MSC Project & Dissertation IT Business

Client-side Web Technologies for Diagrammatical User Interfaces

Supervisor: Gudmund Grov

Jingying Niu
2013/8/13
DECLARATION

I, JINGYING NIU confirms that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed: ..................................................................
Date: ......................................................................
Acknowledgement

I would like to appreciate people who give me support and motivation throughout the project. First, I want to thank to my supervisor Gudmund Grov. He provides useful support and guide to my project. Then, I would thank my second reader Fairouz Kamareddine for judging my report and attending my poster presentation. My mother gives me courage and confidence to accomplish the study of postgraduate. In the end, I thank my friends( Xiemeng Bao and Yanchao Yu) for helping me overcome the obstacles of the project and supporting wonderful advices.
Abstract

This project surveys recent client side web technologies to create diagrammatical user interfaces. In 2000s, a large number of new types of GUI(Graphical User Interface (GUI)) has created, such as mobile interface, web interface and so on. Moreover, due to rapid development of client-side web technologies, it has opened up a new approach to represent Graphical User Interface (GUI)s. However, because different applications have different GUI requirements, it is hard to perform a general comparison between them. This project targets interfaces for genealogy tools as a specific area. Genealogy has a long history, and more and more people want to know their origin and conditions of unfamiliar family members. Hence, a large number of software and web sites for drawing family-trees have been developed. In this project, I explore basic requirements of genealogy tools through surveying different categories of existing tools at first. In the process of survey, basic requirements of genealogy tools are identified. Then, I investigate a set of web technologies to create Graphical User Interface (GUI)s of genealogy tools. The investigated technologies include JavaScript, PHP, JavaFX and Java Applet. Through surveying diverse web technologies used in GUI, I find JavaScript is a more suitable technology than others. Then, I have analysed the four most promising JavaScript libraries which seemed to practising diagrams. The analysed libraries cover JQuery, Dojo, mxGraph and Kinetic. In the end, I chose the two most practising approaches and created two prototypes. One prototype uses JQuery and Kinetic, and one only uses JQuery. These two prototypes were then evaluated based on how ease of use. The evaluation of two prototypes was conducted by a set of testers. Through evaluation, I recommend Kinetic is a more fitting technology for diagrammatical user interfaces.
# Table of Content

DECLARATION .................................................................................................................. I
ACKNOWLEDGEMENT ....................................................................................................... I
ABSTRACT ........................................................................................................................... II
1. INTRODUCTION ......................................................................................................... 3
2. OBJECTIVES .................................................................................................................. 4
3. PROFESSIONAL, LEGAL, ETHICAL AND SOCIAL ISSUES .................................. 5
   3.1 AUTHORITATIVE ISSUES ....................................................................................... 5
   3.2 PRIVACY ISSUES .................................................................................................. 5
   3.3 LEGAL ISSUES ...................................................................................................... 5
   3.4 ETHICAL AND SOCIAL ISSUES ........................................................................... 6
4. LITERATURE REVIEW ................................................................................................. 7
   4.1 GENEALOGY .......................................................................................................... 7
       4.1.1 Concept .......................................................................................................... 7
       4.1.2 Tools For Genealogy ...................................................................................... 8
   4.2 CLIENT-SIDE WEB TECHNOLOGY ..................................................................... 16
       4.2.1 HTML & CSS ................................................................................................ 17
       4.2.2 Web Technology .......................................................................................... 22
5. REQUIREMENT ANALYSIS ....................................................................................... 28
   5.1 OVERVIEW ............................................................................................................ 28
   5.2 FUNCTIONALITY REQUIREMENTS ...................................................................... 28
       5.2.1 Mandatory Requirements .............................................................................. 28
       5.2.2 Optional Requirements .................................................................................. 30
   5.3 WEB TECHNOLOGY REQUIREMENTS ............................................................... 31
6 A COMPARISON OF SELECTED TECHNOLOGIES .................................................. 33
   6.1 LIBRARIES ............................................................................................................ 33
       6.1.1 JQuery .............................................................................................................. 33
       6.1.2 Dojo .................................................................................................................. 40
       6.1.3 mxGraph ......................................................................................................... 42
       6.1.4 Kinetic .............................................................................................................. 47
   6.2 Comparison OF Libraries ...................................................................................... 52
7 PROTOTYPES ............................................................................................................... 60
   7.1 THE JQUERY PROTOTYPE .................................................................................. 60
       7.1.1 Create Function .............................................................................................. 61
       7.1.2 Edit Function .................................................................................................. 62
       7.1.3 Insert Function .............................................................................................. 63
7.1.4 Resize Function ................................................................. 63
7.1.5 Save Function ................................................................. 64
7.1.6 Summary ................................................................. 64
7.2 The Kinetic Prototype ...................................................... 65
  7.2.1 Create Function .......................................................... 67
  7.2.2 Edit Function ............................................................. 67
  7.2.3 Insert Function ............................................................ 68
  7.2.4 Resize Function ............................................................ 69
  7.2.5 Save Function .............................................................. 70
  7.2.6 Summary ................................................................. 70
7.3 Conclusion ................................................................. 71

8 Evaluation ........................................................................... 72
  8.1 Evaluation of Development Process .................................. 72
  8.2 Users Questionnaire ....................................................... 72
  8.3 Testers Evaluation ........................................................... 73
     8.3.1 Function of Creating movable items ......................... 74
     8.3.2 Editing Function ....................................................... 75
     8.3.3 Inserting Function .................................................... 77
     8.3.4 Resizing Function ...................................................... 78
  8.4 Summary ................................................................. 79

9 Conclusion ........................................................................... 80

Reference ............................................................................... 82

Appendix A ............................................................................ 86

Appendix B ............................................................................ 94
List of Tables

Table 1 New Features of HTML 5.0[13] ................................................................. 20
Table 2 Functions of Libraries ............................................................................. 53
Table 3 Comparison Library ................................................................................ 54
Table 4 Explanation of Rating JQuery ................................................................. 55
Table 5 Explanation of Rating Dojo .................................................................... 57
Table 6 Explanation of Rating mxGraph ............................................................... 58
Table 7 Explanation of Rating Kinetic ................................................................. 59
List of Figures

Figure 1 Family-tree [4] ................................................................. 8
Figure 2 Example of GEDCOM[6] ....................................................... 9
Figure 3 Pedigree (TNG)[7] ............................................................. 10
Figure 4 Timeline(TNG)[7] ............................................................... 11
Figure 5 Multimedia(TNG)[7] .......................................................... 11
Figure 6 Interface of Peditmap [8] ..................................................... 12
Figure 7 Bowtie Chart(MyHeritage)[9] .............................................. 13
Figure 8 Close-Family Chart(MyHeritage)[9] ..................................... 14
Figure 9 Ancestor Chart(MyHeritage)[9] .......................................... 14
Figure 10 Descendant Chart(MyHeritage)[9] ..................................... 14
Figure 11 Fan chart(MyHeritage)[9] ................................................ 15
Figure 12 Family-tree(MyHeritage)[9] .............................................. 15
Figure 13 3-Tier Architecture [10] .................................................. 17
Figure 14 HTML example[11] .......................................................... 18
Figure 15 CSS example [14] ............................................................. 21
Figure 16 Internal Style Sheet [14] ................................................... 21
Figure 17 External Style Sheet [14] .................................................... 22
Figure 18 PHP Interaction[19] .......................................................... 25
Figure 19 Use Case Diagram(mandatory) .......................................... 28
Figure 20 Example of Creating family members ................................ 29
Figure 21 Example Table ................................................................. 29
Figure 22 Use case Diagram(optional) .............................................. 30
Figure 23 Browser Support by JQuery [25] ......................................... 35
Figure 24 JQuery Add Function[28] ................................................... 35
Figure 25 JQuery Add Function Result[28] ....................................... 35
Figure 26 JQuery Draggable Function(code)[27] ................................ 36
Figure 27 JQuery Link Function(result)[28] ...................................... 36
Figure 28 JQuery Delete Function(code)[28] ...................................... 37
Figure 29 JQuery Delete Function(result)[28] .................................... 37
Figure 30 JQuery Resize Function[27] ............................................... 38
Figure 31 JQuery Input Function(code)[27] ......................................... 39
Figure 32 JQuery Input Function(result)[27] ..................................... 39
Figure 33 The structure of Dojo[31] .................................................. 40
Figure 34 Create Function(Dojo)[29] ................................................ 41
Figure 35 Insert Function(Dojo code)[32] ......................................... 41
Figure 36 Insert Function(Dojo Result)[32] ....................................... 42
Figure 37 Create New Family Members(mxGraph)[34] ........................ 44
1. Introduction

In recent years, web applications have increased considerably. People could make full use of web applications due to the rapid development of Wi-Fi and 3G. For example, people could search web sites, read real-time news and play games in a bus, hotel, train station and so on. This has led to the boost development of new web technologies.

Genealogy (family-tree) applications are new graphical applications which allow users to create and modify their family trees. People may would like to know their ancestors, situation of unfamiliar family members and components of their family. Tools of genealogy provide a variety of functions to fulfil the diverse demands of people, such as showing condition of every family member, adding family members, changing association among family members and such like. Genealogy is a relatively typical type of diagrams, and it provides the common features to capture. For example, the tree structure is one of diagrammatical structures, the connections among objects are the essentials in diagrams and so on. What's more, there are not any client-side interfaces of genealogy tools. The applications of genealogy are mainly websites and software.

Client-side web technologies have opened a new way to present Graphical User Interface (GUI)s. Therefore, our goal is to analyse client-side web technology for diagrammatical user interfaces, and genealogy provides an ideal area to perform such analysis. The key requirements of genealogy tools become the requirements in analysis. A series of technologies have been surveyed, and they include JavaScript, PHP, JavaFX and Java Applet. JavaScript is the most widely used scripting language, while PHP is the most popular technologies in web pages. JavaFX is able to enhance dynamic web pages, and Java Applet is a common technology which is build on Java. After survey, the most practising technology will be deeply analysed based on the requirements of genealogy tools. Furthermore, two prototypes are developed for the most practising technology.

This report is divided into nine parts. The next section is objectives, which summarize the goals of the project. Following on, related professional, legal, ethical and social issues is discussed. After this section, the literature review introduces related survey of the project, namely genealogy and
client-side web technology. Requirement analysis is the fifth part, which talks about the requirements of the project. Then, the section 6 analyses selected technologies. The two prototypes are discussed in section 7, and they are about how to implement two interfaces of family trees. In section 8, the prototypes are evaluated before I conclude in section 9.

2. Objectives

The main objective of this project is to analyse client-side web technology applied to diagrammatical user interfaces. To achieve this, the project has three parts:

1. The project will identify key requirements of genealogy tools by surveying different kinds of existing tools. The project will then investigate a range of current client-side technologies for diagrammatical user interfaces. The key requirements of genealogy tools are identified as the clues of survey. Following on, the project will deeply analyse the fitting technologies.

2. Two prototypes of family trees interfaces are developed for the most practising approach. One prototype is finished by the most reasonable technology, and the other is implemented by the relatively suitable technology.

3. Evaluation is used to judge how easy to perform the functions of two prototypes. The evaluation consists of four parts. They are whether two prototypes meet the requirements of interfaces of genealogy tools, how easy to use two prototypes, the user experience of two prototypes and the discussion concerned with the evaluation.
3. Professional, Legal, Ethical and Social Issues

3.1 Authoritative Issues

This project aims at analysing client side web technologies for Graphical User Interface (GUI), therefore, this project will involve judgement concerned with web technologies. Depended on the process of the project, this assessment is divided into two parts. One is an assessment based on professional publications, websites and journals. The other is a judgement depended on practical performance of web technologies. The first one is relatively objective, because it collects a variety of viewpoints for a large number of professional materials. The second judgement may be bias because practical performance of web technologies is based on built prototype. This prototype is set up by me, and it is possible that I could not make full use of related web technologies. This will reduce corresponding usability and functionality of web technologies to some extent. This will lead to the judgement of web technologies is biased as well. Although I will evaluate practical performance of web technologies based on the ideas of tester, the judgement of web technologies is still subjective. It just stands for my opinion, and it lacks authority.

3.2 Privacy Issues

In this project, there will have a questionnaire after accomplishing the prototype. In the procedure of collecting viewpoints, all questionnaires stand for the opinions of tester. I should save these information in safe place, and avoid them to be revealed. Meanwhile, I could not make use of theses information to obtain interest. After I employ these information, I need to delete them as soon as possible in order to reveal these information.

3.3 Legal Issues

In the analysed section of this project, I will use some code and examples from public websites and books. These referenced code and example in analysis are open to public. I would obey the copyright of these code and examples. Besides, I will apply some technologies after I chosen the most suitable technologies. These technologies are applied to accomplish prototype. Some technologies may be
open source, but others may be licensed. For example, JQuery is free to use, and mxGraph is licensed. If the most suitable technologies are licensed, I would give up because I should obey the copyright of technologies. Besides this, I may utilize some software to build prototype. Some software are probably open source tool, such as Notepad. Some useful software are not free to use, such as Dreamweaver. Therefore, this project may involve the copyright of technologies and software.

3.4 Ethical and Social Issues

In this project, the prototype is used to draw family-trees. Family-trees should include all family members, but there are some relations which are difficult to express. In some African countries, a man could have more than one wife. This behaviour is accepted by law in local. However, this behaviour cannot be allowed in many countries. Except this, the phenomenon of illegitimate children exists in the society. The status of illegitimate children is the same as the legitimate children. Moreover, a man may have several ex-wives. Sex marriage is more and more popular in recent years. There are a large number of issues concerned with relations of family, and this project may involve these issues. All above introduced conditions are allowed in the prototype of this project.
4. Literature review

In recent years, the development of computer programs has grown rapidly, especially in web sites and supported software. For example, e-commerce websites (Amazon), diagram websites (Cacoo), Adobe Dreamweaver and such like. This section covers two sub-sections, which are genealogy and web technologies. In genealogy, it will introduce concept and tools of genealogy. Here, I will identify which elements are applied for Graphical User Interface (GUI)s, and technologies which may apply in such a Graphical User Interface (GUI).

4.1 Genealogy

4.1.1 Concept

For the definition of genealogy, there are various viewpoints. Because genealogy could show history of descendant of families, pedigree, a particular individual and a family, it is regarded as "The Chambers Dictionary" [1]. Nevertheless, Prinke thought this definition was too simple, and he considered "there is no generally accepted idea of what genealogy is". His views is not accepted by the majority of genealogist [2]. Hoffman is a genealogist, and he defined genealogy as [3]“Let’s record the answers to questions like these: What was their body size and shape? How did they travel and why? Where did they live throughout their lives? How did they view the world, both the physical world as well as spiritual matters?” His idea is a typical viewpoint in genealogy. As a result, there is no confirmed definition concerned with genealogy.

