

**2012**

**University of North Carolina Wilmington  
Master of Science in  
Computer Science and Information Systems  
Proceedings**

**<https://csbapp.uncw.edu/mscsis>**

PROPHET EXPLORER – A FULL-TEXT SEARCH MVC APPLICATION TO COMPARE  
THE SAYINGS OF RELIGIOUS PROPHETS

Zachary J. Wilson

A Capstone Project Submitted to the  
University of North Carolina Wilmington in Partial Fulfillment  
of the Requirements for the Degree of  
Master of Science

Department of Computer Science  
Department of Information Systems and Operations Management

University of North Carolina Wilmington

2012

Approved by

Advisory Committee

---

Dr. Curry Guinn

---

Dr. George Zervos

---

Dr. Tom Janicki, Chair

Accepted By

---

Dean, Graduate School

## Abstract

Prophet Explorer – A Full-Text Search MVC Application to Compare the Sayings of Religious Prophets.  
Wilson, Zachary, 2012. Capstone Paper, University of North Carolina Wilmington.

Prophet Explorer is a web-based application that provides the ability to investigate a collection of sayings that have been attributed to three of the most celebrated *prophets*: Buddha, Jesus, and Muhammad. A main feature of Prophet Explorer is a keyword search that enables the comparison of each *prophet's* sayings in a side-by-side format. This, along with other features, presents a rich exercise in comparative religion and helps answer the important question of “What did these *prophets* say?”

## Table of Contents

	Page
1 Introduction .....	1
2 Literature Review .....	2
2.1 Other Search Techniques .....	2
2.2 Keyword Processing .....	4
2.2.1 Synonyms .....	4
2.2.2 Stopwords .....	5
2.2.3 Stemming .....	6
2.3 Searching Religious Texts .....	6
2.3.1 Biblical Concordances .....	6
2.3.2 Historicity of Sayings .....	6
3 Similar Systems .....	7
4 Development Methodologies .....	8
5 Technologies Employed .....	9
5.1 ASP.NET MVC .....	9
5.2 HTML5 .....	11
5.3 MS SQL Server Full-Text Search .....	12
6 Implementation .....	13
6.1 Database of Sayings .....	13
6.2 Application .....	14
6.2.1 Keyword Search .....	14
6.2.2 Advanced Keyword Search .....	15
6.2.3 Concordance .....	15

6.2.4 Authentic Sayings .....	16
6.2.5 Mobile Devices .....	17
6.3 LIKE vs. CONTAINS .....	17
7 Target Audience .....	19
8 Future Work .....	19
8.1 Adding Prophets .....	20
8.2 Authenticity Highlighting .....	20
8.3 Language Concerns .....	20
9 Conclusion .....	21
10 References .....	21

## Figures

1. Keyword Search Results .....	1
2. MS SQL Server Thesaurus File .....	5
3. Stopwords .....	5
4. Authenticity Labels .....	7
5. Rapid Application Development .....	9
6. HTML Helper Methods .....	10
7. Model View Controller .....	11
8. Full-Text CONTAINS Query .....	12
9. Database Diagram .....	14
10. Multiple Keyword Search .....	15
11. Word Frequency Diagram .....	16
12. Authentic Highlighting .....	16

13.	LIKE vs. CONTAINS .....	18
-----	-------------------------	----

## Appendixes

A.	Desktop and Mobile Comparison .....	23
B.	Keyword Search Pages .....	25
C.	Sayings Use Case .....	26
D.	CRUD Diagram .....	26
E.	Hosting Strategies .....	27
F.	Survey Results .....	27

## 1 Introduction

This project involves the analysis and development of a system that provides the ability to search a collection of sayings that have been attributed to three of the most celebrated *prophets*: Buddha, Jesus, and Muhammad. Since one of the goals of this system is to encourage an exploration of the sayings collection, it has been named Prophet Explorer. A primary feature of Prophet Explorer is a keyword search that allows users to compare the sayings of each *prophet* in a side-by-side format. For example, searching for the word “**good**” would include the results shown in Figure 1 below.

**Figure 1 – Keyword Search Results**

Buddha	Jesus	Muhammad
Let a man overcome anger by love, let him overcome evil by <b>good</b> ; let him overcome the greedy by liberality, the liar by truth!	I will ask you one thing: Is it lawful on the Sabbath to do <b>good</b> or to do evil, to save life or to destroy?	He who makes peace between the people by inventing <b>good</b> information or saying <b>good</b> things, is not a liar.

Another feature of Prophet Explorer is a concordance that allows users to explore the sayings collection by *selecting* a keyword rather than *entering* a keyword. A concordance is an alphabetical index containing the principal words of a collection. Concordances, like indexes in a book, allow for quick and easy lookup of words or topics. For example, to lookup all of the sayings that contain “**good**,” a user can select the letter “**G**,” and then select the word **good** to display the respective sayings.

The keyword search and concordance provide a rich exercise in comparative religion and help answer the important question of “What did these *prophets* say?” This is an important question for those affiliated with Buddhism, Christianity, or Islam, as well as those who have an appetite for religious research. Hence, the target audience for this application includes but is not limited to: Religion professors and students, religious enthusiasts, and anyone with a curiosity for religion.

The inspirations for this system were many but the main inspiration was that there are no *easily found* systems quite like it. There are many systems that allow for searching a particular religious collection, but these systems do not allow for a comparison across religions (e.g., Bible Gateway). Also, there are systems that allow for a side-by-side comparison of texts, but these systems only include texts related to a single religion (e.g., The Five Gospels Parallels). Furthermore, there is literature that shows a comparison of *prophets*, but it is not presented in

a format that is searchable (e.g., Jesus and Buddha: The Parallel Sayings). Knowing all of this was motivation to build Prophet Explorer and to research the best techniques for building an Information Retrieval application.

Information Retrieval describes the “techniques of storing and recovering and often disseminating recorded data especially through the use of a computerized system” (Merriam Webster, 2012). For this project, the data is stored in a MS SQL Server database and the means to recover query the data is provided via a web interface. This type of architecture is typical with business applications and is quite useful because it separates users from the complexities of how the data is stored. These complexities and how they were addressed is one of the topics discussed in this paper.

The paper proceeds as follows. In section 2, a literature review is presented that covers the following topics: other search techniques, keyword processing, and searching religious texts. In section 3, similar systems are discussed and compared. Section 4 describes the development methodologies that were used. In section 5, specific technologies employed are discussed; namely the Model View Controller (MVC) architecture, HTML5, and Full-Text Search (FTS). Section 6 presents the implementation process and the application features. Section 7 discusses the target audience and describes their feedback. Lastly, section 8 discusses future work and potential enhancements to the application.

## **2 Literature Review**

In preparation for undertaking this project, a literature review was conducted to examine what others have previously done in the fields of Information Retrieval and Religious Studies. Since the main feature of Prophet Explorer is a keyword search, it was necessary to research other applications that use keyword searches and determine how they store and retrieve data; this led to topics such as querying relational databases and indexing a large number of documents. Another concern of a keyword search tool is how to process the search phrase by examining the words included and their word properties; this brought about an investigation of synonymy, stemming, and stop-word removal. Also important to investigate was the history of searching religious texts; this led to an exploration of biblical concordances and an examination of textual criticism as it relates to the historicity of sayings.

### **2.1 Other Search Techniques**

The quintessential Information Retrieval system of our era is Google. In 2011, Google processed an average of 4.7 billion searches per day and maintained indexes for 46 billion pages (Google

Annual Search Statistics, 2012). The Google search engine was originally crafted by Sergey Brin and Larry Page while they were students at Stanford. In their seminal paper, they describe the search and ranking algorithms that Google utilizes (Brin & Page, 1998).

Google uses crawling technology to gather web documents and keep them up to date, ultimately storing each in a compressed form. In order to quickly return high quality results, Google maintains indexes and uses a page ranking algorithm (PageRank) to order results. The PageRank algorithm uses the link structure of the web to calculate the rank of each webpage. This rank measures the citation importance of a page by counting the number of backlinks to it. The justification of this ranking system is if a page has a lot of links that point to it, each having a high PageRank, then it must be worth viewing (Brin & Page, 1998).

The Google search algorithm is not directly applicable to Prophet Explorer but the PageRank algorithm does point out the need to rank results based on some sort of criterion. For Prophet Explorer, the criterion is how frequently a keyword appears in a saying. Hence, for a particular result set, the sayings that contain the highest frequency of a keyword are displayed before those with lower frequencies. A secondary way to rank results could be to rank the sayings thought to have a higher likelihood of being authentic, or from a more reputable source, higher in the result set. This has not been implemented but could be a useful addition to the ranking algorithm.

