GospelPreaching.com

Planning Report

Table of Contents

Background

Project Leader

David Nichols
417.847.7596
Nichols316@live.missouristate.edu

Company Information

Contending for the Faith Publications
Allen Bailey, Owner
4216 Abigale Drive
Yukon, OK 73099
214.505.8242
allen.bailey@yahoo.com

CFTF Publications is a small publisher of religious books, tracts, and other media related to the Church of Christ. Allen Bailey is the owner/operator. Writers or extra editors are hired or volunteer on an as-needed basis. The company's main goal is to spread the Gospel and edify existing Christians through the publishing/availability of religious materials. CFTF doesn't have any physical publishing facilities.

System Name and Description

The system is going to be an online portal for CFTF's electronic resources (documents, audio, video, etc.) and a storefront for the company's published goods. The system will simply be known as “GospelPreaching.com”. Along with the system's basic functions mentioned above, it will also provide the ability to categorize and tag electronic materials, post and respond to questions (Q&A), allow users to connect their profiles with social networking sites (Facebook, Twitter, etc.), post comments, and “like” (favorite) articles.

System Stakeholders

Allen Bailey would be the principle user of the site, performing administrative functions in the system. Other users would include the site visitors would be viewing the electronic resources, purchasing published goods, and utilizing the social aspects of the site. There is also the possibility of hiring or finding volunteers to perform the conversion of materials to electronic format. These people would need to have the ability to upload new materials to the site.

Existing System

The current system is a simple static site with a little basic PHP. The current site is only serving as a placeholder for the proposed system and is not adequate for continued use as it lacks many key features CFTF is looking for. Please refer to the figures below and the following list for more details about what some of the shortcomings of the existing system are, as was discussed during informal stakeholder interviews.

Problems with old system:

- It requires someone to have a working knowledge of PHP and HTML to add material to the site.
- It lacks any social or interactive elements for visitors.
- No capabilities to sell books and tracts.
- Very low administrator usability.
- Electronic resources are simply displayed as a list of titles, needs to be displayed in a more logical and visually appealing way. (Figures 2, 3)
- No search function.
- Home page isn't very functional. Doesn't display recent articles, announcements, etc. (Figure 1)
- Doesn't display combined content (text, audio, video, etc.) on one detail page very effectively, or attractively. (Figures 4, 5)

Opportunities:

- Non-technical administrators could upload and manage resources by themselves.
- Provide the opportunity for users to interact socially, helping promote thoughtful discussions and generate interest in materials.
- Introduces the ability to sell the firms publications to a much larger market.
- The site will also be more usable and inviting for readers.

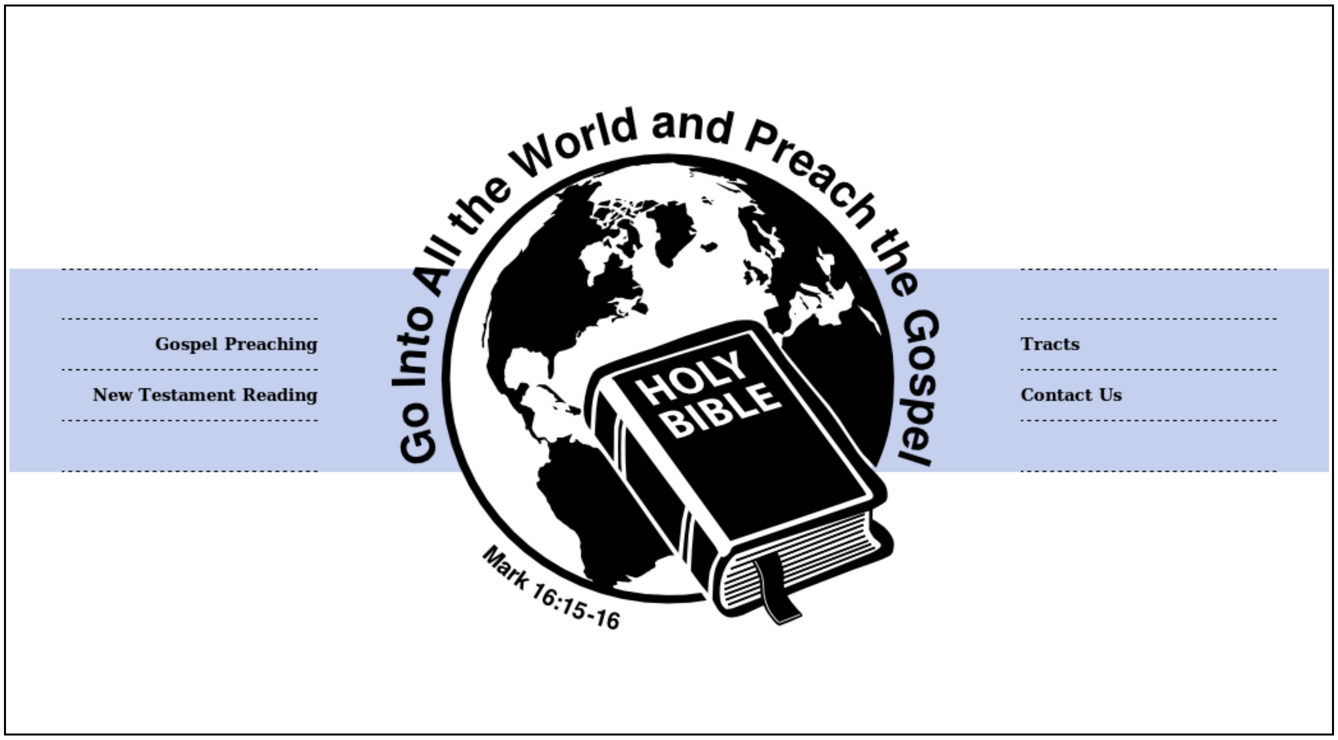
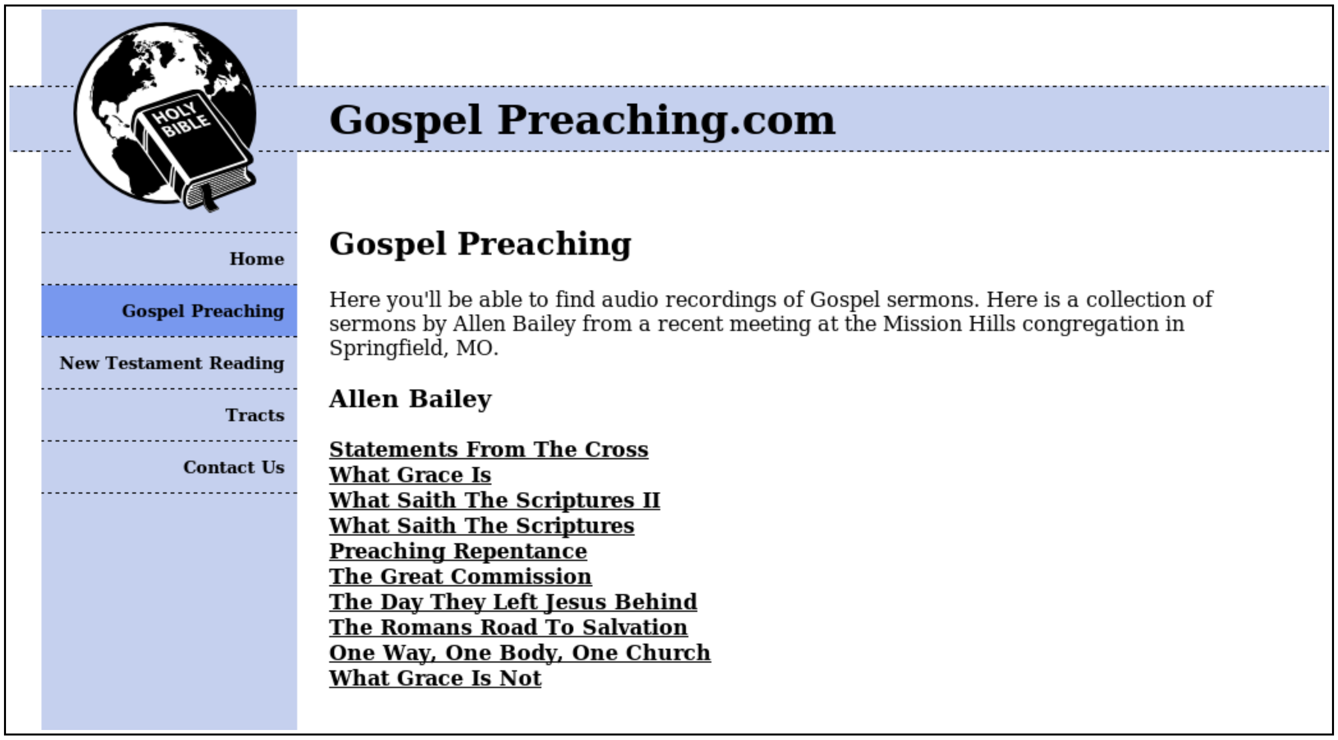