In genealogy, a variety of complicated relationship need to be included. Hoffman [3] claimed a genealogy system just presents children, spouse, parents and ancestors is not enough due to real life families are more complex. In genealogy system, some relationships that are important to history of families are hard to represent. The relationships consist of intergenerational relationships, step-relationships and non familiar relationships. Owing to relatives and blood-line are just the beginning of people , and the influence of some individuals that are outside families is gradually increasing after people are born. Thus, these relationships have become more and more significant.
Diagrams used in genealogy have a long history, and they are often called family-trees. It is one of the earliest diagrams in the world. At the beginning, a family-trees [4] was decorated in the atria of cottages which belonged to roman noble. However, no examples survived. The earliest survived genealogical graph could date from the Middle Ages. Three examples are shown in Figure 1, which shows Noah’s descendants from 11th century at the top of the figure. At the bottom left of the figure, there is another confirmed genealogy. Besides, family-trees of aristocracy were popular based on the record of biblical genealogies. A 12th-century family-trees of the Saxon dynasty is displayed in the bottom right. Genealogy did not only exist in the nobleman and religion, but it was also used in documents of law in ancient time.

![Figure 1 Family-tree [4]](image)

4.1.2 Tools For Genealogy

Here, I show three common tools for genealogy. They all support GEDCOM (genealogical data communication), which is a format of the genealogical data. The example of GEDCOM is shown in Figure 2. It enables genealogical data to flow in different genealogy applications [5]. Three different
types of tools of genealogy are introduced below.

1. The Next Generation of Genealogy Sitebuilding is a web service, which works in a server.
2. Pedimap is a stand-alone application, and it is mainly applied in biology.
3. MyHeritage is a website, and it also includes standalone software version (FamilytreeBuilder).

4.1.2.1 The Next Generation of Genealogy Sitebuilding

The Next Generation of Genealogy Sitebuilding (TNG) [7] is an useful approach to manage and display family-trees in users' computers. It is a web application which runs in a server. Users store information of family members in a database, and display the family-tree though PHP. It does not need any static HTML files, meanwhile, it is a cross-platform application. However, it relays upon a PHP server and MySQL database being installed locally. This is a drawback of this application, while,
PHP server and MySQL database are common sense in computer science, not all individuals are familiar with them and few are likely to have it installed. The majority of people even do not know the meaning of PHP and MySQL. Except this, there is another limitation of TNG. Although users could use TNG without accessing websites or downloading software, users need to have expenditure for TNG. There are a large number of free websites on the web, therefore, most people may not want to pay for it. For this limitation, TNG is popular for business websites.

Even though TNG has its drawbacks, powerful features of TNG are made up core of them. Primarily, it supports [7] GEDCOM, and this means users could import and export family-trees without retyping. After this, uses could search family-trees by name and TNG sustains the advanced search. For example, users could explore family-trees based on a specific date or birth place. Besides, TNG maintain diverse dynamic charts, such as pedigree(see Figure 3), family group and timeline(see Figure 4). Users could filter family members based on their hobby. TNG enables users to customize layout of interfaces. Users could select one from templates , and they could set it by themselves if they do not like the templates. Moreover, TNG allows multimedia, such as videos, photographs and histories(see Figure 5). A TNG protects various right of users. No one would like to reveal private information. Ultimately, TNG still has other features, but above features are its primary characters .

![Figure 3 Pedigree (TNG)[7]](image-url)
4.1.2.2 Peditmap

Peditmap [8] is a user-friendly software tool, which is mainly applied in biology. In addition to show family-trees, it is able to visualize "genetic and phenotypic Data" in pedigrees. Pedigrees are the relevant records of blood linkage of a person’s family. Genetic data consists of "marker scores, Identity-by-Descent probabilities, and marker linkage map positions". Through analyzing linkage in
pedigrees, user could make a decision about breeding. There is a series of genealogical tools like Peditmap in biology, such as HaploPainter, PediGrach, Progeny and so on. These tools are not only to present family-trees, but they are able to perform biological experiments. For example, they could calculate breeding coefficients and store breeding information in a database. Compared with other tools applied in biology, Peditmap has a unique feature: It can mix user-defined phenotypic traits with genetic marker data in order to flexible plant pedigrees. Selfing, cloned and doubled haploid production are all included in condition of pedigrees. In Peditmap, users could select one part of family-trees, and show the same part in different view automatically. Figure 6 shows an interface of running Peditmap.

![Figure 6 Interface of Peditmap](image)

4.1.2.3 MyHeritage

MyHeritage [9] is a powerful web social network service and genealogy website, meanwhile, it supplies two versions for users. One is to provide basic functions, and is free to use. The premium version is able to supply more functions, consequently, it requires users to pay a small amount of money. MyHeritage is the most popular website concerned with genealogy in the world, and it has 64 million users and 1 billion online files. It has become the most common genealogy website, due to its features. At the beginning, it provides DNA tests aimed at helping users to find unfamiliar relatives.
Because some relatives may have lost respective contacts ways, some relatives are hard to recognize unfamiliar family members even they have met each other. However, users need to pay for this service and they could choose diverse types of DNA tests based on their demands. Then, it enables users to tag the face of people in photos and upload tagged photos into their family-tree. Users could select any types of photos, then give a tag for them. Besides, MyHeritage supports smart search and record search. Smart search could let users search by name, and record search enables users to search historical data about people in genealogy. Following on, MyHeritage supplies varied kinds of charts and chart style for users to select. For example, bowtie, close family, ancestors, descendants hourglass and fan chart are included types in charts. Each type has its unique characters, and they are discussed below [9].

- A bowtie chart displays a primary individual at the centre and spouse are next to the main person. Their ancestors are on either side, and their children are below their parents. A unique feature of this kind is its fine proportions and symmetry. Figure 7 is an case.

![Bowtie Chart(MyHeritage)](image)

*A Figure 7 Bowtie Chart(MyHeritage)[9]*

A close family chart presents grandparents of a "main person", and all the descendants of his grandparents. This type shows all cousins of a "main person", and it is convenient to show it to relatives. Figure 8 shows this type.
Ancestor chart shows all the ancestors of a primary individual. This kind is mainly used in sharing with children.

A descendant chart shows all the descendants of a main person. This type is mainly used in sharing with relatives who has the same ancestor.

A hourglass chart combines the ancestor chart with descendant chart.

A fan chart shows all the ancestors of a primary person in fan style (see figure 11). However, this type does not add photos to members in family-trees.
The above features are the unique features of MyHeritage, and MyHeritage also have the same functions as others, such as timeline, showing condition of family members , import family-trees in GEDCOM format and so on. Figure 12 is an example of family-trees when users utilize this website to build family-trees.

FamilytreeBuilder [9] is the standalone version of MyHeritage, and it has the same feature as MyHeritage. Users could decide which version they would like based on their needs. Although Wi-Fi and 3G are available in most places, there are some places that could not connect network. For example, underground, mountain and so on.
4.1.2.4 Features Summary

Here, I summarize some key and optional features which most genealogy tools provide. Key features are common features that tools for genealogy own, while optional features are unique features that respective tool has. Key features are:

1. Users can add, modify and delete family members and relationship in family-tree.
2. Users can see the basic personal information and the relationship between a person and others in family-trees when they do some actions.
3. Users can zoom in and zoom out a family-tree.
4. Users can see the timeline of a family-tree.
5. Users can upload and download a family-tree in the GEDCOM format.
6. Users can search relatives by some parameters.
7. Users can store the file of family-trees.

Optional features:

1. Tools allow user to upload multimedia, such as photos, history and so on.
2. Tools allow users to include the DNA experiments.
3. Tools allow users to show family-trees in different formats.
4. Tools allow users to do gene analysis.
5. Tools allow users to record family events.
6. Tools allow users to customize layout of interface.

4.2 Client-side web technology

In recent years, web applications are widely used in the life of people and they mainly following three layered architecture. This includes the client tier, the business logic tier and the database tier. The web applications provide services through the connection among three layers. The data flow of the three layers is shown in figure 13. The 3-tier architecture is suitable for the large web applications which have a large number of users (50–2000 or more)[10].
This project just focuses on the client-side web technology, and it introduces web technology through two sections. The first sub-section is to present the most common used web technologies---HTML and CSS. Then, it moves to introduce web languages, which covers JavaScript, PHP, JavaFX and Java applet.

4.2.1 HTML & CSS

4.2.1.1 HTML

Markup language is a language for commenting some text with additional information [11]. HTML stands for Hypertext Markup language, which provides the foundation of all content appearing on the World Wide Web (WWW) [12]. HTML is concerned with how a document is displayed in the web browser. HTML is based on SGML (Standard Generalized Markup Language). SGML provides a standard metalanguage for a variety of markup language, but it cannot specify the elements clearly. A markup language based on SGML needs to be able to supply a series of agreed elements which are suitable for a particular situation. HTML therefore provides a set of appropriate elements for displaying documents in World Wide Web [11]. Figure 14 is an example of HTML.
HTML [11] as a language has been developed a long period of time, and was invented in 1989 by Tim Berners-Lee. The prototype of HTML was very simple (a few allowed elements), but the developer added a lot of new elements quickly due to a variety of purposes. Therefore, there arose a need for establishing a standard that all users could accept. HTML version 2.0 was created because of this attempt. Following on, version 3.2, version 4.0 and version 4.01 have been released. These standards are coordinated by the W3C (World Wide Web Consortium). HTML 4.01 is the most widely used today, but HTML 5.0 is in the process of publishing.

HTML 5.0 [13] is still an "experimental technology", and it is stated HTML 5.0 may not be finalized until 2014. Therefore, the list of new features about HTML 5.0 is always changing. However, HTML 5.0 has established a series of new fixed features, in particular for the diagrammatical properties applied in this project. Therefore, I will make use of HTML 5.0 in project. The details of new features are shown in below table.

<table>
<thead>
<tr>
<th>New Feature</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Semantic Layout</td>
<td>A series of new tags are applied in semantic layout. This means the logically grouped sections are the same as the logical html tag. For example, the header and footer of a web page can be used as <code>&lt;header&gt;</code> and <code>&lt;footer&gt;</code>; navigation toolbar could make use of the tag of <code>&lt;nav&gt;</code>; large parts could be gathered in the tag of <code>&lt;section&gt;</code>.</td>
</tr>
<tr>
<td>New User Interface Tags</td>
<td>HTML 5.0 has added a set of new tags in UI. For example, <code>&lt;mark&gt;</code> stands for highlighting, and <code>&lt;figure&gt;</code> and <code>&lt;figcaption&gt;</code></td>
</tr>
<tr>
<td><strong>New CSS 3 Visual Enhancement</strong></td>
<td>For CSS3, it is easier to locate and ignore individual tag in customer own web site. Besides, it supports 2D and 3D animation.</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Flexible Box-model and Multi-column layout</strong></td>
<td>Users could organize the data of a web page in a multiple rows or columns table. The table is called <code>&lt;display: box&gt;</code>. Furthermore, the width and height of columns are variable.</td>
</tr>
<tr>
<td><strong>Intelligent Form Inputs</strong></td>
<td>Users could submit data back to the web site in input fields of forms. Except his, users are able to define the format of input fields by web browsers, and it was defined through JavaScript in HTML 4.01.</td>
</tr>
<tr>
<td><strong>jQuery , Browser events and Customer Attributes</strong></td>
<td>In order to contribute to create dynamic web pages, HTML 5.0 support to alter browser events, use custom data attributes and so on. JQuery has also added some new events and attributes.</td>
</tr>
<tr>
<td><strong>The Chrome Inspector</strong></td>
<td>The Chrome inspector has build-in Google web browser, and it enable users to test and manipulate HTML, CSS and JavaScript code in actual time.</td>
</tr>
<tr>
<td><strong>Canvas Graphics</strong></td>
<td>The Canvas API enables users to fully control each pixel, color, animation and interaction with the web browser. Even though Canvas API is limited to 2D graphics, HTML 5.0 provides WebGL to create 3D graphics.</td>
</tr>
<tr>
<td><strong>Build-in Audio/Video Support</strong></td>
<td>The audio and video in HTML 5.0 will replace the multimedia in Flash. Although different HTML 5.0 web browser has respective file coding system, users could convert it into</td>
</tr>
</tbody>
</table>
demanded format to reach the goal.

**Drag-and-Drop Event**  
This enables users to visually interact with web sites in a innovated way by mouse or touch-screen.

**Storage Database**  
Web Storage API is able to store more complex data structures and customer information online, including Web SQL and IndexedDB as database.

**Offline Detection and Synchronization**  
HTML 5.0 is able to test whether web site is connected to network. If not, the web browser will get the web site from its internal application cache. Once the web site is connected to network, cache will automatically upload and synchronized with the web browser.

**Geolocation**  
Geolocation API provides relevant physical position to users.

**Web workers**  
Web workers are able to run multi threads of JavaScript in running time, and running each thread separately and asynchronously.

**WebSockets**  
WebSockets are planned to be low-level, persistent communication channels between web browser and web server.

**Notification**  
The Notification API is able to present notification messages outside the web browser.

| Table 1 New Features of HTML 5.0[13] |

### 4.2.1.2 CSS

CSS [14]stands for Cascading Style Sheets, which is used to define specific format for tags and applied documents in a web browser. Cascading means the style could override or stack. Figure 15 is a case.
CSS has developed a period of time, and the latest version is CSS 3. However, the most popular version is still CSS 2.1, which is a revision of CSS 2.0. CSS 3 has begun to set up after CSS 2.0 released in 1998, but CSS 3 is still not accomplished now. However, major web browsers support the stable feature of CSS 3.

```css
body {
  font-family: verdana, palatino, georgia, arial, sans-serif;
  font-size:10pt;
}

p {
  font-family: verdana, palatino, georgia, arial, sans-serif;
  font-size:10pt;
}

p.quote {
  font-family: verdana, palatino, georgia, arial, sans-serif;
  font-size:10pt;
  border: solid thin black;
  background: #FA637E;
  padding: .75em;
}
```

*Figure 15 CSS example [14]*

Through working with HTML, CSS is able to make the web pages in a good layout format and keep the coherence of a web site. There are three categories of CSS [11], which are inline styles, internal style sheet and external style sheet. An inline style can be used when a unique style is to be applied to one single line. Users could utilize any style attributes in relevant tag. For example, `<p style="colour: blue">this is a book</p>`. Internal style sheet can be used when a single document uses a special style. Besides, internal style is utilized in the `<head>` section of a web page. Figure 16 is an example.

```html
<head>
  <style type="text/css">  
    p.heading { font-weight: bold; text-decoration: underline; } 
  </style>
</head>
```

*Figure 16 Internal Style Sheet [14]*

An external style sheet is perfect when it is applied to a whole web site. Users could change the appearance of a web site easily through changing the attributes in external style sheet. Each web page which makes use of external style sheet must cover a `<link>` tag in the `<head>` section. Figure 17 is an example.
Figure 17 External Style Sheet [14]

As a result, a web site may own three types of CSS at the same time. The sequence of executing CSS is that inline style is the first; internal style sheet is the second; external style sheet is the third; the default style of web browsers is the last [11].