Another information retrieval system is DBXplorer, a keyword-based search tool for relational databases. Unlike document-based searches such as Google, DBXplorer focuses on extracting enterprise data from relational databases. The biggest challenge in retrieving data from relational databases is that logical units of data are typically spread across multiple tables (Argawal et al., 2002). Because of this, queries involving multiple joins must be performed in order to return results. This presents no issue for those who are familiar with a database's architecture, but for a typical user this complexity should be hidden; hence, the creation of DBXplorer.

DBXplorer supports conjunctive queries, which are queries that retrieve only results containing *all* query keywords. Prior to returning these results, DBXplorer ranks the results based on the number of table joins involved. The reasoning behind this ordering of results is that results containing more joins are typically harder to comprehend (Argawal et al., 2002). This way of ranking results is similar to other methods that give keywords occurring closer together, a higher preference.

Unlike DBXplorer, in Prophet Explorer the data being searched (i.e., the actual saying text) is confined to one data column. The joins that occur are to display the entire saying information (e.g., book and chapter), not to match keywords. Because of this simplicity, Prophet Explorer

can focus more heavily on other aspects of keyword searches such as broadening the result set using word properties.

## 2.2 Keyword Processing

One of the ways to expand the result set of a keyword search is to consider related words such as synonyms, hyponyms, and the like. Also important to consider are words that will not likely influence the search; namely stopwords. Stopwords are words that occur frequently and are not helpful in resolving searches (e.g., the). Another pertinent topic is stemming, or the process of generating inflectional forms of a word (e.g., running, ran, runner). In the sections that follow, each of these topics will be discussed in greater detail.

### 2.2.1 Synonyms

One of the reasons for using word properties such as synonymy is to broaden the result set of a search. For Prophet Explorer, this was brought about by the idea that if someone searches for the word “wise,” sayings that contain synonyms such as “keen” should be also returned. This is important from a comparative perspective because the collection of sayings for each *prophet* will not always have results for a particular keyword. For example, in the sayings collection, both Jesus and Buddha have sayings that contain “wise,” while Muhammad has no sayings containing “wise.” However, Muhammad does have sayings that contain the word “keen.” Hence, by including synonyms, results containing the same ideas can be returned and an easier comparison can be made.

In MS SQL Server 2008, synonyms can be included in a full-text search by using full-text queries that take into account thesaurus files. By default, thesaurus files for each supported language are provided in an XML format. The specific format of these files, as shown in Figure 2, allows for two types of custom sets: expansion and replacement. Expansion sets are typically used to group words that are synonyms like “wise” and “keen.” Replacement sets are used to replace words with other words such as the misspelling of “Budda” with “Buddha.”

**Figure 2 – MS SQL Server Thesaurus File**

```

<XML ID="Microsoft Search Thesaurus">
  <thesaurus xmlns="x-schema:tsSchema.xml">
    <diacritics_sensitive>0</diacritics_sensitive>
    <expansion>
      <sub>wise</sub>
      <sub>keen</sub>
    </expansion>
    <replacement>
      <pat>Budda</pat>
      <sub>Buddha</sub>
    </replacement>
  </thesaurus>
</XML>

```

### 2.2.2 Stopwords

Stopwords refer to a large number of words that occur frequently but add no meaning to a keyword search; therefore, they are usually ignored. Stopwords are typically words that linguists classify as articles, such as “the”, “a”, and “that.” Whether or not these words are included in search is irrelevant, as it will not influence the search results. For instance, a search for the “University of North Carolina Wilmington” should return the same results as the “University North Carolina Wilmington” (Cotter, 2006).

In MS SQL Server 2008 a default stop list is provided with the installation. This stop list contains 155 numbers, symbols, and words. If customizations are needed, the easiest way is to create a custom stop list from the default one and add or remove words. Then, a full-text index can be associated with the newly created list and used in queries. This is exactly what was done for the Prophet Explorer application; the resultant stop list is shown in Figure 3 below.

**Figure 3 – Stopwords**

```

!, #, $, %, &, (, ), *, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, @, a, A, B, C, D, E, F, G, H, I, J,
K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, ^, also, an, and, another, any, are, as,
at, be, because, been, between, both, but, by, came, can, come, could, did, do, does,
each, else, for, from, get, got, had, has, have, he, her, here, him, himself, his, how,
if, in, into, is, it, its, just, like, make, many, me, might, more, most, much, must, my,
no, not, now, of, on, only, or, other, our, out, over, re, said, same, she, should,
since, so, some, still, such, take, than, that, the, their, them, then, these, they,
this, those, through, to, too, under, up, use, very, want, was, way, we, well, were,
what, when, where, which, while, who, will, with, you, your

```

### 2.2.3 Stemming

Stemming is the process of generating inflectional forms of a word. For example, “running”, and “runner,” are all inflectional forms of the word “ran.” In MS SQL Server 2008, inflectional forms of a word can be generated using a “Forms of Inflectional” query. These queries use an algorithm similar to the notable Porter Stemming algorithm to generate declensions and conjugations of words. The algorithm works by reducing a word to its linguistic root then expanding it to include all linguistic forms of the root. For example, the word careers would be reduced to career, then expanded to career’s, careers’, careered, and careering. Afterward, a search would ensue using each of these expansions (Cotter, 2006).

## 2.3 Searching Religious Texts

For the last phase of the literature review, it was important to get a sense of the history of searching religious texts and to investigate prior work done by pioneers in the field of textual criticism. This is no doubt an area that has a long and complex history; hence, this paper makes no attempt to comprehensively explain it. Rather, a few events and individuals that are relevant to, or have influenced this project, are mentioned below.

### 2.3.1 Biblical Concordances

A concordance is an alphabetical index that contains the principal words of a collection. The idea behind a concordance is to provide an easy way to find words within documents; this is an idea that has been around for centuries. In the 1<sup>st</sup> century C.E., Philo Judaeus compiled a dictionary of biblical proper names in Greek. Between the 4<sup>th</sup> and 10<sup>th</sup> centuries C.E., the Masoretes created lists of biblical words in Hebrew. These documents provided inspiration for the first concordance of the Latin Bible published between 1230 and 1244 C.E. (Welberg, 1999). Today, one of the most widely used English concordances is Strong’s Exhaustive Concordance of the Bible. Strong’s concordance was compiled in the late 1800’s and contains an alphabetical listing of every word in the King James Bible (Strong, 2007). These documents, which were painstakingly created, provided the influence for a digital concordance in the Prophet Explorer application.

### 2.3.2 Historicity of Sayings

When dealing with religious texts it is important to understand their historical context and how they evolved. This is a concept that has been realized by researchers who have studied each of the *prophets* hereof. For this project though, only the historicity of Jesus’ sayings has been addressed. This decision was made to limit the project scope for this version of the application; however, future versions will address the historicity of *all* of the sayings (see section 8).

The quest for the historical sayings of Jesus began with Hermann Samuel Reimarus in the 18<sup>th</sup> century. Reimarus was convinced that what the authors of the Gospels reported Jesus to have said is distinguishable from what he actually said; hence, a distinction was made between “the man Jesus and the Christ of the creeds” (Funk, 1993, p. 2). The work of Reimarus inspired David Friedrich Strauss to write the monumental book, *Life of Jesus Critically Examined* (1835). Strauss, like Reimarus, distinguished from the mythical and historical Jesus. Unfortunately for Strauss though, his positions were ill-received in 19<sup>th</sup> century Germany, and cost him a teaching position at the seminary of Tübingen. These struggles by Strauss and his counterparts were chronicled by Albert Schweitzer in his book, *The Quest for the Historical Jesus* (Funk, 1993, p. 3). Each of these researchers influenced the work of The Jesus Seminar and their search for the authentic sayings of Jesus.

In their provocative book, *The Five Gospels – What did Jesus Really Say?*, Robert Funk, Roy Hoover, and The Jesus Seminar examine the sayings of Jesus in the four canonical Gospels and the non-canonical Gospel of Thomas. For each saying, the members of The Jesus Seminar casted ballots to classify its authenticity. Based on the votes, the sayings were placed into one of the four categories listed in Figure 4 below. The results of these votes, as shown by the color coding of sayings, is included in the Prophet Explorer application (see section 6.2.4).

#### Figure 4 - Authenticity Levels

\*\*Jesus undoubtedly said this or something very like it.

\*\*Jesus probably said something like this.

\*\*Jesus did not say this, but the ideas contained in it are close to his own.

\*\*Jesus did not say this; it represents the perspective or content of a later or different tradition.

### 3 Similar Systems

In order to learn what others have done previously in terms of religious applications, an investigation of similar systems was conducted. This was done by searching the web for systems that have to do with searching and comparing religious texts. In this investigation, the systems found to be most relevant to Prophet Explorer were The Five Gospel Parallels and Bible Gateway (The Five Gospel Parallels, 2001 & Bible Gateway, 2012). Each of these systems is a web application that is used to view texts associated with Christianity.

