MULTIMEDIA SURVEY OF ANCIENT BABYLONIAN CIVILIZATION

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
Satya Srinivas Vema
Spring 2016
SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the
Thesis of Satya Srinivas Vema:

Multimedia Survey of Ancient Babylonian Civilization

Carl Eckberg, Chair
Department of Computer Science

Robert Edwards
Department of Computer Science

Michael O’Sullivan
Department of Mathematics and Statistics

Feb 10, 2016
Approval Date
DEDICATION

I would like to dedicate this thesis to all my family members [Vardhani Vema, Rama Krishna Rao Vema, Bala Tripura Sundari Ch, Venkata Lakshmi Ch, Surya Prabha, Lakshmi Swetha Vema, Harika Pingali, Sai Dhruvanth D, Raja Sekhar DVKA] and friends [Kappagantula Aditya, Deepak MV, Bharat S, Varun P, Santosh Varada] who always supported, encouraged and stood by me for accomplishing my work. I would also like to dedicate this thesis to my Professor and Advisor Dr. Carl Eckberg who supported me throughout my thesis and graduate studies at San Diego State University.

A special mention to my mother Vardhani and aunt Sundari. They are my inspiration, guidance and role-models in my career.

A special mention to Kappagantula Aditya, my fellow Masters aspirant, for supporting and guiding me to the right path in building my career as a Web/Salesforce developer.
ABSTRACT OF THE THESIS

Multimedia Survey of Ancient Babylonian Civilization

by

Satya Srinivas Vema

Master of Science in Computer Science

San Diego State University, 2016

The motivation of this thesis is to develop an interactive GIS application which should enhance student activity and interest in learning History of Ancient Babylonia. This tool could be a geographic computer interactive and user friendly tool which could be supplement to the text book reading.

This thesis focuses on creating a GIS multimedia teaching tool incorporated with respective web pages to learn more about Babylonian Civilization which endured from the 18th until the 6th century BC, and was like the Sumerian era that preceded it. This tool provides interactive graphical information about cities, battle fields, and monuments built at the time of each King’s rule, as well as tombs of significant personnel at that specific time and other important places. Users can switch between different Kings/Kingdoms and learn about them with ease. Once a King’s layer (layer in the map) is selected, it could come up with points on it. And clicking on each point would open up a webpage which contains detailed information of the King’s empire with famous locations. Points might be capital/monument/battle field/any other valued place.