CSS [14] includes three levels, and two levels are actual requirements while the third level is a recommended condition. The designers do not need to ensure the level of CSS when they make use of CSS to design. However, they need to know which user agent will be applied in their website, because not all user agents support all the levels of CSS. Nowadays, the supported level of CSS is diversified in web browsers, even though most modern web browsers sustain CSS. Among three levels, the differences are stated as follows:

- CSS 1 ensures the fundamental functionality of style, including limited font and limited position.
- CSS 2 ensures the aural functionality of style on the basis of CSS1. Besides this, CSS2 provides more support to font and position.
- CSS 3 supports presentation-style properties on the basis of CSS 2. This means designers are capable of setting up presentation from web documents.

4.2.2 Web Technology

4.2.2.1 JavaScript

JavaScript is an interpreted object-oriented programming language, which is different from compiled object-oriented programming languages (e.g., Java, C++, C#). It aims to generate dynamic web pages and make sophisticated functions come true (e.g., print, save) by working with HTML. Because it is embedded in HTML documents, it is also called an Embedded Scripting Language. Besides this, JavaScript is the most widely used scripting language in client side applications. The reason for JavaScript's popularity may attribute to its simplicity [16]. Users just need to open a text editor, type in some correct code, save it, load that file in a browser, and it will work. The process of running
JavaScript is that its code will be sent to a client's computer whenever the web page is loaded. The order of execution is line by line through the JavaScript interpreter included in web browser [12]. In the process, users do not need to worry about any compiling cycles, because it does not involve any complex integrated development environment(IDE). Meanwhile, users do not need to worry about any data types, because JavaScript is a loosely typed language.

JavaScript[14] was invented in 1996, and it was released with NetScape2.0. The original reason for creating JavaScript was to improve prototypical scripting languages to moderate level in order to adapt to the improvement of the World Wide Web (WWW). Therefore, in the beginning, JavaScript was planned to be associated with user agents, and it pursued to analyse any messages of any material the user agents proposed and influenced changes in some elements. In order to realize this pursuit, the Netscape team began from setting up the Document Object Model(DOM), which is used to access document links, anchors, form objects and other objects. DOM is [17] " a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents." After the basic JavaScript was constructed, it was taken over by European Computer Manufacture Association (ECMA). ECMA has established the ECMA script standard through abstracting the major features and capabilities of JavaScript. With the rapid boost of JavaScript, the current issues is various DOM(Document Object Model). Because the gap between DOM and JavaScript was ignored by Netscape team, manufacturers have to create a large number of versions of DOM to work with JavaScript, such as JScript[14]. This leads a problem that programmers need to select the most fitting DOM before they coding. The chosen DOM should work with the majority of user agents.

JSON(JavaScript Object Notation)[18] is data format which is mainly used to interchange data among file. It is easy to read and write, meanwhile, it is easy for machines to parse and create. It is built on JavaScript, but it is a completely independent format. The official website say "JSON is built on two structure:

- A collection of name/value pairs. In various languages, this is realized as an object, record, structure, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence."
4.2.2.2 PHP

PHP [19] is the most widely used technology for building server side of dynamic web sites and was created in 1995. At the beginning, PHP was called the Personal Home Page Tools, and it aims to establish a guestbook through collecting information from online forms and presenting it on a web page. Then, a new feature that is able to communicate with database was added after a short time. In 1998, version 3 was published. The original name was changed as it did not sound like professional language and was not suitable the added complex features. PHP 3 was described as “a very programmer-friendly scripting language suitable for people with little or no programming experience as well as the seasoned web developer who needs to get things done quickly.” Following on, PHP 5 which added extensive support for object-oriented programming, was published. For PHP 5, remained true to its roots is one of the most significant attractions of the language. At the same time, PHP is extremely common used in content management system (CMSs), and most popular web sites. For example, Facebook, Wikipedia and so on.

PHP now stands for Hypertext Processor, and it could be used in various ways as follows [19]:

- Uploading files through a web page
- Generating thumbnails from larger images
- Reading and writing to files
- Displaying and updating information dynamically
- Using a database to display and store information
- Making websites searchable

PHP [19] is an embedded language, which means users could add a block of code in ordinary web page. The whole process of interaction between web page and browsers is shown in Figure 18. However, this function is not fit for complex PHP code as it will reduce maintain of web pages. As a result, it is convenient to make use of external PHP files when the code is complicated and repeated.
PHP [15] is regarded as server-side scripting language, while it gets the input from users, it obtains output by the server. However, to run stand-alone PHP applications with users is appropriate for some cases. Command Line Interface (CLI) provides environment for these cases. Even though a text-based CLI is replaced by Graphical User Interface (GUI) nowadays, CLI is still fit for some applications. A CLI is useful for a large number of the offline system-related tasks which demand to support a large web site. Compared with server-based PHP, CLI-based PHP scripts do not need close attention to access file. This may decrease a large amount of time for a web site manager. Because the speed of running CLI-based PHP is much faster than server-based PHP, CLI-based PHP is becoming popular among professional programmers. However, when the users choose CLI-based PHP, it is worth remembering that it removes the possibilities for HTML formatting of PHP output in a browser window because CLI-based PHP does not require a server. Therefore, PHP is not only server-side scripting language, but it could be used without server.

4.2.2.3 JavaFX

JavaFX [20] is a simple, declarative scripting language which aims at contents of web pages. It focused on assisting rich graphical interface, and it is built on the fundament of Java. This means JavaFX could utilize Java libraries, for example, the User Interface (UI) capabilities of JavaFX make use of Java Swing classes. Beside this, JavaFX is statically typed as Java. JavaFX is also an object-oriented language, because it accomplishes the methods (named operations and functions) and attributes (add more details about GUI library).

When Chris Oliver worked for a named SeeBeyond corporation, he started to develop JavaFX [21].
The company demanded richer user interfaces, therefore, Chris Oliver created a called F3 language to realize the needs. After SeeBeyond was obtained by Sun, F3 was renamed JavaFX and it was published in the JavaOne conference in 2007. The first version of JavaFX was a interpreted language and regarded as a model of the second version. The second version of JavaFX changed into compiled language after a short period of time. In the second version, a new set of API libraries had been added. In 2008, a series of changes concerned JavaFX have happened. The compiled version of JavaFX could apply in the NetBeans JavaFX plug-in in March, 2008. Four months later, the preview SDK of JavaFX had been released. Following on, Sun claimed the JavaFX 1.0 SDK will be released in the fall of 2008 at the conference of JavaOne. JavaFX 1.0 SDK was published in 4th Dec 2008. In the conference of JavaOne at 2009, JavaFX 1.2 SDK was released.

JavaFX [20] has a series of obvious strengths compared with other scripting languages. At first, it is convenient and easy for content developers to set up beautiful looking and functional user interfaces due to the simple, declarative syntax of JavaFX. Then, it supports model-view-controller (MVC) pattern because of the bind ability of JavaFX. MVC[22] includes three components, and they are:

- Model: it performs data of application
- View: it is visual representation of data, and it is shown by GUI
- Controller: it is a bridge between a model and the view. It obtains input of users from the view, and transforms these input to a model.

Meanwhile, the above advantage accomplishes and improves the declarative syntax of JavaFX due to attributes of objects are able to be bound to values in model class. Additionally, the concept of trigger in JavaFX makes the development of UI relatively easy, because the setter and getter are replaced by trigger. Trigger means functionality will automatically invoked when a given condition occurred, such as click the button. Following on, JavaFX could run in any places where Java could run. Due to OpenJFX compiler of JavaFX, JavaFX could be turned into JVM (Java Virtual Machine) byte code directly. Besides, the speed of execution of JavaFX is the same as Java. Last but not the least important, for defining, transforming and querying sequences, JavaFX has a competent syntax. Consequently, JavaFX is appropriate for fast-development of prototype of applications.
4.2.2.4 Java Applet

Java applet[23] is a small application which is written by Java, and it works through embedding documents in HTML. This means it is not standalone application. It is different from the majority of Java programs, because it does not need to begin from main method to run. It is called by files of HTML. Moreover, methods of it could be invoked by JavaScript as well. Java applets are simple, because they do not include too many methods. It just includes four methods, and their function will be introduced below.

- the init method includes codes which users put into the constructor normally. It replaces the constructor in Java, and is able to initialise variables.
- the start method is the start point of Java applet, that is Java applet begins from this method to work.
- the stop method is the ending point of Java applet, that is Java applet does not work after this method. Start method and stop method works in pairs.
- the destroy method is not often used, because its function is the same as stop method. The difference between destroy method and stop method is that destroy method could release additional resources.

The above methods are basic methods of a Java applet. Then, there are two kinds of Java applets[23], namely unsigned and signed. Unsigned Java applet could operate within a safety box which covers a series of safe operations, when running in a client. This is because Java applets do not sign a security certificate. A signed Java applets could run all operations. These two types is chosen based on needs of users.
5. Requirement Analysis

5.1 Overview

This part will introduce requirements identified from surveying applications of the family-tree. This will be used in the comparison in section 6 and prototype in section 7. This project is mainly analysing and assessing client-side web technologies for Graphical User Interface (GUI) by a way of building a prototype family-tree interface. Therefore, the functions of a prototype are just fundamental ones, and it is used to test practical performance of client-side web technologies for Graphical User Interface (GUI).

5.2 Functionality Requirements

5.2.1 Mandatory Requirements

In section 4, I summarised a series of common features of genealogy tools. Therefore, the basic functions of the prototype family-tree interface are selected from them. UML will be used to show the requirements, and Figure 19 shows the mandatory requirements of genealogy tools. Create function is the fundamental one, therefore, the relationship between create and other functions is include.

Figure 19 Use Case Diagram(mandatory)
- **Create family members**
  This function enables users to add new family members and move the pointed family members with the mouse. Every family member is considered as an object. The object is presented by rectangles.

- **Edit family members**
  Family members should be linked by their family relationship, delete family members and their corresponding relations. Users should be able to build a family-tree with all family relations. For example, conjugal relations, relations between parents and children, relations of cousin brother and so on. Figure 20 is an example.

- **Inserting personal information by hand**
  Every rectangle represents a family member, and every family member has their personal information. This function enables users to insert personal information. Personal information should be presented using the mouse. The personal information should be shown in the form of table. Users are able to hand-insert personal information into the table. In the table, all the personal data is optional for users to fill in. The table includes below person data.
  - Year of birth: the natal year of family members
  - Gender: the sex of family members
  - Condition: the status of people(s: single, married)

- **Zooming in /out family-tree**
The length of name of people is different, therefore, the width and length of rectangles should be able to be changed. If the size of rectangles is fixed, it is difficult for users to see the long names clearly. This function provides different proportion of rectangles to users, and it is convenient for users to see family members clearly.

- Saving family-tree

After users have drawn a family-tree, they may want to save it. Saving format of diagrams is important for web applications. Most applications of drawing family-trees obey the format of the GEDCOM. However, this is not the mandatory requirement. Users could modify family-trees in other genealogy tools by saving in the form of the GEDCOM.

### 5.2.2 Optional Requirements

The shown requirements of the UML diagram are optional (see figure 22). The optional requirements are able to improve users experience of genealogy tools to some extent.

#### Timeline

This purpose of this prototype is to look the happened events of family members in order of time. It allows users to find significant events easily. For a big family, it could allow family...
members to know what recently happened to their relatives and what events they need to attend. This is a fundamental feature of genealogy tools, but different tools have their respective expressed ways.

- Uploading and exporting family-tree
  
  This prototype could input and output the file of family-trees. It is hard for users to use the same tool in any time and any places. Users may demand to open the file of family-trees in other tools, and exporting function allows user to output the file of family-trees. Uploading function allows users to open file of family from other tools. Therefore, this function allows data flow in different tools, and is also a basic function in key features of genealogy applications.

### 5.3 Web Technology Requirements

In the literature review, we can see a series of scripting languages. The reasons why these scripting languages are surveyed are shown below:

- JavaScript is the most common scripting language, and ninety percents of people that use scripting language employ it.
- PHP is the most widely technologies to build dynamic web pages. It is not only serve in a server, but it could be utilized in offline environment.
- JavaFX is used to design rich Graphical User Interface (GUI). It makes use of GUI library of java.
- Java Applet is a embedded application within documents of HTML, which is accomplished by Java. It is built on fundamental of Java.

After investigating these technologies, the section 6 makes a comparison of selected fitting technologies. The library includes JQuery, Dojo, mxGraph and Kinetic, and they are:

- JQuery is the most widely used JavaScript library.
- Dojo is a JavaScript library which is able to eliminate the influence of various DOMs.
- mxGraph is the only fully client-side diagrammatical JavaScript library
- Kinetic is a relatively popular graphical JavaScript library

These libraries of JavaScript are analysed in section 6. The criteria of comparing four libraries is based
on five aspects., and these aspects are shown as following:

- Ease of Learn: how easy to learn this library for a beginner
- Ease of Use: how easy to use this library to meet the requirements of existing genealogy tools for a beginner
- Universality: whether this library could satisfy the mandatory requirements of genealogy tools
- Focus: the focus of this library
- Licence: whether this library is open to public

The prototype is a way to test whether opinions of technologies of report meet their practical performance. Through fulfilling the basic functions of prototype, the project could test the actual performance of a library. In the end, the project will prove whether my judgement is right through comparing their performance of prototypes.
6 A Comparison of Selected Technologies

6.1 Libraries

There are a large number of libraries for JavaScript to employ, and this report will analyse four widely used client-side libraries. To select JavaScript as the analysed technologies, it is mainly because JavaScript is the most common client-side technology nowadays. The motivation of selecting these four libraries is to keep the universality of selected technologies. These four libraries are JQuery, Dojo toolkit, mxGraph and Kinetic.

- JQuery is the most popular JavaScript library among libraries.
- Dojo toolkit is a common JavaScript that supports various DOM (Document Object Model).
- mxGraph is the only fully client-side diagrammatical library.
- Kinetic is a widely used JavaScript diagramming library.

Every library has its own features, and they will be analysed below. The analysed section of these libraries is based on the basic requirements of interface of family-trees.

6.1.1 JQuery

JQuery [24] is a JavaScript library which is embedded in HTML, and it supports multi-browser (see Figure 23). However, it could not smooth the differences of diverse browsers. For example, the colour of web pages is different in various browsers. It was published in January 2006 at BarCamp NYC by John Resig, and is currently developed by a group of developers whose leader is Dave Methvin. Compared with other libraries, JQuery has the outstanding strength, namely its popularity.