The Five Gospel Parallels is an application that allows users to view the following gospels in a side-by-side format: Mark, Matthew, Luke, John, and Thomas. The system was created by the Department for the Study of Religion at the University of Toronto. According to them, the

system is meant as a “teaching tool for introductory level classes in New Testament and Christian Origins” (The Five Gospel Parallels, 2001). One of the most impressive things about the system is the parallel relationships maintained between the gospels. For a given narrative or saying, if there is a parallel in another gospel, an icon is displayed next to it that serves as a link to the parallel text. Upon clicking this link, the focus of the linked book is taken to the saying. This allows users to compare similar narrations and sayings in multiple gospels. Overall, the system is great for comparing the most notable Christian gospels. Unlike Prophet Explorer though, it is bound to a single religion and does not allow users to perform keyword searches.

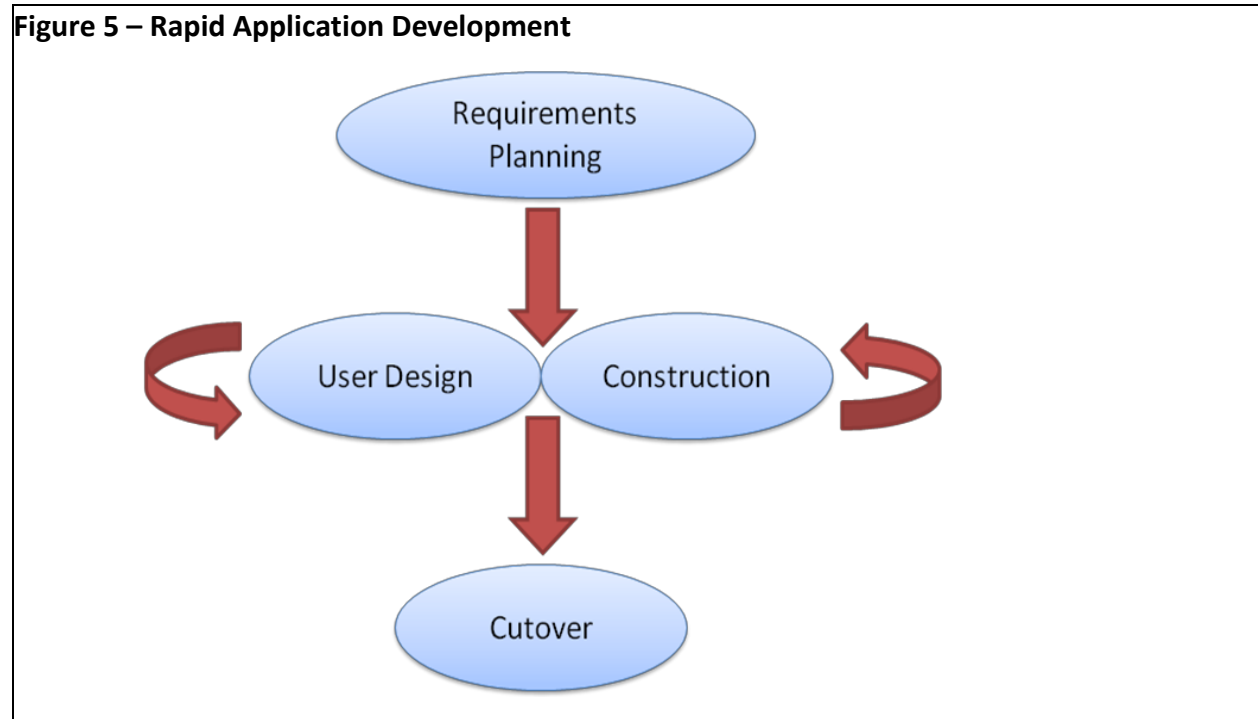
According to the Bible Gateway website, “Bible Gateway is a tool for reading and researching scripture online” (Bible Gateway, 2012). The tool provides numerous translations, which provide flexibility and the prospect of a larger audience. One of the most impressive features of Bible Gateway is the ability to filter the search. Users can search for a particular verse, keyword, or topic all on the same screen and toggle options such as verse headings, words of Jesus in red, footnotes, and citations. There is also an option to “add a parallel.” This option allows the addition of a parallel translation of the Bible next to the current translation. For instance, the King James Version and the New King James Version can be viewed in the same window using this feature. However, this only allows for comparing translations, not intra-comparisons; for example, a comparison of the same story in Mark and Matthew. This lack of intra-comparison is one of the biggest differences between Bible Gateway and Prophet Explorer and speaks to the goals of each. The goal of Bible Gateway is to make the Bible accessible to as many people in as many versions as possible. The goal of Prophet Explorer is to provide a comparative environment that presents only the sayings of *prophets*.

## 4 Development Methodologies

The main development methodology employed for building Prophet Explorer has been Rapid Application Development (RAD). RAD is a development life cycle designed for faster development and higher quality results than more traditional life cycles. RAD is also intended to take advantage of powerful development environments that have evolved recently and allow for speedier development (e.g., Microsoft Visual Studio). These development environments allow for rapid iterations of refinement and a faster transition from a prototype to a production system (Martin, 1991).

The RAD process, more precisely defined, consists of four phases: requirements planning, user design, construction, and cutover. The requirements planning phase consists of defining user and system requirements and is typically accomplished using Joint Application Design (JAD). The user design phase is when developers transform the system requirements into logical

descriptions and create documentation/diagrams to help model the system. The construction phase is when developers actually construct an initial prototype or update an existing one. Lastly, the cutover phase refers to finalizing and delivering the application (Yen & Davis, 1999). Each phase, as well as the iterative nature of RAD, is shown in Figure 5 below.



## 5 Technologies Employed

For this project, there has been a goal to use the latest technologies and standards that are widely used in industry. By doing so, important experience has been gained and the application has benefited. These technologies include ASP.NET MVC, HTML5, and MS SQL Server Full-Text Search. Each of these technologies is quite useful in a web-based information retrieval system and is described in detail below.

### 5.1 ASP.NET MVC

ASP.NET MVC is a framework that implements the Model-View-Controller architectural pattern. The MVC pattern has been around since the early 1980s and provides qualities such as reusability and separation of concerns. Reusability can be realized because MVC applications have three distinct parts that are easily decoupled, thereby enabling an individual part to be plugged into another application. Separation of concerns can greatly reduce the complexity of

an application as the business, presentation, and functional logic are separated. Each of these properties contributes to better maintainability.

*Models* in the MVC pattern are the objects that implement the logic for an application's data domain. Typically, *models* are direct representations of database objects such as a Product *model* that represents a Product table. A *model* such as this could be used to read product data from the Product table, populate the *model's* properties, and perform the necessary CRUD operations as needed. However, a *model* need not be a direct physical representation; it may be used as a conceptual separation. For instance, a *model* could simply read a dataset from the database and send this information to a *view* (ASP.NET MVC Overview, 2009).

*Views* in the MVC pattern are the components that display the user interface of an application. The information in the *view* is typically gotten from the data supplied by the *model*. For instance, in Prophet Explorer sayings are retrieved in a *model* object and displayed on the *view*. One of the best features in ASP.NET MVC that can be used in *views* is HTML helpers. HTML helpers are methods that generate a string and can be used to more easily create HTML elements such as textboxes, hyperlinks, and dropdown lists. This helps to keep *views* clean and speed development as shown in Figure 6 below (ASP.NET MVC Overview, 2009).