Apart from the information about Ancient Babylonian Civilization, students can also customize the tool to select layers of their choice, zoom-in, zoom-out, print command, map tips, distance between points and query builder. This tool was developed using GIS technologies like Map objects – Java Edition and j2sdk. Eclipse as the IDE to develop this tool.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>v</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>vii</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>x</td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2 TECHNOLOGY</td>
<td>3</td>
</tr>
<tr>
<td>3 REQUIREMENTS</td>
<td>8</td>
</tr>
<tr>
<td>4 SOFTWARE PROTOTYPING</td>
<td>11</td>
</tr>
<tr>
<td>5 MOJO TOOLBAR</td>
<td>13</td>
</tr>
<tr>
<td>6 CUSTOM TOOLBAR</td>
<td>15</td>
</tr>
<tr>
<td>7 ECLIPSE IDE AND MOJO CONFIGURATION</td>
<td>19</td>
</tr>
</tbody>
</table>
8 OVERVIEW OF THE APPLICATION ............................................................21
9 SUMMARY ..................................................................................................30
  9.1 Difficulties ............................................................................................30
  9.2 Future Scope .........................................................................................30
REFERENCES ................................................................................................32
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>HTML DOM tree of objects</td>
<td>6</td>
</tr>
<tr>
<td>4.1</td>
<td>Prototyping</td>
<td>12</td>
</tr>
<tr>
<td>5.1</td>
<td>Selection ToolBar</td>
<td>13</td>
</tr>
<tr>
<td>5.2</td>
<td>ZoomPanToolBar</td>
<td>14</td>
</tr>
<tr>
<td>6.1</td>
<td>Custom ToolBar</td>
<td>15</td>
</tr>
<tr>
<td>6.2</td>
<td>Hotlink</td>
<td>15</td>
</tr>
<tr>
<td>6.3</td>
<td>Timeline tool</td>
<td>16</td>
</tr>
<tr>
<td>6.4</td>
<td>Print options screen</td>
<td>17</td>
</tr>
<tr>
<td>6.5</td>
<td>Add layer dialog window with Browse option</td>
<td>17</td>
</tr>
<tr>
<td>6.6</td>
<td>Pointer</td>
<td>18</td>
</tr>
<tr>
<td>6.7</td>
<td>Distance tool</td>
<td>18</td>
</tr>
<tr>
<td>6.8</td>
<td>XY Tool’s pop-up window with Browse option</td>
<td>18</td>
</tr>
<tr>
<td>7.1</td>
<td>Selecting Properties option after right-clicking project’s title</td>
<td>18</td>
</tr>
<tr>
<td>7.2</td>
<td>Adding External Jar files: Java Build Path &gt;&gt; Add External JARs</td>
<td>19</td>
</tr>
<tr>
<td>8.1</td>
<td>After executing the jar file, this map will be displayed</td>
<td>20</td>
</tr>
<tr>
<td>8.2</td>
<td>Clicking on XY button will open up the above Dialog window</td>
<td>21</td>
</tr>
<tr>
<td>8.3</td>
<td>Adding a layer on the map with Browse option to local storage</td>
<td>22</td>
</tr>
<tr>
<td>8.4</td>
<td>Identify tool showing up pop-up window</td>
<td>22</td>
</tr>
<tr>
<td>8.5</td>
<td>Distance toolbar with measurements</td>
<td>23</td>
</tr>
<tr>
<td>8.6</td>
<td>Query builder with the dialog window</td>
<td>24</td>
</tr>
<tr>
<td>8.7</td>
<td>Select Features tool displaying what shapes it will offer to select</td>
<td>25</td>
</tr>
<tr>
<td>8.8</td>
<td>Timeline button, clicking on it launches Timeline website</td>
<td>25</td>
</tr>
<tr>
<td>8.9</td>
<td>Timeline website with all dynasties</td>
<td>26</td>
</tr>
<tr>
<td>8.10</td>
<td>Hotlink tool, clicking on it enables clicking capability on the map</td>
<td>27</td>
</tr>
<tr>
<td>8.11</td>
<td>King’s web page with all the views on left</td>
<td>27</td>
</tr>
</tbody>
</table>
Figure 8.12. Compressed navigation bar on the left hand side of the screen. ..........................28
Figure 8.13. Opening up the navigation after clicking on the navigation button. ......................29
ACKNOWLEDGEMENTS

I would like to extend my heartfelt thanks to my advisor and Thesis chair person Professor Dr. Carl Eckberg, Computer Science Department, San Diego State University, for giving me an opportunity to work on this thesis and for the continuous support of my study and research.

I would also like to express my sincere thanks to Dr. Rob Edwards, Professor, Computer Science Department, San Diego State University for the kind support and his presence on my thesis committee.

Special thanks to Dr. Michael E. O'Sullivan, Professor and Chair, Department of Mathematics and Statistics, San Diego State University for being on my thesis committee and co-operation throughout.
CHAPTER 1

INTRODUCTION

1.1 Overview

As the Information technology’s growth exponentially increased from the past two decades, use of computer/mobile devices/tablets has been reached to a standard that they became part of everyone’s life. Learning standards were also changed drastically. People feel comfortable in learning on a mobile device/computer-based system rather than class-room teaching. This GIS application is intended for study purpose, more conveniently with all the information provided from different sources at one’s hand-clicks.

The objective of this thesis application is to provide an interactive learning GIS tool for history students who are looking to learn about Babylonian Civilization, especially from 18th century BCE to 6th Century BCE for which very little information was documented at any single source on the Internet as well as books. This tool provides all the interactive tools which enable the user to find more information at one place along with ease of interaction [1].

This application is developed by the combination of Java technologies and Map Objects Java Edition which is provided by Environmental Systems Research Institute (ESRI) as an open-source on their official website [2]. Thus GIS (Geographic Information System) developers can find it easy to get the tool and build interactive GIS Map applications. Java technologies like Java Swing, JFrame, JButton, JPanel and others were used as a part of development. Toolbars like ZoomPanToolBar and SelectionToolBar were utilized from Map Objects Java Edition to make the GIS application more interactive and powerful.