Figure 1 – Current system – Home page



Gospel Preaching.com

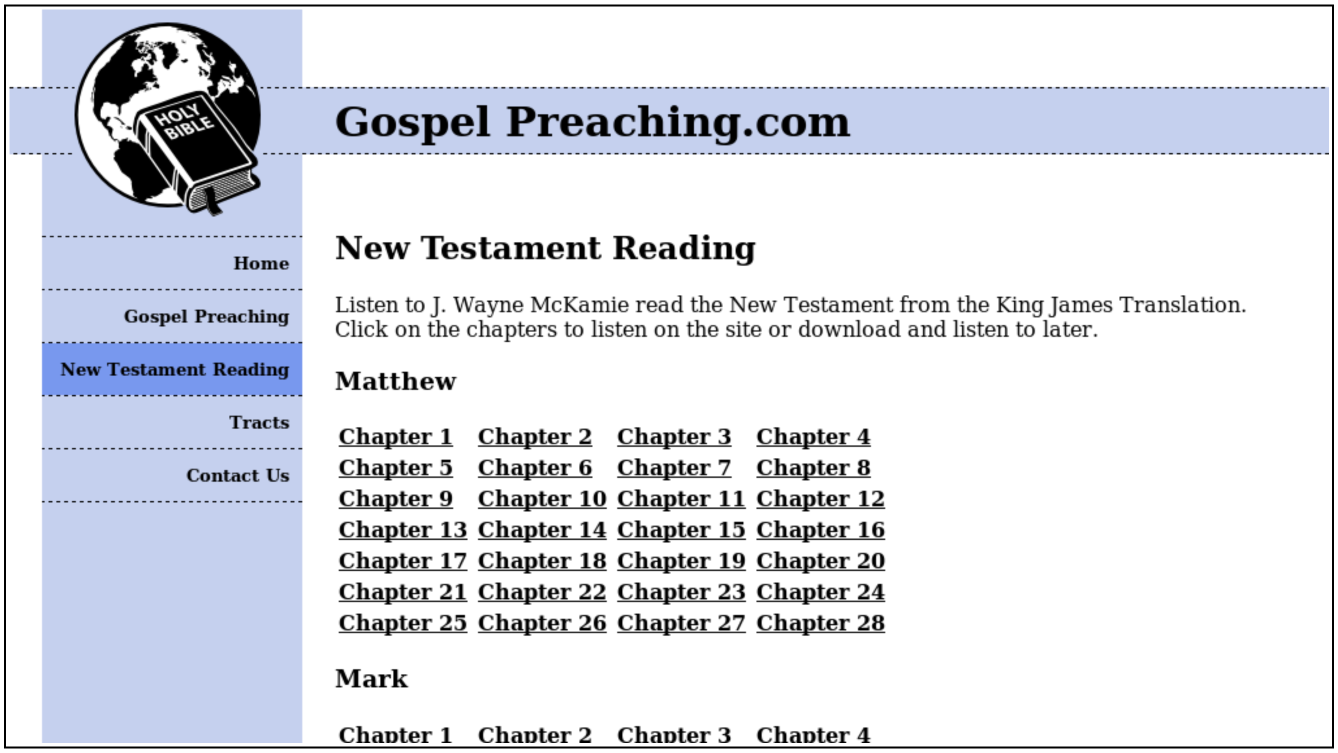
Gospel Preaching

Here you'll be able to find audio recordings of Gospel sermons. Here is a collection of sermons by Allen Bailey from a recent meeting at the Mission Hills congregation in Springfield, MO.

Allen Bailey

Statements From The Cross
What Grace Is
What Saith The Scriptures II
What Saith The Scriptures
Preaching Repentance
The Great Commission
The Day They Left Jesus Behind
The Romans Road To Salvation
One Way, One Body, One Church
What Grace Is Not

Figure 2 – Current system – Article index



Gospel Preaching.com

New Testament Reading

Listen to J. Wayne McKamie read the New Testament from the King James Translation. Click on the chapters to listen on the site or download and listen to later.

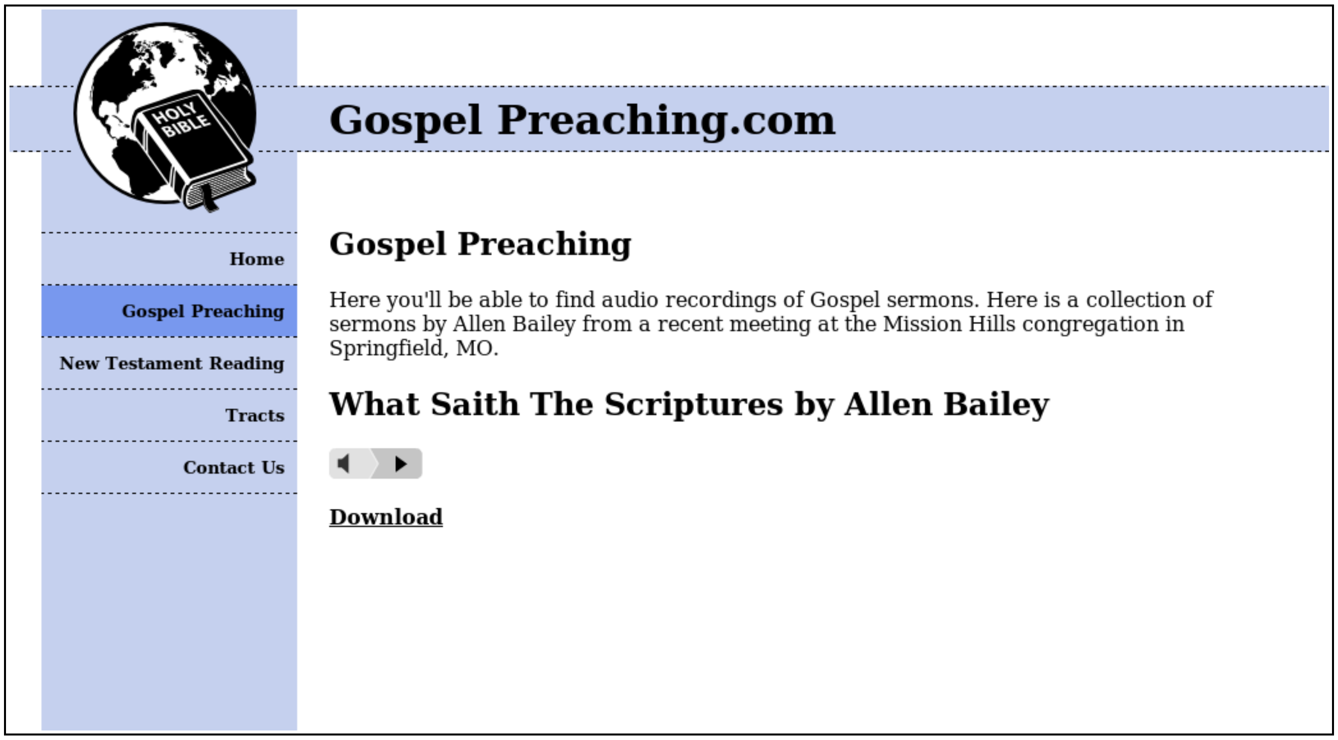
Matthew

Chapter 1	Chapter 2	Chapter 3	Chapter 4
Chapter 5	Chapter 6	Chapter 7	Chapter 8
Chapter 9	Chapter 10	Chapter 11	Chapter 12
Chapter 13	Chapter 14	Chapter 15	Chapter 16
Chapter 17	Chapter 18	Chapter 19	Chapter 20
Chapter 21	Chapter 22	Chapter 23	Chapter 24
Chapter 25	Chapter 26	Chapter 27	Chapter 28