**Figure 6 - HTML Helper Methods**

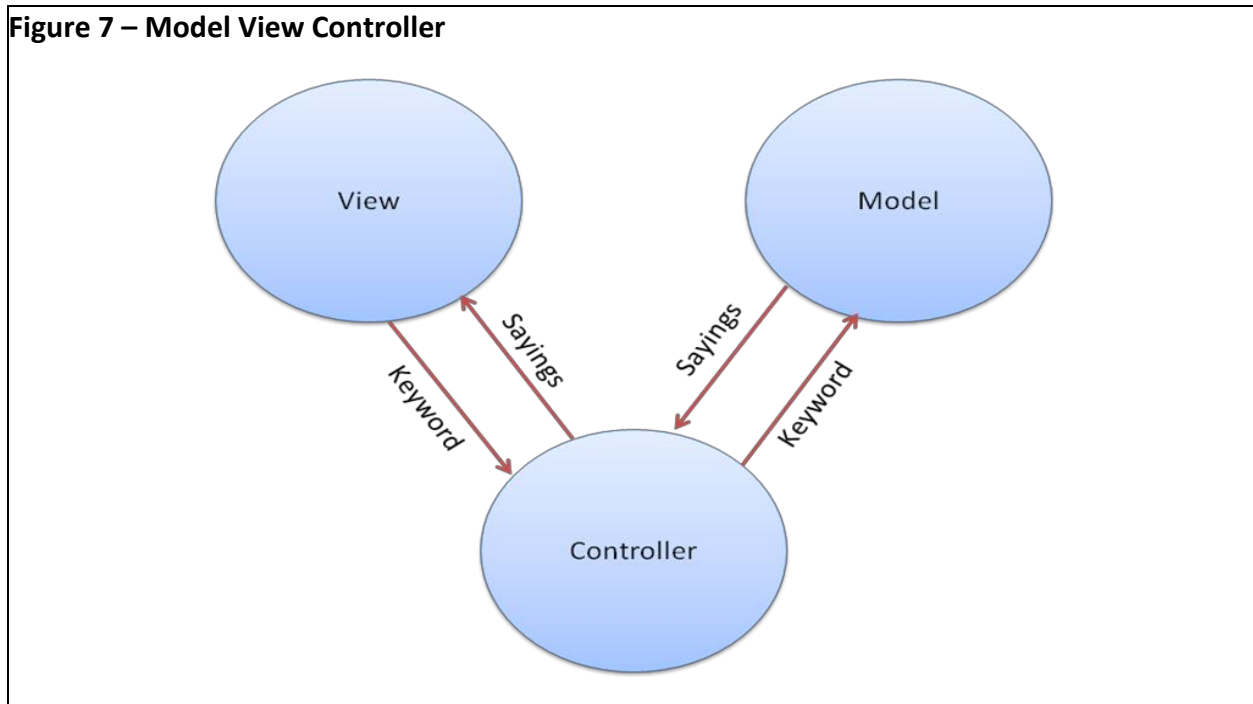
Helper Method	Normal HTML
<code>@Html.TextBox("keyword")</code>	<code>&lt;input type="text" name="keyword" /&gt;</code>

*Controllers* in the MVC pattern are the components that handle user interaction, work directly with the *model*, and specify which *view* to render. Controllers handle user interaction such as button clicks that invoke action events within a controller. Then, a *controller* event may interact with a *model* by sending and/or requesting data. Subsequently, a *controller's* action method can return a *view* in order to render the User Interface (ASP.NET MVC Overview, 2009).

In the Prophet Explorer application, one of the pages that best exemplifies the MVC pattern is the Search page. On this page, users are given the ability to enter a keyword and click a search button to view sayings containing the keyword. Upon clicking this button, the keyword (from the *view*), is sent to a *controller* class that creates a *model* instance with the keyword provided. This *model* then reads all of the sayings from the database and is returned to the *controller*. The

*controller* then sends the *model* back to the *view*, and the *view* displays all of the sayings that contain the keyword. This process is depicted in Figure 7 below.

**Figure 7 – Model View Controller**



## 5.2 HTML5

HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG). Previously, WHATWG was working with web forms, and W3C was working with XHTML 2.0. However, in 2006 they decided to combine their efforts and create a new version of HTML (WHATWG Community, 2012). The rules they established were as follows:

- New features should be based on HTML, CSS, DOM, and JavaScript
- Reduce the need for external plug-ins (e.g., Flash)
- Better markup handling
- More markup to replace scripting
- Be device independent
- The development process should be transparent - create API's

Visual Studio 2012 provides the ability to use HTML5 syntax by default. Because of this, the new HTML5 tags can easily be used in applications. In Prophet Explorer, tags such as <header>, <nav>, <article>, and <footer> have been used throughout. The only issue with using these new tags is that some of the older browsers do not support them. However, with each passing year this becomes less and less of a concern.

### 5.3 MS SQL Server Full Text Search

Full Text Search (FTS) is an indexing and query service provided by MS SQL Server. FTS gives users and applications the ability to run full-text queries on character-based data. Before running these queries, a full-text index must first be created on one or more columns of a table. Also, the column(s) used must be of one of the following data types: char, nchar, varchar, text, ntext, image, xml, varbinary, or filestream (Full-Text Search, 2012).

Once an index is created, users and applications can run full-text queries on the data in these columns. These queries can search for specific words, words beginning with a particular prefix, inflectional forms of a word, a word close to another word, synonym forms of a word, or words using weighted values (Full-Text Search, 2012). For the Prophet Explorer application, queries associated with specific words and synonym forms have been used. An example of a CONTAINS query to match specific words is shown in Figure 8 below.

**Figure 8 – Full-Text CONTAINS Query**

```

SELECT sayingID, sourceTitle, bookTitle, sectionTitle,
       sayingNumber, sayingText, prophetID,
       -- Figure out the keyword count
       (SELECT (LEN(sayingText) -
               LEN(REPLACE(sayingText, @keyword, ''))) /
        LEN(@keyword)) AS [keywordCount]
FROM Saying INNER JOIN
      Section ON Saying.sectionID = Section.sectionID
      INNER JOIN
      Book ON Section.bookID = Book.bookID
      INNER JOIN
      [Source] ON Book.sourceID = Source.sourceID
WHERE CONTAINS(sayingText, @wrapKeyword)
ORDER BY [keywordCount] DESC

```

As shown in the query above, a CONTAINS function is used to return all of the sayings that contain a keyword. The CONTAINS function takes two parameters: the column to be searched and a search phrase. Additionally, the second parameter may be wrapped with a FORMSOF command to perform a Thesaurus or Inflectional search. The main benefits of using these FTS functions include the ability to:

- disregard stopwords
- expand result sets using a thesaurus
- replace search words using expansion sets (e.g., IE to Internet Explorer)
- automatically stem keywords

- generate inflectional forms of a word

Each of these benefits makes FTS preferable to predicate searches for complex information retrieval systems. However, there is additional overhead to consider when using FTS. So much so that when building Prophet Explorer, a decision was made to host the application and database on its own virtual machine. But how much overhead does FTS really involve? Also, is the performance better than the LIKE predicate for small databases? These questions will be addressed in the Implementation section that follows.

## 6 Implementation

The implementation process was iterative and required multiple changes to produce each deliverable. The deliverables include: a database of sayings, a fully-functional application, and a comparison between the LIKE predicate and the CONTAINS function. Each of these deliverables is explained in subsections 6.1 – 6.3.

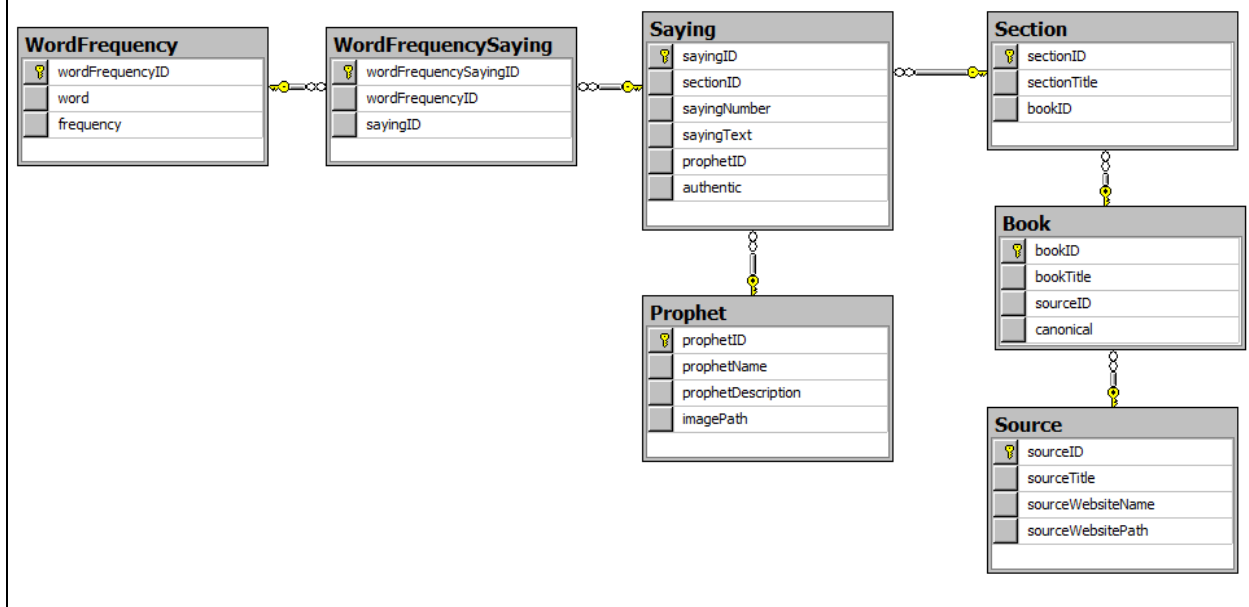
### 6.1 Database of Sayings

In order to get the sayings collection into the database, extensive preprocessing was done. This preprocessing included researching and reading religious texts and devising ways to parse out only the sayings. Each source and each *prophet* was different, but eventually every saying ended up in a text file separated by line breaks. Each text file was devoted to a book, and within the file tokens were used to signify the start of a new chapter or equivalent (e.g., \*Chapter 1). This format was then used as a base to write a program to read in files and insert sayings. The program involved a GUI that required the *prophet* and book to be selected, and incorporated the following logic:

- If a blank line is reached, skip it
- If a line beginning with "\*" is reached, insert a new chapter into the database
- If a line that does not start with "\*" is reached, insert the saying

There were multiple iterations of inserting sayings into the database. This was due to the addition of sources and updates to the database structure. To date there are 2,846 sayings in the database: 909 attributed to Buddha, 964 attributed to Jesus, and 973 attributed to Muhammad. Also there are 6,417 unique words in the Word Frequency table (excluding stopwords). The full database diagram is shown in Figure 9 below.