All the information about the Babylonian civilization was documented and presented as Web pages. Two types of web applications were developed as a part of this. First, a timeline website, which shows all the Babylonian dynasties starting from the First Dynasty of Babylonia (Amorite Dynasty) in 1830 BCE to the Dynasty XI of Babylon (Neo-Babylonian
or Chaldean Dynasty) till 539 BCE [3]. This website provides a list of all the kings that ruled in each dynasty with their reigning period, some short information about his rule/monumental constructions and an appropriate image. Second, a web page for each of the most famous and popular kings reigned in the timeline. These web pages are very responsive and mobile-friendly as it is developed using HTML Bootstrap, JavaScript and CSS which can be more adaptive in any short display devices.

1.2 Motivation

Babylonian Civilization is one of the world’s Ancient river valley civilizations near the banks of the Tigris River in Iraq. Almost all the river valley civilizations are famous for one or the other aspect at which they are good. Babylonian civilization is most famous for the “Code of Laws” written by King Hammurabi at his time of reign dated back to about 1754 BC. This code of Hammurabi consists of 282 laws that are inscribed on 12 tablets. Also Babylonian civilization is famous for “Hanging Gardens of Babylon” which is one of the seven wonders of the Ancient World. King Nebuchadnezzar II constructed this monument for his wife, Queen Amyitis as she missed her homeland which consists of green hills and valleys.

Other Ancient river valley civilizations include the Egyptian civilization along the Nile river and the Harappan civilization along the Indus river in India and Pakistan [4]. A similar application was developed as a thesis for the Indus Valley in consultation with an SDSU history professor, and this project was designed to make all these topics available at a single location: www-rohan.sdsu.edu/~eckberg

Dr. Carl Eckberg from the San Diego State University, Computer Science Department guided me in all the stages of application development and motivated me to build a GIS application based on Ancient Babylonian Civilization.
CHAPTER 2

TECHNOLOGY

2.1 Java

Java was developed by Sun Microsystems in 1995 to be an object-oriented, platform-independent, class-based, secure, fast, robust and high-level programming language. It allows application developers to “Write once, run anywhere”, which means that once Java source code gets compiled, it will produce Java bytecode, which can run on any Java Virtual Machine (JVM) irrespective of its computer architecture.

Various Java features which make Java a more powerful programming language than others are as follows: [5]

- **Secure**: Java has its own mechanism for internal management of memory allocation rather than using pointers (C++ language uses Pointers) which grants access to the data for the program to access if and only if the program has appropriate verified authorization.

- **Object-Oriented**: Java is a pure Object-oriented programming language as it is structured around defining classes with different methods in it, which can be accessed by relative objects. Important concepts of OOP: Inheritance, Polymorphism, Abstraction and Encapsulation

- **Simple**: Coding in Java is easier to write, compile, interpret and debug as it allows automatic memory allocation and garbage collection.

- **Platform-Independent**: As mentioned before, Java’s bytecode enables Java to run on different kinds of environments by having a Java Virtual Machine on them.

- **Multi Threading**: Java provides Multi threading by performing many tasks simultaneously within a program to achieve a certain output. All the required resources for performing Multi threading were provided by Java in a timely manner.

- **Robust**: Java gives more importance at finding errors in the code at the time of compiling, before making the code as executable, so that it will be more reliable and does not crash the entire system.
• **Interpreter:** Bytecode is generated by compiling Java code from the Java Virtual Machine. This bytecode is independent of the machine and can run on any machine that has the Java interpreter.

### 2.2 Map Objects Java Edition

Map Objects Java Edition can be used to create custom-built applications that integrate GIS and mapping capabilities or to enhance the capabilities of existing applications. Map Objects Java Edition enables developers to build Server or Client side GIS or mapping applications without any other ESRI tool or software. It consists of pure Java mapping APIs (Application Programming Interface) which are accessible in the form of Jar files. Jar files provide a number of classes/methods/functions that are essential for manipulating geographical data and querying the spatial information without intervention of any other Software/toolkit. It has more precise control over the behavior of the application; a user can build the application by creating essential components and by adding only those features that he requires; a Geographic Information System (GIS) class at San Diego State University covers Map Objects in depth and has a published material named Notes On Map Objects Java Edition by Dr. Carl Eckberg, which was one of the main criteria for choosing Map Objects [2].