In the survey of websites, JQuery is used by 56.6% of all the websites, moreover, it occupies JavaScript library market share of 91.2%. Besides this, Microsoft announced they will make use of it within Visual Studio in order to better apply ASP.NET AJAX framework of Microsoft and ASP.NET MVC Framework [26]. Moreover, jQuery provides strong extensions to user interface (UI), that is jQuery UI. JQuery UI sustains a variety of interactions and widgets. The interactions include draggable, droppable, resizable, selectable and sortable functions [27].
A draggable function allows users to move elements by using a mouse. Users are able to move target objects easily and freely through given mouse actions, such as left click, double left click and so on.

A droppable function works for draggable function, and it is built on the foundation of the draggable function. When users want to move elements, the computers need to know which one is chosen. The droppable function helps computers find and create targets for moved objects.

A resizable function allows users to change the size of elements by using a mouse. Sometimes the characters of objects are too small to see it clearly. Users are required to enlarge this element. Sometimes objects are too big to see it oddly in web pages. In this circumstance, users are needed to narrow the objects.

A selectable function allows user to select an element alone or from a list. Users could do dozens of actions in websites, but selecting suitable options is the most basic action. Users could select a button, select fitting items from a dropdown list and so on.

A sortable function allows user to change the order of elements in a list or a grid. The layout of websites are usually fixed, but some websites and browser allow users to change the layout of websites. Users could set websites based on their hobby.

Furthermore, jQuery UI [27] supports a series of widgets as well. For example, tooltips, tabs, accordion, progressbar and so on. Tooltips and tabs tools are common tools for users, while accordion and progressbar tools are not familiar by people. Accordion tools enable designers to hide information in content panels aimed at saving limited space. Progressbar tools allows developers to show the current state of processing data to users. In conclusion, jQuery UI plays an important role in jQuery.
JQuery and Jquery UI support a large number of functions, but how could they apply in the Graphical User Interface (GUI) of a family-tree tool? This report will introduce them based on the requirements of applications of family-trees. Firstly, for creating new objects and moving them, Jquery UI can accomplish this task. Jquery could utilize switch() or for() to dynamic add new objects, and Jquery UI uses draggable() to move objects. Figure 24 shows the code of adding new objects.

```javascript
var buttons = [ ];
buttons.push(new primitives.orgDiagram.ButtonConfig("delete", "ui-icon-close", "Delete"));
buttons.push(new primitives.orgDiagram.ButtonConfig("add", "ui-icon-person", "Add"));

options rootItem = rootItem;
options cursorItem = rootItem;
options buttons = buttons;
options hasButtons = primitives.common.Enabled.Auto;
options hasSelectedCheckbox = primitives.common.Enabled.True;
options leavesPlacementType = primitives.orgDiagram.ChildrenPlacementType_Matrix;
options onButtonClick = function(e, data)
{
    switch(data.name)
    {
    case "add":
        var itemZ = new primitives.orgDiagram.ItemConfig();
        itemZ .title = "Z Title";
        itemZ .description = "Z Description";
        itemZ .image = "http://www.basicprimitives.com/demo/images/photos/z.png";
        data.content .items .push(itemZ);
        JQuery(".basicdiagramZ").orgDiagram("update", primitives.orgDiagram.UpdateMode.Refresh);
        break;
    }
}
```

Figure 24 Jquery Add Function[28]

Before Add                              After Add

Figure 25 Jquery Add Function Result[28]
From figure 26, it is clear that move objects using JQuery UI is easy. Users could only make use of a method, namely draggable(). Moreover, users could customize the movement of objects by defining the various options of draggable(). If users want to define the area of movement, user could define the axis option (see Appendix A). If users want to delay the movement, users could define the distance option (see Appendix A). If users would like to count the times of movement, users are able to set start, drag and stop events (see Appendix A). Therefore, for the first requirement, the draggable function of JQuery UI is a recommendable choice, and the way of adding new objects is common.

Then, users are able to link objects and delete objects by JQuery. For deleting objects, there are two situations. In the first case, user only delete object itself, which is easy to performed. And in the second case, the object and the corresponding relations are both removed. However, there is a limitation for linking objects. Each object is allowed to be linked with a fixed number of objects in JQuery. Figure 27 is the relevant example.
Once objects are linked, users need to delete the objects with relevant links. The example is shown in Figure 29. From figure 27, JQuery is not good at linking objects, because of its limitation and the length of executed code. Beside this, the way of deleting function is similar to others.

```javascript
options.rootItem = rootItem;
options.curserItem = rootItem;
options.buttons = buttons;
options.hasButtons = primitives.common.EnabledAuto;
options.addActionCheckbox = primitives.common.EnabledTrue;
options.leavePlacementType = primitives.orgdiagram.ChildrenPlacementType.Matrix;
options.onButtonClick = function(e, data) {
    switch (data.name) {
    case "delete":
        if (data.parentItem == null) {
            alert("You are trying to delete root item!");
        } else {
            var position = primitives.common.indexOf(data.parentItem.items, data.context);
            data.parentItem.items.splice(position, 1);
            JQuery(".basicdiagram").orgdiagram("update", primitives.orgdiagram.UpdateMode.Refresh);
        }
    break;
```

**Figure 28 JQuery Delete Function(code)**[28]

Before Delete                         After Delete

**Figure 29 JQuery Delete Function(result)**[28]

Following on, JQuery UI provides the strong support for resizing objects. Users could apply resizable() to change size of objects depended on their needs. The related code (see figure 30) is so simple that it is admirable to resize elements. Moreover, JQuery UI supplies more extensions for resizing objects. Especially, for a diagrammatical interface, there are mainly three available extensions. At first, users are able to constrain area of enlargement by setting up containment option to be as a parent element (see Appendix A ). Then, users could establish resizable objects to be a maximum and minimum range through define the maxHeight, maxWidth, minHeight and minWidth options for elements (see Appendix A). If users want to resize several objects at the same time, also resize
option is an acceptable choice. This option enables users to zoom in/out objects synchronously (see Appendix A). From above introduced functionality, JQuery UI is a remarkable library for resizing objects. The reasons are that the executed code is easy and the related functions is strong.

```html
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8" />
<title>JQuery UI Resizable - Default functionality</title>
<link rel="stylesheet" href="http://code.jquery.com/ui/1.10.3/themes/smoothness/jquery-ui.css" />
<script src="http://code.jquery.com/jquery-1.9.1.js"></script>
<script src="http://code.jquery.com/ui/1.10.3/jquery-ui-1.10.3.custom.js"></script>
<link rel="stylesheet" href="/resources/demo/style.css" />
<style>
#resizable { width: 150px; height: 150px; padding: 0.5em; }
#resizable h3 { text-align: center; margin: 0; }
</style>
<script>
$(function() {
  $('.#resizable').resizable();
});
</script>
</head>
<body>
<div id="resizable" class="ui-widget-content">
  <h3 class="ui-widget-header">Resizable</h3>
</div>
</body>
</html>
```

*Figure 30 JQUERY Resize Function [27]*

Furthermore, users could utilize tooltip widget to insert personal information in the JQuery UI.

When the mouse moves over the target objects, the related information will be shown. The related example is shown in Figure 31 and 32. Besides this, the target objects, presented style and content could be customized. For target elements, it could be text, forms and textarea. The presented style of message could be decided by users (see Appendix A). What's more, the content of message is not limited to characters, and it could be pictures (see Appendix A). Even though JQuery UI provides this useful tool to insert information, there is still a issue. The inserted information is fixed, and user could not update inserted message. This may lead users could not change the input information unless they create a new target object.
In the end, JQuery could not save diagrams in browsers. However, there are other approaches to save diagrams. For HTML 5, users could save diagrams through canvas. The most common way is to use screenshot to store diagrams as pictures. Besides this, some browsers support export diagrams to be images.

In conclusion, JQuery could meet all the requirements of family-trees applications except the function of saving. In all requirements, JQuery has wonderful performance in creating new objects, resizing objects. JQuery has special functions to implement the demands, namely draggable() and resizable(). Meanwhile, these functions are easy to use and learn. However, JQuery is not specialize in linking objects and dynamic updating inserted message. JQuery has obvious limitations to link objects and update input information. JQuery are hardly to link random number of objects. JQuery could not update inserted information.
6.1.2 Dojo

The Dojo toolkit is a open source library concerned with JavaScript like jQuery, and the main feature of it is that it could smooth the differences among various web browsers. It allows users to employ the same code to obtain the same result in different browsers, because it could eliminate all annoying differences existed different browsers which JavaScript programs have to consider [30]. Dojo includes several components, and the structure of it is shown in figure 33. Dijit is Dojo's the most widely used library for widgets, which replaces web page controls of HTML and allows users to interact with web pages. The most significant feature of Dijit brings to users is the ability of encapsulation, which combines user interface components into standalone tools [31]. What's more, Dijit consists of all the UI components which users are required to apply. For example, tooltip, toolbar, tree and so on. Dojox is different from Dijit, and it provides all new features of Dojo. The most useful feature of Dojox is to support offline tasks. It enables offline tasks works like they run in a server. This is convenient for developers who have to support a large website to finish offline tasks [29]. Base [30] is the foundation of Dojo, which supply all basic functions of applications. What's more, it provides convenient language and AJAX utilities. If Dojo does not include the Base, nothing could be built. Core is built upon the base, which provide extra facilities to parse widgets, manage cookies, advance animation effects and more. The difference between Core and Base is the universality of supported operations. Util is a set of utilities which provides JavaScript unit-test framework and custom version of Dojo for production setting.

Figure 33 The structure of Dojo [31]

To apply Dojo to interfaces of family-trees does not seem feasible, because the focus of its graphs is charts. Family-tree is a tree structure of diagrams. However, Dojo could realize basic requirements of graphs. It is able to create shapes, move shapes, resize objects, insert and update inserted information. For creating shapes, Dojo support rectangles, lines, polygon, circle and texts. The
rectangles are usually representatives of family members in applications of genealogy. Developers could use createRect() to create a rectangle easily (see figure 34). The function of dragging objects is finished by the Base of Dojo, and the relevant function is draggable().

```javascript
var r = s.createRect({ x: 15, y: 80, width: 60, height: 80 })
.setStroke({color: "green"});
```

*Figure 34 Create Function (Dojo)*[29]

To resize objects in Dojo, there are mainly three functions. The first function[32] is onZoom100(evt), and the aim of this function to zoom objects to 100%. The remaining functions are onZoomIn(evt) and onZoomOut(evt), and their effects are the same as their name. Then, Dojo could apply tooltipDialog to insert and update personal information (see figure 35 & 36). The dijit/TooltipDialog [32] displays a tooltip that contains form elements (like a dialog). Although both Dialog and TooltipDialog are modal, TooltipDialog can be closed by clicking anywhere on the screen. In the end, Dojo supports the format of JSON. Users could apply toJson() to transform JavaScript objects into JSON easily. Furthermore, users could use fromJson() to transform JSON strings to JavaScript objects.

```javascript
require([  
  "dijit/TooltipDialog",  
  "dijit/form/TextBox",  
  "dijit/form/Button",  
  "dijit/form/DropDownButton",  
  "dojo/dom",  
  "dojo/ready!"
], function(TooltipDialog, TextBox, Button, DropDownButton, dom){

var myDialog = new TooltipDialog(
        content: '<label for="name">Name:</label><input data-dojo-type="dijit/form/TextBox" id="name" name="name"><br/>
        <label for="hobby">Hobby:</label><input data-dojo-type="dijit/form/TextBox" id="hobby" name="hobby"><br/>
        <button data-dojo-type="dijit/form/Button" type="submit">Save</button>
    );

var myButton = new DropDownButton(
    label: "show tooltip dialog",
    dropdown: myDialog
);

dom.byId("DropDownButtonContainer").appendChild(myButton.domNode);
});
```

*Figure 35 Insert Function (Dojo code)*[32]
In conclusion, Dojo could satisfy four requirements of interfaces of family-trees. However, it could not link objects, which is the essential function in applications of family-trees. Therefore, Dojo is not seemed to be a acceptable library to meets the requirements of genealogy tools.

6.1.3 mxGraph

The mxGraph[33] is a JavaScript visualisation diagrammatical library whose purpose is to present interactive diagrams and graphs. The graphs represents mathematical graphs, and it does not comprise any charts (even though sometimes charts are graphs). Visualisation represents the procedure of creating a helpful visional portrayal of a graph. Visualisation is a significant strength of mxGraph, because extensive components of mxGraph could guarantee the representation of objects be restricted by the experience of users and the available functionality of platforms. Objects may comprise shapes, images, animations, and any graphical components. Another strength of mxGraph is the flexibility of how to realize interaction. In theory[33], if visual feedback spends more than 0.2 seconds in an application, the usability of application will be significantly affected. However, mxGraph could supply the real feeling of an application through putting all interactions on the client side. Moreover, mxGraph is able to supply all the general required functionality for diagrams, and it could collaborate with a diagram. For example, mxGraph is able to create objects, drag objects, resize objects, connect objects, disconnect objects and suchlike. Therefore, mxGraph is widely used in many area of graphs and diagrams. The official website says it could be used in

- process diagrams, workflow and BPM visualization, flowcharts, traffic or water flow, database
- WWW visualization, networks and telecommunications displays, mapping applications and GIS, UML diagrams
- electronic circuits, VLSI, CAD, financial and social networks, data mining, biochemistry, ecological
cycles, entity and cause-effect relationships and organisational charts”.

It is clear that functionality of mxGraph is strong, but there is a obvious limitation. The mxGraph is a commercial library, and users are not free to use it. For this reason, mxGraph provides a series of examples and demos that aim at explaining capable functionality of this library. The project will analyse this library based on these examples and demos, and the related code could not be presented due to the licence of the mxGraph.

For creating new objects, mxGraph provides one relevant functions. The function[33] is mxGraph.inserVertex(parent, id, value ,x, y, width, height, style ), and its purpose is to create a new object in the graphs and diagrams. The meaning of parameters are explained below:

- **parent**: the element which is the direct parent of the new added element in the group structure. Group structure means one object connects with other objects logically.
- **id**: this is a global unique key that is used to identify the object in the group structure. It is usually a string. The id is often used in referencing the objects when an application needs to use this object to do some actions.
- **value**: this is the user object of the new element. User object is what gives diagrams a background. It stores all the logical connections concerned with an element.
- **x, y, width, height**: the x and y stand top left corner of the vertex in a page. The width and height define the size of an element.
- **style**: the appearance of an element. Users could define the style of an element through mxGraph.setCellStyle(style, cells)

Users could add new objects through this function, but it is difficult to show the result of dragging objects through an image and characters. Therefore, this report does not show the result of dragging objects. The official website could show how easy to drag objects. Figure 37 shows the result of creating new objects.
For editing function, mxGraph has given definitions and functions. In mxGraph[33], objects are called cells, and lines are called edges. The mxGraph has a separate mxGeometry class instead of the simple mxRectangle class, because the edges have geometry information as well. The function about edges is `mxGraph.insertEdge(parent, id, value, source, target, style)`, and the parameters have the same meaning as `mxGraph.insertVertex()`. Furthermore, the logical relations of cells are named grouping. In other graph toolkits, grouping is also called sub-graphs. Sub-graphs mean a separate graph is regarded as a child of higher layer graphs. Through grouping, mxGraph enables users to do higher level of actions. For example, users are able to change the order of graphs through layering, and users could also replace a series of grouped cells with only their parent cell. Therefore, mxGraph could logically link cells easily. The mxGraph is able to delete objects, but the example is rare. The figures 38 and 39 shows the outcome of linking and deleting cells.
For resizing objects, mxGraph has two useful functions. The first function is `mxGraph.resizeCell(cell, bounds)`, and it enables users to resize the given object to the defined range. Cell represents the given object, and bounds stand for defined area in advance. The second function is `mxGraph.resizeCells(cells, bounds)`, and it enables developers to resize every element of an element array to the related entry in the array of range. The second method is convenient to enlarge or narrow the whole family-tree. Due to the corresponding code is licensed, the case of resizing is presented in figure 40.
For inserting and updating message, mxGraph has a great performance. Users could insert related information in the dialog. If they want to update information, they could change inserted information and click the save button to store the changes. However, there is a drawback of mxGraph in this function. mxGraph allows user to change the property of objects, but the procedure of processing these transformations from diagrams works on the server. Then, these transformations are sent back to the client side. Figure 41 shows the outcome of updating the specified element.