Figure 9 – Database Diagram



## 6.2 Application

The Prophet Explorer application was written in C# using Visual Studio 2012 and the MVC4 framework. The application has the following *main* features: a keyword search, an advanced keyword search, and a concordance. The application also includes the implementation of flagging authentic sayings and an alternative presentation for mobile devices. Each of these features, along with the work that went into creating them, is detailed below.

### 6.2.1 Keyword Search

The keyword search allows for comparing the sayings of each *prophet* in a tabbed format (see Appendix B). The tabbed format was chosen for spatial reasons and because it allows for less scrolling of the browser window. Keywords are gathered using a text box and are processed using a submit button. Once the submit button is clicked, the application processes the keyword string by separating each word with an AND operator. By using the AND operator, only sayings that contain each word in the keyword string are returned. For example, searching for “overcome anger by love” returns the saying shown in Figure 10. However, searching for “overcome anger by love *joy*” would not return the same saying because it does not contain the word *joy*.

### Figure 10 – Multiple Keyword Search

Let a man **overcome anger by love**, let him **overcome** evil **by** good; let him **overcome** the greedy **by** liberality, the liar **by** truth!

The query used for the keyword search is a CONTAINS query, as shown in Figure 8. This query returns a list of results ordered by the frequency of keyword hits. By using the CONTAINS query, the registered stop-list for the full-text index is used. Hence, words in the stop-list such as “the” are ignored and not used to return results. This may be confusing to users because a search for “the” does not return any results.

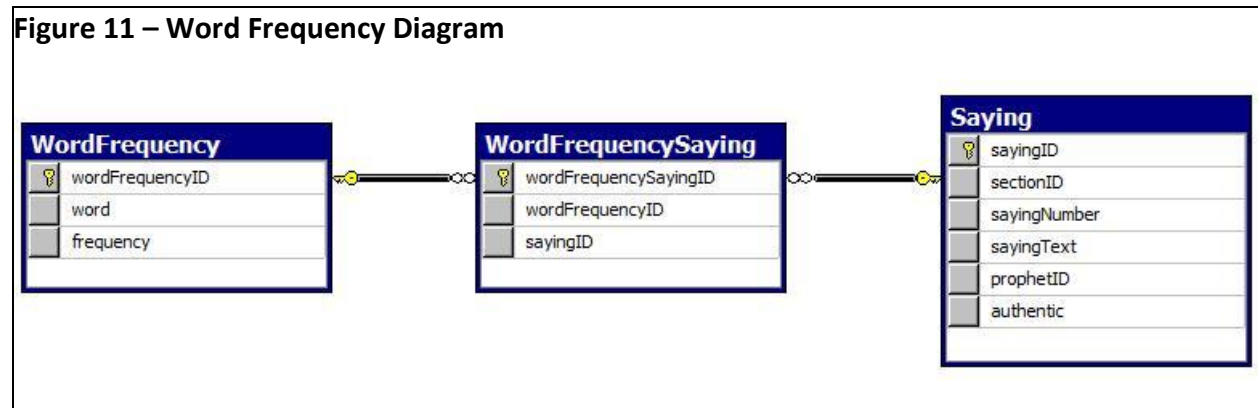
## 6.2.2 Advanced Keyword Search

An advanced keyword search was created to provide additional search options and more flexibility. These additional options include: 1) disregarding results with certain keywords, 2) including results with only *consecutive* keywords, *all* keywords, or *any* of the keywords, 3) including synonyms, 4) choosing what *prophets* to display, and 5) an option to narrow the search by book. Each of these options improves the usability of the application by providing the ability to broaden or narrow the search. Another difference between the default and advanced keyword search is the presentation layout. The layout of the default keyword search is tabbed, while the advanced layout is framed (see Appendix B). The framed layout allows for an easier comparison of sayings because the sayings of each *prophet* are shown in its own scrollable window.

When the option to include synonyms is selected, the program uses a query with a FORMSOF Thesaurus predicate to include the thesaurus file associated with the full-text index. For this project, rather than manually creating the thesaurus file, a program was written in Java to aid in its creation. This program uses WordNet and an API named JAWS (JAWS, 2009). WordNet is a lexical database for English that groups nouns, verbs, adjectives, and adverbs into synonym sets (WordNet, 2012). By using the JAWS API, these synonym sets can be easily retrieved, which is what was done in order to create the Thesaurus file. First, the program reads from the sayings database, which has a table that maintains the unique words in the sayings collection. Each of these words is then used to make a call to the WordNet API and retrieve a synonym set. Then, this synonym set is written to an expansion set as is shown in Figure 2. After the thesaurus file is created, the previous one is replaced and can then be used by used in full-text queries.

## 6.2.3 Concordance

The concordance feature allows users to search the sayings collection by *selecting* a keyword rather than *entering* one. This is accomplished using an interface that displays each letter in the alphabet. Upon clicking a letter, the application generates a list of all of the words in the sayings collection that begin with that letter. Then, by clicking a word in the list, all of the sayings that contain the word are displayed. This is accomplished by querying the three tables shown in Figure 11.



## 6.2.4 Authentic Sayings

As discussed earlier, the quest for the authentic sayings of Jesus is one that has been undertaken by many researchers and as one would expect, there are differing views on which sayings are authentic. For this project, the opinions of The Jesus Seminar have been used. Their book, *The Five Gospels*, highlights each saying of Jesus using color codes that represent its level of authenticity. In order to integrate this into the Prophet Explorer application, each saying of Jesus was surrounded with HTML tags. For instance, a saying in the *red* category was surrounded by a `<span class="headingred">` tag. This allows the application to display the color codes using CSS when the option is selected by the user. This option is presented on the Advanced Search page when Jesus is the only *prophet* selected. An example of a search for the keyword "good" is shown in Figure 12 below (see Figure 4 for a description of the color codes).

**Figure 12 – Authentic Highlighting – Keyword: “good”**

**New King James Bible - Luke - Chapter 6**

27 But I say to you who hear: Love your enemies, do good to those who hate you, 28 bless those who curse you, and pray for those who spitefully use you. 29 To him who strikes you on the one cheek, offer the other also. And from him who takes away your cloak, do not withhold your tunic either. 30 Give to everyone who asks of you. And from him who takes away your goods do not ask them back. 31 And just as you want men to do to you, you also do to them likewise.

## 6.2.5 Mobile Devices

There are many approaches to changing an application’s user interface for mobile devices. For example, some developers use JavaScript to detect the device type and redirect to a new page if a mobile device is detected. Others use CSS Mobile queries to apply different styles to elements when a mobile device is detected. For this application, a new feature in MVC4 called Display Mode has been used to make it more mobile-friendly.

The Display Mode feature allows different *views* to be rendered based on the browser making the request. By default, there is a Mobile display mode that renders *views* with the .mobile prefix before the file extension. For instance, in the Prophet Explorer application, there are two Index *views*: Index.cshtml and Index.mobile.cshtml. The Index.cshtml *view* is shown for desktop browsers; the Index.mobile.cshtml *view* is shown for mobile browsers.

One of the biggest benefits of using the Display Mode feature is that no code changes are needed to render different *views*. In other words, only the *views* change -- not the *models* or *controllers*. This speaks to one of the main benefits of the MVC architecture, which is separation of concerns. Another benefit of the Display Mode feature is that the URL remains the same for desktop and mobile browsers. This helps with Search Engine Optimization and eliminates the need to redirect to another page.

For Prophet Explorer, Mobile *views* were created to give mobile users a better experience by presenting a simpler and light-weight version of the application. This was accomplished by removing images and simplifying the user interface to decrease loading times. Also, some of the features are hidden and only the “have-to-have” features are preserved. These “have-to-have” features are 1) the Keyword Search, and 2) the View Sayings page. The Concordance and the Advanced Keyword Search pages were removed because they do not scale well and are too complicated to use on a mobile device. For a visual comparison between the desktop and mobile versions, refer to Appendix A.