Mark

Chapter 1	Chapter 2	Chapter 3	Chapter 4

Figure 3 – Current system – Audio recording index



Gospel Preaching.com

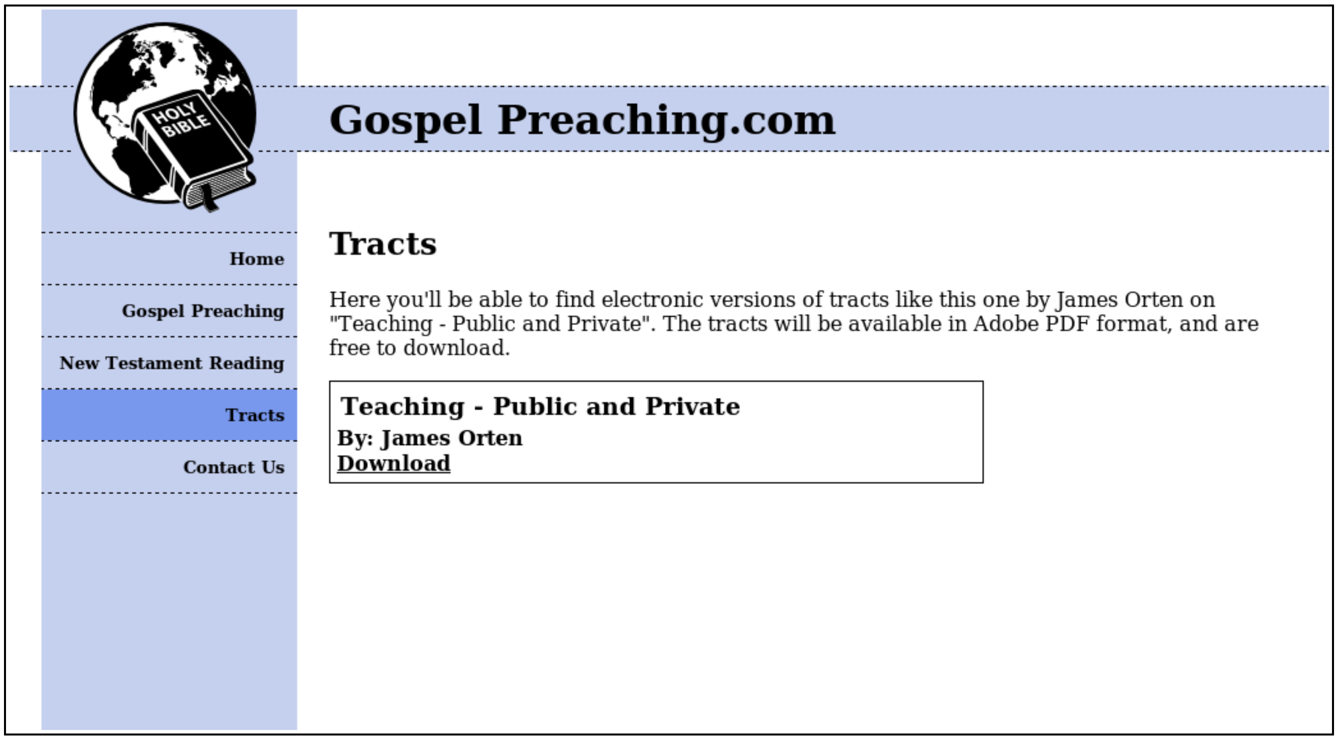
Gospel Preaching

Here you'll be able to find audio recordings of Gospel sermons. Here is a collection of sermons by Allen Bailey from a recent meeting at the Mission Hills congregation in Springfield, MO.

What Saith The Scriptures by Allen Bailey

Download

Figure 4 – Current system – Single article view



Tracts

Here you'll be able to find electronic versions of tracts like this one by James Orten on "Teaching - Public and Private". The tracts will be available in Adobe PDF format, and are free to download.

Teaching - Public and Private
By: James Orten
Download

Figure 5 – Current system – Tract view

Technology

The system would use the Django framework for Python. MySQL would be used for the database and the whole system would run on the existing LAMP server stack. Design should adhere to a strict object-oriented, three layer web design.

There are currently no other systems in the organization to integrate with.

The only work that has been completed beforehand is some proof of concept code to test the viability of the framework and the existing HTML and CSS of the old system.

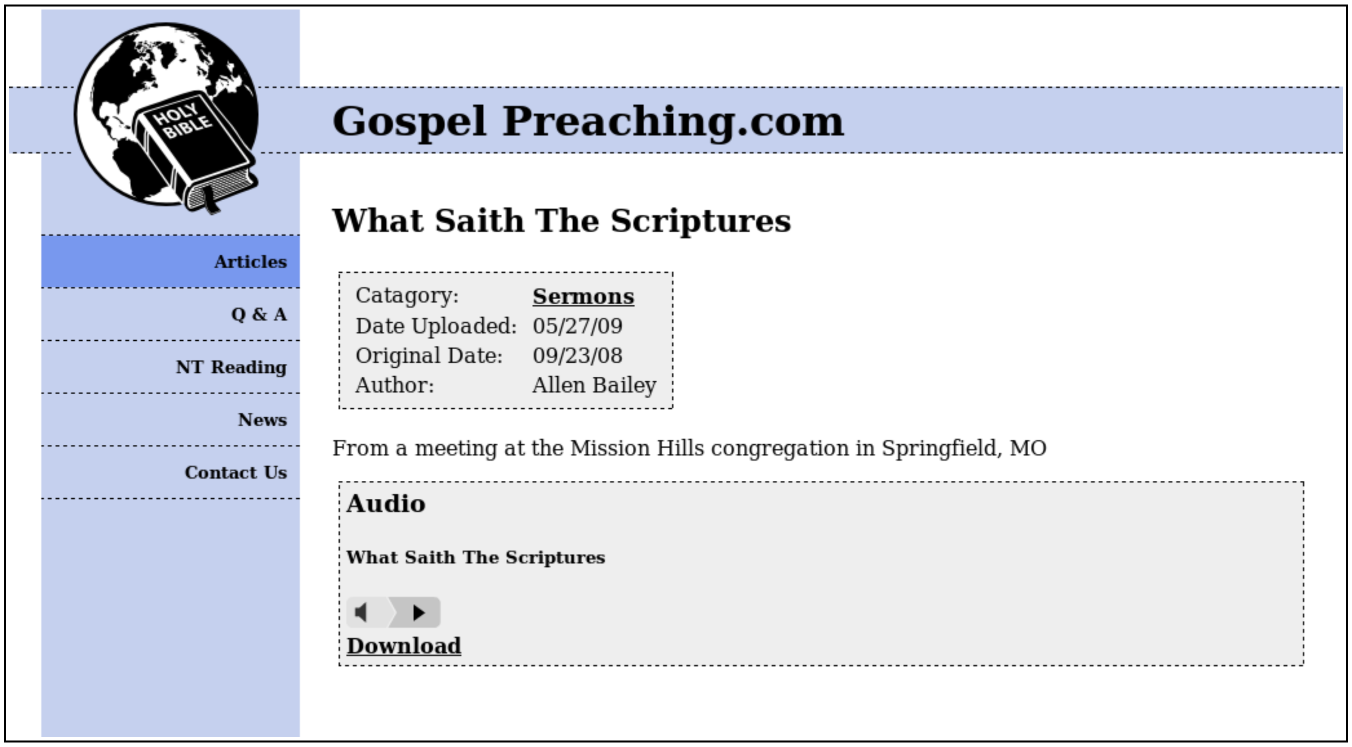
I've used this framework for other client sites in the past and feel comfortable using it for this project. Django is an open-source platform and has a very extensive community including a thorough documentation and tutorial wiki on the project's website. I also know several developers who have a considerable working knowledge of the framework. Therefore, I don't expect to have any difficulties obtaining support if needed.

The use case I used to test the framework and design approach I'll be using is “View Article”. This use case involves both an index to find the article you're looking for, as well as a details page to view the full article. Please refer to figures 6 and 7 below to see the screenshots of the implemented use case.

Please reference Appendix A when reading the following description of the code. The Django code is structured a little bit different than a typical MVC design. It's still implemented using a three layer design though. The models, or data access takes place in models.py. No SQL strings are ever directly manipulated by user created code; the Django framework handles all of that. I simply defined the structure of the classes and all the SQL string conversion will be taken care of automatically based on the connection settings in the main settings.py file. The controller code is mostly handled by the framework. However, there are configuration settings that manage the functionality of the controller code. The most notable of these is the urls.py file which defines which view to display. According to the Django's interpretation of MVC, the views.py file manages the view layer. This interpretation defines the view layer as making the decision of ***what*** to display, instead of ***how*** to display it. This logic would traditionally be found in the controller layer in other MVC frameworks. The final piece of the puzzle is the template system. The templates define *how* the information is to be displayed, and consist of HTML files with template tags. You can look at base.html, index.html, and detail.html for the relevant templates in this use case.

Please refer to figure 8 for a screenshot of the database table used in this use case.