Important features of the Map Objects Java Edition are as follows:

• By using the GIS oriented Swing components which are included in Map Objects Java Edition, you can quickly develop applications that include dynamic symbol control, functional toolbars, query dialogs, Pop up windows, overview and insert maps, intelligent legends, alert boxes, and Java panes which have the ability to display multiple formats of data that make your custom applications easy to develop and even easier to use and understand.

• Map Object Java Edition has a class called AttributeTable class, which displays the entire attribute table for a layer. But by implementing the Java JTable class, much more versatile attribute tables can be created. In general, the embedding of Map Objects Java Objects (MOJO) in Java allows for almost unlimited customizations.

• Legend Editor in Map Object Java Edition allows layers to be displayed in transparent and opaque format. It also has polygon options like Horizontal Fill, and Vertical Fill. These options make it possible to show several polygon layers at the same time because they are transparent with distinguishable patterns other than color.

• Map Objects Java Objects provides predefined toolbars such as “Selection toolbar” for selecting, searching and finding a region of a map and Zoom Pan
toolbar for Zooming in/out, Zoom to full-extent, Panning and Identify feature in a map. Custom toolbars can also be integrated into the application by using Java GUI toolkit.

- Map Object Java Edition has a method to add layers using the GUI, a standard feature in a dynamic interactive map. And also we can remove layers from the map. We can also implement methods like, Promoting/demoting a layer from Table Of Contents (TOC).

### 2.3 Web Technologies

#### 2.3.1 HTML and CSS

HTML is an acronym for Hypertext Markup Language, which is a standard markup language in building web pages with the file extension as ‘html’ or ‘htm’. The building blocks for websites are the ‘Html Elements’ which consists of tags that are enclosed in angle brackets (for an instance, `<head>`). Almost all Html elements will have a start-tag and an end-tag, having the content in between them. Browsers use these tags to interpret the content of the page but will not display them [6].

A few important Html Tags are as follows:

- `<html> </html>`
- `<head></head>`
- `<body></body>`
- `<p></p>`
- `<a href="WebsiteUrl">Click here</a>`
- `<h1></h1>`
- `<span></span>`
- `<div></div>`
- `<img src="ImageUrl" />`
- `<font></font>`
- `<style></style>`

CSS is an acronym for Cascading Style Sheets. It is a style sheet language which is used to describe the presentation of a web document written in a Markup language (HTML). CSS enables the separation of document content from document appearance, such as page layout, colors, font-styles and text representation. Separation makes improvement to the content accessibility and allows more flexibility in the specification of appearance characteristics. CSS makes it possible to separate presentation instructions from the Markup language content in a separate file (file extension as .css) or style section of the content file.
2.3.2 JavaScript

JavaScript is a lightweight, dynamic, interpreted, scripting, high-level programming language. It adds elements to the web pages that are interactive and dynamic in functionality. Syntaxes used in JavaScript are similar to C programming language, even though influenced by Java. As mentioned above, JavaScript is a scripting language, at the client side. JavaScript source code is processed at client’s web browser. The JavaScript code can show an error message based on the criteria mentioned before any data is actually transmitted to the server [6, 7].

JavaScript functions can be accessed within start tag <script> and end tag </script>) or when specific events takes place. Examples include onFocus, onSubmit, onClick, onSave and many more. JavaScript code can be embedded directly in HTML code, as well as by making all JavaScript code in a single file (.js file extension) and including it in the Script element.

![HTML DOM tree of objects](image)

**Figure 2.1. HTML DOM tree of objects [7].**

2.3.3 jQuery

jQuery is a cross-platform, Java DOM manipulation library (Document Object Model i.e., DOM) which is designed to simplify the complexity of client-side scripting of HTML. jQuery simplifies the syntax for searching, selecting, and manipulating these elements. It is also used for event handling that goes beyond general DOM element selection and manipulation. The DOM is the representation of all the elements of a Web page in a tree-
structure representation. For an instance, `<button>` tag generates a Button object in the DOM [6, 7].

