>Byte 28</td>
<td>Version</td>
<td>1000</td>
<td>Integer</td>
<td>Little</td>
</tr>
<tr>
<td>Byte 32</td>
<td>Shape Type</td>
<td>Shape Type</td>
<td>Integer</td>
<td>Little</td>
</tr>
<tr>
<td>Byte 36</td>
<td>Bounding Box</td>
<td>Xmin</td>
<td>Double</td>
<td>Little</td>
</tr>
<tr>
<td>Byte 44</td>
<td>Bounding Box</td>
<td>Ymin</td>
<td>Double</td>
<td>Little</td>
</tr>
<tr>
<td>Byte 52</td>
<td>Bounding Box</td>
<td>Xmax</td>
<td>Double</td>
<td>Little</td>
</tr>
<tr>
<td>Byte 60</td>
<td>Bounding Box</td>
<td>Ymax</td>
<td>Double</td>
<td>Little</td>
</tr>
<tr>
<td>Byte 68*</td>
<td>Bounding Box</td>
<td>Zmin</td>
<td>Double</td>
<td>Little</td>
</tr>
<tr>
<td>Byte 76*</td>
<td>Bounding Box</td>
<td>Zmax</td>
<td>Double</td>
<td>Little</td>
</tr>
<tr>
<td>Byte 84*</td>
<td>Bounding Box</td>
<td>Mmin</td>
<td>Double</td>
<td>Little</td>
</tr>
<tr>
<td>Byte 92*</td>
<td>Bounding Box</td>
<td>Mmax</td>
<td>Double</td>
<td>Little</td>
</tr>
</tbody>
</table>
Table 12.2. Value/Shape Type Mapping

<table>
<thead>
<tr>
<th>VALUE</th>
<th>SHAPE TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Null Shape</td>
</tr>
<tr>
<td>1</td>
<td>Point</td>
</tr>
<tr>
<td>3</td>
<td>PolyLine</td>
</tr>
<tr>
<td>5</td>
<td>Polygon</td>
</tr>
<tr>
<td>8</td>
<td>MultiPoint</td>
</tr>
<tr>
<td>11</td>
<td>PointZ</td>
</tr>
<tr>
<td>13</td>
<td>PolyLineZ</td>
</tr>
<tr>
<td>15</td>
<td>PolygonZ</td>
</tr>
<tr>
<td>18</td>
<td>MultiPointZ</td>
</tr>
<tr>
<td>21</td>
<td>PointM</td>
</tr>
<tr>
<td>23</td>
<td>PolyLineM</td>
</tr>
<tr>
<td>25</td>
<td>PolygonM</td>
</tr>
<tr>
<td>28</td>
<td>MultiPointM</td>
</tr>
<tr>
<td>31</td>
<td>MultiPatch</td>
</tr>
</tbody>
</table>

### 12.4 INDEX FILE

The Index file is denoted by .shx extension. This file is used to store the Index of the feature geometry. The offset in the index file record, which relatively maps to the corresponding record in main file. The Index file is identical to shape file header. The file length stored in the header is the total length of the file, which is in 16 bit words.

### 12.5 DBASE FILE

The dbase file is denoted with the .dbf extension. The dBase table gives non-functional attributes of a feature in a layer. Record numbers are used to associate geometry records with attribute records. Thus, the order of records in the dBase file should be same as the order of records in main file. The file can be opened and read using Microsoft Office Excel [11].
The main requirements for a dBase table are:

- There should be one record per shape feature in the table.
- The order of the records is defined by the order of the shape features in the main file.
- The prefix of the file name should be same as the prefix of shape and Index File. Also the suffix should be dbf.

### 12.6 How to Create Shape File

The tools are implemented in the application to create new shape files when required. ‘Create Layers from Selection’ is used to create a shape file. By using a select feature tool, we select some features from the existing map and use theme-&gt;create layer from selection and mention the name you want to give to the shape file, and the location to store it. The steps followed to create a shape file are explained in Figure 12.1.

![Screen shot of a feature selection tool](image)

**Figure 12.1. Creating shape file by selecting features. (Theme-&gt;create layer from selection).**

In Figure 12.1, we select the select feature tool and chose the features (points) we want to convert to a new shape file. The yellow points are the selected features. Then go to menu named Theme -&gt; create layer from selection, which will open up a dialog as shown in Figure 12.1 to enter a name and the location for a newly created shape file.
CHAPTER 13

SUMMARY AND OBSTACLES

Working on this project was an interesting, challenging, intellectually enriching experience and taught me importance of time management, data collection, design and planning. It not only gave me an opportunity to work with latest technology and enhance my skills but also taught me to focus on user interface and design decisions.

MOJO provides easy extendibility. It will help to add new features or functionality to this tool effortlessly. This tool can be converted to a web based application to provide better accessibility to the users. Since Civil, Constructions and Architectural Engineering was not my department of studies, I had to start learning from zero knowledge. It took a while to understand the terminology. One of the most challenging parts of the thesis was the collection of data. (For example, Evolution Building Codes, Statistics about the buildings).

A major obstacle was to support different browsers. Most of the CSS properties and html5 properties did not work very well with Internet explorer and certain features were not compatible with Chrome and Firefox, many workarounds and validations had to be coded to ensure the application worked on all browsers. However, it was determined that Firefox was best suited for it and was made a requirement for the application.