For saving function, mxGraph [34] supports five formats of saving, namely PNG(Portable Network Graphics), GIF(Graphics Interchange Format), JPG(JPEG File Interchange Format), PDF(Portable Document Format), SVG(Scalable Document Graphics) and XML(Diagram XML Document ). This is convenient for developers to store diagrams and graphs. Meanwhile, variable formats provide a wide range of tools for developers to upload diagrams and graphs. Consequently, mxGraph could achieve the requirements of interface of family-trees beyond comparing. From the above examples,
mxGraph could realize all the functionality, and the technology seems promising for family-trees. Nevertheless, the analysis may be bias due to the related code functions are hardly to find. The analysis is only based on the examples and demos of this library.

### 6.1.4 Kinetic

Kinetic[35] is a HTML 5 canvas JavaScript diagrammatical library. It supports various actions applied to diagrams and graphs, for example, animation, node nesting, layering , filtering and so on. Kinetic could be employed in both mobile and desktop applications. Kinetic [35] has several features, and it is introduced below:

- It supports Object Oriented API and event bubbling.
- It supply high performance event detection via colour map hashing
- It supply node caching to improve draw performance. Nodes can be converted into data URLs, image data, or image objects
- It is ready to use shapes including rectangles, circles, images, text, lines, polygons, SVG paths, and more. Meanwhile, it support customize shapes .
- It is event driven architecture which enables developers to subscribe to attr change events, layer draw events, and more
- It supply pixel ratio optimizations for sharp text and images

To apply Kinetic, developers are required to notice the whole structure of Kinetic. The whole structure of Kinetic is divided into two layers. The stage[35] is a relatively high level, and it is made up of user defined layers. There are two canvas renderers in every layer, that are a scene renderer and a hit graph renderer. The scene renderer is used to show diagrams and graphs, and the hit graph renderer is used to detect happened events. In each layer, it may cover shapes, groups of shapes, groups of other groups. Figure 42 shows a hierarchical structure.
At first, Kinetic provides strong support for shapes and dragging. Kinetic supports diverse shapes, such as rectangles, circles, texts and so on. The below figure is a case of creating a new rectangles. Users could add a new rectangle easily through Kinetic.Rect()[36]. The way of adding other shapes is the same as rectangles, and just the applied functions are different. For example, if users want to create a circle, they need to use the function called Kinetic.Circle(). The meaning of parameters are introduced below:

- **x, y**: the position of object in the page
- **width, height**: the width and height of an object
- **fill**: the colour of an object
- **stoke**: the colour of boundary of an object
- **strokeWidth**: the width of boundary of an object

Kinetic applies setDraggable() to drag and drop objects. The objects could comprise images, lines, shapes, groups and suchlike. The figure 44 is a fundamental example of dragging rectangles. If users want to move other types of objects, they need to change rectangles into the corresponding objects in code(see Appendix A). Furthermore, Kinetic is able to set the boundary of dragging by dragBoundsFunc()(see Appendix A).
Kinetic is capable of linking objects and deleting objects. However, it is a little complex to connect objects in Kinetic. Kinetic does not have the specified function to link objects, but it supply helpful functions about lines. Lines are essentials in the procedure of connecting objects. For example, getSize() enables get the size of lines. There are two functions applied to link objects, namely getOffsetX() and getOffsetY(). The aims of two functions are to get position of lines, then, linking children and their parents is through judging their according positions. Figure 45 is a case of associating objects. Then, Kinetic could delete objects, but it does not have the appointed function to do. In figure 46, developers code a delete function, and register this command in the function of executing.
For resizing function, Kinetic does not have the given function. However, a proper example which could show the resize function of Kinetic is shown in figure 47 and 48. Due to the result of resizing is difficult to present, the difference between before zooming in and after zooming out is not distinct. Developers could set the minimise size of objects at first. Then, users could zoom out the given object by comparing the current size and minimise size. If the current size is smaller than the minimise size, users could not narrow the object. The function of enlarging objects is the same as the function of zooming out.
Developers are able to insert information through layering of Kinetic. Figure 50 is the corresponding code. Developers create a variable called tooltip at first. Then, developers could add tooltip into the layer named tooltipLayer. By Kinetic.Layer(), the application will automatically add 49 circles. Through the order of layers, tooltipLayer is finally added into stage. In the end, developers update the tooltip through the actions on the stage. It is obvious that Kinetic realize this function through the way of addListener() to the corresponding layer. However, Kinetic is not able to update inserted information.
Finally, there are three approaches to save diagrams in Kinetic. The first one is that Kinetic provides a function called toImage(). This function could transform objects into images, and store objects in the format of pictures. The second is that Kinetic allows objects to be saved in JSON. JSON (JavaScript Object Notation) is a data-interchange format which is simple for machines to parse and generate. The third is that Kinetic has a called toDataURL() method. This method requires a callback function, and stores specific objects through data URLs of specific objects. To apply this method, users should notice the domain of the server which hosts the canvas should be the same as the domain of performing code. Hence, users could choose the way of storing depend their demands in Kinetic.

In all, Kinetic could meet all the requirements of interfaces of genealogy. Meanwhile, Kinetic performs well in all functions. Kinetic could create all kinds of shapes and move created elements freely. Then, there are not any limitations to associate and delete objects. Developers could link arbitrary number of objects, and delete them uncomplicatedly. Following on, Kinetic could resize objects through functions. To insert personal information, Kinetic could solve this issue by adding tooltip layer to stage. In the end, Kinetic supply toImage(), JSON and toDataURL() to save diagrams.

6.2 Comparison OF Libraries

Even though core JavaScript could resolve a large number of problems, the functionality of the analysed libraries of JavaScript is discussed. This project mainly analyses applications of libraries for
diagrammatical interfaces. The basic requirements of genealogical applications provide the key clue of the analysis. The below table summarise what could every library do, with respect to the mandatory requirements in section 5.

<table>
<thead>
<tr>
<th>Function of Libraries</th>
<th>JQuery</th>
<th>Dojo</th>
<th>mxGraph</th>
<th>Kinetic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create objects</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Edit objects</td>
<td>×</td>
<td></td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Zoom in/out</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Insert personal information</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Save</td>
<td></td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>

*Table 2 Functions of Libraries*

Then, This project compares these four libraries from five aspects. For all the libraries, I am a beginner.

- The first aspect is how easy to learn this library. Through the length of learning time, adequacy of related materials and its relevant examples, I judge the extent of ease to learn these libraries. The highest rate is 5, and the lowest rate is 1.
- The second aspect is how easy to use this library. After I have studied these libraries, I try to employ these libraries. I assess these libraries from my experience and the actual cost time of utilizing libraries. The rating standard is the same as the first aspect.
- The third aspect is the functionality of these libraries. I assess the functionality of these libraries through how well they apply to basic requirements of interface of family-trees. The standard of judgement is the same as the former one.
- The fourth aspect is the focus of these libraries. Every library may have their skilled functions. The library whose focus is diagram is much better than others, because selected libraries are applied to diagrammatical interface.
- The final aspect is licence of libraries, and it is a significant element for comparison. If a
wonderful library is not free to use, a few persons may spend money to employ it. Even though licensed libraries are usually more powerful than other libraries, a single developer is rarely to use it due to high cost. The licensed libraries are often employed by organisations and big companies. The below table show my assessment for four libraries.

<table>
<thead>
<tr>
<th>Libraries</th>
<th>Ease of Learn</th>
<th>Ease of Use</th>
<th>Functionality</th>
<th>Focus</th>
<th>Licence</th>
</tr>
</thead>
<tbody>
<tr>
<td>jQuery</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>Variable</td>
<td>free</td>
</tr>
<tr>
<td>Dojo</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>DOM</td>
<td>free</td>
</tr>
<tr>
<td>mxGraph</td>
<td>Not sure</td>
<td>Not sure</td>
<td>5</td>
<td>Diagram</td>
<td>Not free</td>
</tr>
<tr>
<td>Kinetic</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>Diagram</td>
<td>free</td>
</tr>
</tbody>
</table>

Table 3 Comparison Library

The above table summarize all assessment of the four libraries. The explanation of judgement will be introduced detailedly in their own table. The first table show the explanation of Jquery, and the following is Dojo, mxGraph and Kinetic.

<table>
<thead>
<tr>
<th>JQuery</th>
<th>Evaluation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ease of Learn</td>
<td>4</td>
<td>JQuery is the relatively simple library to learn among four libraries. Because JQuery is the most widely used library, there are a large number of related materials which could help learners to study and research. Meanwhile, it is easy to find examples about application of JQuery. Except this, official website provides useful introduction to JQuery UI. Enough materials and examples are essential elements in the process of learning. Therefore, I spend one week to learn JQuery.</td>
</tr>
<tr>
<td>Ease of Use</td>
<td>3</td>
<td>Even though JQuery supply enough materials for learners to study, JQuery is a little difficult to use. JQuery provides useful functions to implement the requirements of interface of family-trees, but they cost more time than I expected when they works well together. One function could run well alone, but sometimes it could not run with</td>
</tr>
</tbody>
</table>
other functions. I planned to spend one week to build prototype of JQuery, but it cost two weeks in fact. From my experience of using JQuery, it is not extremely difficult to use it. However, it will spend users a short period of time if they are beginners.

### Functionality

From the above analysis section, it is obvious that JQuery could achieve all requirements of a interface of family-trees except the function of saving. In creating new family members and resizing family members, JQuery has acceptable functionality. Whereas, the disadvantages of JQuery is clear as well. Linking family members based on their family relations is a highly significant part in the procedure of drawing family-trees. However, JQuery could not link random numbers of objects. Beside this, JQuery could not update personal information. Therefore, I asses the functionality of JQuery is 3.

### Focus

JQuery is the most widely used library, and it apply to many websites and web services. For example, Wikipedia, blackberry, Intel and so on. Therefore, it does not have accurate and clear focus. It could be used for mobile applications to process characters. Meanwhile, it could be utilized in websites to response some actions of users. In conclusion, the focus of JQuery is variable.

**Table 4 Explanation of Rating JQuery**

<table>
<thead>
<tr>
<th>Dojo Evaluation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ease of Learn</td>
<td>2</td>
</tr>
<tr>
<td>Dojo is a mature JavaScript library, and it is widely used in many web applications. Therefore, there are enough materials and examples for learners to research. But Dojo includes too many components, it is difficult for beginners</td>
<td></td>
</tr>
</tbody>
</table>
to understand all the components. I spend two weeks to learn Dojo, and I only basically understand Dijit. Dijit is the GUI part of Dojo, so I guess it will cost a long period of time to understand the whole Dojo. Due to the limitation of time of project, I do not learn all the components of Dojo. Too many section of Dojo are a barrier for learner. For above reasons, I rate 2 for Dojo.

<table>
<thead>
<tr>
<th>Ease of Use</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Due to I just learn Dijit, I am hardly to judge the extent of ease to learn the whole Dojo. In the process of utilizing Dojo, it is difficult to begin. I tried to code some basic functions, but it does not work well. I guess maybe I am not familiar with Dojo. From my experience, users are required to pay more attention to Dojo if they want to be a skilled developer. In my opinion, I think Dojo is challenging to learn.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Functionality</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Although Dojo has the strong functionality, the focus of its graphs is charts. Thus, it only meets basic requirements of graphs. Dojo could create all types of shapes and drag them. Dojo enables users to resize objects, show and update inserted information. Moreover, Dojo allows users to transform objects to JSON. Due to Dojo cannot link objects, the judgement of functionality of Dojo is 3.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Focus</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dojo could apply to enormous applications. It could be used in both mobile and desktop application. For graphs, it is usually applied to bar charts, line charts, pie charts, geographical charts and maps. Moreover, it is often used in</td>
</tr>
</tbody>
</table>
rich web application, such as mail, video player and so on.

Table 5 Explanation of Rating Dojo

<table>
<thead>
<tr>
<th>mxGraph</th>
<th>Evaluation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ease of Learn</strong></td>
<td>Not sure</td>
<td>Because mxGraph is licensed, I could not get this library. However, related materials are rare. I just find a useful introduction from github. Beside this, the useful examples are only found in official website. Therefore, I could not know whether it is easy to learn.</td>
</tr>
<tr>
<td><strong>Ease of Use</strong></td>
<td>Not sure</td>
<td>I cannot access this library due to it is not free to use. I need to obey the copyright of this library. I sent email to this corporation in order to get tested version. However, the corporation just send the evaluation version to organisations. The corporation does not provide evaluation version to individuals. For this reason, I do not know whether it is easy to use.</td>
</tr>
<tr>
<td><strong>Functionality</strong></td>
<td>5</td>
<td>mxGraph could fulfil all requirements of interface of family-trees. Meanwhile, practical performance of mxGraph is excellent. For creating new family members, it is easy to add any shapes of objects. At the same time, users are easily to move objects. For editing family members, mxGraph supply many kinds of connections and arrows. When users delete targeted objects, the related connection will be removed simultaneously. Following, users could update inserted information freely. Moreover, users could use it to do different kinds of resizing. In the end, the exported diagrams have six format. Therefore, I judge the</td>
</tr>
</tbody>
</table>
functionality of mxGraph is 5.