The jQuery library is a single large JavaScript file containing all of its common DOM events, effects, and Ajax functions. This file can be included within a Web page by having a link to a local copy or just referring to one of the copies available from different public servers. jQuery also can be used to perform animations and include AJAX interactions to the Web pages.

### 2.3.4 Bootstrap

Bootstrap is a front-end framework based on JS framework, HTML and CSS for developing responsive, mobile-first projects on the web. It makes User Interface development faster and convenient. Bootstrap makes it easier for developing dynamic websites and web applications. We can get Bootstrap on to our website from the following two ways [6, 8].

1. By just downloading Bootstrap from getbootstrap.com. We can find the latest versions of Bootstrap on this website.
2. Or by just including Bootstrap from a Content Delivery Network (CDN). CDN support for Bootstrap's CSS and JavaScript is provided by MaxCDN.

For CSS: `<link rel="stylesheet" href = "http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" >`

For JavaScript:
`<script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js" >`
`</script>`

Bootstrap contains some commonly used interface elements in addition to the regular HTML elements. Examples include buttons with advanced features such as navigation, horizontal and vertical tabs, pagination, buttons with drop-down lists and etc., labels, thumbnails, typographic capabilities and progress bar.
CHAPTER 3

REQUIREMENTS

The main objective of this interactive GIS application is to help users/students to find a more interesting and efficient application about Ancient Babylonian Civilization from 18\textsuperscript{th} Century BCE to 6\textsuperscript{th} Century BCE. The thesis application has been developed under the guidance of Dr. Carl Eckberg from the San Diego State University, Computer Science Department and he was the major contributor of the requirements for this thesis. The requirements gathered have been classified as follows:

- Data Requirements
- Platform Requirements
- Functional Requirements

3.1 DATA REQUIREMENTS

The data that is required to be collected to represent it in the tool is termed as Data requirements. Dr. Carl Eckberg helped in identifying the important information of Babylonian dynasties and kings that is going to be displayed as layers on the application. The data includes the geographic coordinates of the places, the significance of the location; images related to that place/king/dynasty, constructional information of monuments built at those places, war causes/sequences, literature, art and languages spoken by the people at that time of the civilization [9, 10].

The most prominent kings of Babylonian civilization (kings including Kassites and Assyria who were also the part of Babylonian civilization, from wikipedia.com) are as follows [11]:

- Hammurabi - Reigned between 1792 – 1750 BCE [3, 12]
- Nebuchadnezzar_II - Reigned between 605 – 562 BCE
- Kurigalzu_I - Died 1375 BCE
3.2 Platform Requirements

Platform requirements are categorized by what programming language is used to build the application, what programming editor (or) Integrated Development Environment (IDE) used at developing and what kind of Operating system is used. I have used the following software platforms:

- Java (Core Java, Swing, JFrames, JPanels and other Java tools) as a Programming language.
- Map Objects Java Edition as the GIS API
- Eclipse IDE as an IDE
- Windows 10 as an Operating System

3.3 Functional Requirements

Functional requirements refer to functional details of the application along with GUI specifications. Many of the GUI specifications were included in the GIS application itself in the form of toolbars. Some of the important Functional requirements are as follows:

The application,

- Must be capable of providing Zoom-in, Zoom-out, Pan and Pan-in (four directions).
- When executed, should display a base map of the Middle East countries.
- Must have an in-built Print option/button.
- Should have an Identification tool, which helps in identify different places on the map layer.
- Should be capable of Adding/Deleteing layers from the Table of Contents (TOC).
- Should be able to Promote/Demote the layers from TOC.
- Must possess a “Hotlink” button, which should enable “click-on-point” to an external web page, which displays more information about that point.
- Must have a “Timeline” button, on click should open up a Timeline website of Babylonian dynasties dating from 18th century.
- Must have a Help menu-item, which should guide the user with all the GIS application functionalities.
• Should be able to import external .csv files containing the coordinates, which are displayed upon the map.
CHAPTER 4

SOFTWARE PROTOTYPING

Prototyping is the essential aspect of developing any software application. It makes a blue print of the end-product including design aspects, functional aspects and maintenance aspects. This prototyping gives you an idea of how the flow of controls works as a single entity. Software prototype serves as a preliminary model which is refined along the Software development life cycle to meet the initially documented end-user requirements.