I'd like to add one note to this use case. The code uses separate classes for different types of media instead of having a single media class. I did this to try a different approach and see how it would work. This detail will be decided in the next iteration and be reflected the analysis report.



What Saith The Scriptures

Catagory:	Sermons
Date Uploaded:	05/27/09
Original Date:	09/23/08
Author:	Allen Bailey

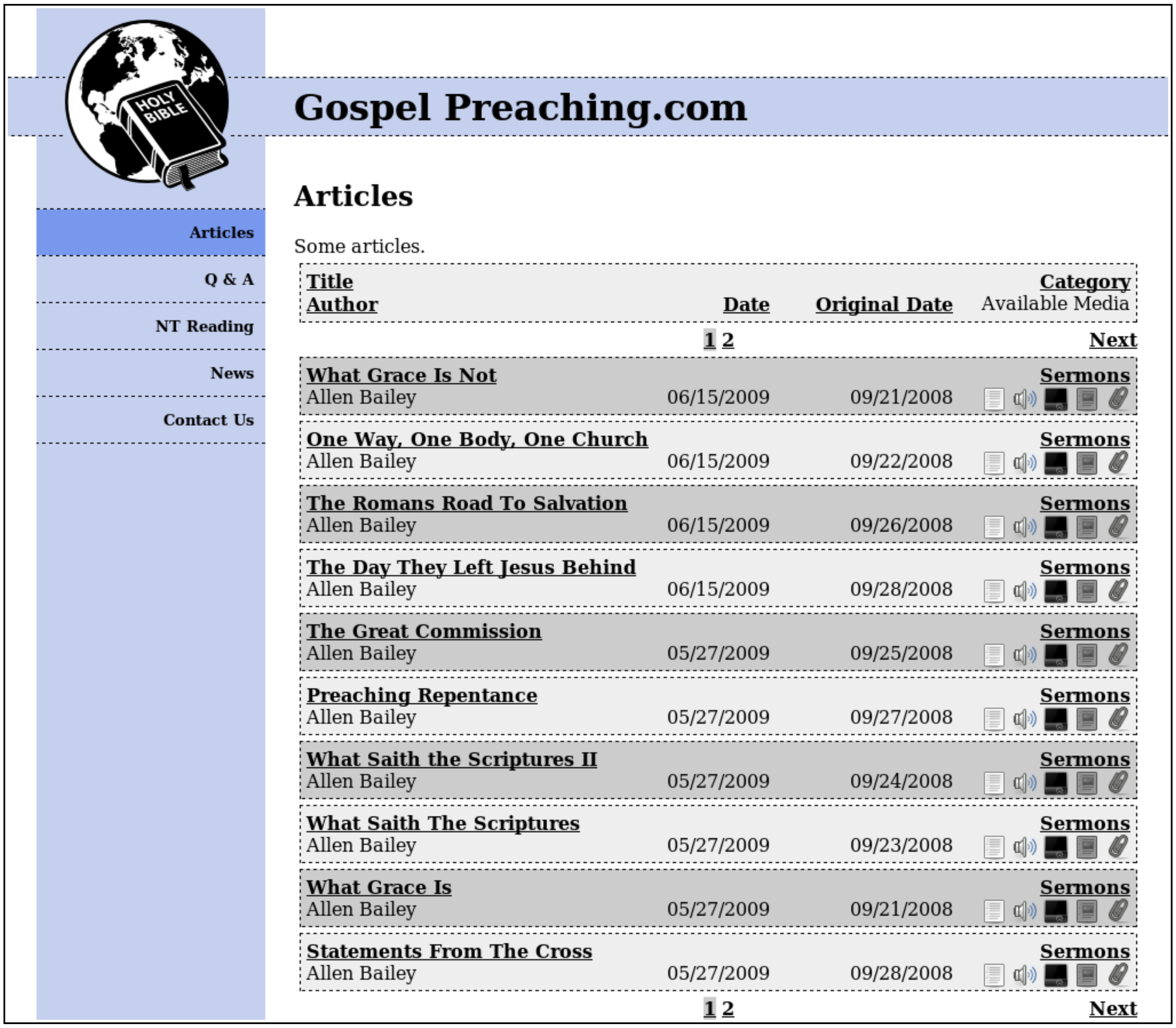
From a meeting at the Mission Hills congregation in Springfield, MO

Audio

What Saith The Scriptures

Download

Figure 6 – Implemented Use Case: View Article – Article Details



Gospel Preaching.com

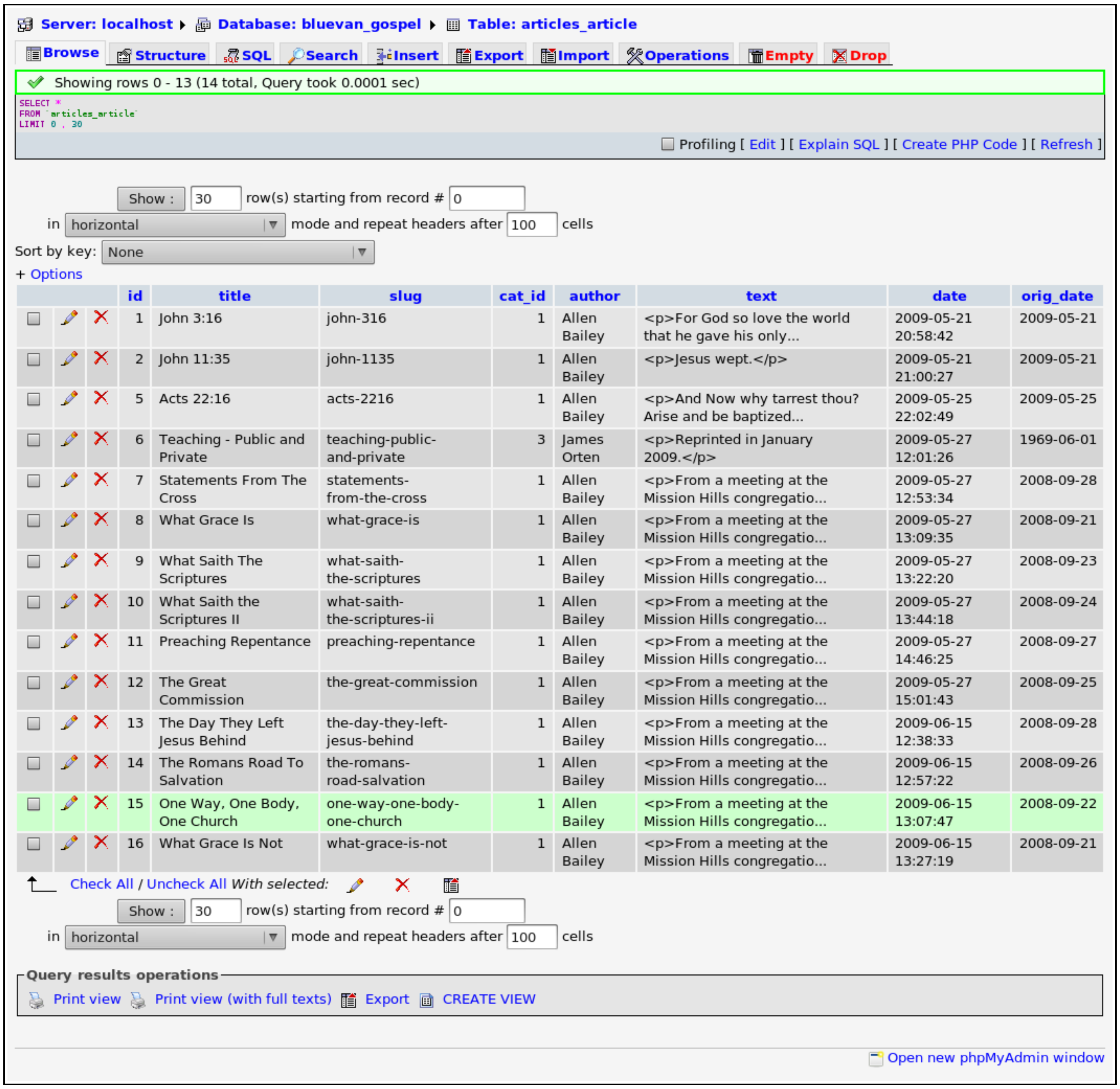
Articles

Some articles.