Professor Eckberg’s & professor Bayasi’s suggestions and support played a key role in overcoming most of the obstacles and making this application a success.
CHAPTER 14

FUTURE ENHANCEMENTS

This tool provides a great framework, which can be expanded to add more features and functionalities. This tool has been developed using object-oriented design in java, which will allow reusing classes, and extra features can be added with minimal code changes. Following are some of the future enhancements for this application.

- Deploying the application on the web as browser hosted applet, so that it can be used without installing it on each machine.
- This tool can be localized into different languages.
- Application can be made for mobile devices and tablets.
- More customized tools can be added as per the requirement.
- More countries can be added.
REFERENCES


APPENDIX A

CONFIG PROPERTIES FILE
This application uses a properties file called config.properties which is under ./properties/ directory in the USASkyscrapers application folder. For more details about the application directory structure refer to Appendix A. The values from properties file is used in displaying different fields of JDialog opened when hot link is clicked on the city point. The format of the properties for each city is as follows:

```
<CityName>.description=<Brief description about the city>
```

This property is used to define the brief description of the city. The value of which is displayed in the Text area of the JDialog.

```
<CityName>.image=<City image file name> (Image file should be under ./images/)
```

This property specifies the image file name of the city. Image file should be placed under ./images/ directory. This image is displayed in the image area of the JDialog.

```
<CityName>.building1=<Building1 Name>
```

This property is used to display the building 1 button name. If this value is not present, first button in the JDialog is disabled.

```
<CityName>.webpage1=<Building1 webpage>
```

This property provides the html page file name of building1. On click of button1, this html page will be displayed on the browser. HTML page should be placed under ./html/ directory. If this property is not present, first button is disabled.

```
<CityName>.building2=<Building2 Name>
```

This property is used to display the building 2 button name. If this value is not present, second button in the JDialog is disabled.

```
<CityName>.webpage2=<Building2 webpage> (Static HTML page corresponding this building. Page should be under ./html/)
```

This property provides the html page file name of building2. On click of button2, this html page will be displayed on the browser. HTML page should be placed under ./html/ directory. If this property is not present, second button is disabled.

```
<CityName>.building3=<Building3 Name>
```

This property is used to display the building 3 button name. If this value is not present, third button in the JDialog is disabled.

```
<CityName>.webpage3=<Building3 webpage> (Static HTML page corresponding this building. Page should be under ./html/)
```

This property provides the html page file name of building 3. On click of button3, this html page will be displayed on the browser. HTML page should be placed under ./html/ directory. If this property is not present, third button is disabled.
This file is loaded when the application is launched and when the csv file is used to load more points. If we modify the file after the application launch, the changes will not take effect until application is re launched.

**Example:** Properties for Seattle is as shown in Figure A.1 and corresponding JDialog is as shown in Figure A.2

![config.properties](config.properties)

Seattle.description=Seattle Washington is a multi-faceted city located in the Pacific Northwest an area that has grown increasingly popular over the last two decades. In addition to its reputation as the espresso cart capital of the world Seattle also has a dining scene that features acclaimed local chefs and internationally recognized cuisine. The city is extremely multicultural and is influenced by the Pacific Rim in art and architecture. The city offers residents a pace of life that can be either relaxed or fast paced and entrepreneurial. Seattle is home to many successful startup businesses such as Microsoft Starbucks Eddie Bauer Nintendo Amazon.com and REI. Water is everywhere both freshwater and salt water and the cities maritime heritage is very noticeable.

Seattle.image=Seattle.jpg
Seattle.building1=Columbia Center
Seattle.webpage1=ColumbiaCenter.html
Seattle.building2=1201 Third Avenue Tower
Seattle.webpage2=ThirdAvenueTower.html
Seattle.building3=Two Union Square
Seattle.webpage3=TwoUnionSquare.html

**Figure A.1.** Config.properties for Seattle.
Seattle, Washington is a multi-faceted city located in the Pacific Northwest, an area that has grown increasingly popular over the last two decades. In addition to its reputation as the espresso cart capital of the world, Seattle also has a dining scene that features acclaimed local chefs and internationally recognized cuisine. The city is extremely multicultural and is influenced by the Pacific Rim in art and architecture. The city offers residents a pace of life that can be either relaxed or fast-paced and entrepreneurial. Seattle is home to many successful startup businesses such as Microsoft, Starbucks, Eddie Bauer, Nintendo, Amazon.com and REI. Water is everywhere, both freshwater and salt water, and the city's maritime heritage is very noticeable.

Figure A.2. Hot link JDialog for Seattle.
APPENDIX B

APPLICATION DIRECTORY STRUCTURE
This application consists of an executable jar file and directories to store webpages, properties, images and csv files. The directory structure should be as shown in Figure B.1

![Figure B.1. Application directory structure.](image)

- USASkyscrapers is the root directory under which all other artifacts required for the application is placed.
- User can add more points to the map by loading USASkyscrapers.csv file, which is under csv directory.
- html directory is under USASkyscrapers and hosts all webpages used in the application. It also has css, js and images directories. All “.css” files used in the webpages are placed under css directory, “.js” files are placed under js directory and image files “.jpg”, “.png” and “.gif” used in html pages are placed under images directory.
- Images displayed on the JFrame window are placed under images directory.
- config.properties, ReadMe.pdf, User_Guide.pdf, WTC_Bcode.pdf and Historical_Images.mov files are placed under properties directory.
- All shapefiles used in the application are placed under shapeFiles directory.
- USASkyscrapers.jar is the executable jar file. On double click of this file application is launched. This file is placed under USASkyscrapers directory.