Below are the important phases of any Software application life cycle:

- **Requirements**: Requirements gathering is the initial stage of any Software development life cycle. It consists of data requirements, platform requirements and functional requirements. All the essential requirements are documented.

- **Design**: In this stage, we will analyze the documented requirements, and create a basic design of the application, in terms of its functionality, features to be included and interface layouts.

- **Development**: Actual coding will begins in this stage. All the functional requirements are implemented through programming to get the desired output.

- **Verification**: In this stage, all the developed programming code is tested for its desired working abilities. Integration of all the small pieces of code in making a complete working application and conducting testing to validate the actual output to expected output. Thus finding flaws of overall application and fixing those.

- **Maintenance**: After the deployment of the product, it is the responsibility of the developer to maintain it as per the future needs and advancements in the technologies used in building it. As this is a thesis application, it is out of scope.
Figure 4.1. Prototyping [13].
CHAPTER 5

MOJO TOOLBAR

Map Objects Java Objects (MOJO) provides some pre-built standard tools that are used in many GIS applications. These tools are categorized into specific toolbars based on the generic functionality they perform. For instance, ZoomPanToolBar will have tools like: Zoom-in, Zoom-out, Zoom-to-full-extent and some other relevant tools. The toolbars which are implemented in this application are:

- SelectionToolBar: com.esri.mo2.ui.tb.SelectionToolBar
- ZoomPanToolBar: com.esri.mo2.ui.tb.ZoomPanToolBar

5.1 SELECTION TOOLBAR

As the name suggests, it consists of tools that perform selective operations. Such as Search, Find, Query builder and others. With the help of this tool bar, a user can Selection a layer, creation of new layers and shape files. Following are the tools that are included in SelectionToolBar:

- **Find**: To initiate dialog box for finding something in the layer, it opens up a dialog window. The dialog window can be provided by a single value. This will find the theme and highlight it in a yellow color on the map.

- **Search**: Opens a dialog for locating features based on a predefined “stored query”.

- **Query Builder**: Clicking on this will open up a dialog window to help the user in querying to the selected database, which can be used to display certain feature on the map.

- **Select Feature**: This tool helps in selecting a specific area on the map. It also has options for selecting the portion on the map in different shapes, such as Circle, Rectangle, Line, and Polygon.
- **Clear All Selection**: Clears all the above selected attributes on the map.
- **Buffer**: Clicking on this tool will create a buffer area of a selected distance around the selected layer.
- **Attributes**: Displays the attributes of the selected feature in a tabular format.

### 5.2 ZoomPan Toolbar

ZoomPanToolBar provides zooming and other relative zoom capabilities to the map. Every tool on this toolbar is subjected to perform a certain task such as zooming, panning, etc. Following are the tools that are included in ZoomPanToolBar.

**Figure 5.2. ZoomPanToolBar.**

- **Previous Extend**: This helps in zooming to previous display of the map that has been stored.
- **Next Extent**: This helps to zoom in the map to the next extent available.
- **Zoom to Active Layer**: This helps in zooming in to the feature selected on the map which is in active state.
- **Zoom to Full Extent**: This helps in zooming map to the extent of all layers within the map.
- **Zoom In**: This tool is used to zoom into a portion, made by click (or) drag a rectangle on the map.
- **Zoom Out**: This tool works by clicking or dragging a rectangle on the map in order to zoom out.
- **Pan**: This tool works by dragging the map to a new location without varying the zoom level.
- **Pan One Direction**: This tool works by dragging the map in any one direction (North, South, East, West)
- **Identify**: This tool is used to identify the selected layer. Also displays the features of that layer in an attribute table.
CHAPTER 6

CUSTOM TOOLBAR

Custom toolbar has been developed using Java Swing and awt API’s with MOJO methods to provide additional functionalities, thus making the GIS application more interactive and powerful. Additional tools include Hotlink, Distance and Importing CSV file with points to project on the map. As per this thesis application, following are the tools included in the Custom toolbar:

![Custom Toolbar](image)

**Figure 6.1. Custom ToolBar.**

- **Hotlink:** The main objective of Hotlink tool is to give user the ability to get custom and detailed information about the selected point of the selected layer in the form of a Web page. In this GIS application, Hotlink tool is represented as a lightning bolt icon for both the tool and cursor value. This tool works with the active layer of the map.