Title / Author	Date	Original Date	Category / Available Media
What Grace Is Not / Allen Bailey	06/15/2009	09/21/2008	Sermons
One Way, One Body, One Church / Allen Bailey	06/15/2009	09/22/2008	Sermons
The Romans Road To Salvation / Allen Bailey	06/15/2009	09/26/2008	Sermons
The Day They Left Jesus Behind / Allen Bailey	06/15/2009	09/28/2008	Sermons
The Great Commission / Allen Bailey	05/27/2009	09/25/2008	Sermons
Preaching Repentance / Allen Bailey	05/27/2009	09/27/2008	Sermons
What Saith the Scriptures II / Allen Bailey	05/27/2009	09/24/2008	Sermons
What Saith The Scriptures / Allen Bailey	05/27/2009	09/23/2008	Sermons
What Grace Is / Allen Bailey	05/27/2009	09/21/2008	Sermons
Statements From The Cross / Allen Bailey	05/27/2009	09/28/2008	Sermons

1 2 Next

Figure 7 – Implemented Use Case: View Article – Article Index



Showing rows 0 - 13 (14 total, Query took 0.0001 sec)

```
SELECT *
FROM `articles_article`
LIMIT 0 , 30
```

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells
Sort by key: None

id	title	slug	cat_id	author	text	date	orig_date
1	John 3:16	john-316	1	Allen Bailey	<p>For God so love the world that he gave his only...	2009-05-21 20:58:42	2009-05-21
2	John 11:35	john-1135	1	Allen Bailey	<p>Jesus wept.</p>	2009-05-21 21:00:27	2009-05-21
5	Acts 22:16	acts-2216	1	Allen Bailey	<p>And Now why tarrest thou? Arise and be baptized...	2009-05-25 22:02:49	2009-05-25
6	Teaching - Public and Private	teaching-public-and-private	3	James Orten	<p>Reprinted in January 2009.</p>	2009-05-27 12:01:26	1969-06-01
7	Statements From The Cross	statements-from-the-cross	1	Allen Bailey	<p>From a meeting at the Mission Hills congregatio...	2009-05-27 12:53:34	2008-09-28
8	What Grace Is	what-grace-is	1	Allen Bailey	<p>From a meeting at the Mission Hills congregatio...	2009-05-27 13:09:35	2008-09-21
9	What Saith The Scriptures	what-saith-the-scriptures	1	Allen Bailey	<p>From a meeting at the Mission Hills congregatio...	2009-05-27 13:22:20	2008-09-23
10	What Saith the Scriptures II	what-saith-the-scriptures-ii	1	Allen Bailey	<p>From a meeting at the Mission Hills congregatio...	2009-05-27 13:44:18	2008-09-24
11	Preaching Repentance	preaching-repentance	1	Allen Bailey	<p>From a meeting at the Mission Hills congregatio...	2009-05-27 14:46:25	2008-09-27
12	The Great Commission	the-great-commission	1	Allen Bailey	<p>From a meeting at the Mission Hills congregatio...	2009-05-27 15:01:43	2008-09-25
13	The Day They Left Jesus Behind	the-day-they-left-jesus-behind	1	Allen Bailey	<p>From a meeting at the Mission Hills congregatio...	2009-06-15 12:38:33	2008-09-28
14	The Romans Road To Salvation	the-romans-road-salvation	1	Allen Bailey	<p>From a meeting at the Mission Hills congregatio...	2009-06-15 12:57:22	2008-09-26
15	One Way, One Body, One Church	one-way-one-body-one-church	1	Allen Bailey	<p>From a meeting at the Mission Hills congregatio...	2009-06-15 13:07:47	2008-09-22
16	What Grace Is Not	what-grace-is-not	1	Allen Bailey	<p>From a meeting at the Mission Hills congregatio...	2009-06-15 13:27:19	2008-09-21

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

Figure 8 – Implemented Use Case: View Article – Database screenshot

Initial Requirements Assumptions

Key Domain Classes:

- Article
  - Title, rich text, and meta information for an article or presentation. Media items can also be attached to enrich the material.

- Author
  - Author, or presenter who created one or many articles in the system.
- Media
  - Audio, video, PDF documents, etc. that are attached to an article.
- Category
  - Major categories of article types, for classification purposes.
- Tag
  - Metadata attached to an article, indicating keywords and subject matter.
- User (profile information)
  - Social networking account information and simple profile information for the site's social functionality.
- Comment
  - User generated comments in response to an article.
- Like
  - A social indicator showing that a user enjoyed an article.
- Question
  - User submitted question for an administrator to answer and display in a Q&A format.
- Book
  - Published product for sale in the online store.
- Order
  - A customer order for published materials.
- Order Item
  - Line item linking an order with a book and indicating the quantity ordered.

Key Business Functions:

- Add, View, Update, Delete Article
- Add, View, Update, Delete Author
- Attach, Delete Media
- Create, View, Update, Delete Category
- Create, View, Delete Tag
- Add, Remove Tag to/from Article
- Register, View, Update, Delete User
- Add, Delete Comment
- Add, Delete Like
- Ask, View, Update, Delete Question
- Add, View, Update, Delete Book
- Create, View, Update Order
- Add, Update, Delete Order Item

Development Methodology

The project will use an agile approach to the Unified Process for the development methodology. The project will also use an object oriented approach to analysis, design, and implementation. The project will be implemented using the three-layer MVC design architecture typical of the Django framework.

Ultimate Project Disposition

The sponsor will actually use the final system. It will be deployed at the the end of the class as a replacement of the old system. This project came about because Allen is a friend of mine. We've been in discussions for a while about creating a system like the one described here. Allen is very enthusiastic about the project and I will likely continue active development after the initial implementation.

System Benefits

Tangible Benefits

Benefit/Cost Saving	Annual Amount	Comments
Non-technical administrators being able to manage resources themselves.	$3,120.00	Approximate current maintenance costs that could be done away with: 3 hours/week @ $20/hour
Online publication sales.	$4,250.00	Preachers' Study CD's: $1,500 + Commentaries: $7,000 = $8,500 annual * 50% estimated increase in sales

Intangible Benefits

- Social interaction, promoting thoughtful discussions on religious subjects.
- Promoting Christianity and edification of members.
- Preservation of religious literary works and multimedia recordings.

Project Schedule and Costs

Phases, Activities, Tasks, and Person Hours

Disciplines	Activities	Tasks	Hours
Business Modeling			9.5
	Understand the business environment		
		Interview stakeholders.	3
		Evaluate existing architecture.	1
	Create the system vision		
		Generate a list of primary business benefits.	0.5
		Develop a list of system capabilities.	2
	Create business models		
		Identify business events.	1.5
		Define information and data flow models	1.5
Requirements			53
	Gather detailed information		
		Analyze current system.	3
		Interview stakeholders.	4
	Define functional requirements		
		Create event decomposition table.	2
		Create use case diagram.	2
		Write use case descriptions.	8
		Create problem domain class diagram.	2
		Create interaction diagram.	1
	Define nonfunctional requirements		
		Define security requirements.	2
		Define reliability requirements.	2
		Define technological requirements.	2
		Define usability requirements.	2
	Prioritize requirements		
		Determine core requirements.	3
		Schedule requirements across iterations.	2
	Develop user interface dialogs		
		Create storyboards.	8
		Determine if changes to existing design are needed.	4
	Evaluate requirements with users		
		Interview stakeholders with requirements.	3
		Revise requirements as needed.	3

Design			67
	Design the support services architecture and deployment environment		
		Plan production server setup.	3
		Find or design acceptable hosting solution.	4
	Design the software architecture		
		Establish model, view, controller architecture details.	4
		Design package diagrams.	3
	Design use case realizations		
		Create design class diagram.	5
		Create sequence diagrams.	12
	Design the database		
		Finalize domain class diagram.	2
		Specify relationships between domain classes.	2
	Design the system and user interfaces		
		Review storyboards.	3
		Design view logic.	5
		Design Django templates.	3
		Design RSS and social networking interfaces.	8
		Review interface design details with stakeholders.	3
	Design the system security and controls		
		Determine security groups and access levels.	8
		Design login page.	2
Implementation			69
	Build software components		
		Build models.	20
		Build view logic.	10
		Build templates.	20
		Write URL configuration files.	5
		Write settings files.	5
	Acquire software components		
		Download and install python libraries.	2
		Download and install Django.	2
		Download and install any needed Django modules.	2
	Integrate software components		
		Configure deployment software on test server.	1
		Deploy full system on test server.	2
Testing			40
	Define and conduct unit testing		
		Define unit tests.	5
		Conduct core function unit tests.	5
		Conduct support function unit tests.	4
	Define and conduct integration testing		
		Define integration tests.	5
		Conduct integration tests.	8
	Define and conduct usability testing		
		Review design specifications against implementation.	2
		Conduct usability testing with users.	3
	Define and conduct user acceptance testing		
		Have administrators test administration functions.	5
		Review non-administrative functions with users.	3