### 6.3 LIKE versus CONTAINS

According to MSDN, a query using the LIKE predicate against millions of rows of text can take minutes to return while the same query using the CONTAINS predicate can take only seconds (Full-Text Search, 2012). However, in the Prophet Explorer database there are not millions of rows of text – there are only thousands. Hence, an experiment was run to compare LIKE and CONTAINS for a small database. For this experiment, queries were run in MS SQL Server using the query window and choosing to include the execution plan and client statistics. The goal of each of these queries was to return all of the sayings that contain the word “good.” This word was chosen because it is one of the most frequently used words in the collection, which forces the queries to return a lot of results.

As expected, the CONTAINS predicate outperformed the LIKE predicate but the performance was only marginally better with an execution time of 1.52 milliseconds compared to 1.58 milliseconds. More interesting were the statistics that describe the query execution plan. For the LIKE query, 99% of the effort was spent performing a clustered index scan as compared to 80% for CONTAINS. This suggests that although the CONTAINS index scan was likely quicker, there is additional overhead associated with joining results and additional functions used to match results. These results are summarized in Figure 13 below.

**Figure 13 – LIKE vs. CONTAINS**

	LIKE	CONTAINS
<b>Query Text</b>	<pre>SELECT sayingText FROM Saying WHERE sayingText LIKE '%good%'</pre>	<pre>SELECT sayingText FROM Saying WHERE CONTAINS(sayingText, 'good*')</pre>
<b>Average Execution Time</b>	1.58 milliseconds	1.52 milliseconds
<b>Query Execution Plan</b>	<b>Filter:</b> 1% cost <b>Clustered Index Scan:</b> 99% cost	<b>Table Valued Function:</b> Cost 7% <b>Merge Join:</b> Cost 13% <b>Clustered Index Scan:</b> Cost 80%

It is also important to note that queries using LIKE and CONTAINS can return different results depending on how they are formatted and the data they are searching. For instance, in the CONTAINS query above, if there was not an asterisk after “good,” then sayings containing “goods” would not be returned. Similarly, if the LIKE query did not contain a percent sign after

“good,” it would only match sayings that end in “good.” Another issue arises when the word being searched for is within another word. For example, if “scattergood” was in the sayings collection, it would be matched by the LIKE query but not the CONTAINS query. Being cognizant of these differences, as well as ensuring the same number of rows are returned, is important when testing the performance of these queries.

## 7 Target Audience

The target audience for the Prophet Explorer application includes but is not limited to: Religion professors and students, religious enthusiasts, and anyone with a curiosity for religion. In order to learn more about the target audience, an effort was made to solicit feedback. This feedback was gotten by: 1) conducting a survey of Philosophy and Religion students, and 2) presenting the application at a religious conference.

The survey was conducted by asking students in PAR 125 - Great Books of the World’s Religions, to experiment with the application then answer ten questions. The questions ranged from ranking the design to whether they found the material to be offensive. Overall, the feedback was positive and encouraging. Seventy percent of the fifty students surveyed ranked the design 4 or 5 on a scale of 1 to 5, and none found the material to be offensive. Also important are the results to the question “Do you think the term *prophet* is an accurate description of Buddha, Jesus, and Muhammad?” Seventy-two percent responded yes, eight percent responded no, and twenty percent were unsure. This was reassuring because there have been talks of changing the name of the application because the term *prophet* is used loosely. For a complete overview of the survey results, refer to Appendix F.

The North Carolina Religious Studies Association (NCRSA) conference was held at the University of North Carolina Wilmington on November 3, 2012. The conference featured religious professors and students who presented their papers. This was a great opportunity to share this application and get feedback from the target audience. Overall, the feedback was very positive. To the question about the term *prophet*, the audience did not themselves find it to be an issue but noted that there are definitely some who may. In fact, one audience member stated, “You have not stepped outside of the realm of prophets.” One criticism they did have is that there are no *prophets* from Judaism represented and other *prophets* from eastern religions could be included. This criticism will be addressed in future versions of the application as is discussed in the following section.

## 8 Future Work

In future versions of the application there are three issues that are currently important to address. The first issue is the need to include additional *prophets* so that all of the major religions are represented. The second issue is the need to apply authenticity labels to the sayings of each *prophet*. The third issue is the need to address language concerns.

### 8.1 Adding Prophets

Adding *prophets* would give the application a larger potential audience and increase the educational value of the application. The Jewish prophet Isaiah would be a good fit because he has numerous sayings recorded in the Old Testament. For eastern religions, it would be interesting to include Confucius, the father of Confucianism, or Lao Tzu, the father of Taoism. In order to include one of these *prophets*, there would have to be a lot of pre-processing work done. That is, the texts that describe these *prophets* would have to be examined and only the sayings retrieved. Once completed, the sayings could be inserted into the database. This process would be similar to what was described previously – input files would be created and the program to insert sayings would be executed using these files. Lastly, there would have to be updates to the *views*. For instance, on the Search *view* there is currently a tabbed panel that includes tabs for Buddha, Jesus, and Muhammad. This panel would have to be expanded to include any additional *prophets*. Besides this work, any additional *prophets* would plug-in relatively easy.

### 8.2 Authenticity Highlighting

Currently, only the sayings of Jesus have been highlighted using authenticity labels. In the future though, it would be interesting to research scholarship in the Buddhist and Muslim communities to figure out which and what types of sayings are considered to be authentic for Buddha and Muhammad. This would no doubt require a lot of research and the consultation of experts in the field. The work would be worth it though because it would provide better insight into what these individuals taught.

### 8.3 Language Concerns

As any scholar will tell you, the best way to understand religious texts is to read them in their original language rather than a translation. Those who are aware of this have what they believe to be a strong criticism against this application, which is that it ignores the original languages of the sayings. This argument is quite facile though, as it fails to acknowledge the main goal of the Prophet Explorer application: *to provide a comparative environment that presents only the*

*sayings of prophets*. In line with this goal, there must be a middle ground to compare these sayings. This middle ground is and will continue to be the English translations.

If the sayings were presented in their original language of Arabic, Coptic, Greek, or Pali, it would greatly inhibit the ability of a general audience to perform comparisons. A better solution would be to include the native texts as a supplement to the English translations. This could work by initially showing the saying in English and switching to the original language upon some user action. This would help those who are familiar with one of the original languages of these texts extract a more precise meaning of the sayings. For example, someone familiar with Greek could more deeply examine a saying of Jesus by pulling up the Greek version.

## 9 Conclusion

The Prophet Explorer application is unique in the sense that it provides the ability to compare the sayings of religious *prophets* using modern day search techniques – this is something that is not currently readily available. There are many applications that allow for an intra-comparison of a religion, but there are no easily found applications that allow for inter-comparisons of religions. Also, unlike other religious search applications, Prophet Explorer ignores narrative elements and presents only the sayings of *prophets*. This was done to differentiate the application and to strip away some of the heightened doctrine that has been applied to these figures. This inspiration came from Thomas Jefferson who cut and pasted sections of the Gospels to try to uncover the true teachings of Jesus. Unlike Jefferson though, we no longer need scissors and paste to uncover truth – we have great technologies at our fingertips.

## 10 References


- Agrawal, S., & Chaudhuri, S., & Das, G. (2002). DBXplorer: A System for Keyword-Based Search over Relational Databases. Retrieved on August 24, 2012 from <http://www.cs.washington.edu/education/courses/cse590g/04wi/papers/dbexplorer.pdf>
- ASP.NET MVC Overview (2009). Microsoft ASP.NET. Retrieved on August 24, 2012 from <http://www.asp.net/mvc/tutorials/older-versions/overview/asp-net-mvc-overview>
- Bible Gateway (2012). Retrieved on August 24, 2012 from <http://www.biblegateway.com/>
- Brin, S., & Page, L. (1998). The Anatomy of a Large-Scale Hypertextual Web Search Engine. Retrieved on August 24, 2012 from <http://ilpubs.stanford.edu:8090/361/1/1998-8.pdf>

- Cotter, H. (2006). SQL Server Full Text Search Language Features. Retrieved on August 24, 2012 from <http://www.simple-talk.com/sql/learn-sql-server/sql-server-full-text-search-language-features/>
- Full-Text Search (2012). MSDN. Retrieved on August 24, 2012 from <http://msdn.microsoft.com/en-us/library/ms142571.aspx>
- Funk, Robert, Hoover, Roy, and The Jesus Seminar. The Five Gospels - What Did Jesus Really Say? Polebridge Press, 1993.
- Google Annual Search Statistics (2012). Statistic Brain. Retrieved on August 24, 2012 from <http://www.statisticbrain.com/google-searches/>
- Java API for WordNet Searching (JAWS) (2009). Southern Methodist University. Retrieved on August 24, 2012 from <http://lyle.smu.edu/~tspell/jaws/index.html>
- Martin, J. Rapid Application Development. Macmillan, New York, 1991.
- Merriam Webster (2012). Retrieved on August 24, 2012 from <http://www.merriam-webster.com/dictionary/information%20retrieval>
- Strong, J. (2007). Strong's Exhaustive Concordance of the Bible. Hendrickson Publishers. Peabody, Massachusetts.
- The Five Gospel Parallels (2001). University of Toronto. Retrieved on August 24, 2012 from <http://www.utoronto.ca/religion/synopsis/>
- Weinberg, B. (1999). Indexes and Religion: Reflections on Research in the History of Indexes. *The Indexer* 21(3) Retrieved on August 24, 2012 from [http://www.theindexer.org/files/21-3/21-3\\_111.pdf](http://www.theindexer.org/files/21-3/21-3_111.pdf)
- WHATWG Community (2012). Retrieved on August 24, 2012 from <http://www.whatwg.org/>
- WordNet: A Lexical Database for English (2012). Princeton University. Retrieved on August 24, 2012 from <http://wordnet.princeton.edu/>
- Yen, D., & Davis, W. Rapid Application Development (RAD). The Information Consultant's System Handbook: System Analysis. 1999.