<table>
<thead>
<tr>
<th>Focus</th>
<th>Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>mxGraph</td>
<td>is the only fully client-side graphical library, and it could apply all</td>
</tr>
<tr>
<td></td>
<td>kinds of shapes and arrows, such as rectangle, circle, parallelogram,</td>
</tr>
<tr>
<td></td>
<td>tape, callout quad arrow, bent right arrow and so on. Moreover, it could</td>
</tr>
<tr>
<td></td>
<td>be used to draw UML diagrams and flow chart. Furthermore, it supports</td>
</tr>
<tr>
<td></td>
<td>iOS, BPMN gateways, BPMN events and BPMN general. The official website</td>
</tr>
<tr>
<td></td>
<td>announced that mxGraph is &quot;Interactive JavaScript HTML 5 Diagramming</td>
</tr>
<tr>
<td></td>
<td>Library&quot;.</td>
</tr>
</tbody>
</table>

**Table 6 Explanation of Rating mxGraph**

<table>
<thead>
<tr>
<th>Kinetic</th>
<th>Evaluation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ease of Learn</td>
<td>3</td>
<td>The tutorial materials about Kinetic are scarce. I just find tutorial materials from the official website. However, the found materials only introduce basic definitions of Kinetic. The relevant examples are more than materials. I explore useful examples from the official website and GitHub. The origins of examples and materials are rare, and this is an obstacle for beginners to learn. I spend two weeks to study it. Due to the related materials and examples are limited, I rate the extent of ease is 3.</td>
</tr>
<tr>
<td>Ease of Use</td>
<td>3</td>
<td>Even though Kinetic does not supply enough materials for learners to study, Kinetic is not very difficult to use. Kinetic supports a large number of useful functions and shapes to implement the requirements of interface of the family-trees. From my experience, the speed of employing Kinetic will accelerate once users understand the structure</td>
</tr>
</tbody>
</table>


of Kinetic. I planned to spend two week to build prototype of Kinetic, and it cost two weeks as planned. However, it will spend developers a little long period of time to skilled use if they are beginners. For above reasons, I judge the extent of ease to use is 3.

<table>
<thead>
<tr>
<th>Functionality</th>
<th>5</th>
<th>Kinetic could realize all the requirements of applications of genealogy. At the same time, Kinetic does a good job at all functions. Except this, Kinetic is supported by HTML 5. All new features of HTML 5 could be executed in Kinetic. HTML 5 has strong features in diagrams and graphs, therefore, the functionality of Kinetic is improved. This are reasons why I give full mark to Kinetic.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focus</td>
<td>Diagram</td>
<td>Kinetic is a JavaScript interactive graphical library. It supports HTML 5 canvas. There are two ways to draw diagrams, that is canvas and SVG. Furthermore, it enables various animations, transitions, node nesting, layering, filtering for mobile and desktop applications. Kinetic support all types of shapes as well, such as rectangle, line, circle, text and so on.</td>
</tr>
</tbody>
</table>

In conclusion, Kinetic seems the best choice considered all aspects. The reasons are that the total mark of Kinetic is the highest, and the focus of Kinetic is diagram. Meanwhile, Kinetic is free to use. JQuery is the second choice, and the functionality of JQuery is relatively acceptable. The main drawback of JQuery is the limitation of linking function. The mxGraph seems a promising selection from its functionality. However, it could not be adopted as it is licensed. Thus, this project will design two prototypes. One is implemented by Kinetic and JQuery, and the other is finished by JQuery. Through assessing the functionality of two prototypes, the evaluation section will prove whether the analysis section of this project is correct and accurate.
7 Prototypes

This section describes two prototypes. The functionality of two prototype are developed to evaluate the selected technologies. Based on the analysis of section 6, the selected technologies are JQuery and Kinetic. Although the functions of two prototypes are the same, the way of performing is different.

7.1 The JQuery Prototype

The JQuery prototype is implemented by JQuery. The JQuery prototype could execute creating functions, editing function, resizing function and the function of inserting personal information. Figure 51 is a activity diagram, which presents the process of drawing a family-tree.

![Activity Diagram(JQuery Prototype)](image)

Figure 51 Activity Diagram(JQuery Prototype)

Figure 52 shows the GUI for the JQuery prototype. At the left of this interface, there is a table, which is used to set the personal information of default family members. At the top of this prototype, there are two buttons, which are used to resize family members. In the centre, there is a canvas that is applied to draw a family-tree. There are three default family members, namely mother, father and
oneself. At the right, there is also a table which is used to add new family members and insert personal information.

![Figure 52 JQuery Prototype](image)

### 7.1.1 Create Function

Based on the above activity diagram, it is clear that users are required to set the personal information for default family members at first. If users do not want to do this step, users could skip it. After this activity, user must insert personal information before adding new family members. Accordingly, the functions of inserting personal information and adding new family members are combined in the JQuery prototype. In this prototype, a rectangle stands for a family member. Figure 53 shows for adding new family members. At first, the prototype ensures the added family members are able to move. Then, users insert personal information at the right table. After inserted personal information, users click the additive button. The related personal information is stored in a array called nnn(see Figure 53). The array is stored in the cache, so the inserted update information could not be changed. After users click the add button, new family members will be presented in the canvas.
The edit function includes two sub-functions, that is the linking function and the deleting function. JQuery is not good at linking objects, and the JQuery prototype connects rectangles through judging the position of lines and rectangles (see Appendix A). To link family members, this prototype will connect family members automatically through assessing the position of the selected family member. When users move family members, the relevant connection will automatically move. Due to limitations of JQuery, this prototype is only able to add brother and sister for oneself. This function is triggered by using the mouse to move a selected family member. To delete family members, this prototype allows users to delete a alone family member and the linked family member. Figure 54 shows related code. This function is triggered by clicking the button of delete in the table of presenting personal information.

7.1.2 Edit Function

Figure 53 Add New Family Member (JQuery Prototype)
7.1.3 Insert Function

This function has been introduced in section 7.1.1. Figure 55 shows the table of showing personal information. When the mouse moves over the family member, the relevant personal information of the moved family member will be shown.

7.1.4 Resize Function

In section 6, the analysis of JQuery claimed that users could use resizable() to resize objects. However, as the experience of the author is limited, the author could not employ this function well. Therefore, this project utilizes the height and width of rectangles to resize family members. When users would like to zoom in family members, the width and height of rectangles will automatically add 10 at every click of the mouse. If users want to zoom out family members, the interface will judge whether the width and height of rectangles is less than 20 at first. If the width and height of rectangles is less than 20, there will be a notice box to remind users the width and height of rectangles is the smallest. If not, users could narrow rectangles. This function is triggered by clicking the buttons of "zoom in" and "zoom out".
The function of creating family member is fulfilled by the author because JQuery does not supply the given method. The function of dragging created family members is fulfilled by the draggable() method of JQuery.

The author finishes the function of editing family members by oneself, and the reason is the same as the create function.

The function of inserting personal information is accomplished by the tooltip of JQuery.

The function of resizing family members is accomplished by the author. Due to the limitation of experience of the author, the author could not apply the resizable() method.

The JQuery prototype cannot perform:

- The function of updating inserted personal information is not realized, because all information is
The function of saving is not fulfilled, as JQuery does not have the related method.

### 7.2 The Kinetic Prototype

The Kinetic prototype is implemented Kinetic, and all functions could be performed well. As the pure client-side applications could not update inserted information, this prototype has added database and PHP. Therefore, this prototype is a web application to some extent. Database (see figure 57) is used to store information of family members, and PHP is used to exchange data between the HTML and the database. Hence, this prototype includes three tiers (see Figure 58). Users could access this prototype by below website: [http://www2.macs.hw.ac.uk/~jn81/familytree81project/index.php](http://www2.macs.hw.ac.uk/~jn81/familytree81project/index.php).

Figure 59 is a activity diagram, which presents the whole process of drawing a family-tree.

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Type</th>
<th>Collation</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>node_id</td>
<td>int(10)</td>
<td>unci</td>
<td>No</td>
<td>None</td>
<td>AUTO_INCREMENT</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>node_x</td>
<td>int(11)</td>
<td>unci</td>
<td>Yes</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>node_y</td>
<td>int(11)</td>
<td>unci</td>
<td>Yes</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>node_name</td>
<td>varchar(64)</td>
<td>latin1_swedish_ci</td>
<td>Yes</td>
<td>NULL</td>
<td>AUTO_INCREMENT</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>parent_id</td>
<td>int(10)</td>
<td>unci</td>
<td>Yes</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>spouse_id</td>
<td>int(10)</td>
<td>unci</td>
<td>Yes</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>yr</td>
<td>smallint(5)</td>
<td>unci</td>
<td>Yes</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>gn</td>
<td>tinyint(3)</td>
<td>unci</td>
<td>Yes</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>ms</td>
<td>tinyint(3)</td>
<td>unci</td>
<td>Yes</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 57 Database Structure (Kinetic prototype)**

**Figure 58 The overall structure of the Kinetic Prototype**
**Figure 59 Activity Diagram (Kinetic prototype)**

Figure 60 is the appearance of the Kinetic prototype. The layout of this prototype is simple because there are only three buttons at the top. The first button is New Diagram, which allows users to get a new canvas. The button of New Node is used to add new family members. After this button, the button of Save Snapshot is employed to save a family-tree as the PNG format of an image.

**Figure 60 Kinetic Prototype**
7.2.1 Create Function

The Kinetic prototype implements Kinetic.Group() and Kinetic.Rect() methods to perform the function of creating family members. The Kinetic.Group() method is used to group multiple shapes. The Kinetic.Rect() method is used to create a rectangle. Every rectangle represents a family member. At first, this prototype creates a group, and the draggable property of this group is true. Therefore, every rectangle in this group is able to move. Then, the Kinetic prototype create rectangles in the new created group. The properties of rectangles have been introduced in section 6. This function is trigged by clicking the button of "New Node".

```javascript
node({id: node_id}) = new Kinetic.Group({ // create new group
  draggable: true // group should be drappable
});
box({id: node_id}) = new Kinetic.Rect({ // create new box
  x: node_x,
  y: node_y,
  width: 80,
  height: 100,
  fill: boxcolor,
  shadowColor: boxcolor[0],
  shadowBlur: 2,
  shadowOffsetX: 1,
  shadowOffsetY: 1,
  stroke: strokeborder,
  strokeWidth: strokeborders
});
```

Figure 61 Create Function(Kinetic Prototype)

7.2.2 Edit Function

After users create new family members, a table will be presented if users click the family members. In the table, users could select parents and a spouse for this family member. Parents and spouse are made up by existed family members. Then, if the spouse of this family member is not empty, this family members and his spouse will be connected by a green line (see figure 60). The way of connection between parents and their children is the same as couples, and the only difference is the connected line is navy (see figure 60). Figure 62 shows how this function works in code. Figure 63 shows the whole procedure of deleting family members. If user want to delete a family member, they need to click it at first. The reason is that the button of "delete" is in the table of showing personal information. Then, the prototype will judge whether the family member is linked or not.
Finally, the family member will be removed from the layer.

7.2.3 Insert Function

Figure 64 shows the whole process of inserting personal information of the Kinetic prototype. The family members should be selected at first. After this, a table which contains personal information will be shown. In this table, users could insert and update existed information. In the end, all operation are stored by clicking the button of "Save". Figure 65 shows how this function is accomplished by code. The prototype judges whether the object is chosen or not. Then, the inserted
information is passed to the corresponding attributes. The function of updating inserted information is fulfilled by the database and PHP. The method of `document.getElementById().value` is able to get the value of corresponding attributes. Finally, the diagram will be redraw in the layer.

![Figure 64 Process of Inserting(Kinetic Interface)](image)

```javascript
node[#{node_id}].on('click', function() { // this function is called when a box is clicked on
box[#{node_id}].setStroke(strokeborderselected); // set border color
box[#{node_id}].setStrokeWidth(strokeborderselected); // set border width
if (selected > -1) selected = (#{node_id}); // and remove the previous selection (if it exists)
box[#{selected}].setStroke(strokeborder);
box[#{selected}].setStrokeWidth(strokeborderwidth);
};

selected = (#{node_id}); // remember which node is selected
// properties to edit - fill these fields in
node[#{node_id}].value = (#{node_id});
node[#{node_id}].value = (#{node_id});
node[#{node_id}].value = (#{node_id});
node[#{node_id}].value = (#{node_id});
node[#{node_id}].value = (#{node_id});
node[#{node_id}].value = (#{node_id});
node[#{node_id}].value = (#{node_id});
node[#{node_id}].value = (#{node_id});
node[#{node_id}].value = (#{node_id});
node[#{node_id}].value = (#{node_id});
node[#{node_id}].value = (#{node_id});
layer.draw(); //redraw the layer
```

![Figure 65 Insert Function(Kinetic Interface)](image)

### 7.2.4 Resize Function

In this prototype, the way of executing resizing function is different from JQuery and the example of the analysis. The width of rectangles could automatically adjust depended on the length of the name of family members. Through this function, users are not required to enlarge or narrow rectangles.
Figure 66 is about how to realize this by code.

```javascript
// adjust box width by the text width
if (name[$x["node_id"]].getText() != '') {box[$x["node_id"]].setWidth(name[$x["node_id"]].getWidth() + 10);};
```

*Figure 66 Resize Function (Kinetic Prototype)*

7.2.5 Save Function

Kinetic provides three approaches to save objects, and this interface adopts the toDataURL() method. The toDataURL() method could stores specific rectangles by data URLs of specific rectangles. The diagram draw on the canvas could be stored in the format of PNG. This function is trigged by clicking the button of "Save Snapshot".

```javascript
// create snapshot which can then be saved
document.getElementById('snapshot').addEventListener('click', function() {
    stage.toDataURL({
        callback: function(imageURL) {
            //var imageURL = imageURL.replace("data:image/png;base64","data:image/octet-stream;base64");
            window.open(imageURL);
        }, mimeType: "image/png", quality: 1.0
    });
}, false);
```

*Figure 67 Save Function (Kinetic Interface)*

7.2.6 Summary

In conclusion, the Kinetic prototype is able to manipulate:

- The function of creating family members is achieved by the Kinetic.group() and Kinetic.Rect() methods. The function of moving the created family members is achieved by the property of the Kinetic.group() method.
- The function of editing family members is accomplished by the author.
- The function of inserting personal information is performed by the layering of Kinetic. PHP and database enables the Kinetic prototype to update inserted information.
- The author achieves the function of resizing family members by oneself.
- The Kinetic prototype applies the toDataURL() method to save family-trees.
7.3 Conclusion

To summarize, the functionality of the Kinetic prototype is better than that of the JQuery prototype. Although the added PHP and database could improve the functionality of the Kinetic prototype, the provided methods of Kinetic is much more than JQuery in diagrammatical user interfaces. The Kinetic could satisfy all basic requirements of genealogical applications, while JQuery cannot achieve this goal. The two prototypes are built to evaluate JQuery and Kinetic, and the evaluation will be based on how easy to execute these functions.
8 Evaluation

This section aims at evaluating the functionality of the two prototypes, and then, the analysis about JQuery and Kinetic will be proved whether it is correct or not. This section is divided into four sub-sections. The first section is the author evaluate the functionality of two prototypes based on the procedure of developing. Then, this section introduces users questionnaire. After this, the evaluation of testers is analysed. Finally, there is a discussion about the evaluation.