Deployment			31
	Acquire hardware and system software		
		Purchase hosting package or setup server hardware.	5
		Install and configure system software as needed.	5
	Package and install components		
		Deploy finished system to production server.	1
		Configure system on production server.	1
	Train users		
		Develop training documentation for system.	5
		Hands-on training.	5
	Convert and initialize data		
		Add initial users to system.	1
		Import existing information.	8
Project Management			22.5
	Evaluate the project’s scope and risk		
		Create initial problem domain class list.	1
		Create initial primary use case list.	1
		Review project scope and evaluate risks and threats.	2
	Confirm the project’s feasibility		
		Calculate economic feasibility.	1
		Evaluate technical feasibility.	1
		Evaluate schedule feasibility.	1
	Develop the project and iteration schedules		
		List out activity tasks.	2
		Review with Dr. Satzinger.	0.5
	Monitor and control the project’s iterations		
		Keep log of time spent on tasks.	5
		Create periodic reports for Dr. Satzinger.	8
Configuration and Change Management			8
	Develop change control procedures		
		Setup SVN procedures.	1
		Develop plan for release to test & production servers.	2
	Manage models and software components		
		Update models as needed.	2
		Keep proper, detailed revision documentation.	3
Environment			16
	Select and configure the development tools		
		Install python and Django on development machine.	1
		Create a SVN repository.	1
		Install system software on test server.	0
		Configure test server.	2
	Tailor the UP development process		
		Setup of deliverables.	3
		Meeting with Dr Satzinger.	3
	Provide technical support services		
		Provide support to test users.	3
		Provide support for development environment.	3
Total			**316**

Project Costs

Project Hours	Hourly Rate	Project Development Cost
316	$20.00	$6,320.00

Licensing costs will be nonexistent since all the development tools are open source. The operating systems will all be Linux and the server software will be open source as well.

Maintenance Hours / Year	Hourly Rate	Maintenance Cost Annually
120	$20.00	$2,400.00

The following five year cost table takes into account the previous two tables on development and maintenance costs, as well as a basic hosting cost of $7.95/month and $10.69/year for a domain name.

	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5
Development Cost	$6,320.00	$0.00	$0.00	$0.00	$0.00	$0.00
Maintenance Cost	$0.00	$2,400.00	$2,400.00	$2,400.00	$2,400.00	$2,400.00
Hosting Cost	$0.00	$106.09	$106.09	$106.09	$106.09	$106.09
Total Costs	$6,320.00	$2,506.09	$2,506.09	$2,506.09	$2,506.09	$2,506.09

Feasibility Study

Organizational and Cultural Feasibility

The owner, Allen Bailey, is the principle stakeholder and is very enthusiastic about the project. The only issue I can see arising would be if the system didn't have a simple enough of an interface. Usability was one point that he stressed, but I don't think it will cause any issues.

Evaluating the Technological Feasibility

Security could be an issue since it is a web based system. Appropriate measures will need to be taken to ensure that the system is kept secure. Since Django is a web framework that I'm very familiar with, I don't think there will be any issues on the implementation side.

Determining the Schedule Feasibility

Since the project has a sort time frame of a single semester and I am the only analyst/developer, it will be important to stick to the project management plans and not get side-tracked. However, with due diligence I think the schedule is certainly feasible.

Assessing the Resource Feasibility

The only resources here are my time and the hosting fees once the project goes live. After talking with Allen, resource feasibility seems reasonable as well.

Determining the Economic Feasibility

	Year 0	**Year 1**	**Year 2**	**Year 3**	**Year 4**	**Year 5**	**Total**
Value of benefits		$7,370.00	$7,370.00	$7,370.00	$7,370.00	$7,370.00	
Discount factor (%10)	1.0000	0.9091	0.8264	0.7513	0.6830	0.6209	
Present value of benefits		$6,700.07	$6,090.57	$5,537.08	$5,033.71	$4,576.03	$27,937.46
Development costs	($6,320.00)						($6,320.00)
Ongoing costs		($2,506.09)	($2,506.09)	($2,506.09)	($2,506.09)	($2,506.09)	
Discount factor (%10)	1.0000	0.9091	0.8264	0.7513	0.6830	0.6209	
Present value of costs		($2,278.29)	($2,071.03)	($1,882.83)	($1,711.66)	($1,556.03)	($9,499.84)
PV of net of benefits and costs	($6,320.00)	$4,421.78	$4,019.54	$3,654.25	$3,322.05	$3,020.00	
Cumulative NPV	($6,320.00)	($1,898.22)	$2,121.32	$5,775.57	$9,097.62	$12,117.62	
Payback Period	1 Year, 5 Months, 22 Days						
5-year ROI	76.60%						

Summary

The economic feasibility alone is very good for this kind of system. No major hurdles to feasibility exist at this time. The favorable feasibility analysis combined with the significant intangible benefits puts the project in good standing.

Project Monitoring/Reporting To Date

Disciplines	Activities	Tasks	Projected	1/14 – 1/21	1/21 – 1/28	1/28 – 2/4	2/4 – 2/11	Total
Business Modeling			9.5	3.5	0	0	2	5.5
	Understand the business environment							
		Interview stakeholders.	3	1			0.5	1.5
		Evaluate existing architecture.	1	0.5				0.5
	Create the system vision							
		Generate a list of primary business benefits.	0.5				0.5	0.5
		Develop a list of system capabilities.	2	1				1
	Create business models							
		Identify business events.	1.5	0.5			0.5	1
		Define information and data flow models	1.5	0.5			0.5	1
Requirements			53	3	0	0	1	4
	Gather detailed information							
		Analyze current system.	3	1				1
		Interview stakeholders.	4	2			1	3
	Define functional requirements							
		Create event decomposition table.	2					0
		Create use case diagram.	2					0
		Write use case descriptions.	8					0
		Create problem domain class diagram.	2					0
		Create interaction diagram.	1					0
	Define nonfunctional requirements							
		Define security requirements.	2					0
		Define reliability requirements.	2					0
		Define technological requirements.	2					0
		Define usability requirements.	2					0
	Prioritize requirements							
		Determine core requirements.	3					0
		Schedule requirements across iterations.	2					0
	Develop user interface dialogs							
		Create storyboards.	8					0
		Determine if changes to existing design are needed.	4					0
	Evaluate requirements with users							
		Interview stakeholders with requirements.	3					0
		Revise requirements as needed.	3					0
Design			67	0.5	2	0	0.5	3
	Design the support services architecture and deployment environment							
		Plan production server setup.	3	0.5				0.5
		Find or design acceptable hosting solution.	4				0.5	0.5
	Design the software architecture							
		Establish model, view, controller architecture details.	4		1			1
		Design package diagrams.	3					0
	Design use case realizations							
		Create design class diagram.	5					0
		Create sequence diagrams.	12					0
	Design the database							
		Finalize domain class diagram.	2					0
		Specify relationships between domain classes.	2					0
	Design the system and user interfaces							
		Review storyboards.	3					0
		Design view logic.	5		0.5			0.5
		Design Django templates.	3		0.5			0.5
		Design RSS and social networking interfaces.	8					0
		Review interface design details with stakeholders.	3					0
	Design the system security and controls							
		Determine security groups and access levels.	8					0
		Design login page.	2					0
Implementation			69	0	4.5	6.5	0	11
	Build software components							
		Build models.	20		0.5	2.5		3
		Build view logic.	10		0.5	1		1.5
		Build templates.	20		0.5	1		1.5
		Write URL configuration files.	5			0.5		0.5
		Write settings files.	5		0.5	0.5		1
	Acquire software components							
		Download and install python libraries.	2		1			1
		Download and install Django.	2		1			1
		Download and install any needed Django modules.	2			0.5		0.5
	Integrate software components							
		Configure deployment software on test server.	1		0.5			0.5
		Deploy full system on test server.	2			0.5		0.5