![Hotlink](image)

**Figure 6.2. Hotlink.**

When a user clicks on the Hotlink icon from toolbar, and clicks on any displayed point on the map, it triggers a custom created webpage containing all the information of the King along with other locations respective to that King’s rule.

This kind of application can be of value if it simply points to web sources like Wikipedia,

If it does so in an organized way, but to be really effective, information should be created by the application author, since that person can tailor the language and content to the target audience he/she has in mind. More copious or alternative treatments can then be made easy to find by optional web-links.
• **Timeline:** Timeline tool helps in launching the Timeline website of the complete Babylonian dynasties [3, 14] from 18th century BCE to 6th century BCE. This tool is made of a JButton and accessed using an action-listener to perform the respective action i.e., opening a website.

![Timeline](image)

**Figure 6.3. Timeline tool.**

List of the dynasties (Source: Wikipedia.com) [1] covered as a part of timeline:

- First Dynasty of Babylonia (Amorite Dynasty)
- Sealand Dynasty (Dynasty II of Babylon)
- Kassite Dynasty (Third Dynasty of Babylon)
- Dynasty IV of Babylon, from Isin (Second Dynasty of Isin)
- Dynasty V of Babylon (Second Sealand Dynasty)
- Dynasty VI of Babylon (Bit-Bazi Dynasty)
- Dynasty VII of Babylon (Elamite Dynasty)
- Dynasty VIII of Babylon
- Dynasty IX of Babylon (Dynasty of E)
- Dynasty X of Babylon (Assyrian)
- Dynasty XI of Babylon (Neo-Babylonian or Chaldean Dynasty)

Timelines are relatively new in the GIS field, but are now extremely popular. They allow the application user to quickly correlate time, location and events. Another way to say this is that the application should nicely combine spatial, temporal, and factual information.

• **Print:** This tool is used to print the map to an external connected printer (or) save as a PDF file in the local system. Clicking on the Print tool will display the following image.
• **Add Layer:** Clicking on this tool will open up a dialog window from which a user can navigate to a location on Local hard drive to add a Shape file (.shp file)/Image file/ARC IMS/ ARC SDE on to the Table of Contents (TOC).

![Add Layer dialog window with Browse option.](image)

Figure 6.5. Add layer dialog window with Browse option.

• **Pointer Tool:** This tool helps a user to revert back the cursor to a normal mouse-cursor-arrow-pointer from a particular tool i.e., remove the special behavior of the previous tool selected. MOJO class, com.esri.mo2.ui.bean.Tool used to add this Pointer tool to the Custom toolbar.
• **Distance Measuring Tool:** This tool helps in measuring the distance between any two points on the map (Miles/Kilometers). First click on the tool, then click on the start-point and drag the cursor till the desired end-point, then release the cursor. You will notice the distance between these two points at the bottom of the map, status bar.

• **XY Tool:** This tool gives the user the ability to create a Points layer dynamically from a selected CSV file, which contains the Longitude and Latitude of the locations that are to be displayed on the map. After clicking the XY button from the toolbar, it displays a Pop up window as shown in the below figure.

![XY Tool’s pop-up window with Browse option.](image-url)
CHAPTER 7

ECLIPSE IDE AND MOJO CONFIGURATION

This GIS application was developed using the Java platform. I have chosen Eclipse IDE (Integrated Development Environment) as a platform to edit, debug, compile and run the complete GIS Application. For GIS applications to run precisely on Eclipse IDE, we need to import the ESRI Jar libraries (esri_mo20.jar, esri_mo20img.jar, esri_mo20res.jar) from external Map Objects folder and make them available for compilation and execution of the complete application.