# Appendixes

## A. Desktop and Mobile Comparison

### A.1. Desktop View



Prophet Explorer
[Home](#)
[Search](#)
[View](#)
[Concordance](#)
[Sources](#)

### About Prophet Explorer


Prophet Explorer is an application that was developed to aid in comparing the sayings of religious *prophets* and is meant to educate users about their teachings. By using the [Search](#) feature and examining the [Concordance](#), users can easily compare what each *prophet* said about certain topics. Users can also view the entire collection of sayings by visiting the [View](#) Sayings page. Each of these features allows for a rich exercise in comparative religion and helps answer the important question of "What did these *prophets* say?" [ [More on our usage of prophet](#) ]

The sayings were acquired from a variety of sources which can be viewed at the [Sources](#) page. From these sources, the sayings were stripped out and processed. In other words, the sayings collection has no narrative elements – only sayings. This helps to differentiate the application from others like it and allows users to more quickly learn about the teachings of each *prophet*.


### About the Prophets



**Buddha**  
 Buddha was born in northern India in the 5th Century BCE. Legends say that at birth he was able to walk and talk, and from a young age practiced meditation. Buddha was the heir to a king...



**Jesus**  
 Jesus was born in Bethlehem, Judea, around 4 BCE in the age of the Roman Empire. Little is known about the early life of Jesus, as he did not become influential until later in life. Around the year...



**Muhammad**  
 Muhammad was born in Mecca, Saudi Arabia in 570 CE and is considered the founder of the religion of Islam. Muhammad was raised mostly by his uncle because his father, mother, and ...

Prophet Explorer - A Saying Search Tool using MVC and Full Text Search  
 University of North Carolina Wilmington  
 Masters of Science in Computer Science and Information Systems  
 Zach Wilson - Capstone Project 2012

Sayings obtained from:  
[Access to Insight](#)  
[Bible Gateway](#)  
[Project Gutenberg](#)  
[The Hadith Collection](#)  
[Sacred Texts Archive](#)

## A.2. Mobile View (iPhone5)

**Prophet Explorer**

**Home** >

**Search** >


**View** >

### About Prophet Explorer


Prophet Explorer is an application that was developed to aid in comparing the sayings of religious *prophets* and is meant to educate users about their teachings. By using the [Keyword Search](#) and [View Sayings](#) features, users can easily compare what each *prophet* said about certain topics. Each of these features allows for a rich exercise in comparative religion and helps answer the important question of "What did these *prophets* say?" [ [More on our usage of prophet](#) ]

The sayings were acquired from a variety of sources which can be viewed at the [Sources](#) page. From these sources, the sayings were stripped out and processed. In other words, the sayings collection has no narrative elements – only sayings. This helps to differentiate the application from others like it and allows users to more quickly learn about the teachings of each *prophet*.


### About the Prophets



**Buddha** was born in northern India in the 5th Century BCE. Legends say that at birth he was able to walk and talk, and from a young age practiced meditation. Buddha was the heir to a king...






**Jesus** was born in Bethlehem, Judea, around 4 BCE in the age of the Roman Empire. Little is known about the early life of Jesus, as he did not become influential until later in life. Around the year...



**Muhammad** was born in Mecca, Saudi Arabia in 570 CE and is considered the founder of the religion of Islam. Muhammad was raised mostly by his uncle because his father, mother, and ...

Prophet Explorer  
 University of North Carolina Wilmington  
 MS in Computer Science and Information Systems  
 Zach Wilson - Capstone Project 2012

## B.1. Keyword Search Page

 <b>Buddha</b>	 <b>Jesus</b>	 <b>Muhammad</b>
-------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------

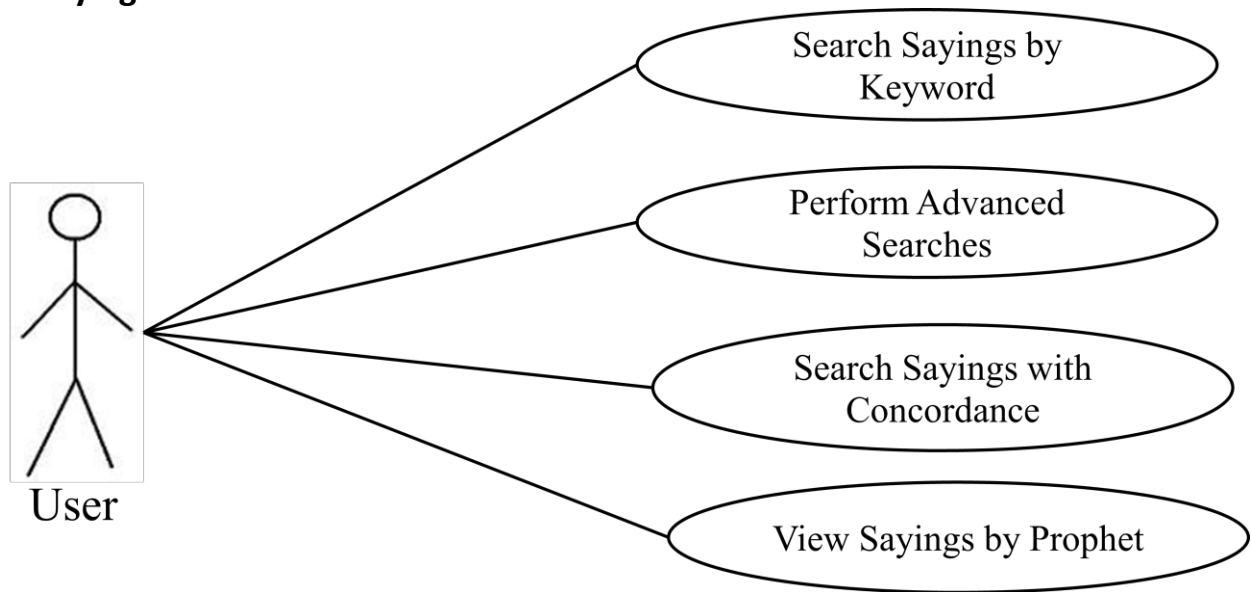
[The Pali Canon - Anguttara Nikaya - Book of the Fours](#)  
 Then it's certain, monks, that that monk didn't suffuse the four royal snake lineages with a mind of **good** will. For if he had suffused the four royal snake lineages with a mind of **good** will, he would not have died after having been bitten by a snake. Which four? The Virupakkha royal snake lineage, the Erapatha royal snake lineage, the Chabyaputta royal snake lineage, the Dark Gotamaka royal snake lineage. It's certain that that monk didn't suffuse these four royal snake lineages with a mind of **good** will. For if he had suffused these four royal snake lineages with a mind of **good** will, he would not have died after having been bitten by a snake. I allow you, monks, to suffuse these four royal snake lineages with a mind of **good** will for the sake of self-protection, self-guarding, self-preservation.

[The Pali Canon - Itivuttaka - Chapter 3. The Group of Threes](#)  
 Intelligent, you should guard your virtue, aspiring to three forms of bliss: praise; the obtaining of wealth; and, after death, rejoicing in heaven. Even if you do no evil but seek out one who does, you're suspected of evil. Your bad reputation grows. The sort of person you make a friend, the sort you seek out, that's the sort you yourself become — for your living together is of that sort. The one associated with, the one who associates, the one who's touched, the one who touches another — like an arrow smeared with poison — contaminates the quiver. So, fearing contamination, the enlightened should not be comrades with evil people. A man who wraps rotting fish in a blade of kusa grass makes the grass smelly: so it is if you seek out fools. But a man who wraps powdered incense in the leaf of a tree makes the leaf fragrant: so it is if you seek out the enlightened. So, knowing your own outcome as like the leaf-wrapper's, you shouldn't seek out those who aren't **good**. The wise would associate with those who are. Those who aren't **good** lead you to hell. The **good** help you reach a **good** destination.