8.1 Evaluation of development process

In the process of developing, the usability of Kinetic is much higher than JQuery as the functionality of Kinetic is stronger. The Kinetic has a large number of methods that could apply to diagrammatical user interfaces, but JQuery does not have. Compared the functionality of the two prototypes, the kinetic prototype is more powerful than JQuery. The reasons are that:

- The Kinetic could use the Kinetic.group() method and required shapes to create items, while JQuery is required users to code by oneself. For a beginner, to apply existed methods is much simpler than coding by oneself.
- Although both Kinetic and JQuery do not have the specific functions to connect objects, it is more convenient for a beginner to apply Kinetic. Kinetic allows users to connect random numbers of objects, while JQuery only links the fixed number of objects. The Kinetic is easy to move linked family members, but it is a little difficult for JQuery because the lines could not move with the family members simultaneously.
- The Kinetic supplies three approaches to save diagrams, nevertheless, JQuery could not save diagrams by itself.

8.2 Users Questionnaire

This evaluation phrase concerned two prototypes with different technologies(e.g. JQuery and Kinetic). Both of prototypes with same functions are designed to build a family tree. Since the efficiency of these prototypes performing functions are different, a questionnaire is made to prove
the view about "the combination of JQuery and Kinetic is a better way to implement the drawing functionality" mentioned before. All participants are required to complete a set of tasks (such as creating items and remove them) in the two ways, and then answer several questions in this questionnaire.

The questionnaire(see Appendix B) covers four parts, and they are introduced:

- The first part includes general questions about users' background, such as the familiarity of applications of family-trees and the familiarity of Graphical User Interface (GUI) (GUI).
- The second part is an instruction about how to use two prototypes. This instruction guides testers how to use two prototypes to perform each function respectively. As the JQuery interface could not save diagrams by itself, the function of saving is not included the compared functions.
- The third part provides a set of function cards. It is main body of the questionnaire. The function cards are used to examine how easy it is to use in the two different ways or prototypes. There are four function cards, including the create function, the edit function, the insert function and the resize function. At the top of each function card, there is a description that introduces the corresponding tasks. In the middle, it is the evaluation of testers about tasks. The rating is from 1 to 6. The higher the score is, the easier the prototype performs a specific function. Each function card provides three questions relating to satisfaction of User interface of each prototype.
- Testers could provide some comments in the 4th part after rating the two prototypes.

8.3 Testers Evaluation

16 testers are willing to take part in this test, and they are all students from Herriot-watt university. They fill in the questionnaire based on their actual experience. In order to keep the diversity of testers, the testers come from different subjects. Among participants, the number of women is equal to that of men. 37.5% participants (6) in total are familiar with applications of family-trees; 62.5% participants in all are familiar with Graphical User Interface (GUI)s. Their evaluation about the functions is shown in below column diagrams.
8.3.1 Function of Creating movable items

Figure 68 claims the evaluation of participants for the create function. The x axis stands for the rating stand, and it is from one to six. The y axis stands for how many the score has been selected. The evaluation for creating function is made up by two sub-sections below.

A Evaluation of Creating Family Members

➢ Figure 68 A clarifies the assessment about how easy to create new family members. It is obvious that most participants think this function of the two prototypes does not have many differences. Even though the majority of participants (62.5% in all people) consider the Kinetic prototype is extremely easy to create new family members, no participants consider it is difficult to create new family members by the JQuery prototype.

B Evaluation of Moving created family members

Figure 68 Evaluation of Create Function
Figure 6B clarifies the evaluation about how easy to move new added family members. It is clear that 75% participants think this function of the Kinetic prototype is highly easy to perform. As the evaluation concerned with the JQuery prototype is various, it is hard to get a explicit conclusion. However, it is obvious that it is easier to perform this function by the Kinetic Prototype.

### 8.3.2 Editing Function

Figure 69 outlines the evaluation of participants for the edit function. The x axis and y axis have the same meaning as the former column diagrams. The evaluation for the create function is made up by four sub-sections. They are:

![Figure 69 A Evaluation of Linking](image)

Figure 69 A illustrates the assessment about how easy to link family members by the two prototypes. It is evident that the majority of participants (87.5% in all) feel it is more acceptable to link family members by the Kinetic prototype.
Figure 69 B Evaluation of Moving linked family members

- Figure 69 B illustrates the evaluation about how simple to move linked family members. Half participants feel the JQuery prototype is complicated to move linked family members. Compared with the JQuery prototype, most people (75% in all) consider the Kinetic prototype is easier to manipulate this function as there are twelve persons selecting the highest score.

Figure 69 C Evaluation of Changing relations

- Figure 69 C describes the judgement concerned with how straightforward to change the relations among family members. It is noticeable that 87.5% testers in all feel the Kinetic prototype is extremely straightforward to change relations, while they hold opposite opinions for this function of the JQuery prototype.
Figure 69 D Evaluation of Deleting

Figure 69 Evaluation of Editing Function

Figure 69 D illustrates the assessment about how easy to delete family members. Through comparing the two prototypes, it is easy to find there is no obvious diversity for this functions of the two prototypes.

8.3.3 Inserting Function

Figure 70 outlines the evaluation about how easy to insert personal information by the two prototypes. It is clear to find that half people feel it is neither easy nor difficult to perform this function of the JQuery prototype. However, 87.5% people feel it is highly straightforward to manipulate this function using the Kinetic prototype.
8.3.4 Resizing Function

Figure 71 illustrates the evaluation of participants for the resize function. The definition of the x axis and the y axis is equal to the former one. The evaluation for the resize function is made up by two sub-sections.

A Evaluation of Resizing family members

- Figure 71 A describes the assessment about how easy to resize family members using the two prototypes. From the figure, it is distinct that half participants think the extent of ease about performing this function by the JQuery prototype is middle, while 81.25% participants think the performance of the Kinetic prototype is better than the JQuery one.

B Evaluation of Updating connection automatically

Figure 71 Evaluation of Resizing Function

- Figure 71 B tells the assessment about how easy to automatically update connections after
resizing family members. It is obvious that the majority of people (12 in all) consider it is extremely easy to automatically update connections utilising the Kinetic prototype. Yet, the opinions of the JQuery prototype is opposite.

8.4 Summary

Eventually, the evaluation proves the analysis concerned with the four libraries (JQuery, Dojo, mxGraph and Kinetic) are correct and accurate. In section 6, the report suggests the Kinetic may be the best choice for diagrammatical user interfaces. This assessment is proved through three aspects:

- The author proves this conclusion from the process of developing the two prototypes. The author considers the functionality of Kinetic is stronger than that of JQuery, and the complexity of Kinetic is less than that of JQuery.

- The evaluation of testers proves using the Kinetic prototype to manipulate tasks is much easier than JQuery. The strengths of the Kinetic prototype are centralized on the functions of editing and resizing. The two prototypes do not have much differences on the remaining functions, but it is obvious that the Kinetic prototype is a little better than the JQuery prototype.

- The comments of testers also prove the assessment. The comments about the JQuery prototype is to advice the author to improve the functionality, while the comments about the Kinetic prototype is to suggest the author to enhance the usability of the interface.

Although the analysis of selected four libraries is proved to be correct, the judgment of the author and testers has limitations. The author judges the two prototypes based on the procedure of developing, but the assessment of the author could be effected by the limited experience, the subjective perspective and the insufficiently found resources. Lacked relevant computer background and misunderstood the manipulations of the functions may influence the judgment of testers.
9 Conclusion

This project aims at analysing client-side technologies for diagrammatical user interfaces. In order to achieve this objective, this project covers four stages, namely the survey stage, the analytical stage, the prototype stage and the evaluation phase.

- At the beginning of the survey phase, the diagram of family-trees is chosen as the analysed diagram. The reason is the project found that genealogy has a large number of common features of diagrams after investigated a series of genealogical applications. Then, this project investigated currently popular web technologies, including HTML, CSS, JavaScript, PHP, JavaFX and Java Applet. After surveyed, this project selects the JavaScript as the analysed technology due to its popularity.

- In the analysed phase, this project analyses the four promising JavaScript libraries, namely JQuery, Dojo, mxGraph and Kinetic. The analysis of the four libraries is based on the basic features of genealogical applications. After analysis, this project compares the four libraries from five aspects, that are ease of use, ease of learn, functionality, focus and licence. Through comparison, Kinetic seems to be the best choice applied to diagrammatical user interfaces, and JQuery is the second.

- The purpose of the prototypes is to prove whether the conclusion of the analytical phase is correct or not. In the prototype phase, two prototypes are developed by Kinetic and JQuery respectively. The functions of the two prototype are the same as the basic features of genealogical applications.

- In the evaluation phase, the judgement is proved to be correct from three aspects. The assessment of the author, the evaluations and comments of testers all demonstrate "The Kinetic library is a relatively satisfying choice for diagrammatical user interface".

In conclusion, this project recommended the Kinetic library could be a acceptable library for diagrammatical user interfaces. However, this conclusion has a set of limitations. Enormous factors could influence the accuracy of this conclusion.

- In the survey phase, the universality of investigated range and the selection of analysed
diagram have great effect on the final conclusion. Except this, the selection of basic features of genealogical tools has the fundamental influence on analysis.

- In the analytical phase, the accuracy and adequacy of found materials and the range of selected technologies could influence the final conclusion.
- In the prototype phase, the experience of the author and the defined functions could affect the functionality of Kinetic and JQuery to be shown.
- In the evaluation phase, the objectivity of the author's assessment, the background and the ability of understanding the performed tasks of tester are able to affect the final conclusion.

Therefore, for the future work, the accuracy and correctness of the final conclusion could be improved. To enhance the accuracy and correctness of the recommendation, there are two ways, namely enlarge the investigated and analysed range and deepen the depth of analysis. People could add new technologies to enlarge the range of surveying and analysing, such as Flash, Raphael library, Joint JS library and so on. Besides this, people could add the analysed functions and compared aspects. People could deepen the depth of analysis, for example, the speed of performing functions, the algorithm of performing functions and so on.
Reference


Appendix A

These are screenshots of analysing selected technologies. These screenshots include related code and its corresponding result.

```
$(function() {
  $( "#draggable" ).draggable({ axis: "y" });
  $( "#draggable2" ).draggable({ axis: "x" });
  $( "#draggable3" ).draggable({ containment: "#containment-wrapper", scroll: false });
  $( "#draggable5" ).draggable({ containment: "parent" });
});
```

*Figure1 Constrain Movement(JQuery)*

```
$(function() {
  $( "#draggable" ).draggable({ distance: 20 });
  $( "#draggable2" ).draggable({ delay: 1000 });
  $( ".ui-draggable" ).disableSelection();
});
```

*Figure2 Delay Movement(JQuery)*

```
$(function() {  
  var $start_counter = $( "#event-start" ),
  $drag_counter = $( "#event-drag" ),
  $stop_counter = $( "#event-stop" ),
  counts = [ 0, 0, 0 ];

  $( "#draggable" ).draggable({
    start: function() {
      counts[ 0 ]++;
      updateCounterStatus( $start_counter, counts[ 0 ] );
    },
    drag: function() {
      counts[ 1 ]++;
      updateCounterStatus( $drag_counter, counts[ 1 ] );
    },
    stop: function() {
      counts[ 2 ]++;
      updateCounterStatus( $stop_counter, counts[ 2 ] );
    }
  });

  function updateCounterStatus( $event_counter, new_count ) {
    // first update the status visually...
    if ( !$event_counter.hasClass( "ui-state-hover" ) ) {
      $event_counter.addClass( "ui-state-hover" ).siblings().removeClass( "ui-state-hover" );
    }
    // then update the numbers
    $( "span.count", $event_counter ).text( new_count );
  }
});
```

*Figure3 Count Movement(JQuery)*
Client-side Technologies for diagrammatical user interfaces

```javascript
options rootItem = rootItem;
options cursorItem = rootItem;
options templates = [getContactTemplate()];
options onItemHeader = onTemplateHeader;

jQuery('.basicDiagram').orgDiagram(options);

function onTemplateHeader(event, data) {
    var hrefElement = data.element.find("[name='Readers']");
    switch (data.renderingMode) {
        case primitives.common.RenderingMode.Create:
            /* Initialize widget here */
            hrefElement.click(function (e) {
                // Block mouse click propagation in order to avoid layout updates before server postback/
                primitives.common.stopPropagation(e);
            });
            break;
        case primitives.common.RenderingMode.Update:
            /* Update widget here */
            break;
    }
    var itemConfig = data.content;
    if (data.templateName == 'contactTemplate') {
        data.element.find("[name='photo']").attr({"src": itemConfig.image });
        hrefElement.attr({"href": itemConfig.href });
        var fields = ['title', 'description', 'phone', 'email'];
        for (var index = 0; index < fields.length; index++) {
            var field = fields[index];
            var element = data.element.find("[name=" + field + "]");
            if (element.text() != itemConfig[field]) {
                element.text(itemConfig[field]);
            }
        }
    }
}

function getContactTemplate() {  
    var result = new primitives.orgDiagram.TemplateConfig();
    result.name = "contactTemplate";
    result.itemSize = new primitives.common.Size(200, 100);
    result.minimizedItemSize = new primitives.common.Size(0, 3);
    result.highlightPadding = new primitives.common.Thickness(2, 2, 2, 0);
    var itemTemplate = jQuery("<div class="b-item-b-corner-all b-item-frame"/>
    + "<div class="b-item-b-corner-all b-item-title-frame" style="top: 2px; left: 2px; width: 216px; height: 20px;"/>
    + "<div class="b-item-title" style="top: 3px; left: 6px; width: 200px; height: 16px;"/>
    + "<div class="b-item-photo-frame" style="top: 26px; left: 2px; width: 50px; height: 50px;"/>
    + "<div class="b-item-photo" style="top: 25px; left: 6px; width: 44px; height: 44px;"/>
    + "<div class="b-name" class="b-item-frame" style="top: 62px; left: 2px; width: 162px; height: 162px; font-size: 12px;"/>
    + "<div class="b-description" class="b-item-frame" style="top: 62px; left: 56px; width: 162px; height: 162px; font-size: 12px;"/>
    + "<div class="b-readers" class="b-item-frame" style="top: 104px; left: 4px; width: 212px; height: 122px; font-family: Arial;"/>
    + "<div style="width: result.itemSize.width + "px"; height: result.itemSize.height + "px"">
        result.itemTemplate = itemTemplate.wrap("<div>").parent().html();
        return result;
    }";/]>
</script>

Figure 4 JQuery Link Function(code)[28]

87
Client-side Technologies for diagrammatical user interfaces

---

**Figure 5** Constrain Resize Area (jQuery)

**Figure 6** Constrain Resize Area (jQuery result)

```html
<html lang="en">
<head>
  <meta charset="utf-8"/>
  <title>jQuery UI Resizable - Constrain resize area</title>
  <link rel="stylesheet" href="http://code.jquery.com/ui/1.10.3/themes/smoothness/jquery-ui.css"/>
  <script src="http://code.jquery.com/jquery-1.9.1.js"></script>
  <script src="http://code.jquery.com/ui/1.10.3/jquery-ui.js"></script>
  <link rel="stylesheet" href="/resources/demos/style.css"/>
</head>
<style>
  #container
  { width: 300px; height: 300px; margin-bottom: 10px; }
  #resizable
  { background-position: top left; width: 150px; height: 150px; }
  #resizable, #container
  { padding: 0.5em; }
</style>
<script>
  $(function()
  {
    $('#resizable').resizable(
      { containment: '#container' }
    );
  });
</script>
<body>
  <div id="container" class="ui-widget-content">
    <h3 class="ui-widget-header">Containment</h3>
    <h3 class="ui-widget-header">Resizable</h3>
  </div>
</body>
</html>

---

88
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Resizable - Maximum / minimum size</title>
  <link rel="stylesheet" href="http://code.jquery.com/ui/1.10.3/themes/smoothness/jquery-ui.css" />
  <script src="http://code.jquery.com/jquery-1.9.1.js"></script>
  <script src="http://code.jquery.com/ui/1.10.3/jquery-ui.js"></script>
  <link rel="stylesheet" href="resources/demos/style.css" />
  <style>
    .Resizable { width: 280px; height: 150px; padding: 5px; }
    .Resizable h3 { text-align: center; margin: 0; }
  </style>
  <script>
    $(function() {
      $('.Resizable').resizable({
        maxHeight: 250,
        maxWidth: 350,
        minHeight: 150,
        minWidth: 200
      });
    });
  </script>
</head>
<body>
<h3 class="ui-widget-content">Resize larger / smaller</h3>
</body>
</html>

Figure 7 Maximum/Minimum Resize (jQuery) [27]

---

Figure 8 Simultaneous Resize (jQuery) [27]
Tooltips can be attached to any element. When you hover the element with your mouse, the title attribute is displayed in a little box next to the element, just like a native tooltip.