Disciplines	Activities	Tasks	Projected	1/14 – 1/21	1/21 – 1/28	1/28 – 2/4	2/4 – 2/11	Total
Testing			40	0	0	0	0	0
	Define and conduct unit testing							
		Define unit tests.	5					0
		Conduct core function unit tests.	5					0
		Conduct support function unit tests.	4					0
	Define and conduct integration testing							
		Define integration tests.	5					0
		Conduct integration tests.	8					0
	Define and conduct usability testing							
		Review design specifications against implementation.	2					0
		Conduct usability testing with users.	3					0
	Define and conduct user acceptance testing							
		Have administrators test administration functions.	5					0
		Review non-administrative functions with users.	3					0
Deployment			31	0	0	0	0	0
	Acquire hardware and system software							
		Purchase hosting package or setup server hardware.	5					0
		Install and configure system software as needed.	5					0
	Package and install components							
		Deploy finished system to production server.	1					0
		Configure system on production server.	1					0
	Train users							
		Develop training documentation for system.	5					0
		Hands-on training.	5					0
	Convert and initialize data							
		Add initial users to system.	1					0
		Import existing information.	8					0
Project Management			22.5	3.5	0	0	8.5	12
	Evaluate the project's scope and risk							
		Create initial problem domain class list.	1	0.75			0.25	1
		Create initial primary use case list.	1	0.75			0.25	1
		Review project scope and evaluate risks and threats.	2				1	1
	Confirm the project's feasibility							
		Calculate economic feasibility.	1				1	1
		Evaluate technical feasibility.	1	1				1
		Evaluate schedule feasibility.	1				1	1
	Develop the project and iteration schedules							
		List out activity tasks.	2				2	2
		Review with Dr. Satzinger.	0.5					0
	Monitor and control the project's iterations							
		Keep log of time spent on tasks.	5				1	1
		Create periodic reports for Dr. Satzinger.	8	1			2	3
Configuration and Change Management			8	0	2	0.25	0	2.25
	Develop change control procedures							
		Setup SVN procedures.	1		1			1
		Develop plan for release to test & production servers.	2		1			1
	Manage models and software components							
		Update models as needed.	2					0
		Keep proper, detailed revision documentation.	3			0.25		0.25
Environment			16	0.5	2	1	1	4.5
	Select and configure the development tools							
		Install python and Django on development machine.	1		1			1
		Create a SVN repository.	1		1			1
		Install system software on test server.	0					0
		Configure test server.	2			1		1
	Tailor the UP development process							
		Setup of deliverables.	3				0.5	0.5
		Meeting with Dr Satzinger.	3	0.5			0.5	1
	Provide technical support services							
		Provide support to test users.	3					0
		Provide support for development environment.	3					0
Total			**316**	**11**	**10.5**	**7.75**	**13**	**42.25**

Appendix A: Code Excerpts

models.py

```
from django.db import models
from datetime import date
from tinymce import models as tinymce_models
import os.path
from django.conf import settings

def get_filename(instance, filename):
    split = filename.split('.')
    ext = ''.join(['.', split[len(split) - 1]])
    month = ''.join([str(date.today().year), '_', str(date.today().month)])
    slug = instance.parent.slug
    t = (None,"audio")[isinstance(instance, Audio)]
    t = (t,"video")[isinstance(instance, Video)]
    t = (t,"docs")[isinstance(instance, Document)]
    t = (t,"other")[isinstance(instance, OtherMedia)]
    fn = ''.join(['article/', month, '/', t, '/', slug, ext])
    i = 1
    if instance.src:
        instance.src.delete()
    while True:
        if os.path.exists(''.join([settings.MEDIA_ROOT, fn])):
            fn = ''.join(['article/', month, '/', t, '/', slug, '_', str(i), ext])
        else:
            return fn

def convert_youtube(url):
    sep = url.find('?v=')
    if sep > 0:
        url = 'http://www.youtube.com/v/%s' % url[sep + 3:]
    sep = url.find('&')
    if sep > 0:
        url = url[:sep]
    return url

class Article(models.Model):
    title = models.CharField(max_length=200)
    slug = models.SlugField()
    cat = models.ForeignKey('Category')
    author = models.CharField(max_length=200)
    text = tinymce_models.HTMLField(blank=True)
    date = models.DateTimeField('date published', auto_now_add=True)
    orig_date = models.DateField('date originally created')
    class Meta:
        ordering = ['-date']
    def __unicode__(self):
        return self.slug

class Audio(models.Model):
    parent = models.ForeignKey('Article')
    title = models.CharField(max_length=200)
    src = models.FileField(upload_to=get_filename, null=True, blank=True)
    class Meta:
        verbose_name_plural = "audio"
        ordering = ['parent','title']
    def __unicode__(self):
        return ''.join([self.parent.slug, "/", self.title])

class Video(models.Model):
    parent = models.ForeignKey('Article')
    title = models.CharField(max_length=200)
    src = models.FileField(upload_to=get_filename, null=True, blank=True)
    class Meta:
        ordering = ['parent','title']
    def __unicode__(self):
        return ''.join([self.parent.slug, "/", self.title])

class YouTube(models.Model):
    parent = models.ForeignKey('Article')
    title = models.CharField(max_length=200)
    url = models.URLField(null=True, blank=True)
    class Meta:
        verbose_name_plural = "YouTube"
        ordering = ['parent','title']
    def save(self):
        self.url = convert_youtube(self.url)
        super(YouTube, self).save()
    def __unicode__(self):
        return ''.join([self.parent.slug, "/", self.title])

class Document(models.Model):
    parent = models.ForeignKey('Article')
    title = models.CharField(max_length=200)
    src = models.FileField(upload_to=get_filename, null=True, blank=True)
    class Meta:
        ordering = ['parent','title']
    def __unicode__(self):
        return ''.join([self.parent.slug, "/", self.title])

class OtherMedia(models.Model):
    parent = models.ForeignKey('Article')
    title = models.CharField(max_length=200)
    src = models.FileField(upload_to=get_filename, null=True, blank=True)
    class Meta:
        verbose_name_plural = "other media"
        ordering = ['parent','title']
    def __unicode__(self):
        return ''.join([self.parent.slug, "/", self.title])

class Category(models.Model):
    title = models.CharField(max_length=200)
    slug = models.SlugField()
    description = tinymce_models.HTMLField(blank=True)
    class Meta:
        verbose_name_plural = "categories"
        ordering = ['title']
    def __unicode__(self):
        return self.slug
```

urls.py

```
from django.conf.urls.defaults import *

urlpatterns = patterns('',
    (r'^$', 'gospel_preaching.articles.views.index'),
    (r'^(?P<cat>[a-zA-Z0-9_-]+)/$', 'gospel_preaching.articles.views.cat'),
    (r'^(?P<cat>[a-zA-Z0-9_-]+)/(?P<slug>[a-zA-Z0-9_-]+)/$',
'gospel_preaching.articles.views.detail'),
)
```

views.py

```
from django.shortcuts import render_to_response, get_object_or_404
from gospel_preaching.articles.models import Article, Category
from django.template import RequestContext
from django.core.paginator import Paginator
from django.conf import settings

def index(request):
    order = request.GET.get('order', '-date')
    page = request.GET.get('page', 1)
    try:
        perpage = settings.ARTICLES_PER_PAGE
    except AttributeError:
        perpage = 10
    pager = Paginator(Article.objects.all().order_by(order), perpage)
    return render_to_response('articles/index.html', {'all_articles':
pager.page(page).object_list, 'order': order, 'page': pager.page(page), 'pager': pager},
context_instance = RequestContext(request))

def detail(request, cat, slug):
    a = get_object_or_404(Article, slug__exact=slug)
    return render_to_response('articles/detail.html', {'article': a}, context_instance =
RequestContext(request))

def cat(request, cat):
    order = request.GET.get('order', '-date')
    page = request.GET.get('page', 1)
    try:
        perpage = settings.ARTICLES_PER_PAGE
    except AttributeError:
        perpage = 10
    c = get_object_or_404(Category, slug__exact=cat)
    pager = Paginator(c.article_set.all().order_by(order), perpage)
    return render_to_response('articles/cat.html', {'cat': c, 'articles':
pager.page(page).object_list, 'order': order, 'page': pager.page(page), 'pager': pager},
context_instance = RequestContext(request))
```

base.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
	<head>
		<meta http-equiv="Content-Style-Type" content="text/css" />
		<title>{% block title %}Gospel Preaching.com{% endblock %}</title>
		<link rel="stylesheet" href="{{ MEDIA_URL }}style/content.css" type="text/css"
/>
		<link rel="stylesheet" href="{{ MEDIA_URL }}style/content_low_res.css"
type="text/css" id="var_style" />
		{% block css %}{% endblock %}
		{% block script %}{% endblock %}
	</head>