The following images show how to add external Jar files to the Eclipse projects:

Figure 7.1. Selecting Properties option after right-clicking project’s title.
Figure 7.2. Adding External Jar files: *Java Build Path >> Add External JARs.*
CHAPTER 8

OVERVIEW OF THE APPLICATION

An overview of the complete GIS application along with Timeline website and Web pages for each stated King will be discussed in this chapter. Each of the tools/website and their respective screenshot will be demonstrated as the flow of controls in the application.

- The initial screen of the application, after executing the Babylonia.jar file will appear as follows:

![Figure 8.1](image)

*Figure 8.1. After executing the jar file, this map will be displayed.*

- XY Tool for adding a CSV on to the map:
Figure 8.2. Clicking on XY button will open up the above Dialog window.

- Add Layer

Figure 8.3. Adding a layer on the map with Browse option to local storage.
• Identify tool from SelectionToolBar: Clicking on “Icon i” from ZoomPanToolBar will change the cursor to an Identify icon, then clicking on any portion of the map will pop-up a dialog window with attributes in it.

Figure 8.4. Identify tool showing up pop-up window.

• Distance tool from the Custom ToolBar:
Figure 8.5. Distance toolbar with measurements.

- Query Builder:

Figure 8.6. Query builder with the dialog window.

- Select Features tool from Selection ToolBar:
Figure 8.7. Select Features tool displaying what shapes it will offer to select.

- Timeline tool: Showing the ToolTipText when hovering with mouse pointer on it.

Figure 8.8. Timeline button, clicking on it launches Timeline website.
Timeline website showing all the eleven dynasties after clicking on the Timeline tool:

Figure 8.9. Timeline website with all dynasties.

- Hotlink tool:
Figure 8.10. Hotlink tool, clicking on it enables clicking capability on the map [15].

- Clicking on any of the displayed points on the map opens up the web page of respective King.

Figure 8.11. King’s web page with all the views on left.
• Responsiveness of the King’s web page:

Figure 8.12. Compressed navigation bar on the left hand side of the screen.
Figure 8.13. Opening up the navigation after clicking on the navigation button.
CHAPTER 9

SUMMARY

9.1 DIFFICULTIES

This thesis application provides an interactive GIS tool showcasing Ancient Babylonian Civilization from 18th to 6th century BCE. To display all the information of each dynasty, a Timeline tool has been developed with all the eleven dynasties along with all the rulers. Also a web page is developed for each famous king with all the information regarding his reign, battles, constructions, arts and literature. More responsive pagination was produced by using Bootstrap technology.

Babylonian Civilization was spread over a very large area in Iraq, Iran, Syria and few other Middle East nations, which made it quite difficult to get the appropriate and accurate information of king/dynasty. Also as this GIS application is based on very ancient civilization, dated back to 18th century, the information is not readily documented in a small number of sources.

And even though information was documented at the time of King’s rule, it got damaged or lost as the course of time. When some data was not available at a particular source or either internet or book which is authentic, it is quite difficult to get the accurate data even though we found it in different sources in different interpretations.

Also this complete Babylonian civilization was not wide spread throughout the Middle East, and it is almost along the banks of rivers Euphrates and Tigris, most of the same places were referred at different timelines of the different dynasties. Thus it for much complexity while creating Points layers since different points have almost no distance between them.

9.2 FUTURE SCOPE

This GIS application is built using Java technologies (JFrame, JPanel, JButton, Swing and etc.), Map Objects Java Edition and Web technologies (HTML, CSS, Bootstrap,
JavaScript, jQuery). This application can be further implemented using future technologies as we progress further into the Technological world. A few enhancements might be in future are as follows:

- Data can be integrated with JavaScript frameworks to look more responsive and intelligent.
- The GIS application tool can be translated into many languages (Especially into native language of Iraq) to provide access for users whose first language is not English.
- Can implement Google APIs into the application to make it more powerful and smart.
- This application can be implemented with dynamic maps like Google maps which will be more helpful to identify the current location more accurately.
- Can be created as a Web based application rather than just running a downloaded Jar file. This might make the application run over web browser itself without any downloading.
REFERENCES