But as it's not a native tooltip, it can be styled. Any themes built with ThemeRoller will also style

WE ASK FOR YOUR AGE ONLY FOR STATISTICAL PURPOSES.

Your age: ___

Hover the field to see the tooltip.

Figure 10 Customized style Result (JQuery)[27]

Figure 11 Customized Content Result (JQuery)[27]
The below is the code of linking objects in analysed section. The code is written by Kinetic.

```javascript
this.render = function(ctx, depth, offsetX, offsetY, $node, $parent, color, zoomFactor) {

  offsetX = offsetX * zoomFactor;
  offsetY = offsetY * zoomFactor;

  var pw = $parent.width();
  var nw = $node.width();
  var pih = $parent.innerHeight();
  var nih = $node.innerHeight();

  // line is drawn from node to parent
  // draw direction
  var leftToRight, topToBottom;

  // node overlaps with parent above or delow
  var overlap = false;

  // canvas attributes
  var left, top, width, height;
  var bColor;

  // position relative to parent
  var nodeLeft = offsetX + nw / 2 < pw / 2;
  if (nodeLeft) {
    var aOffsetX = Math.abs(offsetX);
    if (aOffsetX > nw) {
      // normal left

      // make it one pixel too wide to fix firefox rounding issues
      width = aOffsetX - nw + 1;
      left = nw;
      leftToRight = true;

      // bColor = "red";
    } else {
      // left overlap
      left = -offsetX;
      width = nw + offsetX;
      leftToRight = false;
      overlap = true;
    }
  } else {
    // right overlap
    right = offsetX;
    width = nw - offsetX;
    rightToLeft = true;
    overlap = true;
  }
```

91
// bColor = "orange";

} else {
    if (offsetX > pw) {
        // normal right

        // make it one pixel too wide to fix firefox rounding issues
        width = offsetX - pw + 1;
        left = pw - offsetX;
        leftToRight = false;

        // bColor = "green";
    } else {
        // right overlap
        width = pw - offsetX;
        left = 0;
        leftToRight = true;
        overlap = true;

        // bColor = "yellow";
    }
}

var lineWidth = mindmaps.CanvasDrawingUtil.getLineWidth(zoomFactor,
    depth);
var halfLineWidth = lineWidth / 2;

// avoid zero widths
if (width < lineWidth) {
    width = lineWidth;
}

var nodeAbove = offsetY + nih < pih;
if (nodeAbove) {
    top = nih;
    height = $parent.outerHeight() - offsetY - top;

    topToBottom = true;
} else {
    top = pih - offsetY;
    height = $node.outerHeight() - top;

    topToBottom = false;
}
// fire before draw event
this.beforeDraw(width, height, left, top);

// determine start and end coordinates
var startX, startY, endX, endY;
if (leftToRight) {
    startX = 0;
    endX = width;
} else {
    startX = width;
    endX = 0;
}

// calculate difference in line width to parent node
// and position line vertically centered to parent line
var pLineWidth = mindmaps.CanvasDrawingUtil.getLineWidth(zoomFactor, depth - 1);
var diff = (pLineWidth - lineWidth) / 2;

if (topToBottom) {
    startY = 0 + halfLineWidth;
    endY = height - halfLineWidth - diff;
} else {
    startY = height - halfLineWidth;
    endY = 0 + halfLineWidth + diff;
}
Appendix B

This is a questionnaire which is used to test the practical performance of prototypes. There are two prototypes, namely JQuery and Kinetic.

<table>
<thead>
<tr>
<th>Evaluation Questionnaire for Family-tree Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender:   ☐ Male / ☐ Female</td>
</tr>
<tr>
<td>Familiar with family-tree app: ☐ Yes / ☐ No</td>
</tr>
<tr>
<td>Familiar with graphical interface: ☐ Yes / ☐ No</td>
</tr>
<tr>
<td>Date: ____________________________</td>
</tr>
</tbody>
</table>

**Note:**

1. Each tester could have 4 different function cards, which contains demanded functions individually.

2. Please read each case card to evaluate respective functions of two different interface.

3. Please fill questionnaire for each case card.
## Family-tree Interface Instruction

### JQuery Interface

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add new family members</td>
<td>Users could update information of their own family by left clicking the update button. Users could click add button after they insert personal information. Then, new family members will be showed in the browser. User could use the left click of mouse to move family members.</td>
</tr>
<tr>
<td>Edit family members</td>
<td>Users could move the children to their parents. Then, they will be linked by lines. When the mouse moves over the family members, a table will be showed. In the table, users click the delete button, then, the family member will be deleted.</td>
</tr>
<tr>
<td>Insert Personal Information</td>
<td>Users could hand-insert personal information in the right table in the browser. When the mouse moves over the family members, the personal information will be showed.</td>
</tr>
<tr>
<td>Resize family members</td>
<td>Users could click zoom in/zoom out button to resize family members.</td>
</tr>
<tr>
<td>Kinetic Interface</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Add new family members</td>
<td>User click new node button, then new family member will be presented. Users could use the left click of mouse to move family members.</td>
</tr>
<tr>
<td>Edit family members</td>
<td>When the mouse moves over the family members, a table will be showed. In the table, users could select the spouse and parents for this person. Blue line shows the relationship of parents. Green line shows the relationship of couple. Besides this, users could delete family members through clicking delete button in the table.</td>
</tr>
<tr>
<td>Insert Personal Information</td>
<td>When the mouse moves over the family members, a table will be showed. In the table, users could hand-insert personal information. After users input information, users could click save button to save information. If users want to update information, they could change personal information in the table, then save it.</td>
</tr>
<tr>
<td>Resize family members</td>
<td>The width of rectangles which stand for family members could change automatically depended on the width of name.</td>
</tr>
</tbody>
</table>
**Function Card**

**Description:** Users create new family members through clicking add button, then move created family members by mouse. When user would like to add new family members, they are able to insert personal information for new members.

<table>
<thead>
<tr>
<th>The interface you used for evaluation</th>
<th>JQuery</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>The Interface is easy to:</strong></td>
<td></td>
</tr>
<tr>
<td><em>Create New Family Members</em></td>
<td></td>
</tr>
<tr>
<td>Strongly Agree</td>
<td>6</td>
</tr>
<tr>
<td>Strongly Disagree</td>
<td>1</td>
</tr>
<tr>
<td>Disagree</td>
<td>2</td>
</tr>
<tr>
<td>Strongly Agree</td>
<td>5</td>
</tr>
<tr>
<td><em>Move Family Members</em></td>
<td></td>
</tr>
<tr>
<td>Strongly Agree</td>
<td>6</td>
</tr>
<tr>
<td>Strongly Disagree</td>
<td>1</td>
</tr>
<tr>
<td>Disagree</td>
<td>2</td>
</tr>
<tr>
<td>Strongly Agree</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>The interface you used for evaluation</th>
<th>Kinetic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>The Interface is easy to:</strong></td>
<td></td>
</tr>
<tr>
<td><em>Create New Family Members</em></td>
<td></td>
</tr>
<tr>
<td>Strongly Agree</td>
<td>6</td>
</tr>
<tr>
<td>Strongly Disagree</td>
<td>1</td>
</tr>
<tr>
<td>Disagree</td>
<td>2</td>
</tr>
<tr>
<td>Strongly Agree</td>
<td>5</td>
</tr>
<tr>
<td><em>Move Family Members</em></td>
<td></td>
</tr>
<tr>
<td>Strongly Agree</td>
<td>6</td>
</tr>
<tr>
<td>Strongly Disagree</td>
<td>1</td>
</tr>
<tr>
<td>Disagree</td>
<td>2</td>
</tr>
<tr>
<td>Strongly Agree</td>
<td>5</td>
</tr>
</tbody>
</table>

**Which interface do you think it is the most useful?**  
- □ JQuery  
- □ Kinetic

**Did JQuery interface provide you with all functions you need?**

**Did Kinetic interface provide you with all functions you need?**
**Function Card**

**Description**: Users link the related family members through given relationship. For JQuery interface, users use lines to link them. For Kinetic interface, users link family members through selecting their parents and spouse. Users could change relationships among family members. Then, user delete family members and their relationship through clicking delete button.

<table>
<thead>
<tr>
<th>The interface you used for evaluation</th>
<th>JQuery</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>The Interface is easy to:</strong></td>
<td></td>
</tr>
<tr>
<td>Link family Members</td>
<td>Strongly Disagree 1 2 3 4 5 6</td>
</tr>
<tr>
<td>Move Family Members after linked</td>
<td>Strongly Disagree 1 2 3 4 5 6</td>
</tr>
<tr>
<td>Change relationship</td>
<td>Strongly Disagree 1 2 3 4 5 6</td>
</tr>
<tr>
<td>Delete Family Members</td>
<td>Strongly Disagree 1 2 3 4 5 6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>The Interface you used for evaluation</th>
<th>Kinetic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>The Interface is easy to:</strong></td>
<td></td>
</tr>
<tr>
<td>Link family Members</td>
<td>Strongly Disagree 1 2 3 4 5 6</td>
</tr>
<tr>
<td>Move Family Members after linked</td>
<td>Strongly Disagree 1 2 3 4 5 6</td>
</tr>
<tr>
<td>Change relationship</td>
<td>Strongly Disagree 1 2 3 4 5 6</td>
</tr>
<tr>
<td>Delete Family Members</td>
<td>Strongly Disagree 1 2 3 4 5 6</td>
</tr>
</tbody>
</table>

**Which interface do you think it is the most useful?**  
- [ ] JQuery  
- [ ] Kinetic

**Did JQuery interface provide you with all functions you need?**  
- YES  
- NO  
- PARTIALLY

**Did Kinetic interface provide you with all functions you need?**  
- YES  
- NO  
- PARTIALLY
### Function Card

**Description:** Users could insert personal information (year of birth, gender and status) for every family members. Then, when the mouse moves over the family members, the personal information will be showed. Beside this, users could update personal information.

<table>
<thead>
<tr>
<th>The interface you used for evaluation</th>
<th>JQuery</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>The Interface is easy to:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Insert personal information</strong></td>
<td>Strongly: 1 2 3 4 5 6</td>
</tr>
<tr>
<td></td>
<td>Strongly: Agree</td>
</tr>
<tr>
<td><strong>Show personal information</strong></td>
<td>Strongly: 1 2 3 4 5 6</td>
</tr>
<tr>
<td></td>
<td>Strongly: Agree</td>
</tr>
<tr>
<td><strong>Update personal information</strong></td>
<td>Strongly: 1 2 3 4 5 6</td>
</tr>
<tr>
<td></td>
<td>Strongly: Agree</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>The interface you used for evaluation</th>
<th>Kinetic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>The Interface is easy to:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Insert personal information</strong></td>
<td>Strongly: 1 2 3 4 5 6</td>
</tr>
<tr>
<td></td>
<td>Strongly: Agree</td>
</tr>
<tr>
<td><strong>Show personal information</strong></td>
<td>Strongly: 1 2 3 4 5 6</td>
</tr>
<tr>
<td></td>
<td>Strongly: Agree</td>
</tr>
<tr>
<td><strong>Update personal information</strong></td>
<td>Strongly: 1 2 3 4 5 6</td>
</tr>
<tr>
<td></td>
<td>Strongly: Agree</td>
</tr>
</tbody>
</table>

**Which interface do you think it is the most useful?**  
☐ JQuery  ☐ Kinetic

**Did JQuery interface provide you with all functions you need?**  
[ ] YES  [ ] NO  [ ] PARTIALLY

**Did Kinetic interface provide you with all functions you need?**  
[ ] YES  [ ] NO  [ ] PARTIALLY

---

100
**Function Card**

**Description**: Users could resize the family members in order to see the name of family members clearly. For JQuery, users resize family members through clicking zoom in / out. For kinetic, the width of rectangles could be changed depend on the length of names.

<table>
<thead>
<tr>
<th>The interface you used for evaluation</th>
<th>JQuery</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>The Interface is easy to</strong></td>
<td></td>
</tr>
<tr>
<td>Resize family members</td>
<td>Strongly Agree</td>
</tr>
<tr>
<td>See the name of family members</td>
<td>Strongly Agree</td>
</tr>
<tr>
<td>Update the lines automatically</td>
<td>Strongly Agree</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>The interface you used for evaluation</th>
<th>Kinetic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>The Interface is easy to</strong></td>
<td></td>
</tr>
<tr>
<td>Resize family members</td>
<td>Strongly Agree</td>
</tr>
<tr>
<td>See the name of family members</td>
<td>Strongly Agree</td>
</tr>
<tr>
<td>Update the lines automatically</td>
<td>Strongly Agree</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Which interface do you think is the most useful?</th>
<th>JQuery</th>
<th>Kinetic</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Did JQuery interface provide you with all functions you need?</th>
<th>YES</th>
<th>NO</th>
<th>PARTIALLY</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Did Kinetic interface provide you with all functions you need?</th>
<th>YES</th>
<th>NO</th>
<th>PARTIALLY</th>
</tr>
</thead>
</table>

**Comments on your rating for this interface:**