	<body>
		<div class="main">
			{% block main %}{% endblock %}
		</div>
		{% include "master/side_menu.html" %}
		<div class="header_bar">
			<h1><a href="/">Gospel Preaching.com</a></h1>
		</div>
		<a href="/"><img class="logo"
src="{{ MEDIA_URL }}logos/simple_logo_150.png"/></a>
		<script type="text/javascript">
		<!--
			if(screen.width > 800){
				document.getElementById('var_style').href = '';
			}
		//-->
		</script>
		<script type="text/javascript">
		<!--
			var gaJsHost = (("https:" == document.location.protocol) ? "https://ssl."
: "http://www.");
			document.write(unescape("%3Cscript src='" + gaJsHost +
				"google-analytics.com/ga.js' type='text/javascript'%3E%3C/script
%3E"));
		//-->
		</script>
		<script type="text/javascript">
		<!--
			try {
				var pageTracker = _gat._getTracker("UA-9202863-2");
				pageTracker._trackPageview();
			} catch(err) {}
		//-->
		</script>
	</body>
</html>
```

index.html

```
{% extends "base.html" %}
{% load custom %}

{% block css %}
{% endblock %}
{% block script %}
{% endblock %}

{% block main %}
<h2>{{ cur_app.header }}</h2>
{{ cur_app.description|safe }}
	{% if all_articles %}
		<div class="list_table">
			<div class="list_item_odd">
				<span class="list_title"><a href="{% orderurl 'title' order
%}">Title</a></span>
				<span class="list_cat"><a href="{% orderurl 'cat__title' order
%}">Category</a></span><br />
				<span class="list_author"><a href="{% orderurl 'author' order
%}">Author</a></span>
				<span class="list_dates_media">
					<span class="list_date"><a href="{% orderurl 'date' order
%}">Date</a></span>
					<span class="list_orig_date"><a href="{% orderurl 'orig_date' order
%}">Original Date</a></span>
					<span class="list_media">Available Media</span>
				</span>
			</div>
		{% include "helpers/navigation.html" %}
		{% for article in all_articles %}
		    <div class="list_item_{% cycle 'even' 'odd' %}">
				<span class="list_title"><a href="{% url
gospel_preaching.articles.views.detail article.cat.slug article.slug %}">{{ article.title }}</a></span>
				<span class="list_cat"><a href="{% url gospel_preaching.articles.views.cat
article.cat.slug %}">{{ article.cat.title }}</a></span><br />
				<span class="list_author">{{ article.author }}</span>
				<span class="list_dates_media">
					<span class="list_date">{{ article.date|date:"m/d/Y" }}</span>
					<span class="list_orig_date">{{ article.orig_date|
date:"m/d/Y" }}</span>
					<span class="list_media">
						<img src="{{ MEDIA_URL }}img/text_{{ article.text|striptags|
yesno }}.png" />
						<img src="{{ MEDIA_URL }}img/audio_{{ article.audio_set.all|
yesno }}.png" />
						<img src="{{ MEDIA_URL }}img/youtube_{% if
article.video_set.all or article.youtube_set.all %}yes{% else %}no{% endif %}.png" />
						<img
src="{{ MEDIA_URL }}img/doc_{{ article.document_set.all|yesno }}.png" />
						<img
src="{{ MEDIA_URL }}img/other_{{ article.othermedia_set.all|yesno }}.png" />
					</span>
				</span>
			</div>
		{% endfor %}
		{% include "helpers/navigation.html" %}
		</div>
	{% else %}
		<p>No articles are available.</p>
	{% endif %}
{% endblock %}
```

detail.html

```
{% extends "base.html" %}

{% block css %}
<link rel="stylesheet" href="{{ MEDIA_URL }}style/video.css" type="text/css" />
{% endblock %}
{% block script %}
<script type="text/javascript" src="{{ MEDIA_URL }}scripts/audio-player.js"></script>
<script type="text/javascript" src="{{ MEDIA_URL }}scripts/flowplayer-3.1.0.min.js"></script>
<script type="text/javascript" src="{{ MEDIA_URL }}scripts/flowplayer.controls-3.0.2.js"></script>
{% endblock %}

{% block main %}
<h2>{{ article.title }}</h2>
        <table class="header_block">
                <tr><td>Catagory:</td><td><a href="{% url gospel_preaching.articles.views.cat
article.cat.slug %}">{{ article.cat.title }}</a></td></tr>
                <tr><td>Date Uploaded:</td><td>{{ article.date|date:"m/d/y" }}</td></tr>
                <tr><td>Original Date:</td><td>{{ article.orig_date|date:"m/d/y" }}</td></tr>
                <tr><td>Author:</td><td>{{ article.author }}</td></tr>
        </table>
        {{ article.text|safe }}
        {% if article.audio_set.all %}
                <div class="content_block">
                        <h3>Audio</h3>
                        {% for audio in article.audio_set.all %}
                                <h5>{{ audio.title }}</h5>
                                <object id="{{ article.slug }}" height="24" width="290"
data="{{ MEDIA_URL }}scripts/player.swf" type="application/x-shockwave-flash">
                                        <param value="{{ MEDIA_URL }}scripts/player.swf" name="movie"/>
                                        <param value="playerID={{ article.slug }}&soundFile={{ MEDIA_URL }}
{{ audio.src }}" name="FlashVars"/>
                                        <param value="high" name="quality"/>
                                        <param value="false" name="menu"/>
                                        <param value="transparent" name="wmode"/>
                                </object><br />
                                <a href="{{ MEDIA_URL }}{{ audio.src }}">Download</a>
                        {% endfor %}
                </div>
        {% endif %}
        {% if article.video_set.all or article.youtube_set.all %}
                <div class="content_block">
                        <h3>Video</h3>
                        {% if article.video_set.all %}
                                {% for video in article.video_set.all %}
                                <h5>{{ video.title }}</h5>
                                        <a
                                                href="{{ MEDIA_URL }}{{ video.src }}"
                                                style="display:block;width:425px;height:344px"
                                                id="player{{ video.id }}">
                                        </a>
                                        <!--HTML based control bar: <div id="controls{{ video.id }}"
class="video_controls"></div>-->
                                        <script>
                                        <!--
                                                flowplayer("player{{ video.id }}",
"{{ MEDIA_URL }}scripts/flowplayer-3.1.0.swf", {
                                                        clip: {
                                                                autoPlay: false,
                                                                autoBuffering: true
                                                        },

                                                        plugins: {
                                                                controls: {
                                                                        url:
'{{ MEDIA_URL }}scripts/flowplayer.controls-3.1.0.swf',
                                                                        backgroundColor: '#000000',
                                                                        backgroundGradient: '[0.2, 1.0, 0.6]',
                                                                        buttonColor: '#707070',
                                                                        sliderColor: '#707070',
                                                                        bufferColor: '#A0A0A0',
                                                                        progressColor: '#FF0000',
                                                                        buttonOverColor: '#A0A0A0',
                                                                        volumeSliderColor: '#707070',
                                                                        timeBgColor: '#707070',
                                                                        timeColor: '#FF1010',
                                                                        tooltipColor: '#707070'
                                                                }
                                                        }

                                                });
                                        //-->
                                        </script>
                                        <a href="{{ MEDIA_URL }}{{ video.src }}">Download</a>
                                {% endfor %}
                        {% endif %}
                        {% if article.youtube_set.all %}
                                <h4>YouTube</h4>
                                {% for youtube in article.youtube_set.all %}
                                <h5>{{ youtube.title }}</h5>
                                        <object width="425" height="344" style="display:block;">
                                                <param name="movie"
value="{{ youtube.url }}&hl=en&fs=1&rel=0"></param>
                                                <param name="allowFullScreen" value="true"></param>
                                                <param name="allowscriptaccess" value="always"></param>
                                                <embed src="{{ youtube.url }}&hl=en&fs=1&rel=0"
type="application/x-shockwave-flash" allowscriptaccess="always" allowfullscreen="true" width="425"
height="344"></embed>
                                        </object>
                                {% endfor %}
                        {% endif %}
                </div>
        {% endif %}
        {% if article.document_set.all %}
                <div class="content_block">
                        <h3>Document</h3>
                        {% for doc in article.document_set.all %}
                                <h5>{{ doc.title }}</h5>
                                <a href="{{ MEDIA_URL }}{{ doc.src }}">Download</a>
                        {% endfor %}
                </div>
        {% endif %}
        {% if article.othermedia_set.all %}
                <div class="content_block">
                        <h3>Other Media</h3>
                        {% for media in article.othermedia_set.all %}
                                <h5>{{ media.title }}</h5>
                                <a href="{{ MEDIA_URL }}{{ media.src }}">Download</a>
                        {% endfor %}
                </div>
        {% endif %}
{% endblock %}
```

Legal Notes

Other Notes

I want to give credit to the amazing Django web framework and thank the amazing people that work on it. “It lets you build high-performing, elegant Web applications quickly.” Please check out the Django project's site: <http://www.djangoproject.com/>