## B.2. Advanced Keyword Search Page

Buddha	Jesus	Muhammad
<p><b>The Pali Canon - Anguttara Nikaya - Book of the Fours</b></p> <p>Then it's certain, monks, that that monk didn't suffuse the four royal snake lineages with a mind of <b>good</b> will. For if he had suffused the four royal snake lineages with a mind of <b>good</b> will, he would not have died after having been bitten by a snake. Which four? The Virupakkha royal snake lineage, the Erapatha royal snake lineage, the Chabyaputta royal snake lineage, the Dark Gotamaka royal snake lineage. It's certain that that monk didn't suffuse these four royal snake lineages with a mind of <b>good</b> will. For if he had suffused these four royal snake lineages with a mind of <b>good</b> will, he would not have died after having been bitten by a snake. I allow you, monks, to suffuse these four royal snake lineages with a mind of <b>good</b> will for the sake of self-protection, self-guarding, self-preservation.</p>	<p><b>New King James Bible - Matthew - Chapter 12</b></p> <p>33 Either make the tree <b>good</b> and its fruit <b>good</b>, or else make the tree bad and its fruit bad; for a tree is known by its fruit. 34 Brood of vipers! How can you, being evil, speak <b>good</b> things? For out of the abundance of the heart the mouth speaks. 35 A <b>good</b> man out of the <b>good</b> treasure of his heart brings forth <b>good</b> things, and an evil man out of the evil treasure brings forth evil things. 36 But I say to you that for every idle word men may speak, they will give account of it in the day of judgment. 37 For by your words you will be justified, and by your words you will be condemned.</p> <p><b>New King James Bible - Matthew - Chapter 7</b></p> <p>15 Beware of false prophets, who come to you in sheep's clothing, but inwardly they are ravenous wolves. 16 You will know them by their fruits. Do men</p>	<p><b>Hadith Collection - Volume 1 - Book 02 Belief</b></p> <p>I was shown the Hell-fire and that the majority of its dwellers were women who were ungrateful." It was asked, "Do they disbelieve in Allah?" (or are they ungrateful to Allah?) He replied, "They are ungrateful to their husbands and are ungrateful for the favors and the <b>good</b> (charitable deeds) done to them. If you have always been <b>good</b> (benevolent) to one of them and then she sees something in you (not of her liking), she will say, 'I have never received any <b>good</b> from you.'</p> <p><b>Hadith Collection - Volume 1 - Book 02 Belief</b></p> <p>Whoever said "None has the right to be worshipped but Allah and has in his heart <b>good</b> (faith) equal to the weight of a barley grain will be taken out of Hell. And whoever said: "None has the right to be worshipped but Allah and has in his heart <b>good</b> (faith)</p>

**C. Sayings Use Case**



**D. CRUD Diagram**

Entity Process	Prophet	Saying	Section	Book	Source	WordFrequency	WordFrequencySaying
Add Prophet	C						
View Prophets	R						
Add Source					C		
Add Book				C			
Add Section			C				
Add Saying		C					
Change Saying Text		U					
Remove Saying		D					
View Sayings	R	R	R	R	R		
Keyword Search	R	R	R	R	R		
Advanced Search	R	R	R	R	R		
Concordance Search	R	R	R	R	R	R	R

## E. Hosting Strategies





Currently, the Prophet Explorer application is hosted on a virtual machine on a server at the University of North Carolina Wilmington. In the future though, the application will need to be moved because permissions to access the server will be lost upon graduation. This move will be nontrivial due to the technologies employed in this project.

The main technology employed that influences where to host the application is MS SQL Server Full-Text Search. Because the thesaurus file and stop-list have been customized, shared hosting will not be an option. Shared hosting companies have already been contacted, and they have made it clear that these files cannot be customized in a shared environment. Consequently, a dedicated or virtual server will have to be purchased. Based on pricing concerns, the best option is a Virtual Private Server (VPS). These environments can be purchased relatively cheaply (roughly \$25/month), and allow for isolation from other applications.





The VPS that is purchased will also need version 4.5 of the .NET Framework and MVC 4. However, these technologies can be downloaded without much effort so this is not much of a concern. This leaves MS SQL Server as the only real concern. Because of this, a VPS that is preloaded with MS SQL Server will likely be purchased. A domain name will also need to be purchased with DNS mappings to the VPS. Once all of this is accomplished, the application should be transferred relatively easily.


## F. Survey Results

1. Rank the website CONTENT on a scale of 1 to 5 (1 being the lowest, 5 being the highest)





		Response Total	Response Percent
1		0	0%
2		1	2%
3		10	20%
4		25	50%
5		14	28%
<b>Total Respondents</b>		<b>50</b>	<b>100%</b>

2. Rank the website DESIGN on a scale of 1 to 5 (1 being the lowest, 5 being the highest)

		Response Total	Response Percent
1		1	2%
2		2	4%
3		12	24%
4		20	40%

5		<b>15</b>	<b>30%</b>
<b>Total Respondents</b>		<b>50</b>	<b>100%</b>

3. Rank the USABILITY of the website on a scale of 1 to 5 (1 being the lowest, 5 being the highest)

		<b>Response Total</b>	<b>Response Percent</b>
1		<b>0</b>	<b>0%</b>
2		<b>1</b>	<b>2%</b>
3		<b>8</b>	<b>16%</b>
4		<b>24</b>	<b>48%</b>
5		<b>17</b>	<b>34%</b>
<b>Total Respondents</b>		<b>50</b>	<b>100%</b>

4. The name of the application is Prophet Explorer. Is there another name you would propose?

<b>Total Respondents</b>	<b>32</b>
(skipped this question)	18

1. Religion Explorer
2. Text's of World Religions
3. No
4. No
5. It's a good name and easy to remember
6. I think the name suits this application well. No change necessary.
7. Prophet Finder
8. Sounds good to me
9. N/A
10. I can't think of any others, but I think you can be more creative.
11. N/A
12. N/A
13. Nope, that's a good name
14. I'm not good at making up names, but I would suggest a new one.
15. I like this one, it's easy to say and to the point
16. No, I really enjoy the name.

17. Prophet quest
18. I like the name of Prophet Explorer.
19. I can't think of a better one. Prophet Explorer is pretty self explanatory.
20. No
21. Maybe something catchier that sounds less like Internet Explorer. Prophet Journeys?
22. No, but I'm not sure prophet is the best word to describe all the religious figures represented.
23. No
24. No, I love that! Super creative!
25. Prophet Explorer seems like it would be a great name to continue using.
26. No
27. No, I think Prophet Explorer suites it best.
28. No, I like the name "Prophet Explorer".
29. No
30. No
31. The only major concern I have is if Buddha is truly a prophet, but I'm not a PAR student...prophet to me sort of signifies divinity.
32. I really like the name.

5. Do you find material on the website to be offensive?




		Response Total	Response Percent
Yes		0	0%
No		50	100%
<b>Total Respondents</b>		<b>50</b>	<b>100%</b>

6. Do you think other people may find material on the website to be offensive?





		Response Total	Response Percent
Yes		2	4%
No		34	68%
Maybe		14	28%
<b>Total Respondents</b>		<b>50</b>	<b>100%</b>

7. Do you think the term prophet is an accurate description for Buddha, Jesus, and Muhammad?







		Response Total	Response Percent

Yes		<b>36</b>	<b>72%</b>
No		<b>4</b>	<b>8%</b>
Not Sure		<b>10</b>	<b>20%</b>
<b>Total Respondents</b>		<b>50</b>	<b>100%</b>



8. What is a concordance?

		<b>Response Total</b>	<b>Response Percent</b>
A collection of religious sayings		<b>5</b>	<b>10%</b>
An alphabetical index of words in a collection		<b>43</b>	<b>86%</b>
A website for religious research		<b>1</b>	<b>2%</b>
A book that describes multiple religions		<b>1</b>	<b>2%</b>
<b>Total Respondents</b>		<b>50</b>	<b>100%</b>

9. What Bible translations have you read in the past or be most likely to read?

		<b>Response Total</b>	<b>Response Percent</b>
King James		<b>15</b>	<b>30%</b>
New King James		<b>12</b>	<b>24%</b>
New International Version		<b>18</b>	<b>36%</b>
New American Standard Bible		<b>12</b>	<b>24%</b>
None / Not Sure		<b>12</b>	<b>24%</b>
Other		<b>50</b>	<b>100%</b>
<b>Total Respondents</b>		<b>50</b>	

10. Would the Prophet Explorer website be useful to you in PAR 125?

		<b>Response Total</b>	<b>Response Percent</b>
Yes		<b>44</b>	<b>88%</b>
No		<b>6</b>	<b>12%</b>
<b>Total Respondents</b>		<b>50</b>	<b>100%</b>